Development of a Calibration Procedure for the Eye Tracker, Eyelink 1000 Plus

Project Work

Systems Neuroscience & Neurotechnology Unit
Saarland University of Applied Sciences
Faculty of Engineering

Submitted by : Asha Babu

Matriculation Number : 5006025

Course of Study : Neural Engineering

First Supervisor : Prof. Dr. Daniel J. Strauss

Second Supervisor : Dr. Farah I. Corona-Strauss

Saarbrücken, September 30, 2024

Abstract

Humans make 3-5 eye movements per second, and these movements are crucial in helping us deal with the vast amounts of information we encounter in our everyday lives [1]. Researchers have been showing interest in developing new systems to monitor and track eye movements which helps to ease interpreting the psychological and cognitive functions. A recently developed and widely used advanced eye-tracking device is the Eyelink 1000 Plus by SR-Research which has high spatial resolution and sampling rates. This project report is a development of the calibration procedure of the eye tracker, Eyelink 1000 Plus used for recording pupil dilation, as an indicator of cognitive and emotional responses. Eyelink 1000 plus has introduced multiple features which give information about pupil size and gaze location. Calibrating equipment before performing the task is critical to achieving precise tracking of pupil size and movement, which helps to effectively study the participants' cognitive, attention and emotional state. The report outlines different calibration and validation steps performed in the eye tracker and discusses the analysis of the recorded pupil size data using the phantom eye with known pupil size. From the recordings we have observed that, after the calibration, there occurs a validation process which will be passed only when the calibration is successful following which we get good results in the cognitive and emotional experiments. The study aimed to design and implement calibration, familiarize with the procedures and analyze the data using phantom eyes with known pupil size.

Zusammenfassung

Der Mensch macht 3-5 Augenbewegungen pro Sekunde, und diese Bewegungen sind entscheidend dafür, dass wir die riesigen Mengen an Informationen, denen wir im Alltag begegnen, verarbeiten können [1]. Forscher haben Interesse an der Entwicklung neuer Systeme zur Überwachung und Verfolgung von Augenbewegungen gezeigt, die die Interpretation der psychologischen und kognitiven Funktionen erleichtern sollen. Ein kürzlich entwickeltes und weit verbreitetes fortschrittliches Eye-Tracking-Gerät ist das Eyelink 1000 Plus von SR-Research, das eine hohe räumliche Auflösung und Abtastrate aufweist. Dieser Projektbericht befasst sich mit der Entwicklung des Kalibrierungsverfahrens des Eyelink 1000 Plus, der zur Aufzeichnung der Pupillenerweiterung als Indikator für kognitive und emotionale Reaktionen verwendet wird. Eyelink 1000 Plus verfügt über mehrere Funktionen, die Informationen über die Pupillengröße und die Blickposition liefern. Die Kalibrierung des Geräts vor der Durchführung der Aufgabe ist entscheidend für eine präzise Verfolgung der Pupillengröße und -bewegung, was zu einer effektiven Untersuchung des kognitiven, aufmerksamen und emotionalen Zustands der Teilnehmer beiträgt. Der Bericht beschreibt die verschiedenen Kalibrierungs- und Validierungsschritte des Eye Trackers und erörtert die Analyse der aufgezeichneten Pupillengrößendaten anhand eines Phantomauges mit bekannter Pupillengröße. Anhand der Aufzeichnungen haben wir festgestellt, dass nach der Kalibrierung ein Validierungsprozess stattfindet, der nur bei erfolgreicher Kalibrierung durchlaufen wird, woraufhin wir gute Ergebnisse bei den kognitiven und emotionalen Experimenten erhalten. Ziel der Studie war es, die Kalibrierung zu konzipieren und durchzuführen, sich mit den Verfahren vertraut zu machen und die Daten unter Verwendung von Phantomaugen mit bekannter Pupillengröße zu analysieren.

Declaration

I hereby declare that I have authored this work independently, that I have not used other than the declared sources and resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. This work has neither been submitted to any audit institution nor been published in its current form.

Saarbrücken, September 15, 2024

Asha Babu

Contents

1 Introduction

1.1 Motivation

It is important to get accurate data after a cognitive experiment, also calibration plays an equal and important role in improving accuracy. The primary objective of this project is to gain a comprehensive understanding of the EyeLink 1000 Plus system and its calibration processes. By conducting pre-experiment calibrations, I aim to enhance data accuracy and reliability. Additionally, this study will explore the correlation between pupil diameter and data quality, providing valuable insights into potential influencing factors. Calibration can reduce inconsistencies or inaccuracies of data especially, calibration with a phantom eye can help refine gaze positions.

1.2 Acknowledgments

I want to express my sincere gratitude to Dr. Farah Corona Strauss for guiding me in this project and for their invaluable guidance, expertise and support throughout this project. I also thank the Systems Neuroscience and Neurotechnology Unit for providing the necessary equipment and resources—special thanks to Sophie Bonnair and Valentina Serbanescu for their assistance with the EyeLink 1000 Plus system.

2 Problem Analysis and Goals

2.1 State of the Art

Eye tracking has become an invaluable tool for understanding attention, visual behaviour, and human behaviour in several diverse fields, including psychology, neurophysiology, user experience, market research/neuromarketing, etc. The technology can also be used for medical analysis and screening, providing a new interaction method [4]. Historically, eye-tracking systems were invasive and immobile, and therefore useful only in very limited experiments. While there are still some limitations of eye tracking, recent advancements in the technology have allowed for much smoother setup and more universal applications [4].

An eye tracker is a device that measures eye positions and movements to determine where a person is looking and how their eyes respond to visual stimuli. It is used to analyze gaze direction, fixation duration, and eye movement patterns, providing insights into attention, visual processing, and cognitive state [15]. The retina has an area of dense nerves and high visual activity called the fovea. The lens of the eye focuses light on the fovea and a person moves their eyes to aim the lens and fovea where they want to look. By tracking and analyzing these eye movements, researchers can apply eye-tracking technology to gain valuable insights into human behaviour, physiology, psychology, perception, and visual attention [5].

Most video-based eye tracking systems consist of an infrared-sensitive camera, infrared light (IR) illumination, and a sophisticated algorithm for pupil centre detection and artefact rejection, see Fig 2.1.1. Image processing and data collection is handled by dedicated hardware or software. Infrared-based illumination has several advantages: the illumination is largely invisible to the participant, and artefacts from artificial light sources can be easily filtered out by wavelength. Most modern video eye trackers use the P-CR (pupil – Corneal Reflection) technique to deal with small relative movements between the eye tracker camera and the eye [5]. Certain characteristics of the pupil and cornea are unique under IR illumination, making it easy to detect the eye selectively and to reject “false eyes” from the camera view.

2 Problem Analysis and Goals

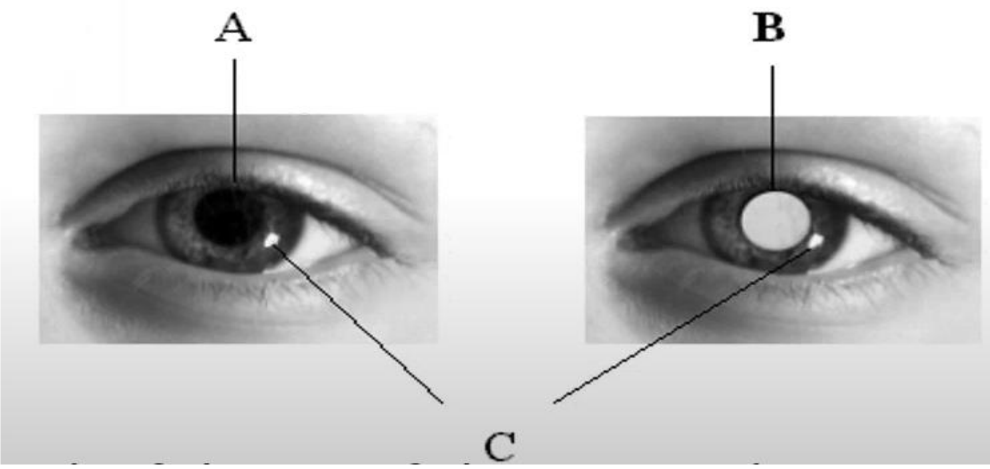


Figure 2.1.1: Dark pupil(A) and bright pupil(B) with Corneal Reflection(C)[7]

Figure 2.1.1 shows the dark pupil technique (A) and bright pupil technique (B) along with the Corneal Reflection (C), which were the video-based eye tracking system's basic principle. The EyeLink 1000 Plus eye tracker, which we are going to address in this project report uses the Dark pupil technique, looking to find a dark circle in the images of their processing. The EyeLink 1000 Plus is a state-of-the-art eye-tracking system widely used for precise pupil size, saccades, and gaze tracking in various fields, including psychology, neuroscience, and usability testing. However, the current calibration procedure poses several challenges that can affect the accuracy and reliability of the data collected.

This study aims to develop ease of calibration procedures, which are complicated for new users and require substantial time to familiarise themselves with the process. This can lead to errors in setting up the device, prolonging experiment setup time. The documentation of the calibration procedure can develop a more efficient and accurate calibration procedure for the EyeLink 1000 Plus, which reduces setup time and improves precision. Moreover, it helps to improve the user experience by simplifying the calibration workflow, ensuring that new users can complete the calibration process within 5 minutes with minimal training.

2 Problem Analysis and Goals

We measured the distance from the camera to the monitor to understand the effect on the units we recorded during the experiment. Experimenting with different phantom eyes (Artificial Eyes) of increasing diameter to understand the change in the unit values recorded.

2.2 Recent Advances in Research

The EyeLink 1000 Plus is a high-performance eye-tracking system for research and clinical applications. It offers exceptional accuracy, precision, and versatility, making it a popular choice among researchers and clinicians worldwide. EyeLink 1000 Plus has industry-leading accuracy and precision with a spatial resolution of 0.25-0.5 degrees and a sampling rate of up to 2000 Hz. The system is highly customisable, allowing users to choose from various mounting options, interchangeable lenses, and head-fixed or head-free tracking modes. It comes with user-friendly software that simplifies data collection, analysis, and visualization. In applications such as investigating attention, perception, decision-making, language processing, and social behaviour, examining neural correlates with eye movements and visual information processing are widely used.

EyeLink 1000 Plus Core System with the addition of a Fiber Optic Camera Upgrade and Long-Range Mount. This allows the same system to be used both inside a regular behavioural laboratory and integrated with fMRI and MEG scanners and other neurophysiological recording techniques. long-range eye-tracking solution is compatible with all MRI scanners and multiple

head coils. We have a complete range of solutions available for Siemens, Philips, and General Electric Scanners, from 1.5T to 13T, including the new 7T Siemens Terra systems. The EyeLink 1000 Plus is also compatible with all MEG systems such as CTF, Elekta, and 4D Yokohama, in both upright and supine positions, as well as with new OPM-MEG systems. Finally, our system is compatible with many scanner peripherals, including EEG [11].

The most-used software tool is Tobii Software. This is probably because Tobii is the most-used eye tracker in software engineering studies. The ease of using the provided software tool of the eye tracker instead of searching for a compatible software tool and maybe needing to make some modifications to the data set may also be a reason for this. Another reason may be that the researchers are not aware of other software tools that may be better suited for their needs. This may imply that there may be better software tools available than Tobii Software, but most researchers do not research properly if such a software tool exists [12].

3 Basic Principle of Eyelink 1000 Plus

Pupillometry is a technique that records changes in the diameter of the pupil size of the pupil changes in response to changes in luminance, a change that has a latency of approximately 200 ms [11]. IR light source illuminates the eye to provide corneal reflection. When the corneal reflection occurs the high-speed video camera takes up to 2000 pictures of the eye each second. Thus, the host PC with the eye tracking software runs RTOS for accurate timing which receives and processes the images from the camera. Resulting in the gaze data by recording what pixel in the screen the person is looking at.

EyeLink camera is a digital black and white camera consisting of a sensor that has thousands of pixels. These sensors are lens which focuses the light on and underneath the sensors there are photodiodes which depend on how much light or photons fall on these pixels and generate a voltage, higher or lower depending on the level. Bright things will produce higher levels of voltage and dark things produce lower levels of voltage. Here the technique used for tracking is the Pupil-CR method. Tracking just the pupil is not optimal. The location on the camera sensor could be changed due to the eye rotating or the person moving their head. To separate eye movement from head movement at least need two points of reference. That is the reason for selecting the Pupil-CR technique. When the eye rotates the position of the Pupil and Corneal Reflection changes but remains relatively constant with minor head movement. In Figure 3.1 we can observe that when the eye rotates the CR stays relatively fixed in space since IR illumination doesn't move. The CR However we can also do Pupil-only mode in the eye tracker for tracking

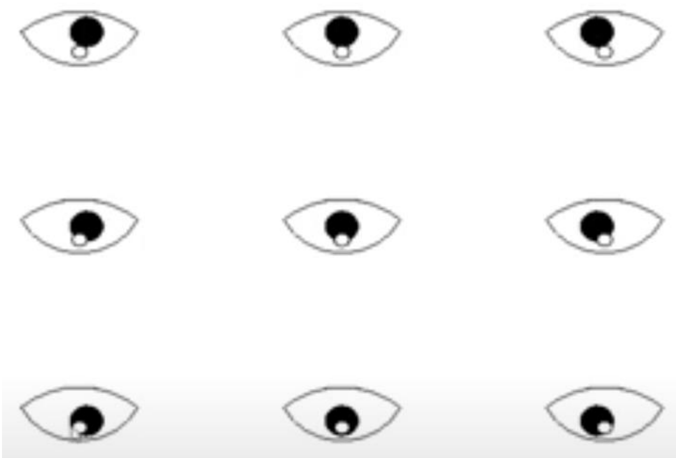


Figure 3.1: When the Position of CR eye rotates.[6]

3 Basic Principle of Eyelink 1000 Plus

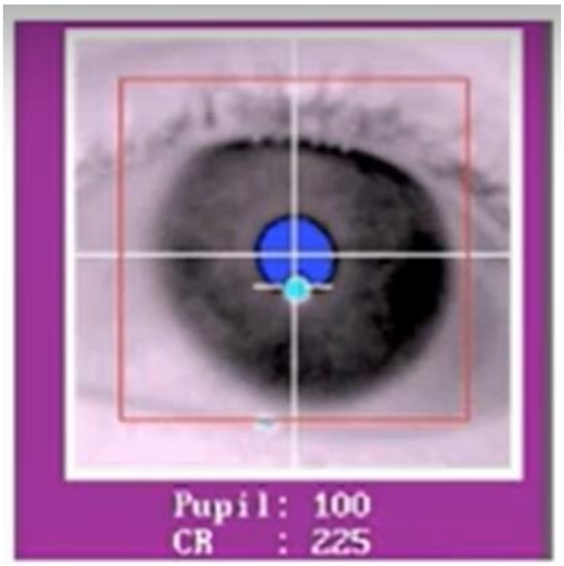


Figure 3.2: Pupil and CR detected in the eye tracker.[7]

Figure 3.2 illustrates how the pupil and corneal reflection are detected in an eye-tracking system. The relationship between the pupil and CR changes as the eye rotates but stays the same if the head moves concerning the camera. The Host PC software subtracts the CR location from the pupil location, eliminating confound caused by small head movements. The Centre of the pupil is estimated based on thresholded pixels. Similarly, corneal reflection is calculated based on pixel threshold. Before applying the thresholds, we have to focus the eyes by rotating the knob in the camera. Figure 3.3 is an example of poor and good focus. Focusing minimizes any ambiguity as to identify the pupil or CR In Figure 3.2 we can see the values under the image are the threshold values for the pupil and CR. Pupil and CR thresholds are greyscale values varying from 0 (black) to 255(white). Here, the idea is to feed the host PC how black or white something must be considered as Pupil or CR. Figure 3.4 illustrated with a red box, shows the option to increase or decrease the Pupil and CR thresholds in the eye tracker monitor along with the auto threshold button automatically setting the threshold by sensing the pixel with the camera.

3 Basic Principle of Eyelink 1000 Plus

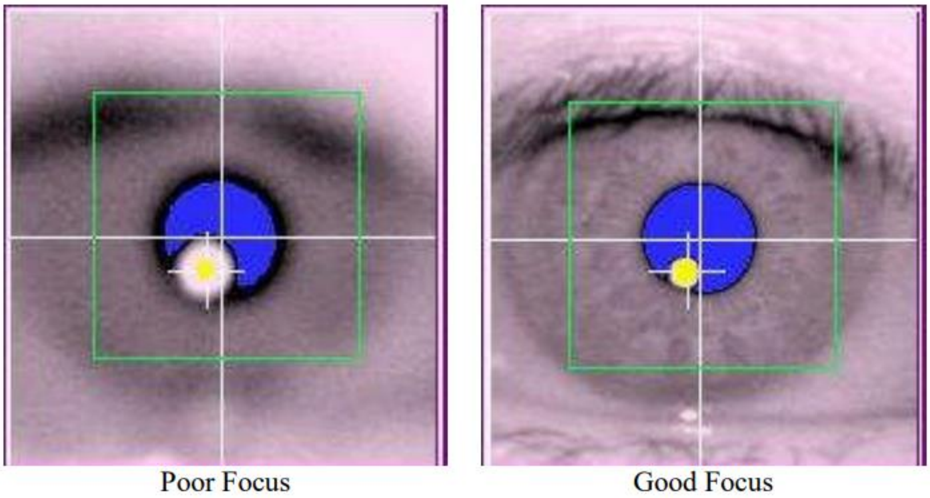


Figure 3.3: Focusing the Camera.[6]

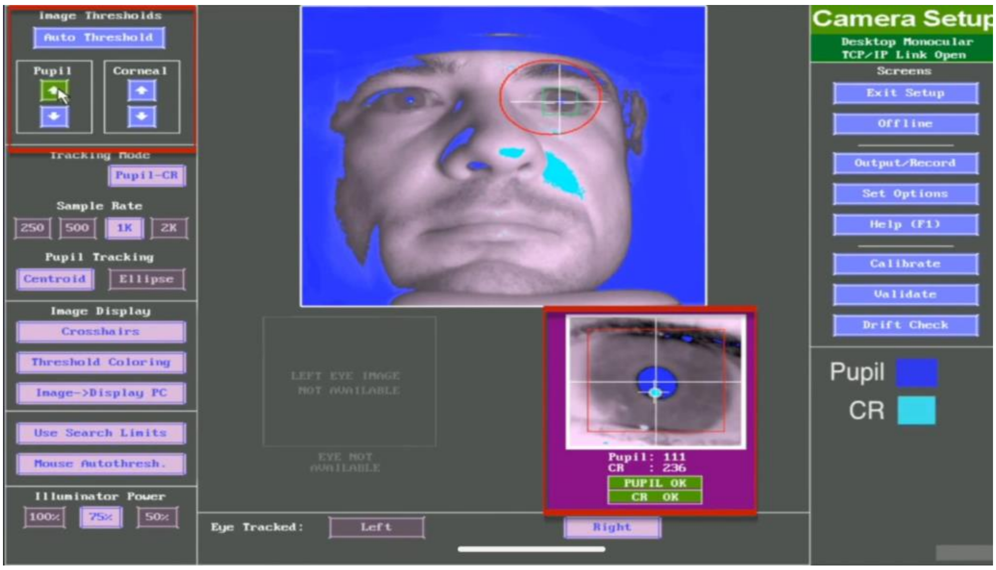


Figure 3.4: Pupil and CR adjusting in the monitor.[13]

Figures 3.5 and 3.6 show how the ideal Pupil and CR should look on the screen while setting up the threshold. There is an option called auto threshold which sets the thresholds automatically by sensing the image from the camera. We have to make sure that the thresholds are not too high or low, the black thin border around the Pupil and white thin border around the Corneal Reflection are important to understand the better threshold values.

