Gezamenlijke opleiding UHasselt en KU Leuven

2021•2022

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Research on the optimal setup for eye tracking and optimization of instruction set with eye tracking

PROMOTOR :

Prof. dr. ir. Eric DEMEESTER

PROMOTOR :

Bart LAMBERIGTS

Jeroen Van Caekenberghe, Jonas Vannieuwenhuijsen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

2021•2022

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Research on the optimal setup for eye tracking and optimization of instruction set with eye tracking

PROMOTOR :
Prof. dr. ir. Eric DEMEESTER

PROMOTOR :
Bart LAMBERIGTS

Jeroen Van Caekenberghe, Jonas Vannieuwenhuijsen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Acknowledgement

We would like to express our gratitude to everyone who helped us in the creation of this Master's thesis.

First of all, we would like to thank our external supervisor Bart Lamberigts from Arkite for giving us the opportunity to work on this thesis. The guidance and feedback played an important role in the realization of this work. Furthermore, we would like to thank our internal supervisor Prof. Dr. Ir. Eric Demeester for his suggestions throughout this project. In addition, we would like to thank Tobii for the use of the eyetracker and the license.

Contents

Acknowledgement **1**

List of tabels **5**

List of figures **8**

Abstract **9**

Abstract in het Nederlands **11**

1 Introduction **13**

2 Theoretical background **15**
2.1 Various application of eye trackers 16
2.1.1 Pro-saccadic task 16
2.1.2 Smooth pursuit task 17
2.1.3 Marketing and user research 17
2.1.4 Human performance in industry 17
2.2 Algorithms to determine fixations and saccades 17
2.2.1 Velocity-based algorithms 17
2.2.2 Dispersion-based algorithms 18
2.2.3 Area-based algorithms 18
2.3 Gaze-based interfaces 18

3 Setup, methodology and results **21**
3.1 Setup 21
3.1.1 Hardware setup 21
3.1.2 Software setup 21
3.2 Methodology 24
3.2.1 Experimental procedure 24
3.2.2 Data quality measurements 25
3.2.3 Tests, results and discussion 27

4 Test application **37**
4.1 Introduction 37
4.2 Test application 38
4.2.1 Subscribe 38
4.2.2 Calibrate 39

4.2.3 Instruction Set 39
4.3 User Experience Questionnaire 41
4.3.1 UEQ results 41
4.4 Time difference 43
4.5 Optimization 44

5 Conclusion **45**

Literaturelis **48**

A Appendix - Code subscribe function **49**

B Appendix - Test Application All Screens **51**

C Appendix - User Experience Questionnaire in Dutch and English **59**

List of Tables

3.1 Tobii Eye Tracker 4L Data Streams [1]. 22
3.2 Gaze calibration results in pixels. 28
3.3 Luminous intensity results in pixels. 30
3.4 Horizontal, vertical and operating distance shifts results in pixels. 33
3.5 Horizontal and vertical head rotation test results in pixels. 34
3.6 Explanation of all different conditions. 35
3.7 Different types of glasses and lenses test results in pixels. 36

4.1 Time difference between eye tracker version and mouse-only version 43

List of Figures

1.1 Example of a possible Human Interface Mate (HIM) setup. a) Working station where projection of the HIM will be projected to, b) monitor where instruction set will be shown, c) the HIM. 14
1.2 Tobii 4L eye tracker attached to a monitor with a 3D-printed mount. 14

2.1 Dark pupil and bright pupil tracking [2]. 15
2.2 Visual representation of cascades and fixations from eye movements [3]. 16
2.3 Sample two-state HMM. The first state represents higher-velocity saccade points; the second state represents lower-velocity fixation points [4]. 18
2.4 An example of a user scanpath on the HCW Travel web page, which is segmented into its visual elements – This webpage was used for the review of scanpath analysis techniques [5]. 19

3.1 Visual representation of the hardware setup. 22
3.2 Stimulus points during measurement phase. 24
3.3 (a) good accuracy and precision, (b) good accuracy and bad precision, (c) bad accuracy and good precision and (d) bad accuracy and precision. 25
3.4 Illustration of the distance between successive gaze points to calculate precision via RMS. 26
3.5 Illustration of the distance between the gaze point and the mean of all gaze points to calculate the standard deviation precision via standard deviation. 27
3.6 Position of test subject during the gaze calibration test. 27
3.7 Nine different calibration phases. 28
3.8 Plot of accuracy, precision and standard deviation precision in pixels for all nine different calibration types. 29
3.9 Measured and validation data plot obtained by eight point calibration. 29
3.10 LM-100 of AMPROBE®. 30
3.11 Plot of accuracy, precision and standard deviation precision in pixels for different illuminations on tracker. 30
3.12 Plot of accuracy, precision and standard deviation precision in pixels for different illuminations on subject. 31
3.13 Measured data plot and validation data plot obtained by direct sunlight on the eye tracker. 31
3.14 Starting position of test subject during calibration for the horizontal, vertical and operating distance shift tests. 32
3.15 Plot of accuracy, precision and standard deviation precision in pixels for shifting in horizontal, vertical and operating distance. 33

3.16 Illustration of horizontal and vertical head rotation. 34
3.17 Plot of accuracy, precision and standard deviation precision in pixels for horizontal and vertical head rotation. 35
3.18 Reflecting sunglasses (left), measured and validation data plot obtained by eye tracking data during tests with reflecting sunglasses (right). 36

4.1 Test subject going through the instruction set of the test application. 37
4.2 Visual process of a calibration point. 39
4.3 Example of instruction screen. 40
4.4 Visual representation of moving average. 40
4.5 Heatmap obtained by data of instruction one. 41
4.6 Assumed scale structure of the UEQ [6]. 42
4.7 Benchmark graph for eye tracking test application. 42
4.8 Comparison of eye tracker version and mouse-only version. 43

Abstract

Arkite is a company that transforms workstations into a digital and interactive environment by providing operator guidance systems. Their main product is called the HIM. Via a projector placed above the workspace, the HIM visually shows instructions for a given assembly line and where to find different objects and tools. This interaction with the operator works fine, but may be experienced as slow. One way to possibly speed up this process is by using an eye tracking sensor.

In this research, the ideal eye tracker setup is searched for. The eye tracker should be accurate, easy to use and robust in different conditions. The accuracy and precision of the eye tracking data are determined for the following: number of calibration points, luminous intensity, head location, head rotation and different types of glasses and lenses. The proposed eye tracking procedure is evaluated by means of a test application that allows the user to go through the instructions using an eye tracker. This test application will be compared with a mouse-only version by means of a User Experience Questionnaire.

The following can be concluded; to obtain the best result, the operator should be positioned at the middle of the eye tracker, 34.5 cm above it and with an operating distance of 65 cm. The operator should be looking to the middle of the screen, with a luminous intensity of 200 lux. When comparing the eye tracker version and the mouse-only version, it can be seen that the eye tracker version scores better for all scales, except for dependability.

Abstract in het Nederlands

Arkite is een bedrijf dat werkstations omvormt tot een digitale en interactieve omgeving door operatorbegeleidingssystemen aan te bieden. Hun belangrijkste product heet de HIM. Via een projector die boven de werkruimte wordt geplaatst, toont de HIM instructies voor een bepaalde assemblagelijn waar verschillende voorwerpen en gereedschappen te vinden zijn. Deze interactie kan als traag ervaren worden. Een manier om dit proces mogelijk te versnellen is het gebruik van een eyetracker.

In dit onderzoek wordt gezocht naar de ideale eyetrackingopstelling. De eyetracker moet nauwkeurig, gemakkelijk te gebruiken en robuust zijn. De nauwkeurigheid en precisie van de eyetracking-gegevens zijn bepaald voor volgende parameters: aantal kalibratiepunten, lichtsterkte, locatie van het hoofd, hoofdrotatie en verschillende soorten brillen. De voorgestelde eyetrackingprocedure wordt geëvalueerd aan de hand van een testtoepassing waarmee de gebruiker de instructies kan doorlopen met behulp van een eyetracker en zal vergeleken worden met een versie zonder eyetracking door middel van een User Experience Questionnaire.

Het volgende kan geconcludeerd worden; voor het beste resultaat moet de operator zich in het midden van de eyetracker bevinden, 34.5 cm erboven, een werkafstand van 65 cm en een lichtintensiteit van 200 lux wordt aanbevolen. Wanneer de eyetrackervariant met de muisvariant vergeleken wordt, kan er gezien worden dat de eyetracker beter scoort op alle gebieden, met uitzondering op betrouwbaarheid.

Chapter 1

Introduction

This master's thesis is in cooperation with Arkite, a company that transforms workstations into a digital and interactive environment by providing operator guidance systems. Their main product is called the HIM (Human Interface Mate). Via a projector placed above the workspace, the HIM will visually show real-time picking and assembly instructions for an assembly line, and will project where to find the different objects that are necessary inside the workspace. Assembly errors by the operator will also be handled, and correct executions will be validated by a smart sensor inside the HIM. By doing this, the amount of human errors will be significantly reduced. The HIM can be adapted for different assembly lines for full optimization. The full setup of a workstation consists of a working table, a projector, a monitor and the HIM itself. An example of this is shown in figure 1.1. The monitor will be placed in front of the operator to show the necessary instructions. To cycle through the instructions displayed on the monitor, the operator has to use a mouse and keyboard setup. Doing it this way, it can be experienced as slow in certain assembly lines, for example when the operator is holding something with both of his hands. This is why Arkite is interested in a different way of cycling through these instruction sets that is easier to use and, more importantly, will speed up the process.