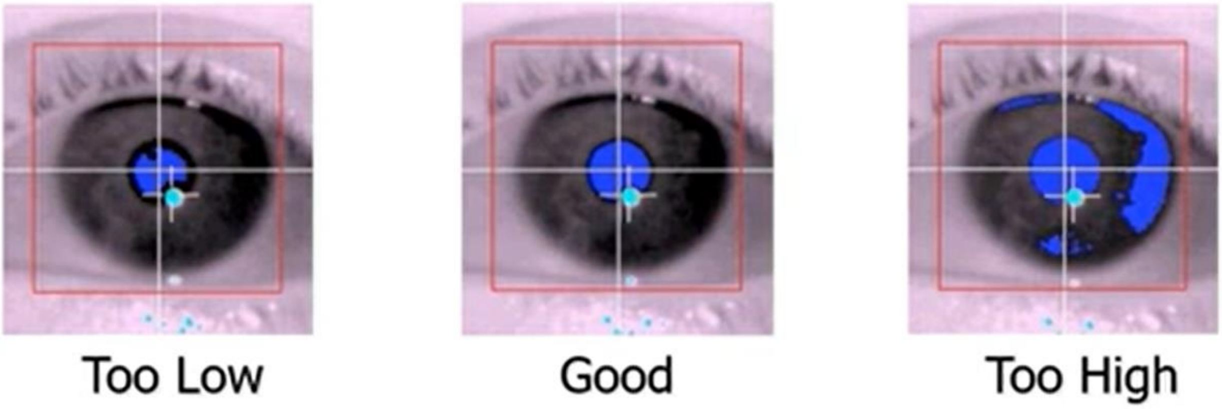


Figure 3.5: Different pupil thresholds.[7]

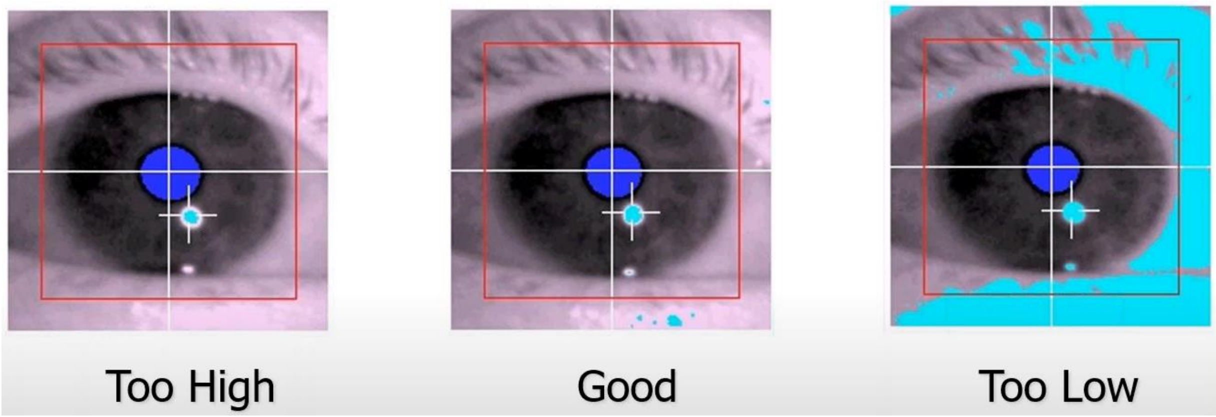


Figure 3.6: Different CR thresholds [7]

The idea to adjust the threshold is to maximise the amount of the pupil that we have threshold reasons that will become apparent soon and then maximise corneal reflection that we are thresholds so that the image processing algorithm has a maximum amount of data that is more data is equal to less noise. The EyeLink digital camera senses the photons that fall on the pixels

3 Basic Principle of Eyelink 1000 Plus

and generates voltage stored in a spreadsheet containing the voltage for each pixel. The round pupil will not map since the camera pixels are square. Small changes in brightness can impact whether or not pixels are included.

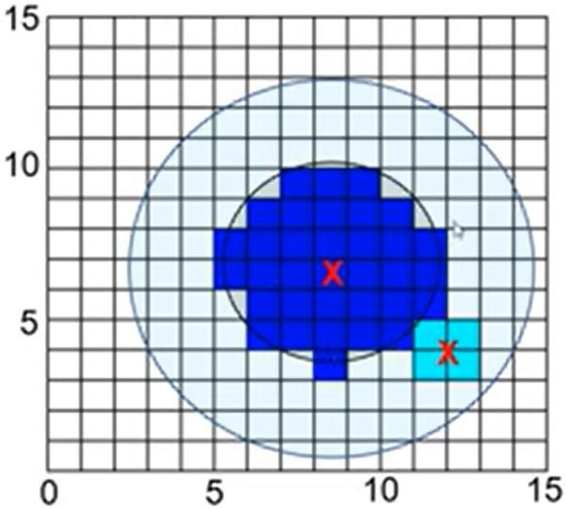


Figure 3.7: Pixels detected on the camera[7]

Figure 3.7 shows how the pupil is detected as pixels in the camera and the software computes the centre of gravity (red cross) of both Pupil and CR blobs. These values are in camera sensor pixel units and it is what you see when you plot them. In the figure, we can calculate the pupil centre coordinates (8.5, 6.7) and the CR coordinates (12. 4). Here we have the two coordinates, which is the heart of eye tracking data. We get this data by subtracting the location of the corneal reflection from the location of the pupil. This is the raw data which is stored from the eye tracker to get the gaze data. By adjusting the threshold, we grab as many pixels to minimize the sample-to-sample noise.

4 Components of EyeLink 1000 Plus

The EyeLink 1000 Plus consists of 3 critical components, an infrared light source, a high-speed video camera and the host PC + EyeLink software, shown in the figure4.1.

Figure 4.1: EyeLink 1000 Plus infrared light source, a high-speed video camera and the host PC + EyeLink software, [13]

At the heart of the EyeLink 1000 Plus is the unique high-speed camera, capable of recording binocular eye movements at up to 2000 frames per second. The camera images are processed by software running on a dedicated Host PC, using a real-time operating system to ensure outstanding temporal precision. Choose either a desktop Host PC – perfect for standard laboratory settings, or a laptop Host PC for a smaller footprint and increased flexibility. Gaze data are returned from the Host PC to the stimulus presentation computer and other external devices via an Ethernet link and/or analog voltage, with an end-to-end delay of less than 3 ms – perfect for gaze-contingent tasks [13]. The camera can be transferred between multiple mounts, allowing for a truly flexible eye-tracking solution that can meet a wide range of research needs such as desktop mount, tower mount, arm mount, long-range mount(fMRI/MEG), and non-human primate mount [13]. In our experiments, we used a desktop mount setup. The Desktop Mount sits below the tracked area and is perfect for

4 Components of EyeLink 1000 Plus

traditional screen-based eye tracking. It allows the camera to operate at a range of 40-70cm, so no electronics need to be near the participant's head, making it perfect for research involving EEG or TMS. It supports both the Head-Stabilized and Remote head free-to-move modes and is the ideal choice for most standard screen-based eye-tracking scenarios [6].

Figure 4.2: EyeLink 1000 Plus desktop-mounted Camera and Infrared light source [13]

The EyeLink 1000 Host PC performs real-time eye tracking at 250, 500, 1000, or 20001 samples per second with no loss of spatial resolution, while also computing true gaze position on the display viewed by the subject. On-line detection and analysis of eye-motion events such as saccades and fixations are performed. These events can be stored in a data file on the Host PC, sent through the Ethernet link to the Display PC with a minimal delay, or output as analog signals (if the analog/digital I/O card is installed). From the Host PC, the operator performs subject setup, monitors performance, and can communicate with applications running on a

4 Components of EyeLink 1000 Plus

Display PC [6].

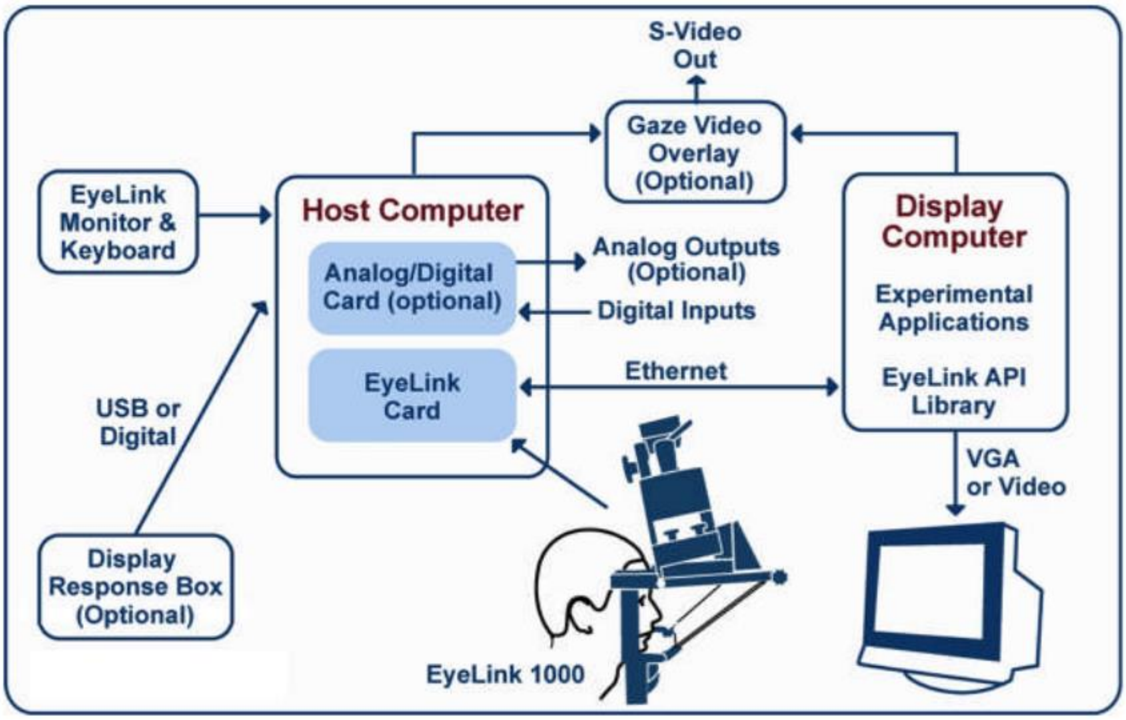


Figure 4.3: Typical EyeLink 1000 Plus Configuration[6]

A typical EyeLink setup is depicted in Figure 4.3. This figure illustrates the Tower Mount. The system consists of two computers – one, the Host PC is dedicated to data collection. The second PC is referred to as the Display PC and is generally used for the presentation of stimuli to a participant. The two computers are connected via an Ethernet link that allows the sharing of critical information from the Host PC to the Display PC, such as the occurrence of eye events, or images from the camera. Similarly, the Display PC can communicate with the Host PC, allowing Display PC applications to direct the collection of data. An optional EyeLink button box is attached directly to the Host PC allowing for the accurate synchronization of participant responses with the eye movement data. Message passing also allows events collected by I/O devices on the Display PC (e.g., button boxes, microphones, etc.) to be accurately noted in the data file [6].

4 Components of EyeLink 1000 Plus

Figure 4.4: image to show the central line of the camera.

To set up the camera, we have to first position the head of the participant by adjusting the manual head foxing arms on both sides and then we adjust the camera to align the central line which is shown in Figure 4.4 to the centre of the face by partitioning the participant face into two equal parts. Here in this image, we used a phantom eye of diameter 7 mm attached to the spectacle. We used both eye detection modes, we can also use a single eye detection mode. In addition, we can also use pupil-only and pupil-CR detection modes.

4 Components of EyeLink 1000 Plus

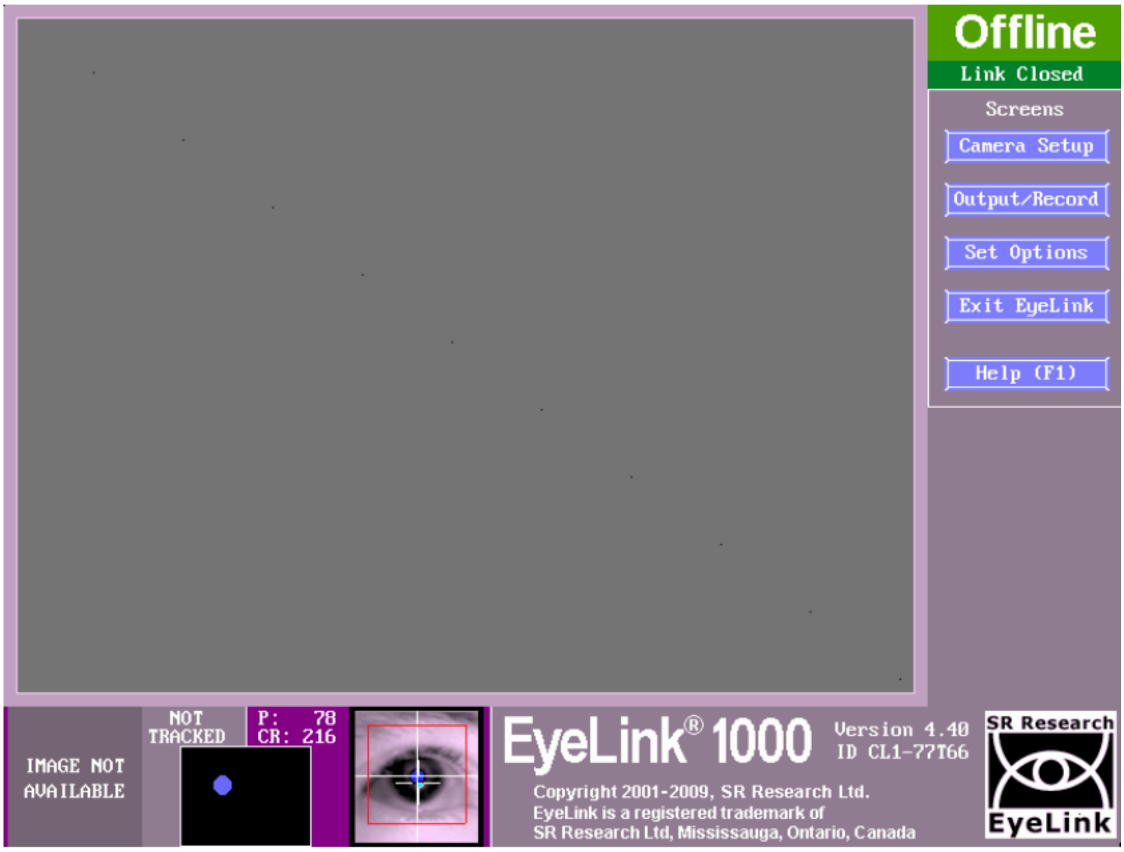


Figure 4.5: Offline Screen Image.[7]

Figure 4.5 is the offline screen image of the EyeLink 1000 Plus software. By selecting the exit EyeLink option, we can set up the distance from the participant's eye to the top and bottom of the screen, which is some information we have to provide the eye tracker before the experiment. To provide this information, start the eye tracker, which will automatically boot into the EyeLink software. Here as we see in Figure 4.5 click on exit setup to exit from the main screen to the setup window. From the file manager click the configuration button shown in Figure 4.6, and then click on the screen settings button first we have to measure the screen dimensions shown in Figure 4.7, and measure the width and height of the display monitor in millimetres, make sure we are measuring only the screen excluding the edges.

4 Components of EyeLink 1000 Plus

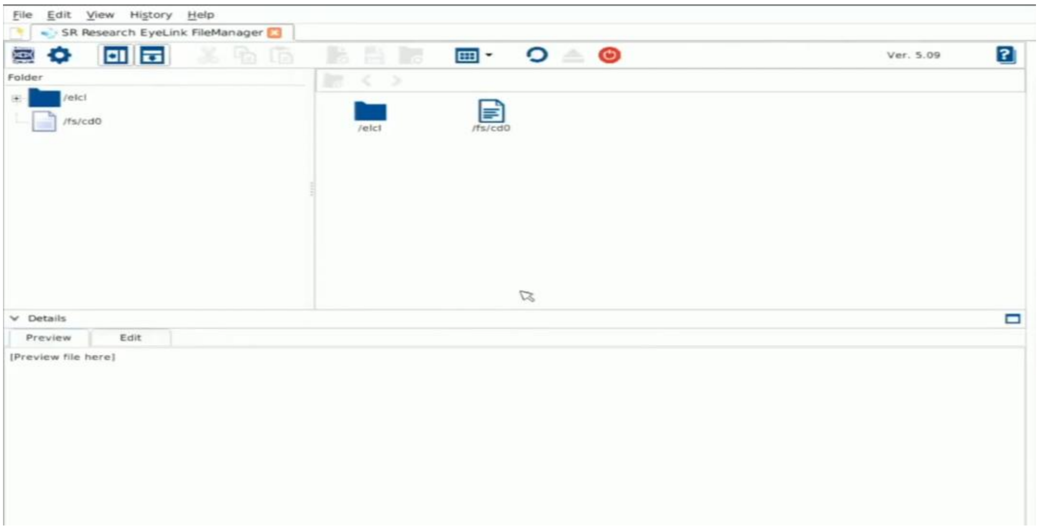


Figure 4.6: Configuration window.[14]

Figure 4.7: Screen Settings Window.[14]

Next, we measure the eye-to-screen distance using a laser meter. Here we have to put the participant in position and measure from the eye directly to the top of the screen by placing a blank white paper for reference. Repeat the same to the bottom of the screen. Then we have to feed them in the settings shown in Figure 4.8. Every distance should be in millimetres.

Figure 4.8: Eye-to-screen distance window.[14]

Clicking on the next button opens the window to add the screen to the camera distance. We measured this with a measuring tape. While in the setup we have to move the camera to the side or the front or back according to the participant's height. This is shown in the figure 4.9. Also, note that the measurements should be taken in millimetres. These are the basic setups of the device before starting the recordings.

Figure 4.9: Camera-to-screen distance window.[14]

5 Calibration and Validation Procedure of EyeLink 1000 Plus

The idea of calibration is to make sure the eye tracker determines the spatial accuracy. It provides a mathematical model that maps the difference between the centre of the Pupil and CR, which is expressed in camera pixel coordinates onto screen pixel coordinates. The regression model is used to map the low-level eye tracker data onto the gaze coordinate reference frame. It is achieved by presenting several targets like 13, 9 and 5 depending upon what calibration we choose. In our experiments, we choose the 9-target model for our experiment. These targets have a known location. We ask the participants to look at these targets, accept the location and repeat the procedure till the 9 targets are accepted. From this pair of inputs, they created a regression model that allows us to predict one from the other thus we can predict where the participant looking at the screen, and which pixel the person looking at. The eye tracker regression model is nonlinear and is fairly complicated. A good calibration is critical for spatially accurate data.

Figure 5.1 depicts the button for selecting calibration on the right side of the screen or pressing C on either keyboard. Make sure that the pupil tracking settings should be selected. There are two types of pupil tracking settings. One is a centroid and the other one is an ellipse. For our experiments, we used centroid throughout. The option is on the left side of the eye tracker monitor.

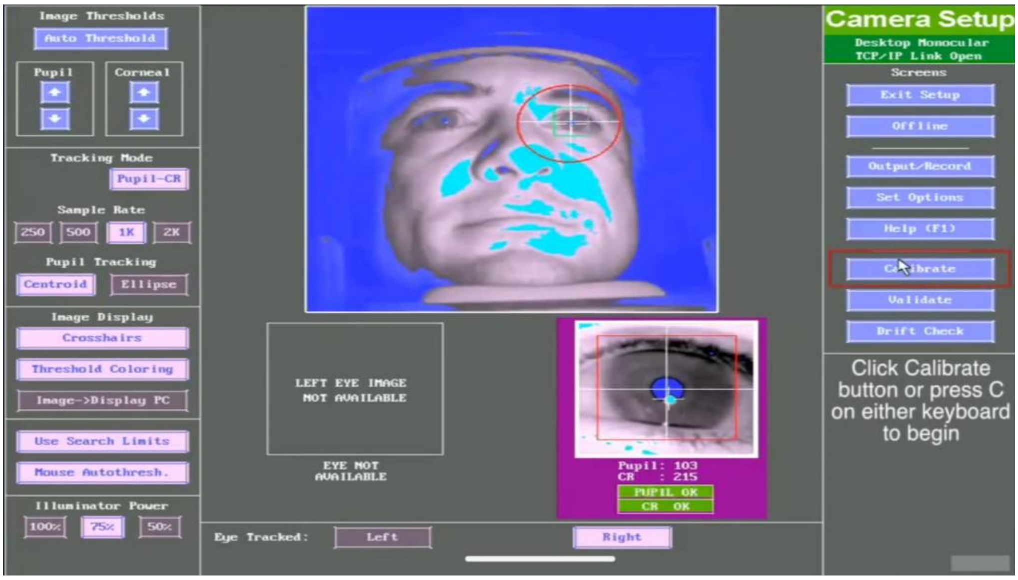
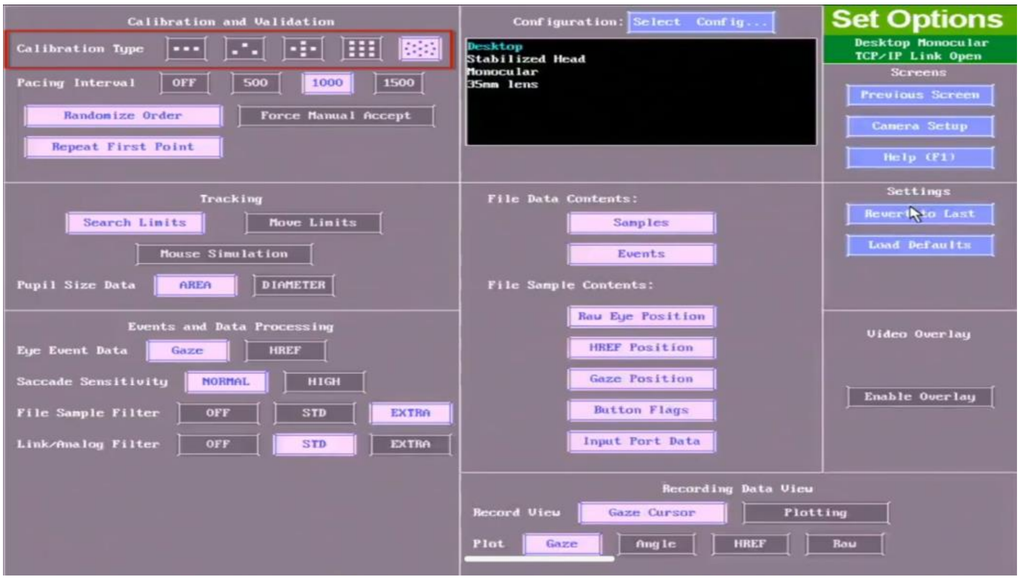


Figure 5.1: Calibrate button in the monitor.[13]

5 Calibration and Validation Procedure of EyeLink 1000 Plus

After selecting the pupil tracking settings we have to select the calibration type and pupil size data settings in the set options situated on the right side of the monitor in Figure 5.1. Here for calibration, we have 4 options 3, 5, 9 and 13-point target. As the number of the targets increases the accuracy also increases. There are 2 settings for pupil size data, one is area and the other is diameter. These are shown in Figure 5.2 on the top of the screen calibration target and towards down are the pupil size data settings. In our experiments, we used a 9-point calibration model and Area as the pupil size. If we select a 9-point calibration settings we automatically set the 9-point validation settings with 9 targets.

Figure 5.2: Calibration model and pupil data size settings.[13]



A target will appear on the screen when we begin calibration, as shown in Figure 5.3. Instruct the participant to fixate the dot in the centre of the target and follow it as it moves around the screen. Instruct them to continue fixating the target until it has moved to its new position. Figure 5.4 shows the host PC calibration screen, the cursor on the screen represents the current position of the ye in the camera image. When the participant is fixating on the first target and the gaze cursor is steady, click the Accept Fixation button or press the spacebar to start calibration. As the participant moves their eyes according to the target continue to click on the Accept Fixation button for the 9 times or select autocalibration. During calibration, the host PC will draw a crosshair representing the position of the eye on the camera for each of the calibration targets.

5 Calibration and Validation Procedure of EyeLink 1000 Plus

Figure 5.3: Appeared target on the screen while calibration.[13]

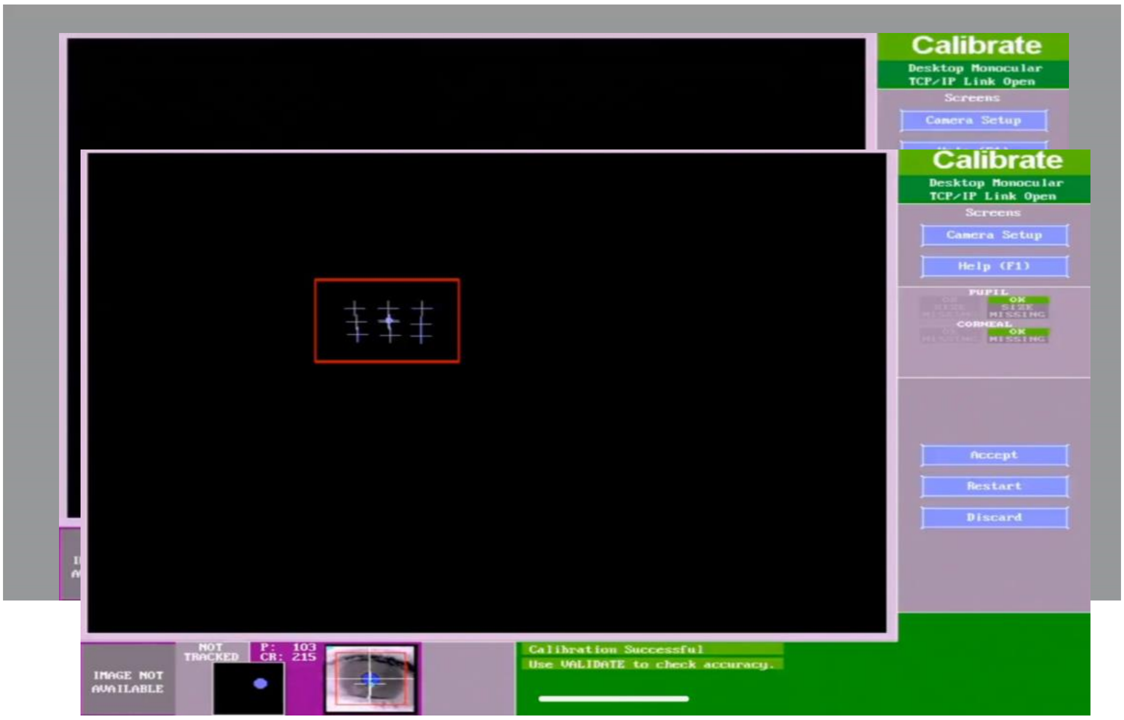


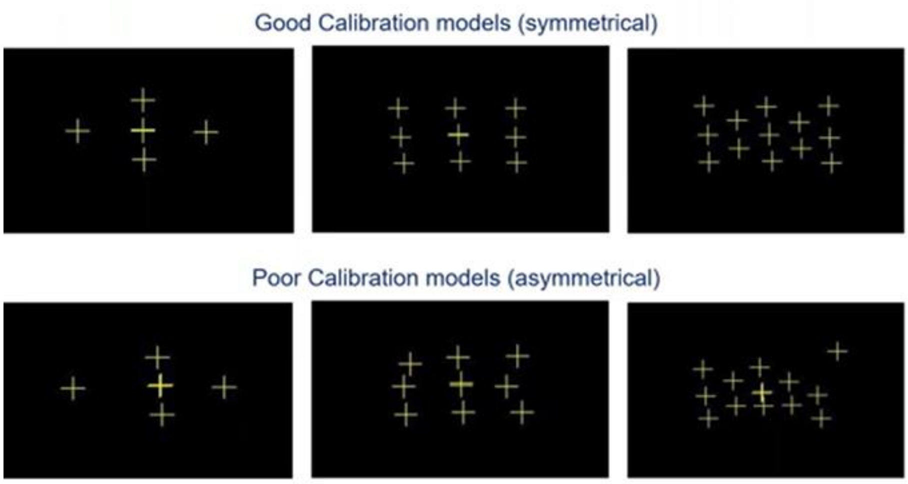
Figure 5.4: Host PC calibration screen with cursor.[13]

In a 9-point target, the crosshair grid should look like a 3*3 grid, as shown in Figure 5.5. If a crosshair looks out of position during calibration. You can press the backspace or undo the last pat to step back through previous targets. Figure 5.6 shows the comparison between a poor calibration grid and a good calibration grid. When the calibration is good the grid looks symmetrical, else the grid looks asymmetrical.

5 Calibration and Validation Procedure of EyeLink 1000 Plus

Figure 5.5: 3*3 calibration grid in the Host PC.[13]
Figure 5.6: Pixels detected on the camera[7]

After calibration, we soon begin the validation by accepting the calibration and clicking the validate option or pressing 'V' on the keyboard. Validation will appear identical to the calibration procedure but on the Host PC, the gaze cursor reflects the measured position of the eye on the Display PC monitor as shown in Figure 5.7. When the participant is fixating on the first target and the gaze cursor is steady, press the spacebar or click Accept Fixation to start Validation. Validation will report the accuracy for each point, in degrees of visual angle. For healthy adult participants with normal vision, use 0.5 degrees or less average error and 1.0 degrees or less maximum error. If the error value obtained is higher than the criteria, we may need to make adjustments to the camera setup or participant setup.



5 Calibration and Validation Procedure of EyeLink 1000 Plus

Figure 5.7: Validation gaze cursor in the Host PC.[13]

Like calibration, validation also consists of 9 targets. The participant should fixate on the dot and follow the dots wherever it moves on the Display PC. By accepting those fixations, we get points with the error values illustrated in Figure 5.8. Once the validation looks good, as shown in Figure 5.8, click Accept or press Enter to go back to the camera setup screen. From the camera setup screen on the Host PC click on the output record or press O to proceed with our experiment.

5 Calibration and Validation Procedure of EyeLink 1000 Plus

Figure 3.3.8: Validation of gaze cursor position with error values in the Host PC.[13]

6 Materials Required for the Experiment

Phantom Eyes

Apart from the setup, we used phantom eyes (artificial eyes) with known pupil diameters to record the pixel units. The phantom eyes are shown in Figure 6.1, with decreasing order of pupil size. The pupil sizes are measured by using a vernier calliper by opening the plastic coating over it, which is shown in Figure 6.2.

Figure 6.1: Phantom eyes with a decrease in diameter.

Figure 6.2: Measuring the diameter with a vernier calliper.

6 Materials Required for the Experiment

Figure 6 .2 is an illustration of measuring the small size pupil of the phantom eye. As we can see it is measured 3.5 mm in the scale. Likewise, we measured all the 5 phantom eyes. After the measurements, we got our phantom pupil sizes as 3.5mm, 5mm, 6mm, 7mm and 9mm. Furthermore, we convert the diameter into to area since we use the pupil size data setting as Area. The equation for the area of the circle is given (1).

$$Area\ of\ the\ circle = \pi * r^2 \qquad (1).$$

Since we have the pupil's diameter, we must modify the equation to (3).

$$r = \frac{Diameter(d)}{2} \qquad (2).$$

$$Area\ of\ the\ artificial\ pupil = \pi * \left(\frac{d}{2}\right)^2 \qquad (3).$$

Table 3.4.1 is the conversion of the pupil size of the phantom eyes to the area in millimetre square. We used this data to plot the graph against the pixel units we obtained after using each phantom eye.

Pupil Diameter (mm)	Area (mm^2)
3.5	9.6211
5	19.6350
6	282743
7	38.4845
9	63.6173

Table 6.1: Conversion of pupil size of phantom eyes into Area.

Pupil size fluctuates continuously and is readily influenced by many non-cognitive factors. The Pupil size has a robust relationship with the rather vague and slippery concept of “mental effort” which is probably involved in most tasks. EyeLink systems report the Pupil area or diameter in an “arbitrary unit” which converts to the pupil size in mm using conversion factor. The units reflect the number of thresholded pixels for example the dark blue pixels. The diameter is derived from the area, so it doesn’t matter which one we choose, but be consistent in choosing one throughout the experiment since it would be easy while processing the signal.

To convert arbitrary Area units to mm, follow the given steps:

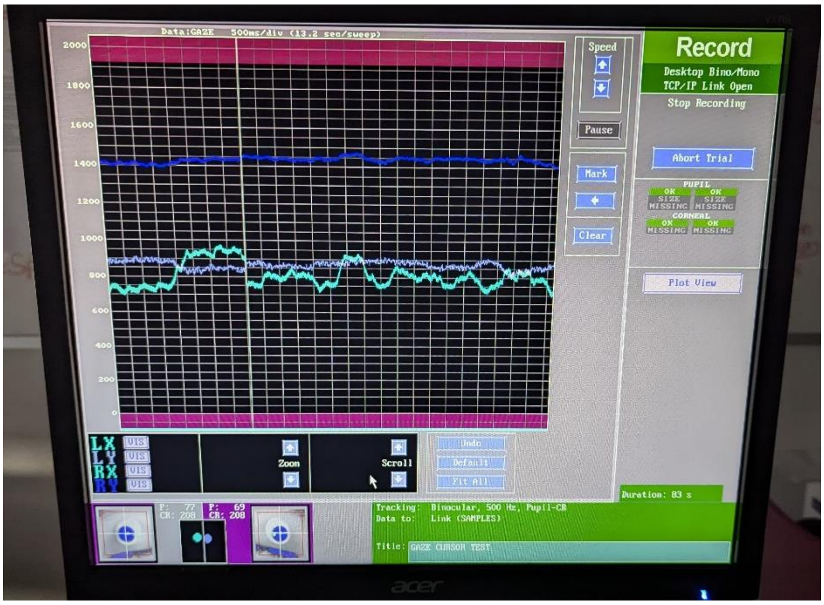
1. Use an artificial eye with a known pupil size. (we used a 7 mm diameter)
2. Place it at the eyelids of the participant and instruct not to move the eyes randomly.
3. Check the pupil is detected on the screen of the Host PC.