One way to possibly speed up this process is using an eye tracking sensor. With this tracker, cycling through the instruction set could be done by only using your eyes. To test this theory, a certain type of eye tracker had to be selected for this application. In this case, the Tobii 4L eye tracker was chosen. Tobii is one of the global leaders in eye tracking. They offer both commercially available trackers and trackers for building professional products. The commercially available tracker is mainly used as a gaming accessory. For this research, the choice is made to use the professional Tobii eye tracker to have full access and control when developing applications with the tracker. Both trackers are USB peripherals. The Tobii Eye Tracker 4L (figure 1.2) is a screen-mounted eye tracker for professional products and applications.

Since the HIM will be used in different kinds of environments, finding the optimal way to use this eye tracker is necessary. Knowing different parameters that could have an impact on the quality of the sensor output will be very useful. For example, the operator could be wearing glasses or the illumination conditions could be challenging. Will this have an impact or not? The methods and results of testing these different parameters and more can be found in this paper.

In this thesis, a test application is made to illustrate the use of an eye tracker on an instruction set. For this test, the operator will work with just a computer monitor that will show an

Figure 1.1: Example of a possible Human Interface Mate (HIM) setup. a) Working station where projection of the HIM will be projected to, b) monitor where instruction set will be shown, c) the HIM.

Figure 1.2: Tobii 4L eye tracker attached to a monitor with a 3D-printed mount.

instruction set. These instructions can then be cycled through by only using your eyes. The operator will have to assemble a small Lego set to demonstrate the usage of the hands while assembling something small. The instruction of this Lego set will be presented on the monitor.

Chapter 2 is a theoretical background to give the reader background information about key concepts around eye tracking and previous research done around the same topics. In chapter 3, the setup, methodology and results will be discussed. Multiple test for the ideal eye tracker setup are conducted. First, the hardware and software setup of these test are discussed. Next, the methodology and the procedure of each of these tests are talked about. Finally, the results of all the test are discussed. In chapter 4, the test application, made to test whether eye tracking would be more valuable than the standard way of going through an instruction set, is fully discussed. Finally, in chapter 5, a conclusion is made of the 2 conducted tests.

Chapter 2

Theoretical background

Charles Bell was the first person to attribute the movements of the eyes to the brain. The different movements and their effects were described by him, resulting in the development of eye tracking [7]. Various ways to measure eye movement began to be developed. However, all these methods were expensive and effortful. This slowed down further research at that time.

In present-day society, thanks to improvements in the eye tracking technologies, it has become more affordable and more user-friendly for researchers and users to use the trackers. A high accuracy of eye tracking and gaze-direction can be achieved by using video-based eye tracking technology. This technology relies on image processing software to identify the precise location of the pupil and corneal reflection. There are two ways to determine these locations, “dark pupil” and “bright pupil” tracking. The first exploits the phenomenon that the pupil becomes a dark circle relative to the rest of the eye when infrared light is shone on it. The latter makes use of the well-known red-eye effect (light shone through the pupil and reflected by the retina) [8][9]. With this information, the participant must look at a series of points on a screen to calibrate the eye tracker. This is done in a calibration phase. If all of this is done correctly, the point of gaze can be effectively estimated from the corneal reflection along with the location of the pupil [7].

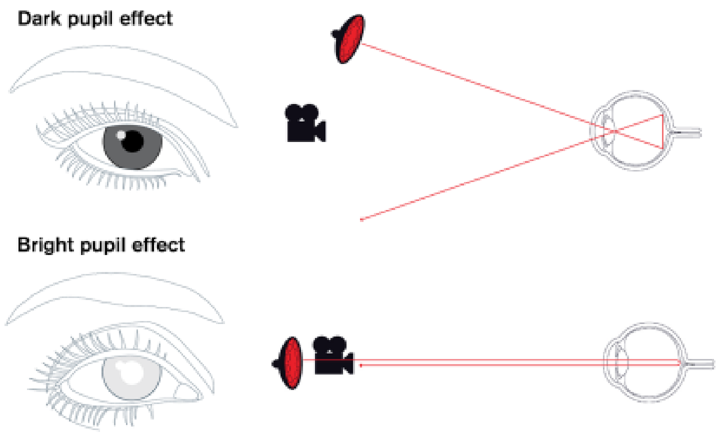


Figure 2.1: Dark pupil and bright pupil tracking [2].

There are two important eye movements, fixations and saccades. Fixation is a period of time the eye will fixate on a visual target. Fixation is possible thanks to the fovea, which is a small depression within the neurosensory retina where visual acuity is the highest. The fovea itself is the central portion of the macula, which is responsible for central vision [10]. To acquire high quality information, the eye will have to frequently move between fixations because the fovea is so small. An average fixation will last 180-300 ms [11]. The movement when the eye goes from one fixation to the next is called a saccade. Both movements are illustrated in figure 2.2. Understanding these movements is a key part of eye tracking studies and is important in eye tracking technologies.

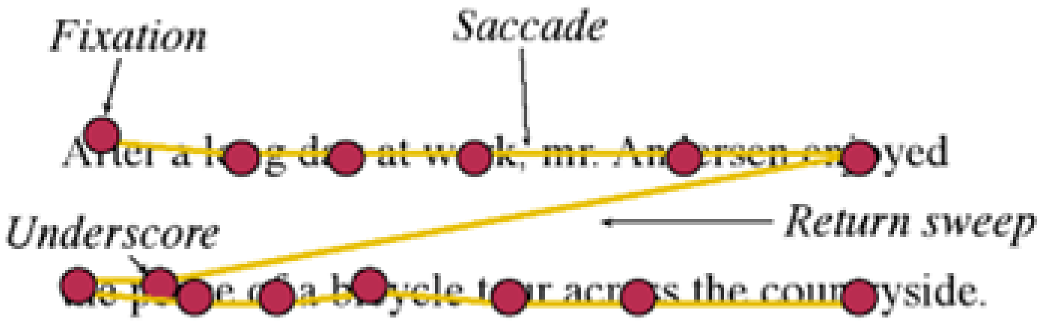


Figure 2.2: Visual representation of cascades and fixations from eye movements [3].

"Tobii" and "SR Research EyeLink" are two of the largest manufacturers among the many commercially available eye trackers. An eye tracker from Tobii is used in this study. All of Tobii's eye trackers are video-based eye trackers. They currently offer two types of eye trackers, one for commercial use and one for use in professional products and applications where control of the software/firmware is required. These two types of eye trackers are both USB peripherals [12]. Tobii also offers integration-based eye trackers, such as compact integration platforms designed as embeddable screen components for new products and devices. They call this the IS5 technology platform. Tobii IS5 can be integrated into PCs, desktop monitors, PC peripherals and medical devices. Tobii is also working on implementing eye tracking in VR or AR headsets, driver monitoring systems, and using RGB cameras to provide head tracking in real time [12]. SR Research provides eye trackers mainly for research purposes. Their main product is called the "EyeLink 1000 Plus" which is a highly customizable eye tracking device with multiple mount and head-free and head-fixed tacking modes. SR Research also provides a way to integrate this eye tracker in MRI and MEG systems to conduct research in the medical radiology field [13].

2.1 Various application of eye trackers

Eye tracking already has a lot of applications. Knowing about these gives a better understanding about using eye trackers. Inspiration from previous work is taken to successfully conduct the research done in this paper.

2.1.1 Pro-saccadic task

The pro-saccadic task is an experiment in which a target rapidly changes its location. The 'saccadic latency' is the time that the patient's brain needs to react and change the direction of the eye. It is an interesting parameter to determine, because decision-making and visual processing contributes to this parameter. This task is used in detecting the Parkinson's disease, because people with this disease tend to have increased saccadic latencies [14].

2.1.2 Smooth pursuit task

In this experiment, patients should follow an object located in their visual field with their eyes. When the object moves gradually, less than 100°/s, a fluid pursuit of the eyes should be noticeable. One application of this task is to detect patients with schizophrenia and bipolar disorder. Because they have difficulties keeping up with smooth pursuit of the eyes, they must make up for lost time saccades [14].

2.1.3 Marketing and user research

The goal of marketing is to provide consumers with as much information as possible in the most efficient way possible. By using eye tracking, researchers can examine how consumers distribute their visual attention in relation to different types of advertisements. This will allow the advertising to be adjusted so that the information about the product reaches the consumer in a more efficient manner [15].

2.1.4 Human performance in industry

Eye tracking can be used in the industry to monitor or improve employee performance. In a pilot study of the German Heart Center in Berlin the aim was to determine attention, perception, and stress levels of perfusionists during operations [16]. In another study, the inspectors at Denso] wore eye tracking glasses while performing an inspection, revealing their visual patterns, and explaining their high quality and efficiency, which allowed them to reduce the time to train new employees [17].

2.2 Algorithms to determine fixations and saccades

This section describes different types of algorithms to determine if the gaze data are saccades or fixation. This information may be of interest to Arkite to see how their instructions are interpreted.

2.2.1 Velocity-based algorithms

Velocity Threshold Identification (I-VT)

The velocity values are calculated for each gaze location sample. These values are then compared to a specific threshold. Fixations are the samples below the threshold value, the samples above the threshold are marked as saccades [18].

HMM Identification (I-HMM)

To determine the most likely identification for a given protocol, the Hidden Markov Model (HMM) uses a probabilistic approach. This model includes fixation and transition probabilities.

The most well-known HMM is the two-state HMM. The first state, which contains a clustering around higher velocities, represents saccade points, while the other state represents fixation points, which contains a clustering around lower velocities [4].

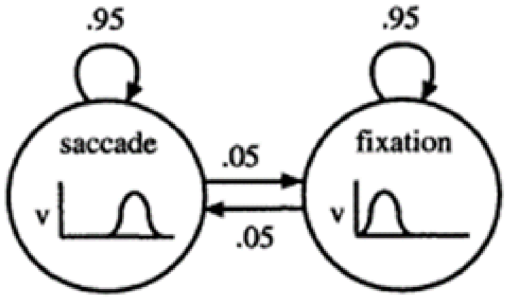


Figure 2.3: Sample two-state HMM. The first state represents higher-velocity saccade points; the second state represents lower-velocity fixation points [4].