6 Materials Required for the Experiment

4. Record for a few seconds.
5. Take the average of this arbitrary unit to create a conversion factor.

Note that the arbitrary value can be influenced by the distance from the participant's eye to the camera, thresholds of the Pupil and Corneal Reflection. Please note these values for better comparison. In our experiment, we also compare the change in the arbitrary units while the participant changes. We have noted the distance from the eye to the camera and the threshold values. Figure 6.3 is a typical Host PC screen when we record the data. The values in the y-axis are the arbitrary units, and the x-axis is the time. On the right, you can see the Pupil and CR are given 'ok' with green colour highlighted, which means the pupil and CR detected. On the bottom left, we can see the artificial eyes that we attached to the left and right eyelids of the participant along with the threshold values. Further, we acquire this data in the form of a Matlab file to process the pupil data. Figure 6.4 is the plot of the raw data of a trial in Matlab. According to our data collection, the pupil size data was in the 45th (left eye) and 46th (right eye) channels.

Figure 6.3: Host PC screen while recording.



6 Materials Required for the Experiment

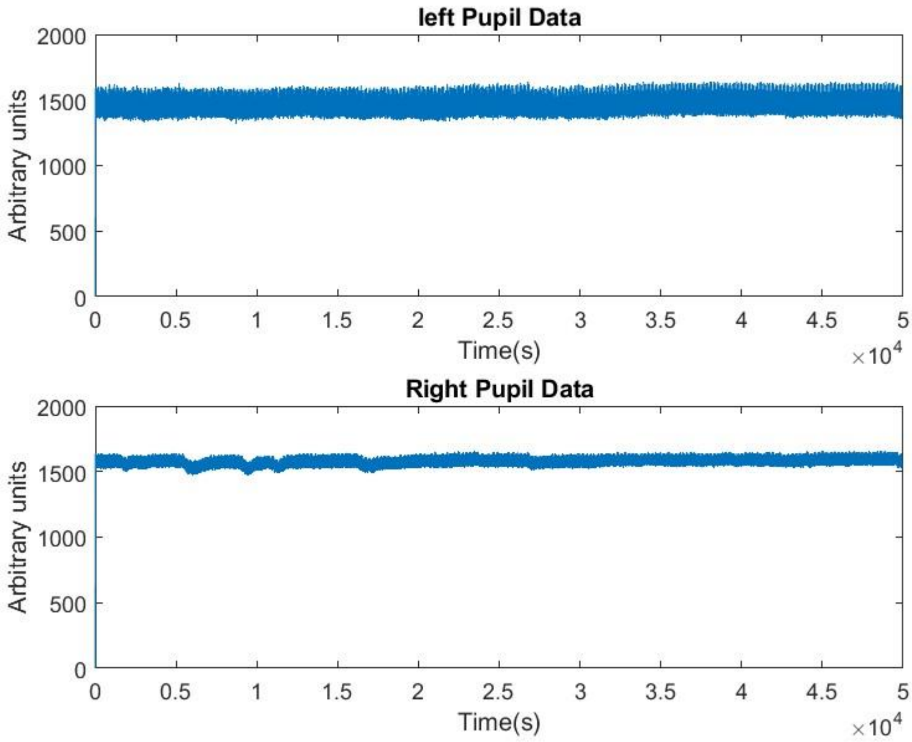


Figure 6.4: Raw data plotted in Matlab.

If an artificial pupil is used, we can find the conversion factor by deriving an equation from the area of a circle. The area of a circle is given in equation (1), from that equation we can compare the area of a known pupil average arbitrary value with the participant's pupil arbitrary value. If an artificial pupil of 'd' mm gives an arbitrary value of 1000 and a participant's pupil 'x' mm gives an arbitrary value of 800, formulating these values with the equation of area (3). Substitute the equation of the area and compare the arbitrary units we got accordingly, which is shown in equation (4)

$$\pi * \left(\frac{d}{2}\right)^2 \rightarrow 1000 \tag{4}$$

$$\pi * \left(\frac{x}{2}\right)^2 \rightarrow 800 \tag{5}$$

Comparing equations (4) and (5) we get a general equation (7)

$$\frac{d}{\sqrt{1000}} = \frac{x}{\sqrt{800}} \tag{6}$$

6 Materials Required for the Experiment

$$x = \sqrt{800} * \frac{d}{\sqrt{1000}} \qquad (7)$$

If we work with diameter, then we can skip the square root that we took for both sides. If we accidentally mix up the Area and Diameter in the samples. The formula (8) given is to convert area to diameter and the formula (9) is to convert diameter to area.

$$Pupil\ \_diameter = 256 * \frac{\sqrt{pupil_{area}}}{\pi} \qquad (8)$$

$$Pupil\_area = \pi * \left(\frac{\sqrt{pupil_{diameter}}}{256}\right)^2 \qquad (9)$$

Artificial illumination

We used artificial lights during the experiments since previous studies have suggested that images' light and dark properties can generate pupil-size changes of around the same magnitude as psycho-sensory changes and should be considered when employing eye-tracking experiments. When a bright light hits the eye, the pupil constricts and pupil size decreases. When it is dark the pupil dilates and the size increases. This reflex is called the Pupillary light reflex. To avoid this non-cognitive change, we introduced constant ambient illumination by using two coloured LED Video Lights with stands at both sides of the display screen shown in Figure 6.5. It has two types of lighting options, ‘I’ is the cold white and ‘II’ is the warm white colour. For the experiment, we used a cold white colour. Controlling luminance levels is not always possible, so we also used curtains to avoid ambient light. Pupil are normally somewhere in the middle of their range in typical light conditions.

6 Materials Required for the Experiment

Figure 6.5: Image of Artificial Illumination and the Curtains used.

7 Experimental Overview.

We conducted three experiments with different participants, collected data, and analysed them in Matlab. This section will explain the procedure that we followed during the experiments and the tasks that were involved during the experiment. Figure 7.1 is the whole set-up of the space where the participant sits. The golden rod above the chair is used to fix the head of the participant to avoid head movements.

Figure 7.1: Experimental Setup.

Experiment 1

In the first set of experiments, we had 22 participants with an age group between 20-30 years. The goal of this experiment was to understand the effect of threshold values and distance to the arbitrary unit with a pupil size of 7mm diameter, which is unchanged throughout the experiments. During the experiment, we fixed the participants' heads and told them to sit comfortably without moving their heads. We measured the distances from participants' eyes to the camera, camera to screen, and participants' eyes to the top of the screen, middle of the screen and bottom of the screen. We fed these values to the software which was explained in section 4. We also focused the eyes with the EyeLink camera and set the threshold values. After all these steps we calibrated and validated the participant's eyes by following the steps given in section 4. We used the area and centroid settings throughout the experiments. The participant was told to close their eyelids and we attached the artificial eyes to their eye. Instructed them not to move the eyeballs for 10 seconds. Recorded the data for at least 10 seconds and repeated one more time with the same participant for reproducibility, this time we kept the pupil inclined. Figure 7.2 is a picture of one of the participants with the artificial eyes attached, we took the picture with their consent.

7 Experimental Overview.

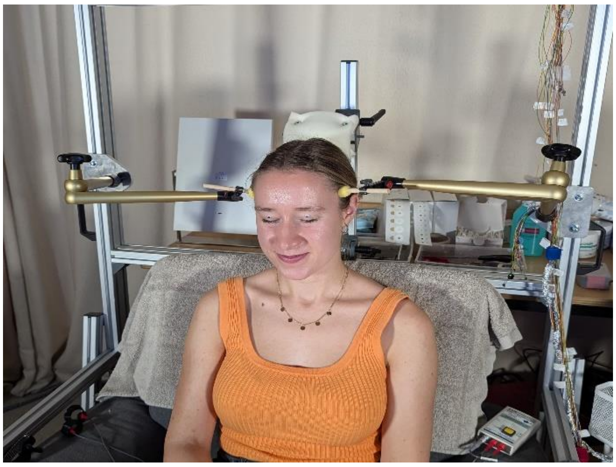


Figure 7.2: A participant with artificial eyes attached.

Experiment 2

In the 2nd experiment, we had 2 participants aged between 20 and 25. Here we used different-sized phantom eyes, having diameters of 3.5, 5, 6, 7, and 9 millimetres. Here the thresholds and the distance were also noted but the main focus is to see the effect of different pupil sizes in the arbitrary units. We followed all the steps that we followed in experiment 1 and the position of the participant was also the same, the only thing that changed was the phantom eyes. We recorded for 10 seconds for each eye after placing and repeated the measurement for reproducibility. In between the changing of the artificial eyes gave some break for the participant. Figure 7.3 is a picture of a participant during the experiment. The participant was told not to move the eyeballs. We used the area and centroid settings throughout the experiments, so when we plot, we convert the diameter into areas which are given in Table 6.1.

7 Experimental Overview.

Figure 7.3: A participant during the experiment.

Experiment 3

The third experiment was the continuation of the second experiment. In this experiment, only one participant was told to do some tasks with the different artificial eyes to analyse how the movement of the eyeballs affected the data. We deployed three types of movements, which were left-right, up-down and rotation. Figure 7.4 shows the document of the participant data with the pupil size, thresholds, distances, and movement illustrated with symbols. The participant is instructed to follow the order of the movement. Also in the document, we have glued the pupils which were cut open from the artificial eye to measure the actual diameter using a vernier calliper. The data was further analysed and the findings will be discussed below in the 4th section. We used the area and centroid settings throughout the experiments, so when we plot, we convert the diameter into areas which are given in Table 6.1.

7 Experimental Overview.

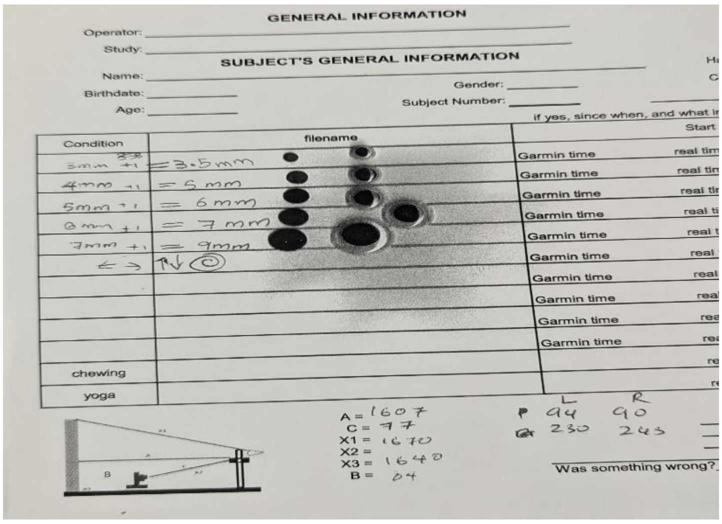


Figure 7.4: The physical data we measured.

8 Results

Before the analysis, we have to preprocess the raw data that we get from the device. This is done using Matlab signal processing. The sampling frequency of the data was 4800Hz, after plotting the raw data on Matlab Figure 6.4 we observed that the data was oversampled. So, the first step was down-sampling the data. Since the device sampling frequency is set to 500Hz we used the downsampling factor as 10. Matlab has an inbuild function to downsample the data which is y = downsample(x,n). Here the letter x denotes the data to be sampled and n denotes the downsampling factor which we choose is 10. Figure 8.1 compares the data before and after downsampling. The change is very visible in Figure 8.1.

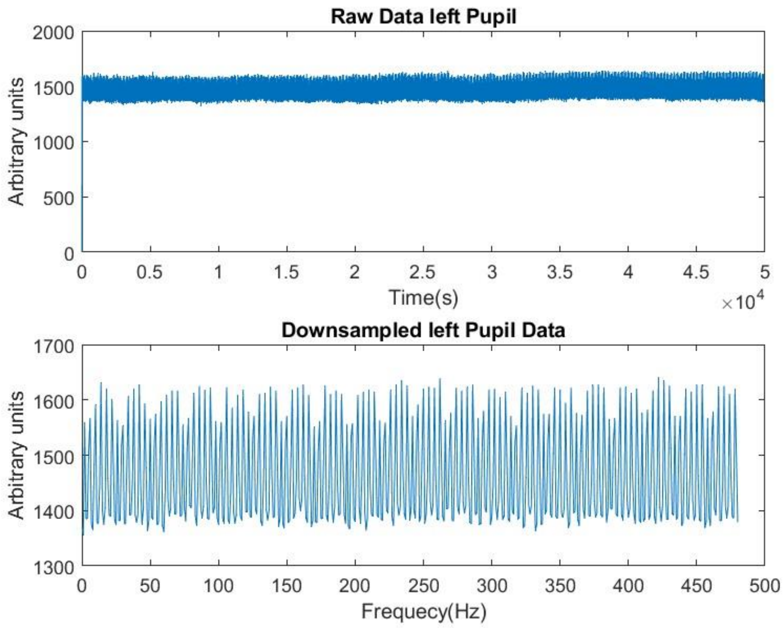


Figure 8.1: Comparison of raw and downsampled data.

The next step we did in the preprocessing was smoothing the data using a Matlab inbuild function called smoothdata which is given as B = smoothdata(**A**, method, window),. In the first space, we enter the data to be sampled, method refers to a different method of smoothing for example there are a few methods for smoothing data they are movement, move median, gaussian, lowes, loess, rlowess, rloess, sgolay. We used rloess, they are Robust quadratic regression over each window of A. This method is a more computationally expensive version of the method "loess", but it is more robust to outliers. The third parameter is to select window, for

the experiment, we took 0.1 times the downsampler frequency(450Hz). The next step of processing includes cutting or trimming the signal into a standard range of time. Here we took 1

8 Results

second for every data, so it will be more precise. Figure 8.2 shows the plot of the data after smoothing and trimming the samples to 1s. For trimming first, we analyse the raw data and identify stable data which lasts for 1 second. Find out the sample points and apply the sample while cutting.

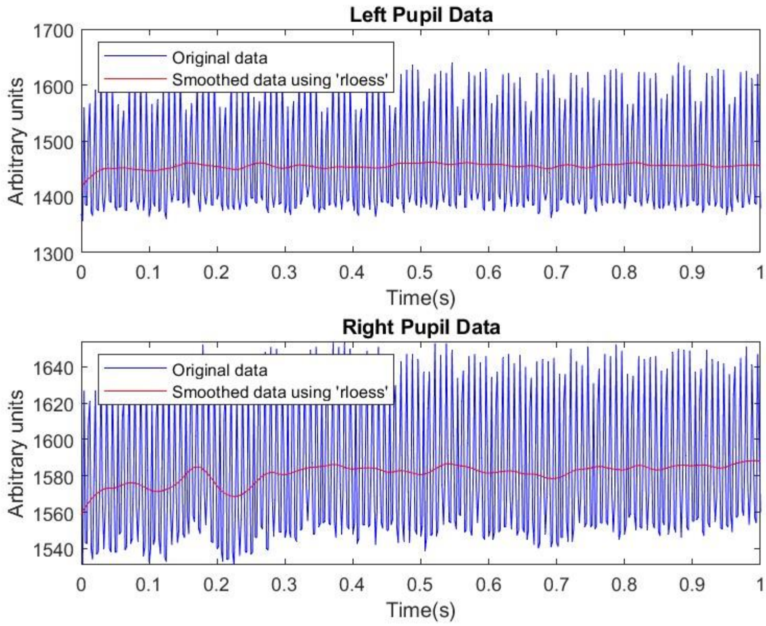


Figure 8.2: Data after applying smoothing and trimming.

Now the data is ready for analysis. Take the average of this section's data using a built-in function called mean M = mean(A), which returns the mean of the overall elements of A. Our main goal is to identify any relation between the Pupil- CR threshold and the arbitrary values. To do that we have to call all the values of each participant in the program and plot the average value of this sectioned data. For plotting we used 2D and 3D scattered plots since we have 3 parameters. The plot given in Figure 8.3, is a 2D scatter plot of CR vs arbitrary units and Pupil vs arbitrary units of left and right pupil. Here the pupil diameter is the same for all the samples, the change is the Pupil and CR thresholds. From the figure we can observe even if we have the same pupil diameter for every participant, the thresholds are not the same, this may depend on the height of the person, the physical properties of their eyes and the distance to the camera.

8 Results

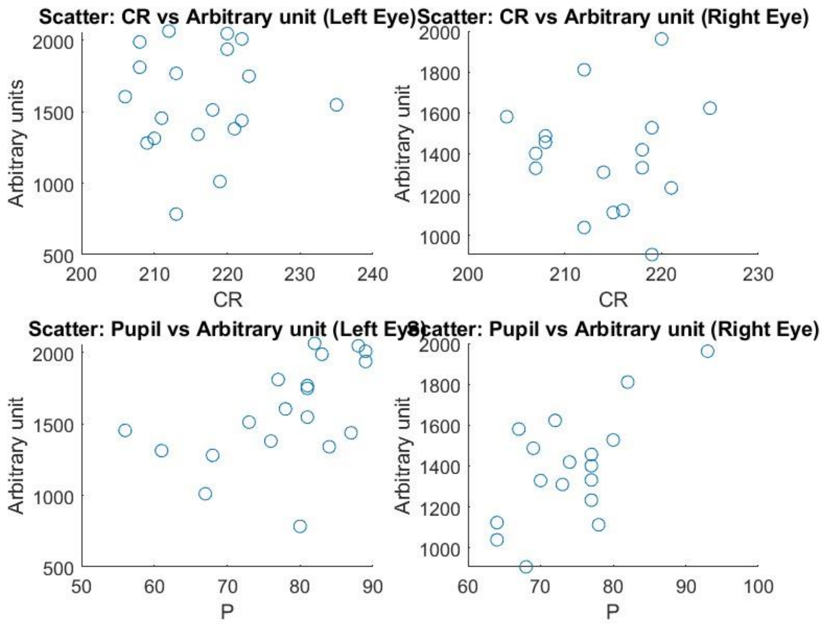


Figure 8.3: 2D Scattered Plot of Pupil/CR vs Arbitrary units.