2.2.2 Dispersion-based algorithms

Dispersion-Threshold Identification (I-DT)

As mentioned in before, fixations have a low velocity, so they tend to cluster closer together. Therefore, I-DT fixations are clusters of consecutive points within a certain range. A minimum duration of 100-200 ms is often used in this technique, as fixations have a duration of at least 100 ms [4].

2.2.3 Area-based algorithms

Area-Of-Interest Identification (I-AOI)

In area-of-interest identification, target regions are specified. These are regions of interest (ROI) that represent units of information. Only fixations that occur in these target areas are identified. There is a threshold to distinguish them between fixations and saccades in these areas. These target regions are later used in analyses such as tracing [19].

2.3 Gaze-based interfaces

In this research, eye tracking will be used to interact with instructionset-interfaces. This is a topic known as gaze-based interaction. Eye gaze data will be used to interact with a computer system. The authors of paper [20] categorized gaze-based interaction into four groups, which are named diagnostic, passive, active and expressive.

Diagnostic analysis mainly happens offline. Saccades and fixation are an important part of this. An ordered set of fixations followed by saccades is called a scanpath. These scanpaths are a way of representing eye movement [21]. Figure 2.4 shows an example of a scanpath of a user on a traveling website. The fixations are illustrated with circles and the saccades with lines. The longer the fixation, the larger the circle will be. With this, a way of analyzing expertise in a certain function is possible. Experts tend to make shorter fixations and use a larger visual span, as defined by Ericsson in [22]. Knowing this, it is possible to analyze the training development to see if a given person is making progress or not by comparing the scanpaths before and after training. With diagnostic analysis, tracking dynamic areas of interests (AOI's) also becomes possible. An example of this is in flight simulations and training, where the study of visual monitoring is especially important, as shown in paper [20]. Being able to analyze the gaze at important AOI's in this example is a good way of evaluating the flight training and simulation. Visualizing eye movement is also a part of the diagnostic analysis. As mentioned before, scanpaths are a common way of representing eye movement. Another popular technique for visualization is called Heatmaps. The different levels of gaze intensity are shown in heatmaps. This makes for a different and beneficial visual representation of eye movement. Heatmaps give a better understanding of eye movement in a given setting.

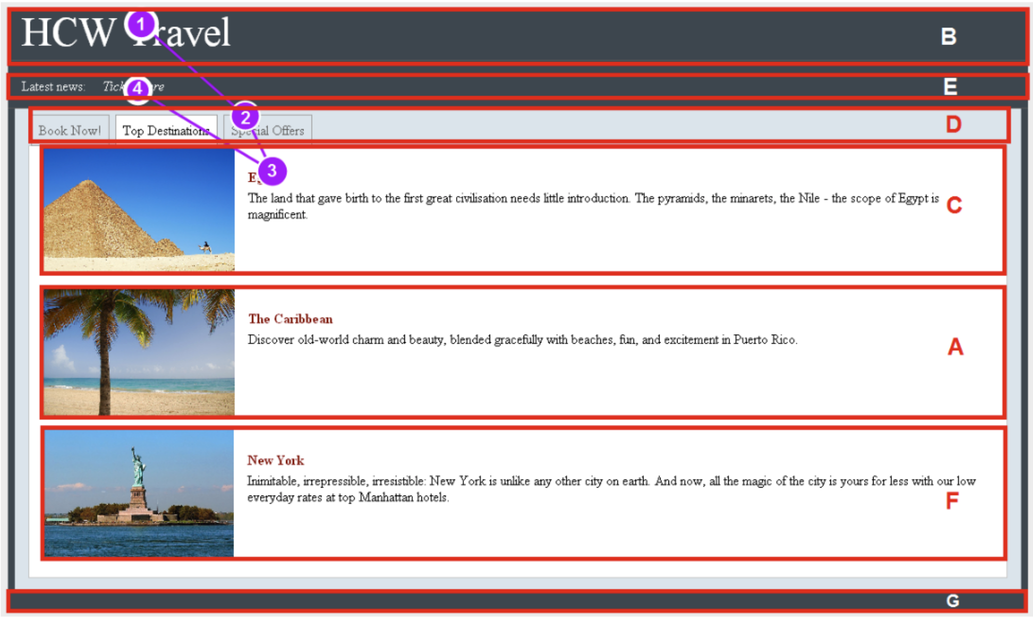


Figure 2.4: An example of a user scanpath on the HCW Travel web page, which is segmented into its visual elements – This webpage was used for the review of scanpath analysis techniques [5].

Active applications are part of the active group where real time data from the eye track is used. These real time data points make it possible to interact with an interface. The idea here is to use dwell time to interact with a computer system. Gaze-based selection is a commonly used example that uses dwell time. To make a selection, there has to be a fixation for a certain period of time. Another example is gaze gestures. This is mainly used for gaze-based eye typing. Another topic in this group is gameplay and interaction in 3D environments. In this topic, eye tracking is used in first-person shooters to adjust the viewport or more specifically to aim the gun to where the user is looking.

The next group is the passive applications group. This is the opposite of an active application because the user is not actively choosing to select something with his or her eyes. Rather, the system is using the user's gaze data to continuously respond to it. An example of a passive application is a gaze-contingent display (GCD). "GCDs attempt to balance the amount of information displayed against the visual information processing capacity of the observer through real-time eye movement sensing." [23].

The final group is called the expressive group. In this group, gaze synthesis is responsible for modeling eyes in 3D models. For example, in video game characters or in CGI movies when realistic eye movement is needed.

Chapter 3

Setup, methodology and results

3.1 Setup

To conduct the research and to eventually develop a test application, a specific setup had to be designed that fits for the use case of this project. As previously shown in figure 1.1, the operator will stand straight while working and looking at the instruction monitor. The eye tracker will be placed at the bottom of this monitor. The work environment and the physical characteristics of HIM users can differ a lot, so to find the optimal way of working with the eye tracker for this setup is useful. In this case, the setup can be divided in two categories, a hardware setup and a software setup. Both of these will be discussed in detail in the next section.

3.1.1 Hardware setup

The most important part of the hardware setup is the type of eye tracker. For this research, the Tobii 4L eye tracker was provided by Arkite. The Tobii 4L is a USB-based eye tracker that connects to a host PC. This tracker is meant to be mounted on a screen and support screen sizes up to 27". When using this tracker, the developer has full control over the software / firmware during the development stage. This is because the 4L is designed to be used in professional products and applications like Healthcare, Industrial or other markets.

To find the optimal use of the eye tracker, tests were conducted. To make all these tests reliable, the setup of the eye tracker and the monitor had to be consistent. To achieve this, we chose to use the same setup for every test. This setup is also used in the testing of the test application. The monitor used is an Acer XF240H, consisting of an LCD-display with a size of 24 inches and a ratio of 16:9. The monitor is adjustable to make sure it can be used standing or seated. This monitor is wired to a laptop running the test or test program via an HDMI cable. The laptop is wired to the tracker via a USB cable. The Tobii 4L eye tracker must be placed at the bottom of the monitor. To achieve this, a 3D-printed holder was designed to make sure the tracker is placed at the exact same position every time. The full hardware setup can be seen in figure 3.1.

3.1.2 Software setup

To have full access over the software / firmware of the Tobii 4L eye tracker, the Tobii Platform Development Kit (PDK) is needed. This PDK is provided with the tracker. It consists of two

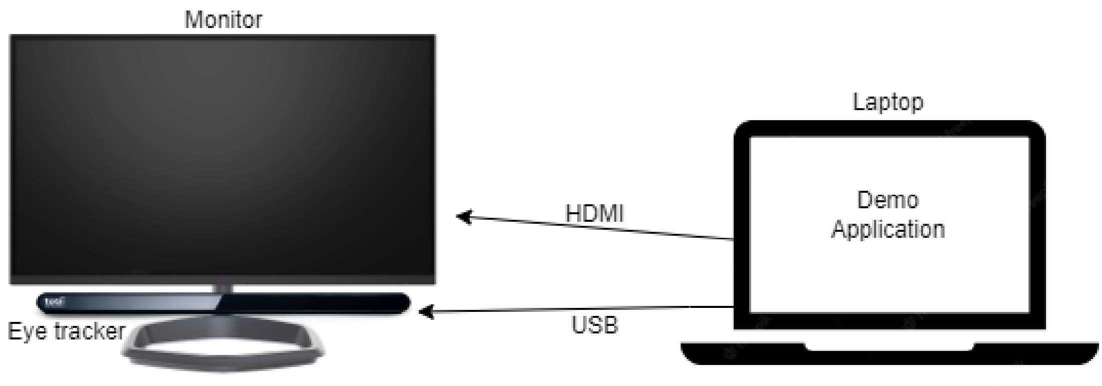


Figure 3.1: Visual representation of the hardware setup.

important parts, the Tobii Platform Runtime and the Tobii Stream Engine.

The Tobii Platform Runtime is a service that is responsible for the interaction between the application and the device. This Runtime is specific for this device. To install this service, the platform executable (included in the PDK) has to be run as administrator. Next, the service has to be started, this is done manually from the Services App on Windows.

The Tobii Stream Engine is a low-level SDK that is intended to have control over the tracker resources. It is responsible for all the software communication with trackers from Tobii. The Stream Engine is an API written in C. A documentation of this API is also provided in the PDK.

Tobii Stream Engine

The Tobii Eye tracker 4L is capable of providing different kind of data streams. All these data streams are shown in table 3.1

Table 3.1: Tobii Eye Tracker 4L Data Streams [1].

Core data streams	Combined (left right eye) gaze point, filtered for interaction
	User presence
	Gaze origin
	User position guide
Advanced data*	Left and right eye separate unfiltered gaze point
	Left and right eye separate unfiltered gaze origin
	Left and right eye separate unfiltered pupil diameter
Configuration data*	Screen calibration and display setup

* these data streams require a separate license key

In this research, only two of these data streams were used. First, the “combined gaze point” data for knowing where the user is looking on the screen. Second, the“screen calibration and display setup” data stream is used in order to calibrate the tracker for every user. This is necessary for getting the optimal quality of gaze data. A license key was provided for research purposes from Tobii to access the necessary data streams.

To access these streams in the code environment, the dynamic library (DLL) of the StreamEngine SDK must be added to the project. A dynamic library is a library that consist of other libraries, each with its own functionalities. These functionalities are only launched during the execution of the program. This means the actual size of the program will be reduced, and less memory will be needed. This DLL consists of 6 header files. These 6 files provide access to different functionalities from the data streams. An overview of these six files is shown in the next list. This list is from the following source [24].

- **tobii.h** – collects the core API functions of Stream Engine. It contains functions to initialize the API and establish a connection to a tracker, as well as enumerating connected devices and requesting callbacks for subscriptions. There are also functions for querying the current state of a tracker, and to query its capabilities,

- **tobii_streams.h** - functionality for managing data stream subscriptions. There are several types of data streams in the API, and tobii_streams.h contains functions to subscribe to and unsubscribe from these streams, as well as data structures describing the data packages,

- **tobii_wearable.h** – functionality related to wearable devices, such as VR headsets. It contains a specialized data stream with different data from the regular streams, as well as functions to retrieve and modify the lens configuration of the device,

- **tobii_licensing.h** – provide access to stream engine functionality restricted by license. The functionality made available by stream engine is controlled by license files generated by Tobii,

- **tobii_config.h** – functionality to configure the state of the tracker, such as calibration and display area setup. To modify the state of the tracker, a license on, at least, config level is required,

- **tobii_advanced.h** – advanced features, requiring a professional license to use.

The tobii_wearable.h and the tobii_advanced.h files were not used during this project because the functionalities of these libraries were not needed for the applications in this research.

Usage of the StreamEngine

First of all, it is important to understand that in order to have a continuous data stream between the tracker and the program, a thread is needed. Since the library does not create treads, the program has to do it itself. This is done by calling the *tobii_device_process_callbacks* function at an interval of at least 10 times per second. This function will receive any data packages that are incoming from the Tobii device, processes them, and calls any subscription callbacks with the data. It will not wait for data if there is no new data to process. Before processing callbacks in a self-made thread, the *wait_for_callbacks* function should be used. This function will put the calling thread to sleep until there are new callbacks available to process from the device connections.

3.2 Methodology

3.2.1 Experimental procedure

To find the optimal setup for eye tracking, different tests were conducted. These tests consist of a calibration phase and a measurement phase. During the calibration phase, stimuli points are displayed that gradually get smaller in order to get the focus on one point. In the measurement phase, there are nine stimuli points displayed at specific validation locations, as shown in figure 3.2. The test subjects look at each stimulus point for 3.75 seconds, thereby obtaining 150 data points per stimulus point. Each measurement is performed three times. The accuracy and precision of the eye tracking data were determined for the following different parameters:

- amount of calibration points,
- luminous intensity,
- horizontal, vertical and operating distance,
- head rotation,
- different types of glasses and lenses.

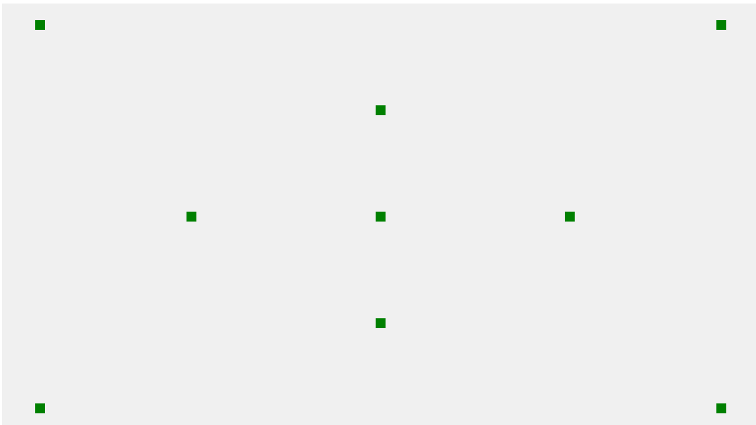


Figure 3.2: Stimulus points during measurement phase.

3.2.2 Data quality measurements

Accuracy is defined as the degree in which the average of the measured gaze points differs from the real stimulus position. Precision is the degree to how close the measured gaze points of the same stimuli point are to each other. The following formulas in this section were inspired by paper [25].

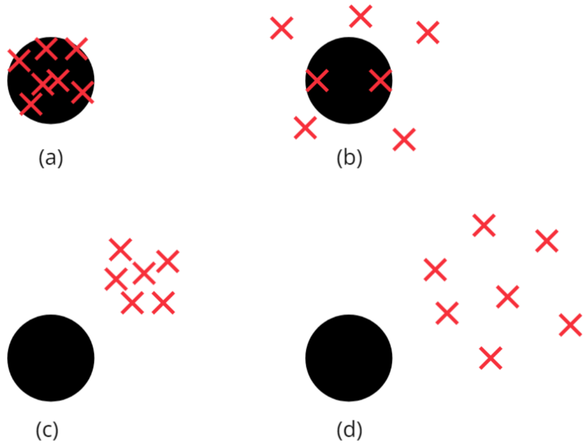


Figure 3.3: (a) good accuracy and precision, (b) good accuracy and bad precision, (c) bad accuracy and good precision and (d) bad accuracy and precision.

Accuracy

The accuracy is calculated as the mean of the distance between the gaze points and the stimuli points in pixels of the screen. The lower the accuracy in pixels, the higher the accuracy of the gaze point.

$$GazeCoordinateX = \frac{GazeCoordinateXLeft + GazeCoordinateXRight}{2} \tag{3.1}$$

$$GazeCoordinateY = \frac{GazeCoordinateYLeft + GazeCoordinateYRight}{2} \tag{3.2}$$

$$DistanceToStimuli = \sqrt{(GazeCoordinateX - StimuliCoordinateX)^2 + (GazeCoordinateY - StimuliCoordinateY)^2} \tag{3.3}$$

$$AccuracyInPixels = mean(DistanceToStimuli) \tag{3.4}$$

$$AccuracyInMillimeters = AccuracyInPixels * PixelLenght \tag{3.5}$$

Precision

The precision is calculated via the Root Mean Square (RMS) (equation 3.6) from the distance between successive gaze points, as illustrated in figure 3.4.

$$PrecisionInPixels = \sqrt{\frac{d_1^2 + d_2^2 + ... + d_n^2}{n}} \tag{3.6}$$

$$PrecisionInMillimeters = PrecisionInPixels * PixelLentgh \tag{3.7}$$

Where d is the distance between successive gaze points and n the number of gaze points.

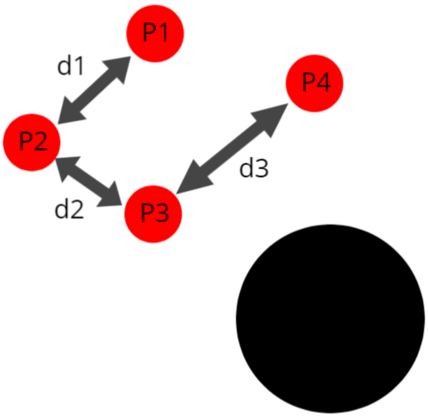


Figure 3.4: Illustration of the distance between successive gaze points to calculate precision via RMS.

Standard deviation precision

Another way to describe the precision is by using the standard deviation precision. It is calculated via the standard deviation from the gaze points, as illustrated in figure 3.5.

$$MeanX = \frac{\sum GazeCoordinateX}{n} \tag{3.8}$$

$$MeanY = \frac{\sum GazeCoordinateY}{n} \tag{3.9}$$

$$\overline{x} = (MeanX, MeanY) \tag{3.10}$$

$$StandardDeviationPrecisionInPixels = \sqrt{\frac{d_1^2 + d_2^2 + ... + d_n^2}{n}} \tag{3.11}$$

$$StandardDeviationPrecisionInMillimeters = PrecisionInPixels * PixelLentgh \tag{3.12}$$

Where d is the distance between the gaze point and the mean of all gaze points and n the number of gaze points.

For both the precision and the standard deviation precision, the smaller the precision in pixels, the more precise the gaze points are.

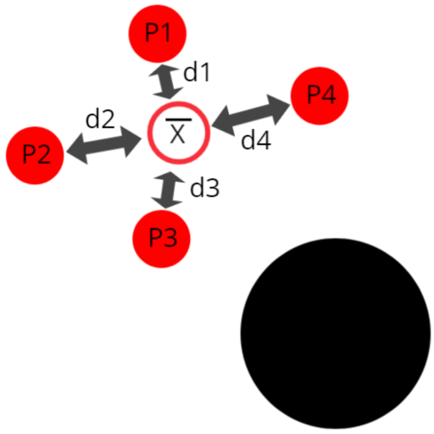


Figure 3.5: Illustration of the distance between the gaze point and the mean of all gaze points to calculate the standard deviation precision via standard deviation.

3.2.3 Tests, results and discussion

Gaze calibration

As mentioned earlier, every test consists of a calibration and a validation phase. Each test will be performed three times, 150 data points per calibration stimulus (8 calibration stimulus per test) so this makes for a total of 4050 measurement points to validate. For each measurement, the test subject will take the same posture as shown in figure 3.6. The eye tracker attached to the monitor has a perpendicular distance of 10.5 cm to the work surface. The test subject's eyes have a perpendicular distance of 34.5 cm to the work surface and 58 cm to the monitor. During this test, a light intensity of 300 lux is used. This will be explained the next test discussion. In the gaze calibration test, the number of calibration points during the calibration phase are changed. They are going from one up to nine calibration points. In figure 3.7, the nine different calibration configurations can be observed.