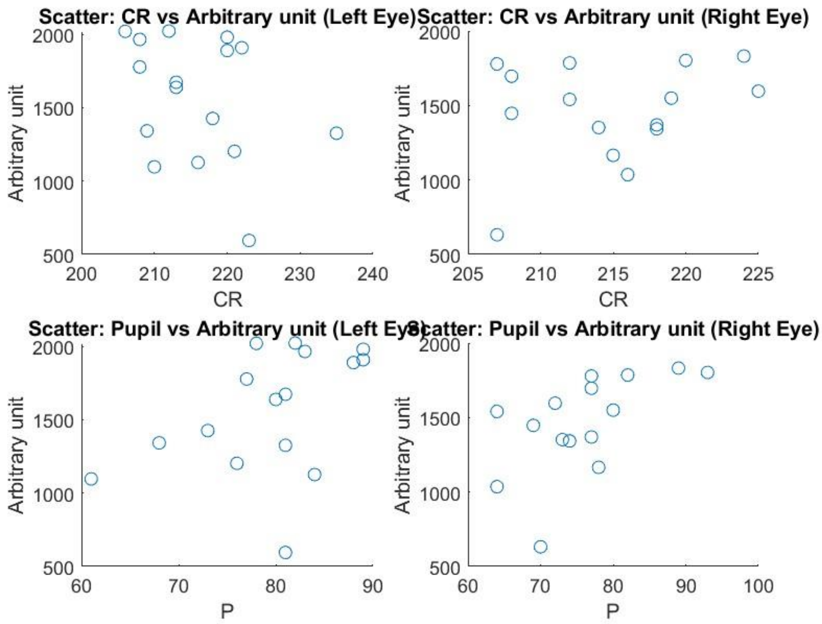


Figure 8.4: 2D Scattered Plot of Pupil/CR vs Arbitrary units of repeated data.

8 Results

To finalise the observation, we also repeated the measurements in the same participants. Figure 8.4 is a 2D scattered plot of the repeated measurements. After observing closely, we can say that the pupil vs arbitrary plots reflects some linear relationship. By observing two plots when the pupil threshold increases the arbitrary unit also tends to increase, that's how we got almost linear graphs. When looking at the Corneal Reflection thresholds it is very tough to predict a similar relation or some properties. Below Figure 8.5 concludes the relationship in one plot, 3D Scatter plots consisting of all three parameters.

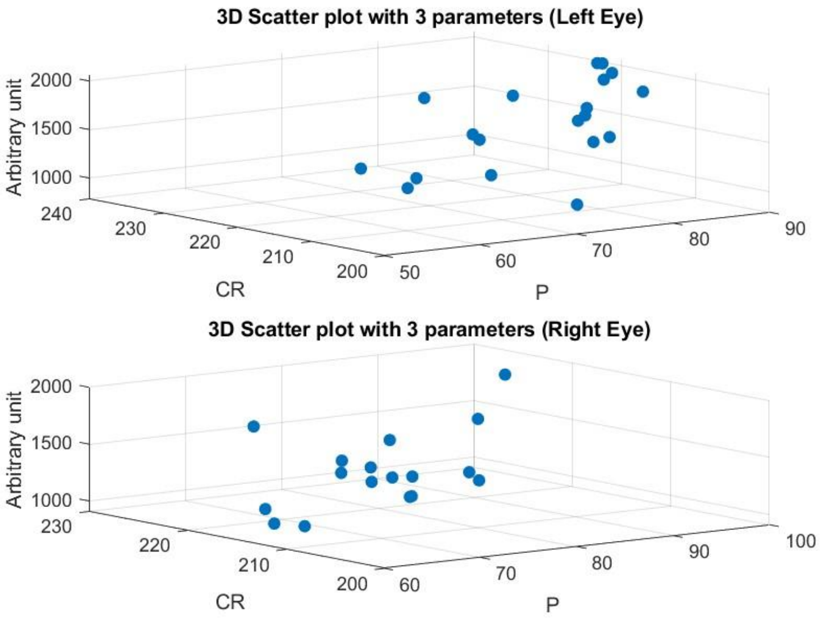


Figure 8.5: 3D Scattered Plot of Pupil vs CR vs Arbitrary Units of repeated data.

From Figure 8.5, we can observe a slight linear relation between all the values. For confirmation, we also plotted the repeated measurements, which are given in Figure 8.6 show a similar relation in the left eye, but we cannot predict the right eye also has the same after the observation. We can conclude after keen observations that when the pupil threshold increases the arbitrary units also increases. CR thresholds show no preferable relations with arbitrary units.

8 Results

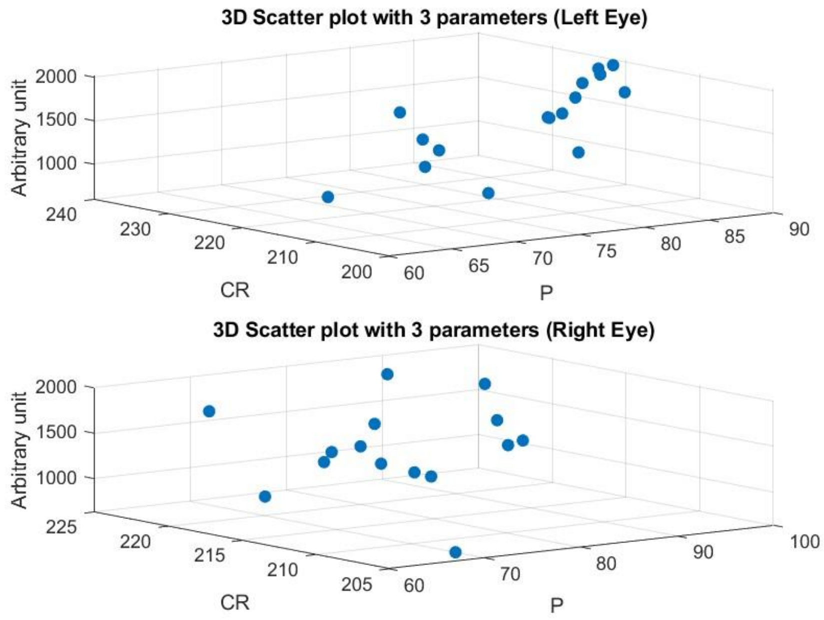


Figure 8.6: 2D Scattered Plot of Pupil vs CR vs Arbitrary units.

The next task is to plot the distance against arbitrary units to see if any major differences and their effect on the arbitrary unit. Figures 8.7 and 8.8 are the plots of the distance vs arbitrary unit. Here we assumed that we could see some notable differences but we couldn't observe any differences. It was showing an increase after 820mm but we cannot fully rely on this. Since the first sample itself shows varying results for every point. So, it is hard to come up with a constant conclusion for this result. But when you look at the data the pupil size was unchanged which was 7mm and when distance changes we get a different arbitrary unit. Both plots show similar results with different values for different distances. Neither it is decreasing when an increase in distance nor does it increase when an increase in distance. These were the two possibilities that we expected. Now we can see the analysis of the experiment 2. Here we had two participants, we noted their Pupil-CR threshold and distances.

8 Results

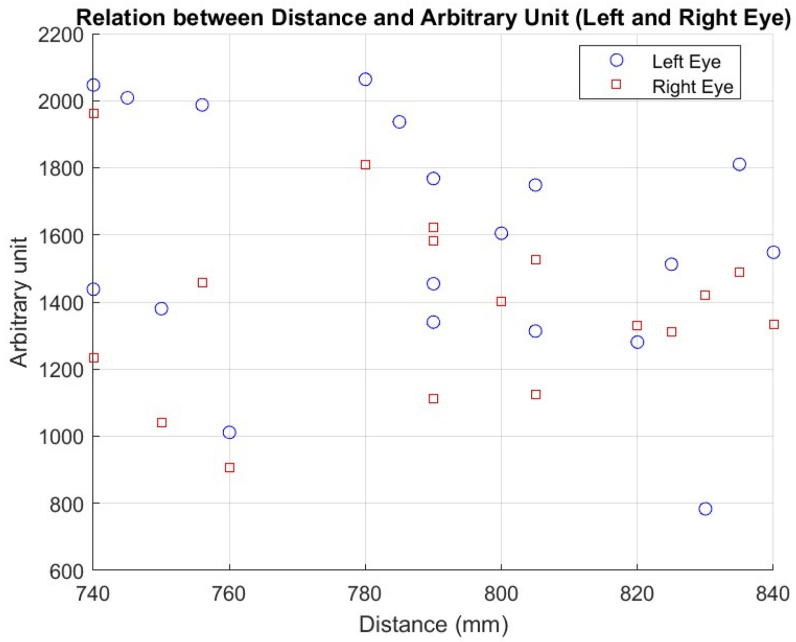


Figure 8.7: 2D Scattered Plot of distance vs Arbitrary units.

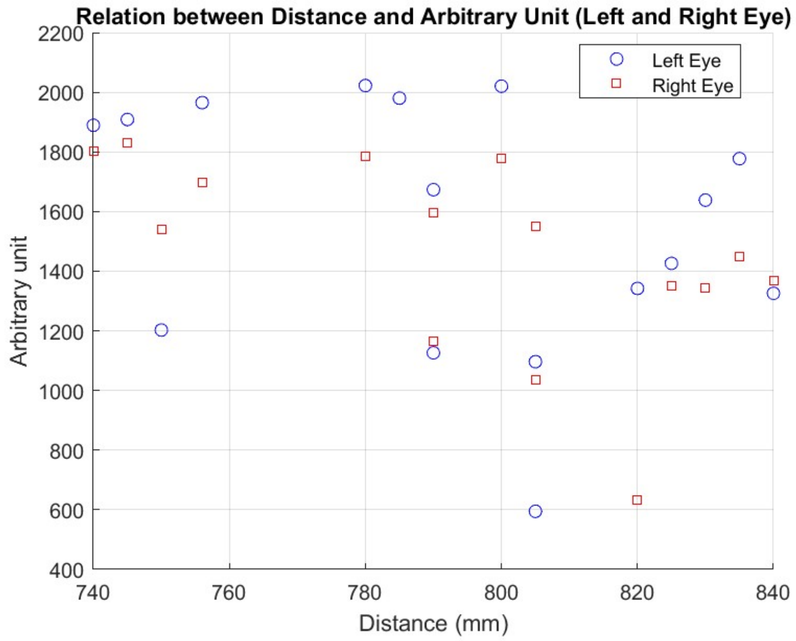


Figure 8.8: 2D Scattered Plot of Distance vs Arbitrary Units of repeated data.

8 Results

Figure 8.9 is the plot of different pupil size areas vs arbitrary units. The diameter of the Pupils is 3.5, 5, 6, 7and 9 mm which is given in table 6.1. the second plot in Figure 8.9 is the repeated measurement of the same participant with an inclined placement of the Pupils. Looking at the parameters like distance and thresholds, the participant eye to camera distance was 773mm, pupil threshold left was 120 and CR left was 227. Similarly, the pupil threshold right was 117 and CR was 228. Note that the Pupil-CR thresholds of both eyes should be similar for better results. A slight difference will not be affected too much, but the Pupil values for both eyes and CR for both eyes getting the same value can improve the accuracy.

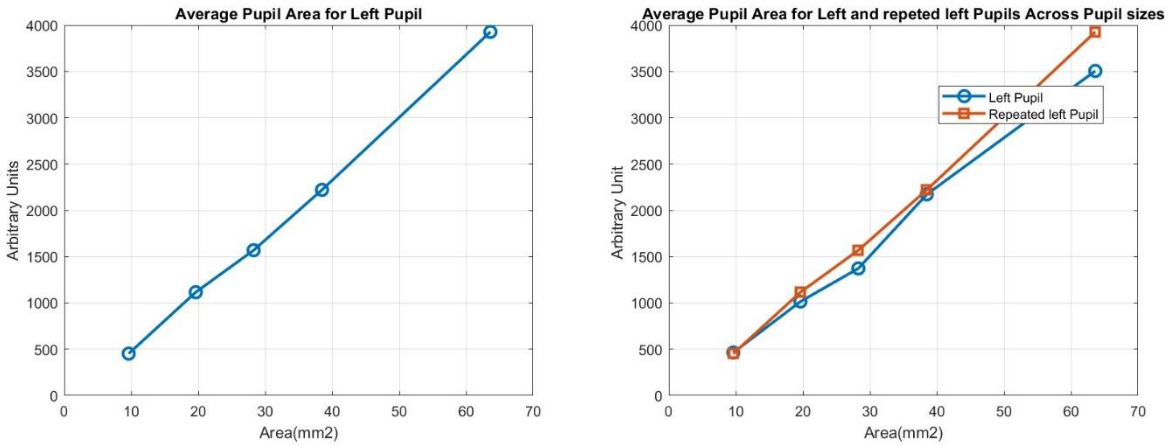


Figure 8.9: Line Plot of Different Pupil Areas in the left phantom eyes compared to two left phantom eyes data.

In Figure 8.9 we can observe a very prominent increase in the arbitrary value while the pupil size increases. In the first image, we can see the linear relation between the arbitrary unit and the pupil area. In the second image, we can see that the values slightly changed since we kept them in an inclined manner. From Figure 8.9 and Table 8.1, we can see they increased while we kept inclined. From this, we understood that when the pupil is in different positions it can affect the arbitrary units. When the pupil size increases then the arbitrary unit also increases. The pupil size of 7mm and the 9 mm has a drastic change. The increase is linear.

Pupil Diameter (mm)	Area (mm^2)	Arbitrary units		Arbitrary units inclined	
		Left	Right	Left	Right
3.5	9.6211	464	409	453	456
5	19.6350	1014	975	1115	1008
6	282743	1372	1355	1569	1435
7	38.4845	2172	2174	2221	1822
9	63.6173	3505	3179	3926	3438

Table 8.1: Pupil Area and Arbitrary unit normal and Inclined.

8 Results

For reproducibility, we also conducted the same experiment with different participants with a camera-eye distance of 770mm, the Pupil threshold left was 94 and CR left was 230. Similarly, the pupil threshold right was 90 and CR was 243. We observed similarities with the above plots so it is clear that when the pupil area increases the arbitrary unit also increases. Figure 8.10 and Table 8.2 represents the plot and the corresponding value is given in the Table.

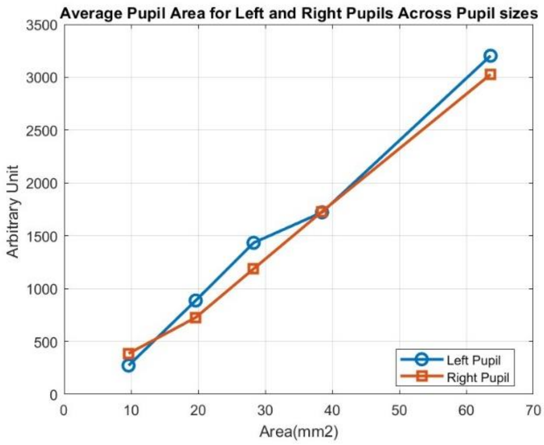


Figure 8.10: Line Plot of Different Pupil Areas vs Arbitrary Units.

s

Pupil Diameter (mm)	Area (mm^2)	Arbitrary units	
		Left	Right
3.5	9.6211	272	385
5	19.6350	887	726
6	282743	1433	1187
7	38.4845	1721	1729
9	63.6173	3202	3027

Table 8.2: Pupil Area and Arbitrary unit.

Compared to the above table 8.1 the values in this table 8.2, the values in 8.2 is decreased. It may be because of the change in participants and the parameters also changed accordingly. it is concluded that the arbitrary unit and pupil size have a linear relationship.

8 Results

After a close observation, we can implement an equation for the line we got. The general equation for the line or slope is (10).

$$Y = mx + c \quad (10).$$

Where:
m is the slope (rate of change of the arbitrary unit per mm^2 increase in pupil area),
c is the y-intercept (the value of the arbitrary unit when the pupil area is zero

Therefore, we can implement this mathematical expression into a linear regression model. Simply put, we can fit this data into equation (10). We use the polyfit(x,y,n) function which returns the coefficients for a polynomial p(x) of degree n that is the best fit (in a least-squares sense) for the data in y. We used polyval(p,x) and an inbuilt function to evaluate the polynomial p at each point in x.

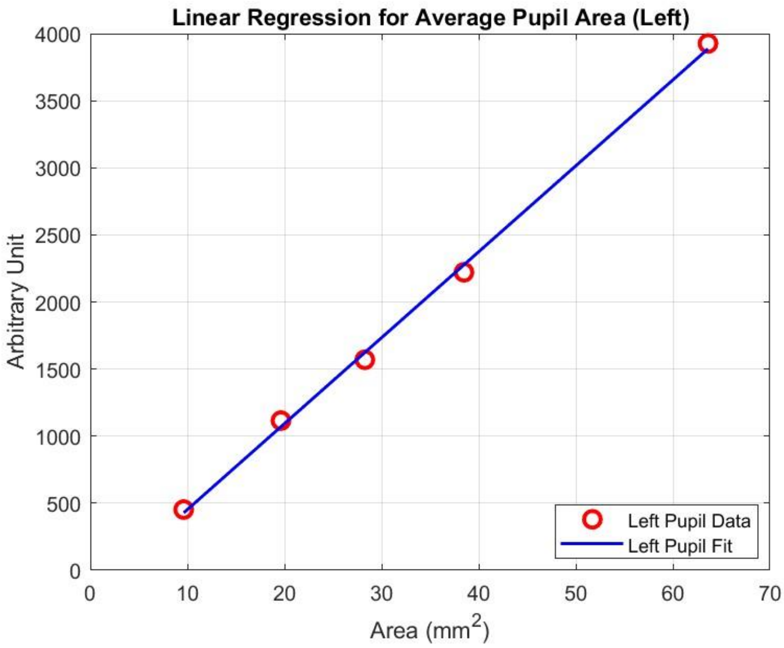


Figure 8.11: Linear Regression of Pupil Areas vs Arbitrary Units of the left eye.

Figure 8.11 illustrates a linear regression between the average pupil area on the x-axis and some measured responses on the y-axis labelled Arbitrary Units. The graph's red circles represent the measured data points for the left pupil's area. The corresponding arbitrary unit values are plotted on the y-axis. The blue line represents the linear fit to the data points, indicating a linear

8 Results

relationship between the pupil area and the arbitrary unit. The equation for the fitted line is provided as (11):

$$y = 64.0448 * x - 187.6555 \tag{11}$$

Where y is the arbitrary unit. x is the pupil area in mm^2. 64.0448 is the slope of the line, which means that for every additional square millimetre in pupil area, the arbitrary unit increases by about 64.04. 187.6555 is the y-intercept, representing the value of the arbitrary unit when the pupil area is zero. Based on this equation, we can also find the pupil size when we know the arbitrary units or vice versa. This shows a strong linear relationship between the pupil area and the measured output, with the response increasing significantly as the pupil area grows.