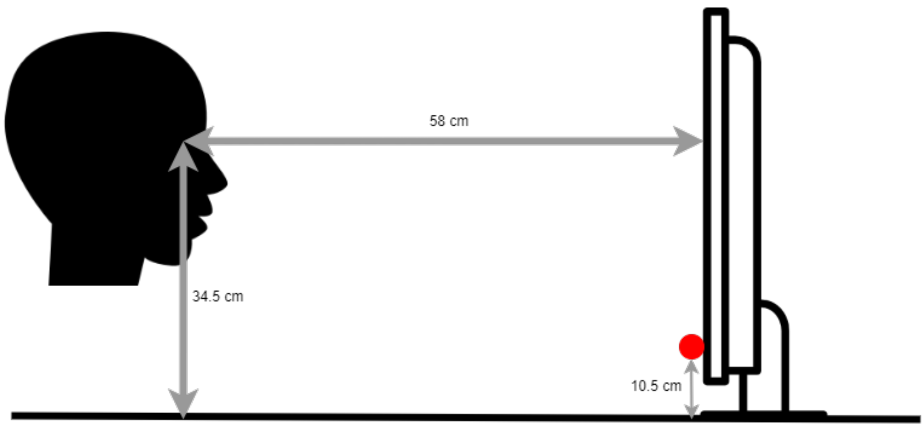


Figure 3.6: Position of test subject during the gaze calibration test.

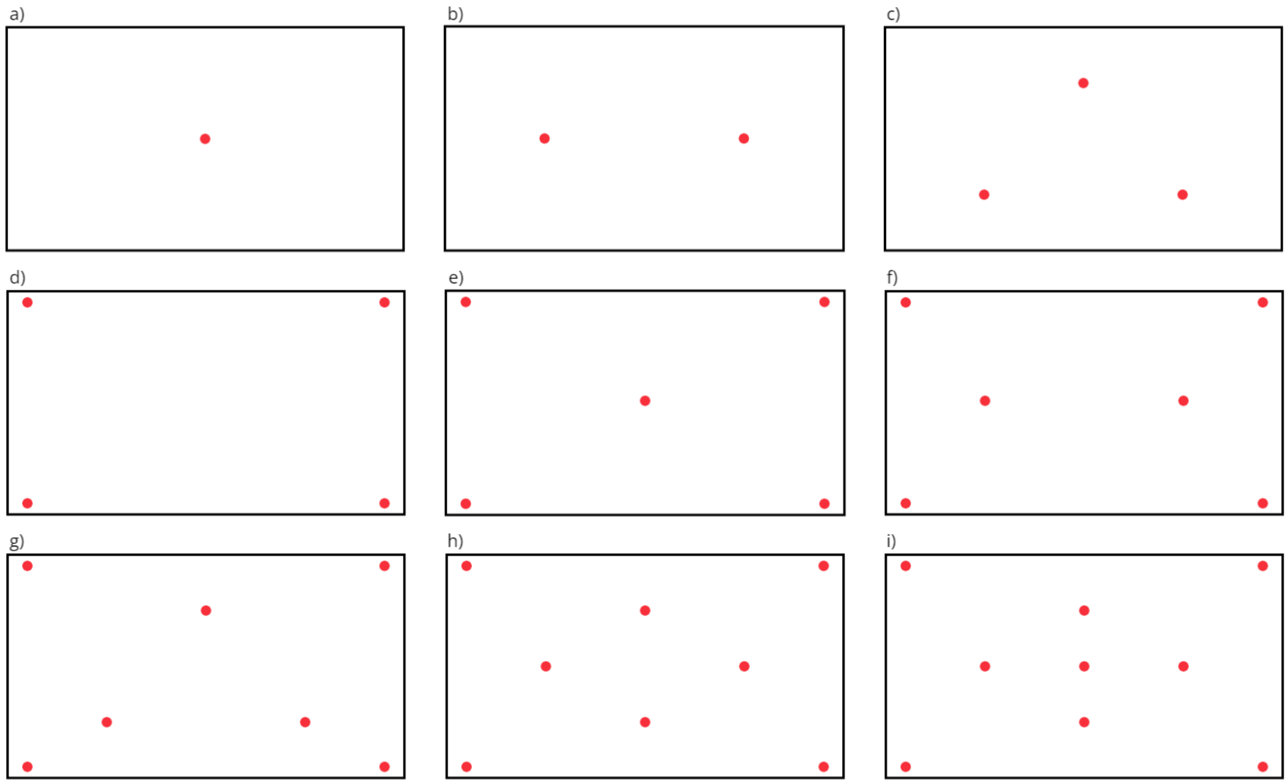


Figure 3.7: Nine different calibration phases.

For each test, the accuracy, precision and standard deviation precision are calculated. These will be calculated for each point, and then the average of the nine points are taken, which can be seen in table 3.2.

Table 3.2: Gaze calibration results in pixels.

	Accuracy (pixels)				Precision (pixels)				Standard deviation precision (pixels)			
Calibration points	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
1	64.90	68.48	69.90	67.76	2.74	2.96	2.61	2.77	21.35	25.20	20.11	22.22
2	20.40	16.71	20.08	19.06	2.45	2.36	2.12	2.31	13.65	14.19	16.69	14.84
3	20.18	26.48	23.32	23.33	2.31	2.56	2.53	2.47	17.15	17.04	14.12	16.10
4	18.47	16.50	16.30	17.09	2.59	2.51	2.54	2.55	18.32	16.97	19.86	18.38
5	14.30	11.58	11.45	12.44	2.48	2.41	2.58	2.49	13.28	16.23	19.40	16.30
6	12.92	14.09	11.94	12.98	4.01	2.58	2.52	3.04	15.27	16.64	14.71	15.54
7	14.12	13.91	14.71	14.25	2.30	2.29	2.50	2.36	14.13	14.69	14.61	14.48
8	13.14	9.35	11.59	11.36	2.32	2.40	2.50	2.41	16.28	15.04	18.67	16.66
9	10.31	14.48	13.95	12.91	2.14	2.22	2.28	2.21	14.68	18.70	15.92	16.43

It can be concluded that when the number of calibration points increases, the accuracy improves, but an additional 3.4 seconds of time is added per calibration point. As seen in figure 3.8 it averages out to a mean error of about 12 pixels. For both the precision and the standard deviation precision, there is no connection with the number of calibration points. The precision is very constant with values fluctuating between 2.2 and 3.1 pixels, which means that all consecutive measuring points are very close to each other. The highest accuracy was achieved with the test of eight calibration points, therefore all subsequent tests will be preceded by the eight point calibration. Figure 3.9 shows the accuracy and precision of the eight point calibration.

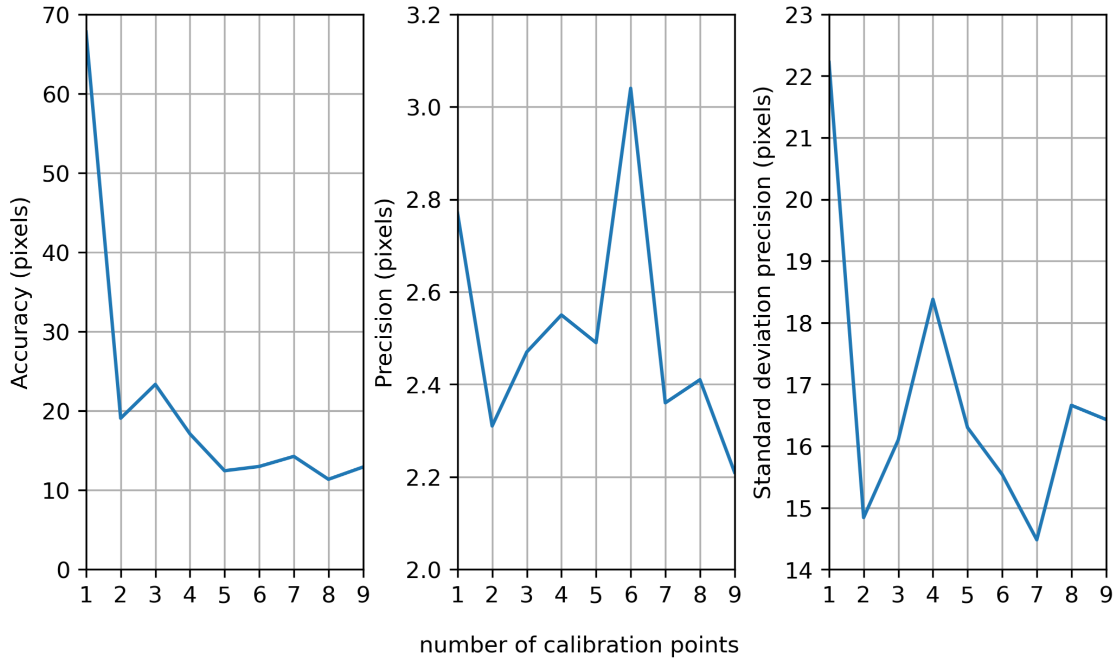


Figure 3.8: Plot of accuracy, precision and standard deviation precision in pixels for all nine different calibration types.

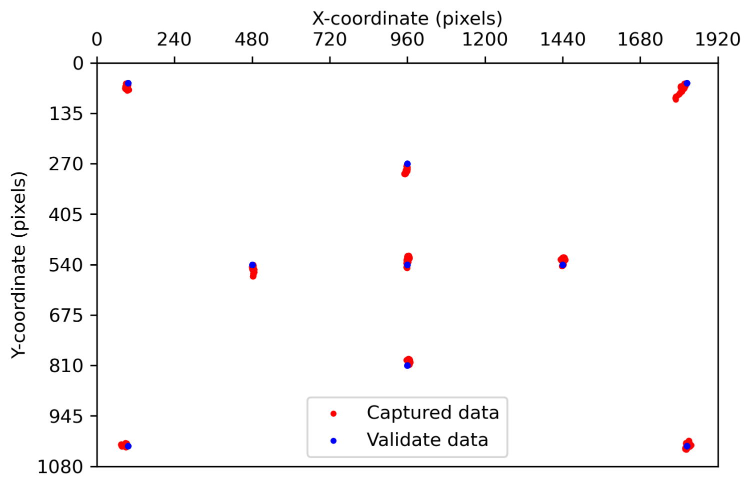


Figure 3.9: Measured and validation data plot obtained by eight point calibration.

Luminous intensity

During these tests, the test subject will hold the same position as in the gaze calibration test, as shown in figure 3.6. The light intensity will differ for each test, this will be obtained by dimmable LED lights present in the test room. The light intensity will be measured by a LM-100 of AMPROBE® (figure 3.10), calibrated and provided by IDEWE vzw. There are two additional measurements done in these tests where instead of LED light, direct sunlight is used. During the first test, direct sunlight fell on the eye tracker, during the second test, direct sunlight fell on the eyes of the test person. As with the previous test, accuracy, precision and standard deviation precision are calculated for these results, which can be seen in table 3.3.

Figure 3.10: LM-100 of AMPROBE®.

Table 3.3: Luminous intensity results in pixels.

Illumination (lux)		Accuracy (pixels)				Precision (pixels)				Standard deviation precision (pixels)			
On tracker	On subject	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
7	70	18.28	17.00	18.48	17.92	2.40	2.29	2.33	2.34	25.24	12.46	16.62	18.11
30	70	18.28	15.42	16.94	16.88	2.62	2.53	2.33	2.49	18.30	15.80	16.91	17.00
90	100	19.67	16.75	18.80	18.41	2.54	2.62	2.37	2.51	16.25	16.72	14.42	15.80
150	140	16.25	15.50	16.84	16.20	3.72	2.50	2.49	2.90	25.02	17.26	14.51	18.93
250	300	15.00	17.65	15.58	16.08	2.45	2.75	2.67	2.62	16.30	16.24	21.10	17.88
700	250	17.87	17.01	18.75	17.88	2.50	2.78	2.79	2.69	17.12	17.34	17.99	17.48
1400	270	18.50	17.61	15.35	17.15	2.90	2.51	2.44	2.62	17.14	15.70	17.38	16.74

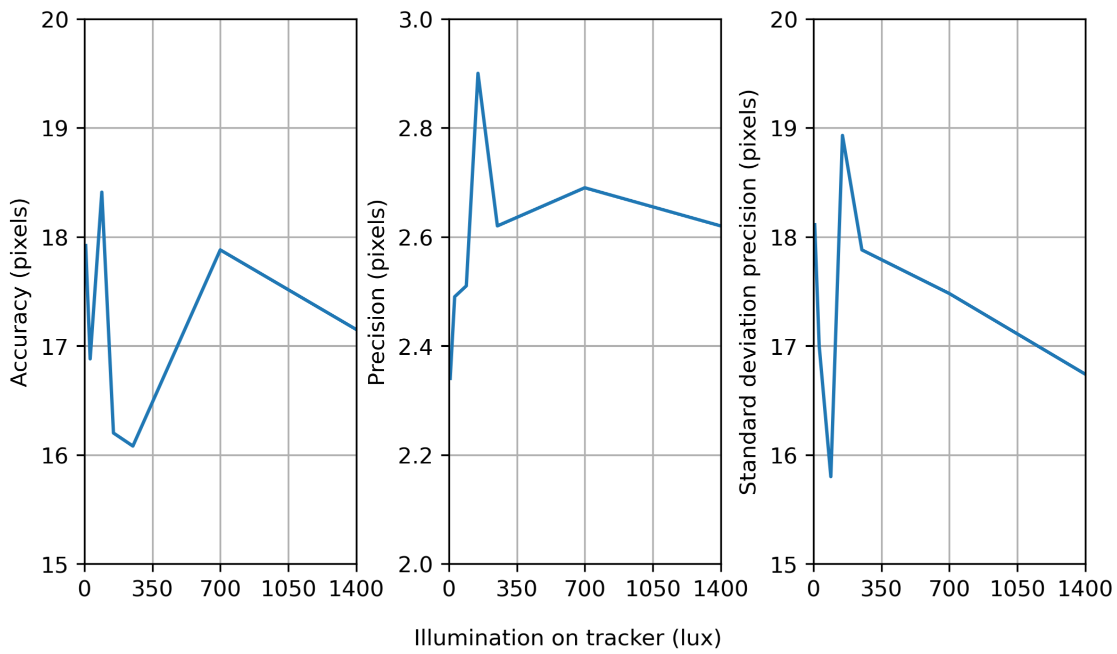


Figure 3.11: Plot of accuracy, precision and standard deviation precision in pixels for different illuminations on tracker.

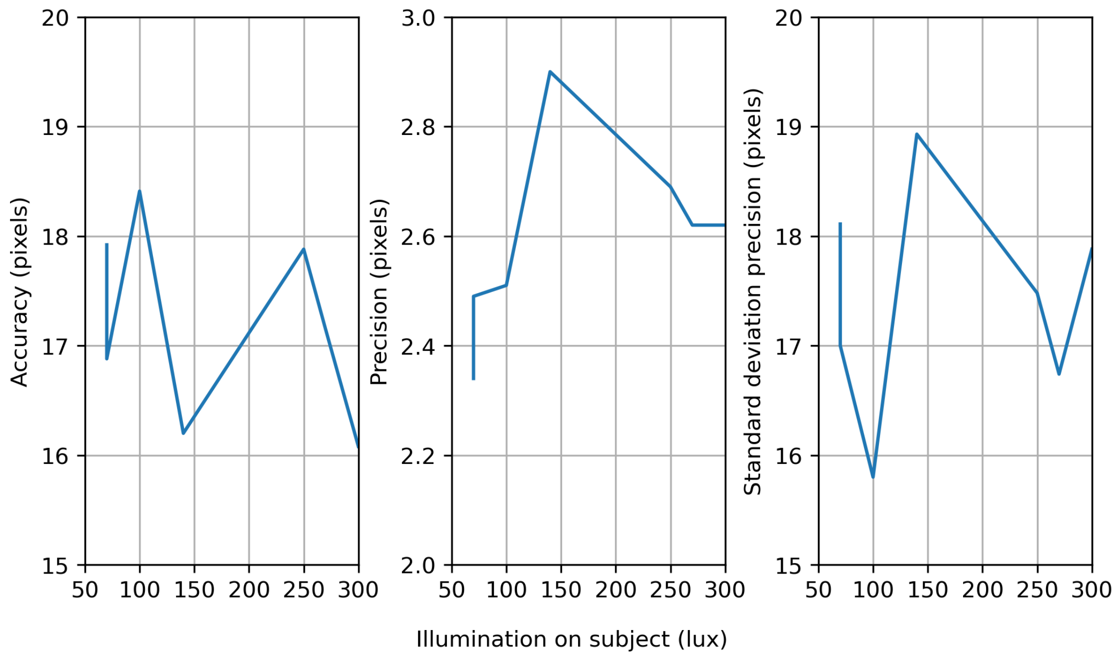


Figure 3.12: Plot of accuracy, precision and standard deviation precision in pixels for different illuminations on subject.

Looking at the results and graph (table 3.3 and figures 3.11, 3.12), it can be concluded that there is no direct relationship with the light intensity on the tracker or test subject and the accuracy or precision. For all measurements, a fairly good accuracy and precision is obtained. Looking at the two separate tests where direct sunlight is used, we see that when there is a lot of sunlight (± 30000 lux) on the eyes of the test person, no measurement is possible. When there is a lot of sunlight (± 30000 lux) on the tracker, the measurements are negatively affected, as shown in figure 3.13. In this test, a mean accuracy, precision and standard deviation precision of 132.21, 151.06 and 580.49 were obtained respectively. Therefore, it can be concluded that in outdoor situations this form of eye tracking is not recommended.

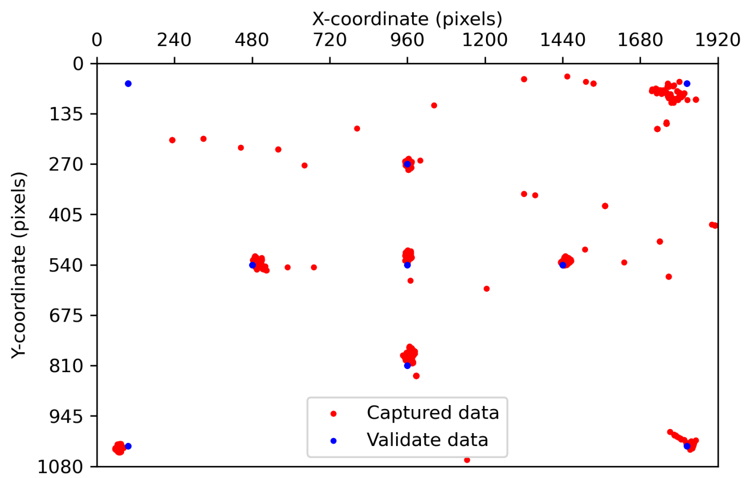


Figure 3.13: Measured data plot and validation data plot obtained by direct sunlight on the eye tracker.

Horizontal, vertical and operating distance

During these tests, the test subject will do shifts in the x, y and z directions. These shifts are done separately from each other in order to measure the influence of each direction. Figure 3.14 shows the starting position, where the calibration is done, after this there will be no more calibration for the shifts. For the first test, the horizontal shift, the test subject will take a starting position where x = 0 cm, y = 45 cm and z = 60 cm. For the second test, the vertical shift, the test subject will take an initial position where x = 0 cm, y = 40 cm and z = 65 cm. For the final test, the z-shift or operating distance, the test subject takes an initial position of x = 0 cm, y = 40 cm and z = 85 cm. As with the previous test, accuracy, precision and standard deviation precision are calculated for these results, which can be seen in table 3.4.