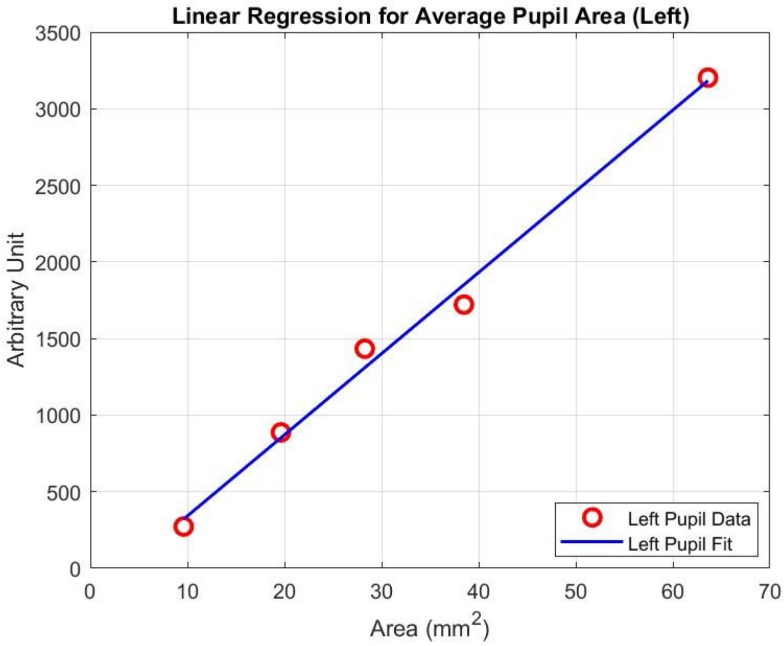


Figure 8.12: Linear Regression of Pupil Areas vs Arbitrary Units of left eye.

$$y = 52.9876x + -188.2565 \tag{12}$$

Equation (12) represents the linear regression line shown in Figure 8.12 of repetition measurement. This line is a statistical model that describes the relationship between the average pupil area (x) and an arbitrary unit (y). y Represents the predicted value of the arbitrary unit for

8 Results

a given average pupil area. 52.9876, This is the slope of the line. It indicates how much the arbitrary unit changes for every one-unit increase in the average pupil area. In other words, based on this model, if the average pupil area increases by 1 mm^2, the arbitrary unit is expected to increase by 52.9876 units. x Represents the average pupil area in mm^2. -188.2565 is the y-intercept of the line. It represents the predicted value of the arbitrary unit when the average pupil area is 0 mm^2. In this case, it's negative, suggesting that the arbitrary unit has a baseline value of -188.2565 when the pupil area is not measured. The regression line shows a positive relationship between the average pupil area and the arbitrary unit. As the average pupil area increases, the arbitrary unit also tends to increase. The slope of 52.9876 indicates that this increase is relatively steep. The equation provides a mathematical representation of the trend observed in the data. It allows you to estimate the arbitrary unit for a given average pupil area based on the linear relationship established by the regression analysis.

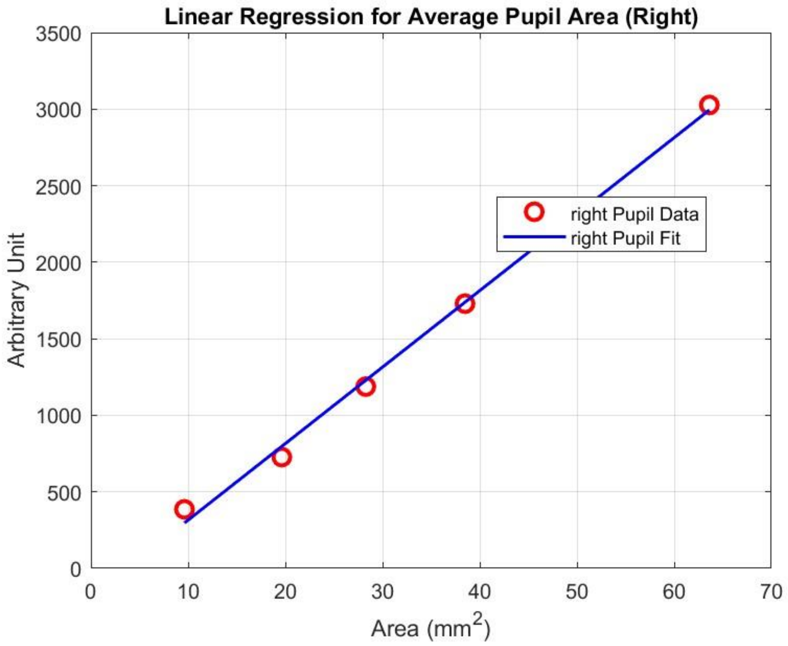


Figure 8.13: Linear Regression of Pupil Areas vs Arbitrary Units of right eye.

$$y = 49.9629x + -183.8841 \tag{13}$$

The equation (13) represents the linear regression line plotted in Figure 8.13 is the repetition measurements of another participant. This line is a statistical model that describes the relationship between the average pupil area (x) and an arbitrary unit (y). y Represents the

8 Results

predicted value of the arbitrary unit. 49.9629 This is the slope of the line. It indicates how much the arbitrary unit changes for every one-unit increase in the average pupil area. In other words, based on this model, if the average pupil area increases by 1 mm^2, the arbitrary unit is expected to increase by 49.9629 units. x Represents the average pupil area in mm^2. -183.8841 This is the y-intercept of the line. It represents the predicted value of the arbitrary unit when the average pupil area is 0 mm^2. In this case, it's negative, suggesting that the arbitrary unit has a baseline value of -183.8841 when the pupil area is not measured. Figure 8.13 visually confirms the linear relationship between the average pupil area and the arbitrary unit. The blue line (right pupil fit) represents this relationship. The positive slope of 49.9629 indicates that the arbitrary unit also increases as the average pupil area increases. This is visually evident in the upward slope of the line. The red dots (right pupil data) represent individual data points. The closeness of these points to the line suggests a strong linear relationship. This equation can be used to predict the arbitrary unit for any given average pupil area within the range of the data. For example, if the average pupil area is 50 mm^2, the predicted arbitrary unit would be:

$$y = 49.9629 * 50 - 183.8841$$
$$y \approx 2416.13$$

The equation can predict the arbitrary unit based on the average pupil area.

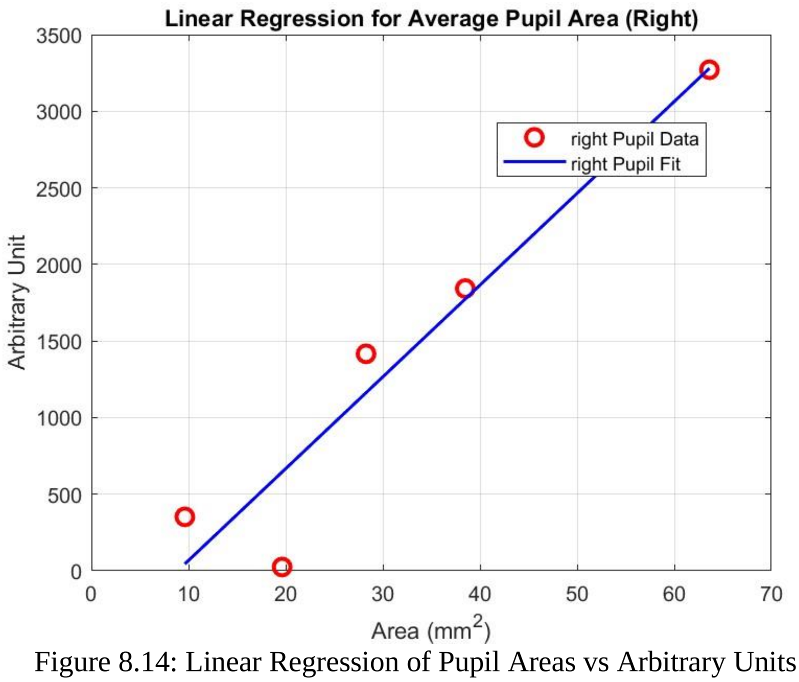


Figure 8.14: Linear Regression of Pupil Areas vs Arbitrary Units of the right eye.

$$y = 57.7948x + -455.4954 \tag{14}$$

8 Results

The equation (14) represents the linear regression line plotted in Figure 8.14 is the repetition measurement of the second participant. This line is a statistical model that describes the relationship between the average pupil area (x) and an arbitrary unit (y). y Represents the predicted value of the arbitrary unit. 57.7948 This is the slope of the line. It indicates how much the arbitrary unit changes for every one-unit increase in the average pupil area. In other words, based on this model, if the average pupil area increases by 1 mm^2, the arbitrary unit is expected to increase by 57.7948 units. x Represents the average pupil area in mm^2. -455.4954 This is the y-intercept of the line. It represents the predicted value of the arbitrary unit when the average pupil area is 0 mm^2. In this case, it's negative, suggesting that the arbitrary unit has a baseline value of -455.4954 when the pupil area is not measured. The figure visually confirms the linear relationship between the average pupil area and the arbitrary unit. The blue line (left pupil fit) represents this relationship. The positive slope of 57.7948 indicates that the arbitrary unit increases as the average pupil area increases. This is visually evident in the upward slope of the line. The red dots (left pupil data) represent individual data points. The closeness of these points to the line suggests a strong linear relationship. This equation can be used to predict the arbitrary unit for any given average pupil area within the range of the data. For example, if the average pupil area is 50 mm^2, the predicted arbitrary unit would be:

$$y = 57.7948 * 50 - 455.4954$$

$$y \approx 2834.74$$

The equation and the accompanying image show a positive linear relationship between the average pupil area and an arbitrary unit. Based on the average pupil area, the equation can predict the arbitrary unit.

8 Results

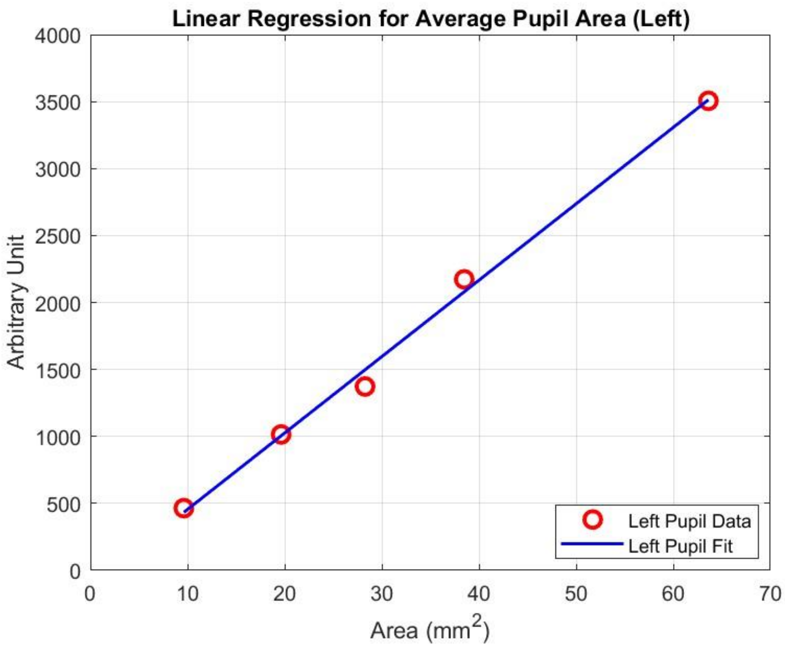


Figure 8.15: Linear Regression of Pupil Areas vs Arbitrary Units of the left eye.

$$y = 57.0339x + -114.9016 \qquad (15)$$

The provided equation, y = 57.0339x + -114.9016, represents a linear regression model that describes the relationship between the average pupil area (x) and a dependent variable (y) measured in arbitrary units. Y is the dependent variable (in arbitrary units). X is the independent variable, representing the average pupil area (in mm²). 57.0339 is the slope of the line, indicating the rate of change of y concerning x. In this context, it means that for every 1 mm² increase in pupil area, the dependent variable increases by 57.0339 units. -114.9016 is the y-intercept, represents the value of y when x is 0. In this case, it's the estimated value of the dependent variable when the pupil area is 0 mm². The provided Figure 8.15 visually illustrates this linear relationship. The red circles represent the actual data points, showing the observed pupil areas and their corresponding values of the dependent variable. The blue line is the linear regression line, fitted to the data points. This line represents the equation y = 57.0339x + -114.9016. The slope of the line (57.0339) determines the steepness and direction of the line. A positive slope indicates a positive relationship between the two variables, meaning that as the pupil area increases, the dependent variable also increases. The y-intercept (-114.9016) is the point where the line intersects the y-axis. It represents the predicted value of the dependent variable when the pupil

8 Results

area is zero. The equation and figure show a linear relationship between the average pupil area and the dependent variable. The equation provides a mathematical model to predict the value of the dependent variable for any given pupil area within the range of the data.

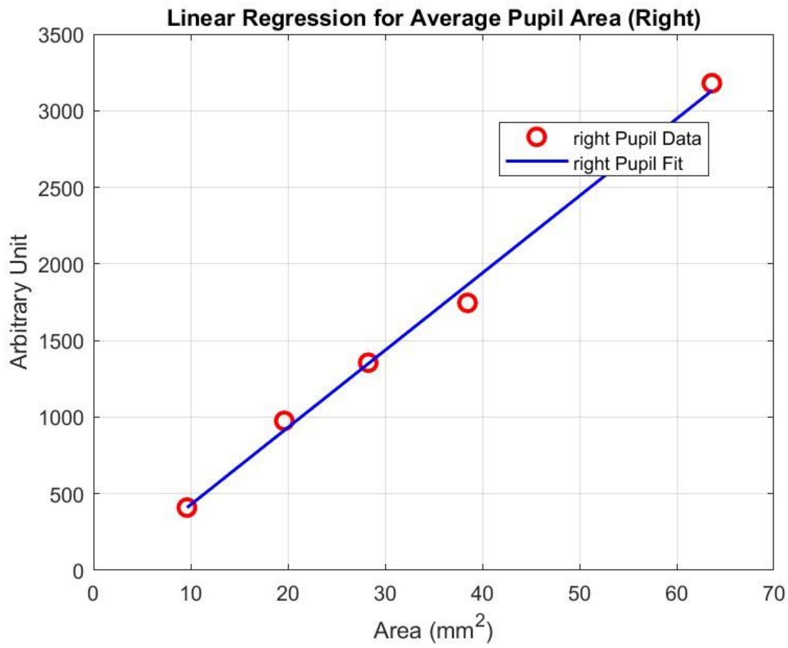


Figure 8.16: Linear Regression of Pupil Areas vs Arbitrary Units of the right eye.

$$y = 50.3986x + -75.9431 \tag{16}$$

The provided equation (16) represents a linear regression model that describes the relationship between the average right pupil area (x) and a dependent variable (y) measured in arbitrary units. y is the dependent variable (in arbitrary units). X is the independent variable, representing the average right pupil area (in mm²). 50.3986: The slope of the line, indicating the rate of change of y concerning x. In this context, it means that for every 1 mm² increase in the right pupil area, the dependent variable increases by 50.3986 units. -75.9431 is the y-intercept, representing the value of y when x is 0. In this case, it's the estimated value of the dependent variable when the right pupil area is 0 mm². The provided figure 8.16 visually illustrates this linear relationship. The red circles represent the actual data points, showing the observed right pupil areas and their corresponding values of the dependent variable. The blue line is the linear regression line, fitted to the data points. This line represents the equation (16). The slope of the line is 50.3986, which determines the steepness and direction of the line. A positive slope indicates a positive relationship between the two variables, meaning that as the right pupil area increases, the dependent variable also increases. The y-intercept is -75.9431 is the point where the line intersects the y-axis. It represents the predicted value of the dependent variable when the right pupil area is zero. The equation and figure show a linear relationship between the average right pupil area and the dependent variable. The equation provides a mathematical model to predict the value of the dependent variable for any given right pupil area within the range of the data.

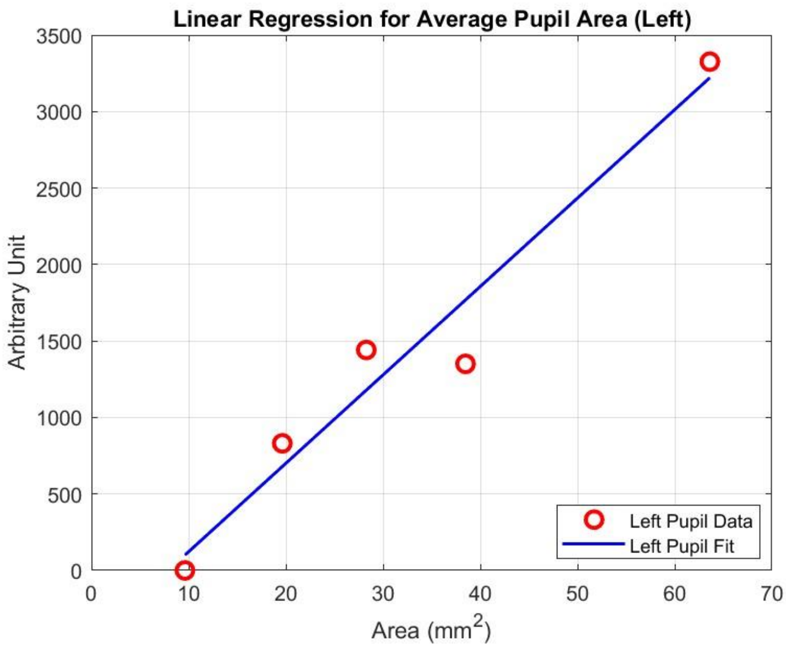


Figure 8.17: Linear Regression of Pupil Areas vs Arbitrary Units of the leftt eye.

$$y = 57.7948x + -455.4954 \tag{17}$$

The figure 8.17 shows a linear regression analysis of the relationship between the average pupil area (right) and an arbitrary unit. The blue line represents the regression line, which is a straight line that best fits the data points. The equation of the regression line (17), where y is the arbitrary unit and x is the average pupil area (left). The slope of the regression line is 57.7948 indicating the rate at which the arbitrary unit changes concerning the average pupil area (right). In other

8 Results

words, for every 1 mm² increase in the average pupil area (right), the arbitrary unit increases by 57.7948 units. The y-intercept of the regression line is -455.4954, which indicates the value of the arbitrary unit when the average pupil area (left) is 0. However, it is important to note that this value may not have a meaningful interpretation in this context, as it is unlikely that the average pupil area (leftt) would ever be 0.

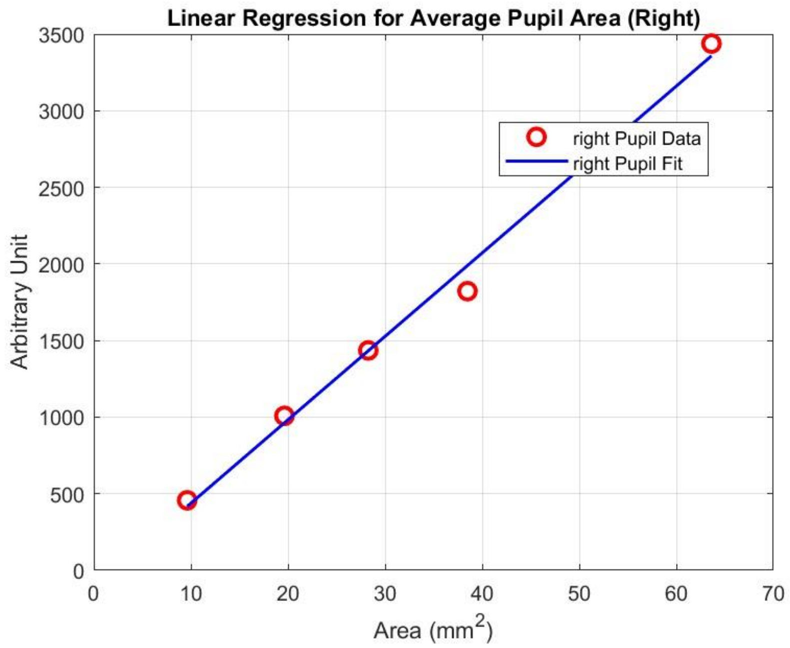


Figure 8.18: Linear Regression of Pupil Areas vs Arbitrary Units of the right eye.

$$y = 54.4538x + -106.4333 \quad (18)$$

The figure 8.18 shows a linear regression analysis of the relationship between the average pupil area (right) and an arbitrary unit. The blue line represents the regression line, which is a straight line that best fits the data points. The equation of the regression line is (18), where y is the arbitrary unit and x is the average pupil area (right). The slope of the regression line is 54.4538, which indicates the rate at which the arbitrary unit changes concerning the average pupil area (right). In other words, for every 1 mm² increase in the average pupil area (right), the arbitrary unit increases by 54.4538 units. The y-intercept of the regression line -106.4333 which indicates the value of the arbitrary unit when the average pupil area (right) is 0. However, it is important to note that this value may not have a meaningful interpretation in this context, as it is unlikely that the average pupil area (right) would ever be 0.

8 Results

The plots discuss multiple linear regression analyses involving pupil areas of the left and right eyes against measured arbitrary units. These analyses follow a consistent pattern where the pupil area serves as the independent variable (x), and the arbitrary unit is the dependent variable (y). Key equations are derived to express the relationship between pupil area and arbitrary units.
Each regression line equation takes the form (10). Multiple regression equations are presented for different sets of data (left eye, right eye, and repeated measurements). Each indicates a positive linear relationship between the pupil area and arbitrary units. This means that as the pupil area increases, the arbitrary unit tends to increase. The slope values range from approximately 49 to 64, indicating how sensitive the arbitrary unit is to changes in the pupil area. For example, in one case, an increase of 1 mm^2 in the pupil area increases 64.04 arbitrary units. The y-intercepts are often negative, ranging from -75 to -455. This might suggest that the arbitrary unit has a baseline below zero when the pupil area is not measured, although a pupil area of zero may not have real-world significance. Visual representations (graphs) suggest a strong correlation between the pupil area and arbitrary units, as the data points closely follow the regression line, indicating a good fit for the model. The regression equations provide a tool for predicting arbitrary unit values based on given pupil areas. For example, using one of the equations, if the pupil area is 50 mm^2, the arbitrary unit can be estimated with reasonable accuracy.

The results provide a robust analysis using linear regression to model the relationship between pupil areas and arbitrary units. The use of standard linear regression techniques (like `polyfit` and `polyval`) and interpretation of the regression slopes and intercepts are correctly applied. The findings demonstrate strong linear relationships across multiple sets of data, implying that pupil area is a reliable predictor of the arbitrary units measured. One observation to consider is the practicality of negative y-intercepts. While they mathematically make sense, their real-world interpretation might require careful consideration. The report could further explore whether these baseline values (at zero pupil area) are meaningful or just a result of the regression fitting process. The statistical models presented are valuable for making predictions but are limited to the range of data presented. Extrapolation beyond the measured pupil area might yield less reliable predictions.

8 Results

The next task was to move the eyes with the attached artificial eyes. Figure 8.12 shows the data samples while the participant moves their eyes. Moving eyes to the left-right, up-down and rotating are the moving tasks that the participant had to do for 3 sets each. When analysing the plot, we can say that the first exercise starts with a stable value for the arbitrary value, the task was moving the eyes left and right. The second exercise started after 10 seconds we can see a small transition there and the second exercise was up-down movement. It shows a similar pattern shown in the first exercise. Both patterns had a small variation not continuing steadily. The third task is the one which brings an entire difference to the whole plot. The movement is circularly rotating the eyes. It lasts longer since it has to cover the 360 degrees. Here we can observe the movement is bringing a remarkable change in the plot. Almost from the 16 seconds it started and took 24 seconds to complete the 3 sets. Here we can see some increase and decrease in the plot, which implies while rotating the eyes the arbitrary value can be affected. By observing different data, we could see similar patterns in the same task, thus we concluded that when we change the eye position the arbitrary value can be changed and it has an impact on the overall experiment.

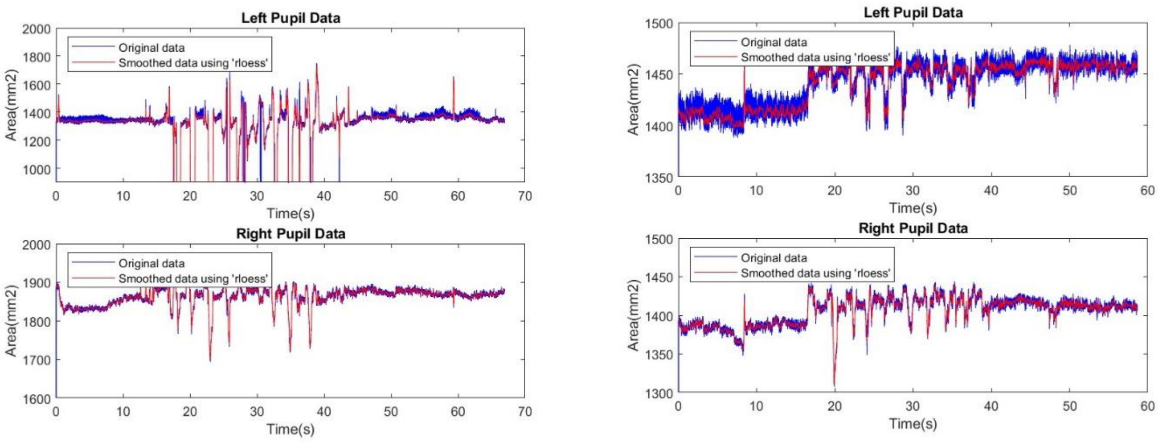


Figure 8.19: Plots of the moving tasks.

9 Discissions

The results of the experiments conducted with the EyeLink 1000 Plus system provide significant insights into both the strengths and limitations of its calibration process and data analysis capabilities. The primary finding of a linear relationship between pupil size and arbitrary units recorded by the system indicates that the EyeLink 1000 Plus is highly sensitive to changes in pupil size. This is an essential aspect for applications where pupil dilation is used as an indicator of cognitive load, emotional state, or attentional shifts. The linearity observed means that for consistent experimental conditions, changes in pupil size can be accurately tracked and interpreted.

However, some challenges were encountered during the calibration process, particularly regarding the Corneal Reflection (CR) thresholds. Unlike pupil size, CR thresholds did not exhibit a consistent or predictable relationship with the arbitrary units, making the calibration less reliable in some cases. This could be attributed to various factors, such as individual differences in eye anatomy, movement of the participant’s head, or even slight variations in the positioning of the equipment. Since CR is a key reference point for tracking eye movements, the variability in CR measurements highlights the need for more robust and adaptable calibration procedures. Addressing these inconsistencies will be crucial for improving the accuracy of gaze tracking, especially in studies requiring precise spatial resolution.

Moreover, the use of phantom eyes with different pupil diameters was effective in confirming that the system can accurately reflect changes in pupil size. The controlled environment with constant illumination reduced external interference, ensuring that the measured changes were primarily due to the varying pupil sizes. However, one notable observation was the influence of eye movements, particularly rotational movements, on the accuracy of measurements. Even slight deviations from a fixed gaze resulted in significant fluctuations in arbitrary unit values. This suggests that in real-world applications, where participants are more likely to move their eyes naturally, maintaining strict control over eye stability during calibration is crucial.

The linear regression analysis conducted on the pupil size data further supports the reliability of the EyeLink 1000 Plus system in detecting changes in the pupil area. The strong linear relationship, as demonstrated by the regression equation, shows that the system produces predictable and consistent measurements of arbitrary units in response to varying pupil sizes. This linearity is particularly valuable for experiments that depend on precise quantification of pupil dilation, such as those investigating cognitive load or emotional responses. The accuracy of the linear fit confirms that the EyeLink system can be relied upon for tracking pupil size under controlled conditions, despite the challenges posed by other variables like corneal reflection and eye movements.

10 Conclusions and Future Work

This project has successfully demonstrated the functionality and capabilities of the EyeLink 1000 Plus system for eye-tracking research, particularly in monitoring pupil size as a proxy for cognitive and emotional states. The system was able to consistently track pupil size across different experimental conditions, affirming the linear relationship between pupil area and the arbitrary units used by the system for data representation. These findings are promising for research areas such as cognitive psychology, neurophysiology, and usability studies, where pupil size changes can provide crucial insights into participants' cognitive load and attention.

However, the study also uncovered areas where improvements are needed. The calibration process, though effective, sometimes produced inconsistent results when it came to adjusting the CR threshold Since CR serves as a vital reference for correcting head movements and ensuring accurate gaze tracking, this limitation could affect the precision of data in experiments where high spatial accuracy is required. Additionally, the influence of eye movements on arbitrary unit values underlines the importance of participant compliance during calibration and data collection phases. These challenges need to be addressed to fully unlock the potential of the EyeLink 1000 Plus system.

In conclusion, the linear regression analysis conducted as part of this study adds further validation to the functionality of the EyeLink 1000 Plus system, particularly in its ability to track pupil size with a high degree of accuracy. The clear linear relationship between pupil area and the arbitrary units used by the system provides a robust method for interpreting changes in pupil size, which is critical in applications like cognitive psychology, neurophysiology, and emotional state analysis. This linearity means that the system can reliably detect even small changes in pupil size, making it a valuable tool for studies involving cognitive load and attention shifts.

Moving forward, several key areas should be the focus of future research and system improvements. First, refining the calibration procedure is essential, particularly with the Corneal Reflection (CR) measurements. One possible avenue of improvement is the development of more adaptive algorithms that can account for individual differences in eye structure, head movement, and positioning. These algorithms could dynamically adjust the calibration settings in real-time, reducing the variability seen in CR thresholds and improving the overall accuracy of the system.

Another area for future research is the effect of external lighting conditions on the accuracy of pupil size measurements. While the current study controlled for lighting by using artificial illumination, real-world environments are rarely as consistent. Investigating how variations in

10 Conclusions and Future Work

ambient light affects pupil measurements and developing compensatory algorithms to normalize these effects could significantly enhance the versatility of the EyeLink 1000 Plus system in more diverse settings.

In addition, further exploration of the impact of natural eye movements (such as saccades, blinks, and rotations) on the system's measurements is warranted. Eye movements are inevitable in most research scenarios, and developing methods to compensate for these movements would make the system more robust. For example, integrating head-tracking systems or using advanced machine learning models to predict and adjust for eye movements in real time could help maintain the accuracy of data even during more dynamic experiments.

Enhancing the user-friendliness of the calibration process should be a priority. Many first-time users may struggle with the complexity of the current setup, leading to longer setup times and potential errors. Simplifying the calibration workflow and automating certain aspects of the process could reduce the burden on the researcher and make the system more accessible to a wider range of users. Additionally, developing more intuitive user interfaces or providing real-time feedback during the calibration process could further minimize errors and improve efficiency.

Exploring non-linear models or incorporating machine learning algorithms that adapt to real-time changes might provide even more accurate predictions of pupil size across diverse scenarios.Moreover, integrating the linear regression models directly into the system's feedback mechanisms during real-time tracking could improve both data quality and usability. This would allow the system to dynamically adjust based on the relationship between pupil area and arbitrary units, potentially increasing the precision of measurements even in less controlled environments.

By refining the use of linear regression alongside other system improvements, future research can not only enhance the predictive capabilities of the EyeLink 1000 Plus system but also expand its application across a wider range of experimental setups.

.

Appendix

 # Program for Experiment 1.

```
clear all;                    % clears all the variables which are loaded in the workspace
close all;                    % closes all the plots which are opened previously
clc;                          % clears all the commands in the command window

data = load('s05_2.mat');        % here 's05_2.mat' change the data

 figure                       % used to plot the graph in a figure
 subplot(2,1,1);                % this function plots the graph in 1st row of 2*1
 plot(data.y(45,:));            % this function plot the data in the channel 45
 xlabel('Time(s)');             % function used to label the x axis
 ylabel('Arbitrary Unit');       % function used to label the x axis
 title('left Pupil Data Before');% function used to give a title for the graph

 subplot(2,1,2);                % this function plots the graph in 2nd row of 2*1
 plot(data.y(46,:));            % this function plot the data in the channel 46
 xlabel('Time(s)');
 ylabel('Arbitrary Unit');
 title('Right Pupil Data Before');
 %%

 pupil_size_L  =45;              % here write down tha channel mumber for left pupil size

 left_pupil    = data.y(pupil_size_L,30240:35040);%here trims left pupil size data to 1s

 pupil_size_R  =46;              % here write down tha channel mumber for right pupil size

 right_pupil   = data.y(pupil_size_R,30240:35040);%here trims right pupil size data to 1s

fs            = 4800;           %here write down the sammpling srequency

downsample_factor = 10;           %here write down the downampling rate of 10

left_pupil_d   = downsample(left_pupil, downsample_factor); %here applies downsamling to the trimed data of left pupil

right_pupil_d  = downsample(right_pupil, downsample_factor);%here applies downsamling to the trimed data of right pupil

fs_downsampled = fs / downsample_factor;%here updating the sampling frequency after down sampling

t = (0:length(left_pupil_d)-1) / fs_downsampled;%here updated the time period for ploting the data
```

Appendix

```
smooth_left_d  = smoothdata(left_pupil_d, 'rloess', 0.1 * fs_downsampled); %here we use smoothing
function tp preprocess the left pupil data


smooth_right_d = smoothdata(right_pupil_d, 'rloess', 0.1 * fs_downsampled);%here we use smoothing
function to preprocess the right pupil data

figure
subplot(2,1,1);
plot(t, left_pupil_d, 'b', t, smooth_left_d, 'r');% this function used to plot the left pupil data against the
smoothed left pupil data
legend('Original data', 'Smoothed data using "rloess"', 'Location', 'NW'); % this function used to indicate
the contents merged in the graph
xlabel('Time(s)');
ylabel('Arbitrary Unit');

title('Left Pupil Data Before');

subplot(2,1,2);
plot(t, right_pupil_d, 'b', t, smooth_right_d, 'r');% this function used to plot the right pupil data against
the smoothed right pupil data
legend('Original data', 'Smoothed data using "rloess"', 'Location', 'NW');
xlabel('Time(s)');
ylabel('Arbitrary Unit');
title('Right Pupil Data Before');


Area_left = mean(smooth_left_d);  % here we use mean function to get the average value of left pupil
Area_right = mean(smooth_right_d);% here we use mean function to get the average value of right pupil

disp(['Mean of smoothed left pupil data before: ', num2str(Area_left)]);  % this function used to display
the average value which is calculated previously for the left
disp(['Mean of smoothed right pupil data before: ', num2str(Area_right)]);% this function used to display
the average value which is calculated previously for the right


%%
d          = zeros(1,22);       % creates a 1*22 array with zeros to store distance value of 22 participants
p_L        = zeros(1,22);     % creates a 1*22 array with zeros to store left pupil threshold value of 22
participants
cr_L       = zeros(1,22);     % creates a 1*22 array with zeros to store left CR threshold value of 22
participants
p_R        = zeros(1,22);     % creates a 1*22 array with zeros to store right pupil threshold of 22
participants
```

Appendix

```
cr_R        = zeros(1,22);     % creates a 1*22 array with zeros to store right CR threshold value of 22
participants
Area_L      = zeros(1,22);   % creates a 1*22 array with zeros to store Left arbitrary unit of 22
participants
Area_R      = zeros(1,22);   % creates a 1*22 array with zeros to store right arbitrary unit of 22

participants
Area2_L     = zeros(1,22);       % creates a 1*22 array with zeros to store left repeated arbitrary unit of
22 participants
Area2_R     = zeros(1,22);       % creates a 1*22 array with zeros to store right repeated arbitrary unit of
22 participants
d(1)        = 790;             % this is how we enter distance to 1st column.
p_L (1)     = 56;              % this is how we enter left pupil threshold to 1st column.
cr_L (1)    = 211;             % this is how we enter left CR threshold to 1st column.
Area_L(1)   = Area_left;       % this is how we enter left arbitrary unit to 1st column.
Area_R(1)   = Area_right;      % this is how we enter right arbitrary unit to 1st column.
p_R(1)      = 67;              % this is how we enter right pupil threshold to 1st column.
cr_R(1)     = 204;             % this is how we enter right CR threshold to 1st column.

save('pupil_data.mat', 'd', 'p_L', 'cr_L','Area_L','p_R','cr_R','Area_R','Area2_R','Area2_L'); % Here saves
all the values we entered in a file and we call this file in every participant program file.
```