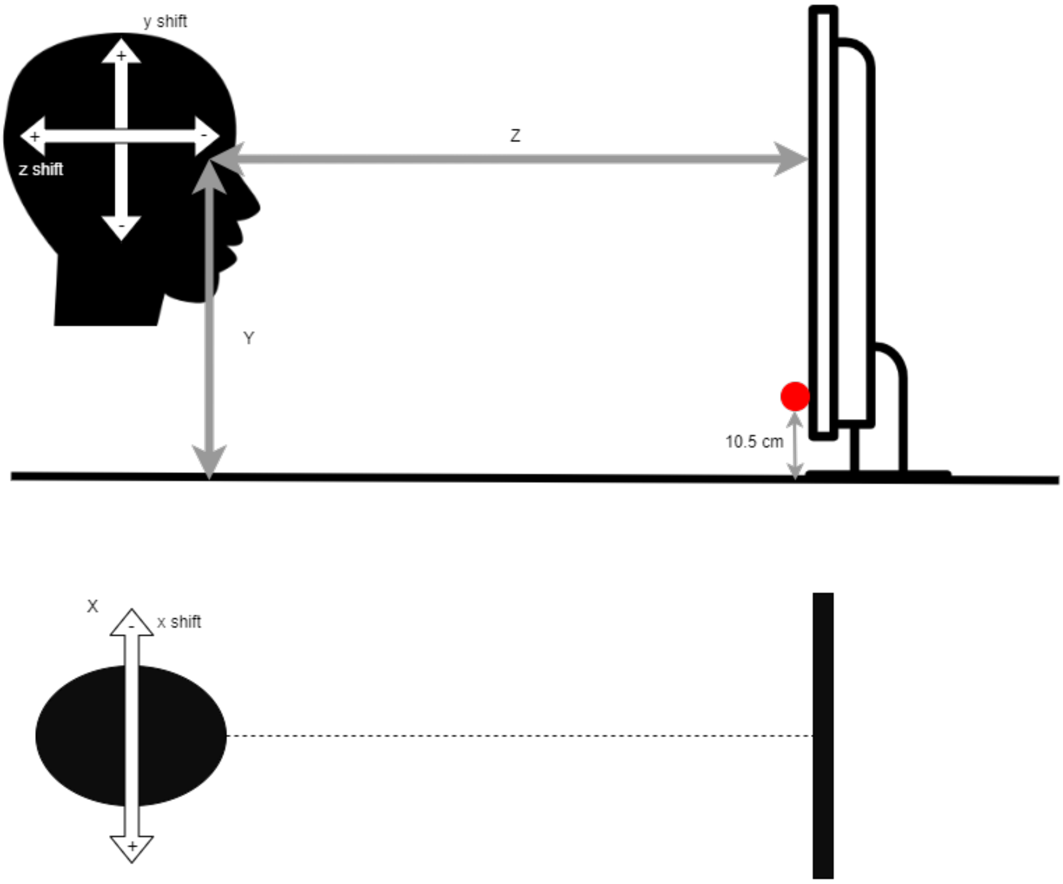


Figure 3.14: Starting position of test subject during calibration for the horizontal, vertical and operating distance shift tests.

As shown in figure 3.14 and table 3.4, the ideal working area according to the x-axis is within ten centimeters to the left and right of the calibration point. This is where the highest accuracy and precision is achieved.

Looking at the y shift, it can be concluded that the ideal working area is between -5 and +15 cm from the calibration point. This results in a working area of 24.5 cm to 44.5 cm above the eye tracker. When looking at the accuracy and precision, it can be said that +5 cm from the calibration state (34.5 cm above the eye tracker) the highest accuracy (12.18 pixels) and precision (2.49 pixels) is achieved.

Table 3.4: Horizontal, vertical and operating distance shifts results in pixels.

Shift (cm)	Accuracy (pixels)				Precision (pixels)				Standard deviation precision (pixels)			
x shift	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
-20	34.86	34.56	29.60	33.01	5.81	13.96	5.54	8.44	45.43	65.19	44.64	51.75
-10	25.83	22.62	26.68	25.04	4.96	2.82	3.75	3.84	15.98	15.62	18.57	16.72
0	16.57	15.10	13.17	14.95	2.52	2.70	2.39	2.54	18.79	14.90	17.03	16.91
10	17.52	15.08	13.26	15.29	2.90	3.40	3.95	3.42	15.66	18.41	14.47	16.18
20	19.99	21.85	20.07	20.64	8.10	5.47	5.50	6.36	25.50	20.61	26.50	24.20
y shift	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
-15	24.00	25.81	26.57	25.46	3.09	3.48	16.08	7.55	16.33	16.30	23.36	18.66
-10	21.45	20.78	24.53	22.25	2.92	2.96	3.06	2.98	16.61	14.87	16.69	16.06
-5	17.37	19.70	19.06	18.71	2.88	2.90	2.87	2.88	15.17	18.71	15.26	16.38
0	12.87	13.94	16.55	14.45	2.80	2.64	4.08	3.17	17.50	13.94	14.88	15.44
5	13.63	12.93	9.97	12.18	2.51	2.38	2.57	2.49	23.55	16.20	19.77	19.84
10	14.22	17.74	14.26	15.41	2.76	2.82	2.84	2.81	16.60	13.32	13.98	14.63
15	16.28	20.38	22.52	19.73	5.22	3.43	3.16	3.94	18.33	19.31	16.47	18.04
z shift	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
-20	129.95	103.60	136.38	123.31	62.81	85.81	101.62	83.41	103.98	134.63	164.03	134.21
-10	39.14	40.14	37.18	38.82	22.18	12.06	12.61	15.62	58.88	40.55	38.06	45.83
0	25.59	28.22	24.43	26.08	4.80	6.20	9.96	6.99	20.83	27.21	23.23	23.76
10	28.32	24.42	28.22	26.99	7.11	9.57	11.18	9.29	19.27	20.48	28.37	22.71
20	13.79	16.50	18.92	16.40	2.86	3.17	3.27	3.10	19.77	15.26	15.25	16.76

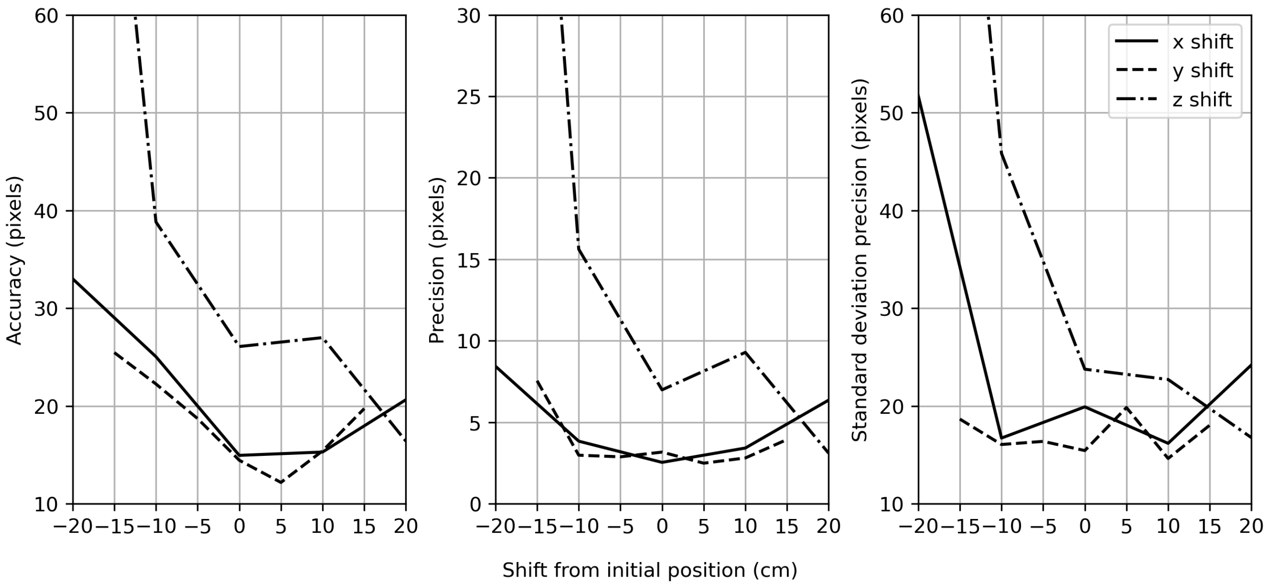


Figure 3.15: Plot of accuracy, precision and standard deviation precision in pixels for shifting in horizontal, vertical and operating distance.

Looking at the operating distance, it can be concluded that the closer the test subject gets to the eye tracker, the more accurate and precise the measurements will be. The ideal operating distance is a distance of 65 cm from the eye tracker. When the test person moves to a distance greater than 95 cm from the eye tracker, it can be concluded that the measurements are no longer accurate and precise.

Head rotation

During the head rotation test, the test subject will apply the calibration in his initial position and then rotate 5° per test and take the measurement. These rotations are measured using the accelerometer and magnetometer of a Samsung Galaxy S20. This is done in both horizontal and vertical rotation (figure 3.16) until it is no longer possible to take the measurement. As with the previous test, accuracy, precision and standard deviation precision are calculated for these results, which can be seen in table 3.5.