 # Program for Experiment 2.

```
clear all;                                        % helps to clear all the datas in the workspace before
starting the program
close all;                                        % helps to close all the plots
clc;                                              % helps to clear commands in the command window

% Load data
data4          = load('s4in.mat');                       % Here we can load the recorded data by using
load function
data5          = load('s5in.mat');
data6          = load('s6in.mat');
data7          = load('s7in.mat');
data8          = load('s8in.mat');

fs             = 4800;                            % Sampling frequency
downsample_factor = 10;                           % Factor to downsample the data
fs_downsampled    = fs / downsample_factor;              % Downsampled frequency

% Initialize arrays to store the areas for each subject
```

Appendix

```
Area_left        = zeros(1, 5);                                        % 1x5 array for left pupil areas
Area_right       = zeros(1, 5);                                        % 1x5 array for right pupil areas

% Process Pupil Size 4
left_pupil4      = data4.y(45, 4800:9600);                             % Extract data for left pupil
right_pupil4     = data4.y(46, 38400:43200);                           % Extract data for right pupil

left_pupil_d4    = downsample(left_pupil4, downsample_factor);         % Downsample left pupil
data
right_pupil_d4   = downsample(right_pupil4, downsample_factor);        % Downsample right pupil
data
smooth_left4     = smoothdata(left_pupil_d4, 'rloess', 0.1 * fs_downsampled); % Smooth left pupil data
smooth_right4    = smoothdata(right_pupil_d4, 'rloess', 0.1 * fs_downsampled);% Smooth right pupil
data
Area_left(1)     = mean(smooth_left4);                                 % Calculate and store mean area for left
pupil (Pupil Size 4)
Area_right(1)    = mean(smooth_right4);                                % Calculate and store mean area for
right pupil (Pupil Size 4)

% Process Pupil Size 5
left_pupil5      = data5.y(45, 33600:38400);                           % Extract data for left pupil
right_pupil5     = data5.y(46, 33600:38400);                           % Extract data for right pupil
left_pupil_d5    = downsample(left_pupil5, downsample_factor);         % Downsample left pupil
data
right_pupil_d5   = downsample(right_pupil5, downsample_factor);        % Downsample right pupil
data
smooth_left5     = smoothdata(left_pupil_d5, 'rloess', 0.1 * fs_downsampled); % Smooth left pupil data
smooth_right5    = smoothdata(right_pupil_d5, 'rloess', 0.1 * fs_downsampled);% Smooth right pupil
data
Area_left(2)     = mean(smooth_left5);                                 % Calculate and store mean area for left
pupil (Pupil Size 5)
Area_right(2)    = mean(smooth_right5);                                % Calculate and store mean area for
right pupil (Pupil Size 5)

% Process Pupil Size 6
left_pupil6      = data6.y(45, 38400:43200);                           % Extract data for left pupil
right_pupil6     = data6.y(46, 38400:43200);                           % Extract data for right pupil
left_pupil_d6    = downsample(left_pupil6, downsample_factor);         % Downsample left pupil
data
right_pupil_d6   = downsample(right_pupil6, downsample_factor);        % Downsample right pupil
data
smooth_left6     = smoothdata(left_pupil_d6, 'rloess', 0.1 * fs_downsampled); % Smooth left pupil data
smooth_right6    = smoothdata(right_pupil_d6, 'rloess', 0.1 * fs_downsampled);% Smooth right pupil
data
```

Appendix

Area_left(3)   = mean(smooth_left6);   % Calculate and store mean area for left pupil (Pupil Size 6)
Area_right(3)   = mean(smooth_right6);   % Calculate and store mean area for right pupil (Pupil Size 6)

% Process Pupil Size 7
left_pupil7   = data7.y(45, 9600:14400);   % Extract data for left pupil
right_pupil7   = data7.y(46, 56160:60960);   % Extract data for right pupil
left_pupil_d7   = downsample(left_pupil7, downsample_factor);   % Downsample left pupil data
right_pupil_d7   = downsample(right_pupil7, downsample_factor);   % Downsample right pupil data
smooth_left7   = smoothdata(left_pupil_d7, 'rloess', 0.1 * fs_downsampled); % Smooth left pupil data
smooth_right7   = smoothdata(right_pupil_d7, 'rloess', 0.1 * fs_downsampled);% Smooth right pupil data
Area_left(4)   = mean(smooth_left7);   % Calculate and store mean area for left pupil (Pupil Size 7)
Area_right(4)   = mean(smooth_right7);   % Calculate and store mean area for right pupil (Pupil Size 7)

% Process Pupil Size 8
left_pupil8   = data8.y(45, 28800:33600);   % Extract data for left pupil
right_pupil8   = data8.y(46, 19200:24000);   % Extract data for right pupil
left_pupil_d8   = downsample(left_pupil8, downsample_factor);   % Downsample left pupil data
right_pupil_d8   = downsample(right_pupil8, downsample_factor);   % Downsample right pupil data
smooth_left8   = smoothdata(left_pupil_d8, 'rloess', 0.1 * fs_downsampled); % Smooth left pupil data
smooth_right8   = smoothdata(right_pupil_d8, 'rloess', 0.1 * fs_downsampled);% Smooth right pupil data
Area_left(5)   = mean(smooth_left8);   % Calculate and store mean area for left pupil (Pupil Size 8)
Area_right(5)   = mean(smooth_right8);   % Calculate and store mean area for right pupil (Pupil Size 8)

% Display or use Area_left and Area_right as needed
disp(Area_left);   % displays the average left arabitrary unit
disp(Area_right);   % displays the average right arabitrary unit
d1 = 3.5;   % enters the diameter1 of the artificial eyes
d2 = 5;   % enters the diameter2 of the artificial eyes
d3 = 6;   % enters the diameter3 of the artificial eyes
d4 = 7;   % enters the diameter4 of the artificial eyes
d5 = 9;   % enters the diameter5 of the artificial eyes

Appendix

```
area = zeros(1,5);                          % create an 1*5 array to store the 5 pupil areas
area(1) = pi*(d1/2)^2;                        % finds area for diameter1
area(2) = pi*(d2/2)^2;                        % finds area for diameter2
area(3) = pi*(d3/2)^2;                        % finds area for diameter3
area(4) = pi*(d4/2)^2;                        % finds area for diameter4
area(5) = pi*(d5/2)^2;                        % finds area for diameter5

disp(area);                                % displays the area array

figure;                  % Create a new figure window
subplot(2, 1, 1);           % Create a 2-row, 1-column subplot and use the first section
plot(area, Area_left, '-o', 'LineWidth', 2, 'MarkerSize', 8);  % Plot 'Area_left' vs 'area' with a line and
circle markers, line width 2, marker size 8
xlabel('Area(mm2)');              % Label the x-axis as 'Area(mm²)'
ylabel('Arbitrary Units');         % Label the y-axis as 'Arbitrary Units'
title('Average Pupil Area for Left Pupil');  % Add a title to the plot for the left pupil
grid on;                  % Turn on the grid for better visualization

subplot(2, 1, 2);           % Use the second section of the 2-row, 1-column subplot
plot(area, Area_right, '-s', 'LineWidth', 2, 'MarkerSize', 8); % Plot 'Area_right' vs 'area' with a line and
square markers, line width 2, marker size 8
xlabel('Area(mm2)');              % Label the x-axis as 'Area(mm²)'
ylabel('Arbitrary Unit');          % Label the y-axis as 'Arbitrary Unit'
title('Average Pupil Area for Right Pupil');  % Add a title to the plot for the right pupil
grid on;                  % Turn on the grid for better visualization


figure;
plot(area, Area_left, '-o', 'LineWidth', 2, 'MarkerSize', 8);
hold on;
plot(area, Area_right, '-s', 'LineWidth', 2, 'MarkerSize', 8);
hold off;

xlabel('Area(mm2)');
ylabel('Arbitrary Unit ');
title('Average Pupil Area for Left and Right Pupils Across Pupil sizes');
legend('Left Pupil', 'Right Pupil', 'Location', 'best');
grid on;

p_right = polyfit(area, Area_right, 1);  % Linear regression for the right pupil data.
fitted_right = polyval(p_right, area);  % Computes the fitted values for the right pupil.

figure;  % Opens a new figure window.
plot(area, Area_left, 'ro', 'LineWidth', 2, 'MarkerSize', 8);  % Plots the left pupil data points.
hold on;
plot(area, fitted_left, '-b', 'LineWidth', 1.5);  % Plots the linear regression line for the left pupil.
```

Appendix

```
hold off;

xlabel('Area (mm^2)');  % Labels x-axis.
ylabel('Arbitrary Unit');  % Labels y-axis.
title('Linear Regression for Average Pupil Area (Left)');  % Adds title to the plot.
legend('Left Pupil Data', 'Left Pupil Fit', 'Location', 'best');  % Adds a legend.
grid on;  % Turns on the grid.

disp(['Left Pupil Regression: y = ' num2str(p_left(1)) 'x + ' num2str(p_left(2))]);  % Displays left pupil
regression equation.
disp(['Right Pupil Regression: y = ' num2str(p_right(1)) 'x + ' num2str(p_right(2))]);  % Displays right
pupil regression equation.
```

 # Program for Experiment 3.

```
clc;                          % Clear the command window
clear all;                    % Clear all variables from the workspace
close all;                    % Close all figure windows

% Load data
data4 = load('Exp2_3.5mm_mov.mat');  % Load the dataset for 3.5mm pupil size
data5 = load('Exp2_4mm_mov.mat');    % Load the dataset for 4mm pupil size
data6 = load('Exp2_5mm_mov.mat');    % Load the dataset for 5mm pupil size
data7 = load('Exp2_6mm_mov.mat');    % Load the dataset for 6mm pupil size
data8 = load('Exp2_7mm_mov.mat');    % Load the dataset for 7mm pupil size

pupil_size_L  = 45;                % Define the channel number for left pupil size
left_pupil    = data7.y(pupil_size_L, :);  % Extract the left pupil size data from dataset (data7)
pupil_size_R  = 46;                % Define the channel number for right pupil size
right_pupil   = data7.y(pupil_size_R, :);  % Extract the right pupil size data from dataset (data7)
fs            = 4800;              % Set the sampling frequency to 4800 Hz
downsample_factor = 10;            % Set the downsampling factor to 10
left_pupil_d  = downsample(left_pupil, downsample_factor);  % Downsample left pupil data
right_pupil_d = downsample(right_pupil, downsample_factor); % Downsample right pupil data

fs_downsampled = fs / downsample_factor;  % Update the sampling frequency after downsampling

t = (0:length(left_pupil_d)-1) / fs_downsampled;  % Create the time vector for plotting

smooth_left_d  = smoothdata(left_pupil_d, 'rloess', 0.1 * fs_downsampled);  % Smooth the
downsampled left pupil data using 'rloess'
smooth_right_d = smoothdata(right_pupil_d, 'rloess', 0.1 * fs_downsampled); % Smooth the
downsampled right pupil data using 'rloess'
```

Appendix

```
figure                                 % Create a new figure window
subplot(2, 1, 1);                      % Create the first subplot (2 rows, 1 column, 1st plot)
plot(t, left_pupil_d, 'b', t, smooth_left_d, 'r'); % Plot the original and smoothed left pupil data
legend('Original data', 'Smoothed data using "rloess"', 'Location', 'NW');  % Add a legend to the plot
xlabel('Time(s)');                     % Label the x-axis (time in seconds)
ylabel('Area(mm2)');                   % Label the y-axis (pupil area in mm²)
ylim([900, 2000]);                     % Set the y-axis limits for better visualization
title('Left Pupil Data');              % Add a title to the plot

subplot(2, 1, 2);                      % Create the second subplot (2 rows, 1 column, 2nd plot)
plot(t, right_pupil_d, 'b', t, smooth_right_d, 'r'); % Plot the original and smoothed right pupil data
legend('Original data', 'Smoothed data using "rloess"', 'Location', 'NW');  % Add a legend to the plot
xlabel('Time(s)');                     % Label the x-axis (time in seconds)
ylabel('Area(mm2)');                   % Label the y-axis (pupil area in mm²)
ylim([1600, 2000]);                    % Set the y-axis limits for better visualization
title('Right Pupil Data');             % Add a title to the plot

%% Section for the second set of data
clc;                                   % Clear the command window
clear all;                             % Clear all variables from the workspace
close all;                             % Close all figure windows

% Load data
data7 = load('Exp2_real_eyes.mat');       % Load the dataset with real eye data

pupil_size_L  = 45;                    % Define the channel number for left pupil size
left_pupil    = data7.y(pupil_size_L, :); % Extract the left pupil size data from the dataset
pupil_size_R  = 46;                    % Define the channel number for right pupil size
right_pupil   = data7.y(pupil_size_R, :); % Extract the right pupil size data from the dataset
fs            = 4800;                  % Set the sampling frequency to 4800 Hz
downsample_factor = 10;                % Set the downsampling factor to 10
left_pupil_d  = downsample(left_pupil, downsample_factor);  % Downsample left pupil data
right_pupil_d = downsample(right_pupil, downsample_factor); % Downsample right pupil data

fs_downsampled = fs / downsample_factor;  % Update the sampling frequency after downsampling

t = (0:length(left_pupil_d)-1) / fs_downsampled;  % Create the time vector for plotting

smooth_left_d  = smoothdata(left_pupil_d, 'rloess', 0.1 * fs_downsampled);  % Smooth the
downsampled left pupil data using 'rloess'
smooth_right_d = smoothdata(right_pupil_d, 'rloess', 0.1 * fs_downsampled); % Smooth the
downsampled right pupil data using 'rloess'

figure                                 % Create a new figure window
```

Appendix

```
subplot(2, 1, 1);                       % Create the first subplot (2 rows, 1 column, 1st plot)
plot(t, left_pupil_d, 'b', t, smooth_left_d, 'r'); % Plot the original and smoothed left pupil data
legend('Original data', 'Smoothed data using "rloess"', 'Location', 'NW');  % Add a legend to the plot
xlabel('Time(s)');                      % Label the x-axis (time in seconds)
ylabel('Area(mm2)');                    % Label the y-axis (pupil area in mm²)
%ylim([900,2000]);                      % (Commented out) Y-axis limits for the left pupil plot
title('Left Pupil Data');               % Add a title to the plot

subplot(2, 1, 2);                       % Create the second subplot (2 rows, 1 column, 2nd plot)
plot(t, right_pupil_d, 'b', t, smooth_right_d, 'r'); % Plot the original and smoothed right pupil data
legend('Original data', 'Smoothed data using "rloess"', 'Location', 'NW');  % Add a legend to the plot
xlabel('Time(s)');                      % Label the x-axis (time in seconds)
ylabel('Area(mm2)');                    % Label the y-axis (pupil area in mm²)
%ylim([1600,2000]);                     % (Commented out) Y-axis limits for the right pupil plot
title('Right Pupil Data');              % Add a title to the plot
```

List of Figures

List of Figures

List of Tables

List of Abbreviations

CR – Corneal Reflection
**P - ** Pupil
HMD - Helmet-mounted displays
VR - Virtual Reality
EEG - Electroencephalogram
fMRI - Functional Magnetic Resonance Imaging
MEG - Magnetoencephalography
MRI - Magnetic Resonance Imaging
7T - 7 Tesla

Bibliography

[1] JOUR AU - Holmqvist, Kenneth AU - Nyström, Marcus AU - Andersson, Richard AU - Dewhurst, Richard AU - Jarodzka, Halszka AU - van de Weijer, Joost PY - 2011/01/01 SP - T1 - Eye Tracking: A Comprehensive Guide To Methods And Measures VL - ER - J. Malmivuo and R. Plonsey. Bioelectromagnetism. Principles and Applications of Bioelectric and Biomagnetic Fields, 1995.

[2] Schlegelmilch, K., & Wertz, A. E. (2019). The effects of calibration target, screen location, and movement type on infant eye-tracking data quality. Infancy, 24(4), 636–662.

[3] Rosengren, William and Nyst, Marcus and Hammar, Bjrn and Stridh, Martin, Suitability of calibration polynomials for eye-tracking data with simulated fixation inaccuracies,2018.

[4] Mark A. Mento, This is how eye-tracking technology works, https://www.bitbrain.com/blog/eye-tracking-technology, May 5, 2020

[5] Ignace Hooge, Kenneth Holmqvist, Marcus Nyström, pupil is faster than the corneal reflection (CR): Are video based pupil-CR eye trackers suitable for studying detailed dynamics of eye movements? Vision Research, Volume 128, 2016.

[6] EyeLink® 1000 Plus User Manual, https://www.hse.ru/mirror/pubs/share/560338728.pdf

[7] https://www.youtube.com/watch?v=CnXuZSCd11Q&t=1419s

[8] Adhanom, I.B., MacNeilage, P. & Folmer, E. Eye Tracking in Virtual Reality: a Broad Review of Applications and Challenges. *Virtual Reality* **27**, 1481–1505 (2023). https://doi.org/10.1007/s10055-022-00738-z

[9] Benjamin T. Carter, Steven G. Luke, Best practices in eye tracking research, International Journal of Psychophysiology, Volume 155,2020 https://doi.org/10.1016/j.ijpsycho.2020.05.010.

[10] Andrew T. Duchowski, Springer International Publishing, Eye Tracking Methodology, Theory and Practice, 2017

[11]https://www.sr-research.com/fmri-meg-systems/

[12] Timon Dörzapf, Exploration of software tools used for analyzing eye tracking data in software engineering, October 10, 2023

Bibliography

[13]https://www.sr-research.com/eyelink-1000-plus/

[14]https://www.youtube.com/watch?v=M6vN22Asxq4&t=144s

[15]Duchowski, A. T. (2017). *Eye Tracking Methodology: Theory and Practice*. Springer.