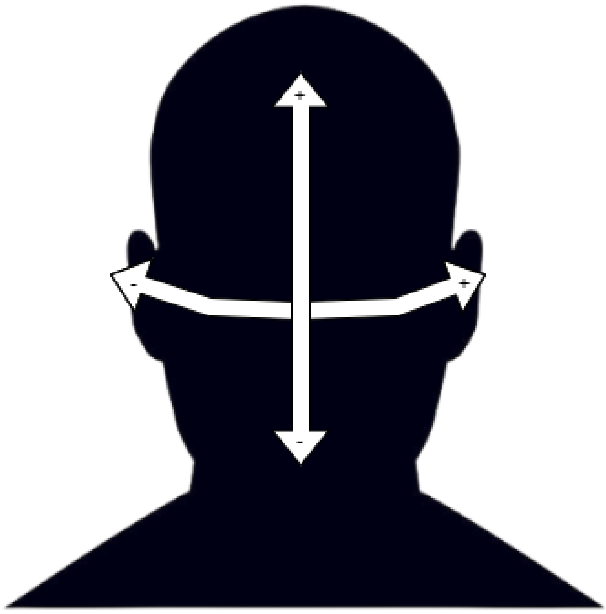


Figure 3.16: Illustration of horizontal and vertical head rotation.

Table 3.5: Horizontal and vertical head rotation test results in pixels.

Angle (°)	Accuracy (pixels)				Precision (pixels)				Standard deviation precision (pixels)			
Horizontal	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
-20	121.07	90.66	76.17	95.97	10.32	42.72	10.88	21.31	84.25	75.33	69.35	76.31
-15	28.04	24.17	37.22	29.81	8.17	2.65	53.66	21.49	67.30	20.87	69.03	52.40
-10	20.75	17.85	24.68	21.09	2.18	2.29	4.98	3.15	14.39	26.45	18.20	19.68
-5	17.66	21.17	16.15	18.33	2.32	2.43	2.56	2.44	16.02	16.42	17.08	16.51
0	16.00	16.62	17.98	16.87	2.26	2.36	2.39	2.34	21.92	24.00	17.18	21.03
5	17.92	22.51	20.53	20.32	2.21	2.70	2.58	2.50	13.02	19.56	12.95	15.18
10	25.93	31.44	22.92	26.76	4.01	4.99	5.10	4.70	16.37	30.45	49.42	32.08
15	21.80	30.40	38.00	30.07	3.07	61.03	39.64	34.58	18.26	60.33	59.22	45.94
20	110.51	88.51	124.69	107.90	10.74	8.47	15.81	11.67	86.66	132.82	106.81	108.76
Vertical	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
-30	26.90	28.55	31.17	28.87	2.32	2.31	2.54	2.39	14.57	17.87	15.24	15.89
-25	17.56	22.42	25.91	21.96	2.20	2.21	2.12	2.18	15.13	15.46	14.05	14.88
-20	20.45	18.72	22.45	20.54	2.19	2.98	2.27	2.48	15.63	26.35	19.46	20.48
-15	17.41	19.18	18.87	18.49	2.24	2.14	2.05	2.14	14.37	17.40	15.05	15.61
-10	18.87	18.46	20.14	19.16	2.05	2.09	2.38	2.17	15.05	13.28	25.66	18.00
-5	16.97	17.73	16.50	17.07	2.15	2.32	2.06	2.18	13.21	27.41	21.70	20.77
0	16.57	15.69	15.29	15.85	2.04	2.77	3.42	2.74	19.63	19.55	16.85	18.68
5	16.32	17.25	18.30	17.29	2.09	4.01	2.43	2.84	14.36	14.84	15.25	14.82
10	35.35	37.11	39.30	37.25	22.11	15.33	23.93	20.46	42.75	37.59	46.32	42.22

Looking at the results in table 3.5 and figure 3.17, it can be concluded that accurate results are obtained up to a horizontal rotation of 15° in both directions. When looking at the precision, it is only valid up to a 10° rotation in both directions.

For the vertical rotation, it can be seen that an accurate and precise measurement can be taken up to an angle of 30° downwards. For the upward rotation, only a 5° angle applies, before the measurements are no longer accurate or precise. This is presumable because the cheeks and eyelashes will block the IR-light reflected by the eyes.

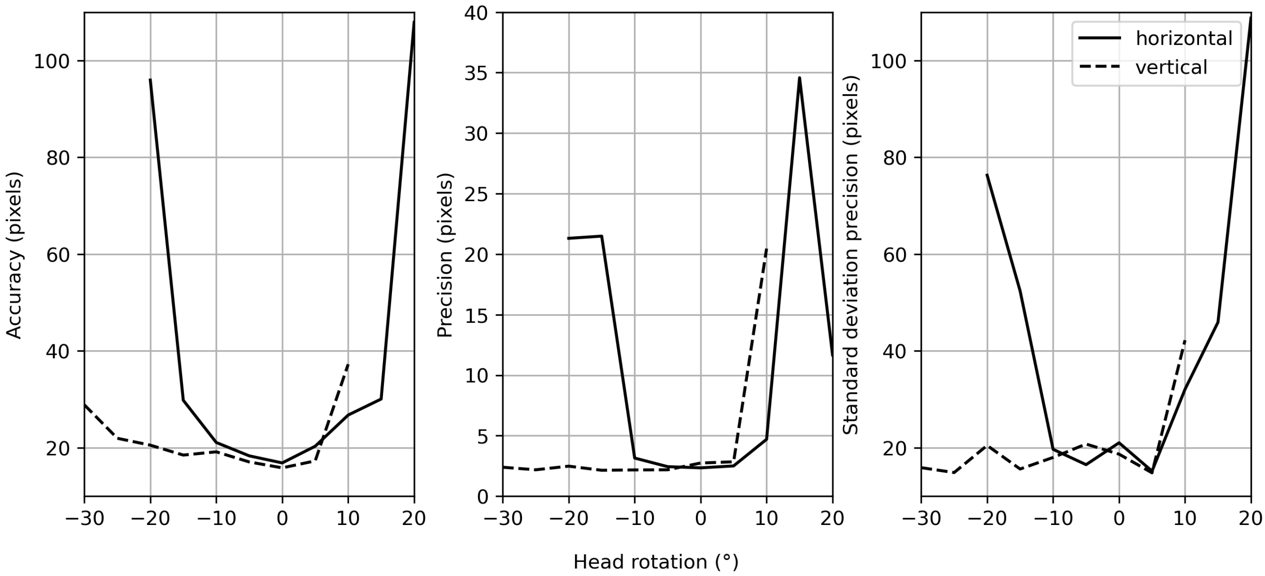


Figure 3.17: Plot of accuracy, precision and standard deviation precision in pixels for horizontal and vertical head rotation.

Different types of glasses and lenses

During these tests, the test subject will put on different types of glasses or lenses, this allows to see if certain conditions are not suitable for eye tracking or provide less data quality. Table 3.6 shows all the different conditions in which these tests were conducted. The results of these tests are shown in table 3.7.

Table 3.6: Explanation of all different conditions.

Condition 1	Calibration with glasses, measurement without glasses
Condition 2	Calibration without glasses, measurement with glasses
Condition 3	Reflecting sunglasses
Condition 4	Sunglasses
Condition 5	Lenses
Condition 6	Blue light filter glasses
Condition 7	Safety glasses
Condition 8	Calibration with reading glasses, measurement without
Condition 9	Calibration without reading glasses, measurement with
Condition 10	One eye taped off

Looking at the results in table 3.7, it can be seen that when the test subject does the calibration without glasses and wears glasses during the measurements that the precision is lower than in the other tests. With this test, the accuracy is also slightly lower.

When looking at the test with the reflective sunglasses, it is noticeable that both the accuracy and precision are low (figure 3.18). From this, it can be concluded that these are not applicable to eye tracking.

Finally, the accuracy is slightly lower for people who would only have one eye.

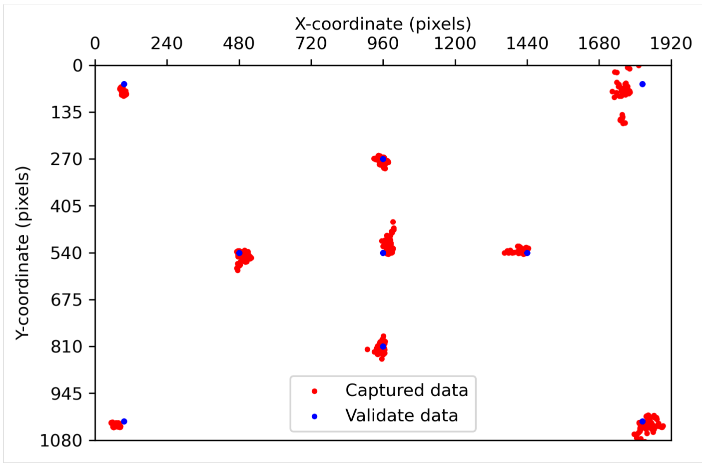


Figure 3.18: Reflecting sunglasses (left), measured and validation data plot obtained by eye tracking data during tests with reflecting sunglasses (right).

Table 3.7: Different types of glasses and lenses test results in pixels.

Condition	Accuracy (pixels)				Precision (pixels)				Standard deviation precision (pixels)			
	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.	Test 1	Test 2	Test 3	Avg.
Condition 1	18.50	17.23	16.85	17.53	2.11	2.54	2.06	2.24	15.14	19.19	13.71	16.01
Condition 2	20.22	16.85	19.80	18.96	5.51	2.57	9.20	5.76	37.42	23.68	24.50	28.53
Condition 3	30.90	25.66	33.21	29.92	36.57	44.17	35.41	38.72	63.86	60.68	92.02	72.19
Condition 4	16.33	14.57	19.98	16.96	2.38	2.33	2.70	2.47	17.01	21.93	22.41	20.45
Condition 5	11.98	17.67	13.05	14.23	2.50	2.81	2.46	2.59	21.12	20.31	15.36	18.93
Condition 6	15.17	14.30	18.09	15.85	2.61	2.34	2.25	2.40	14.58	19.28	21.38	18.41
Condition 7	15.64	12.46	11.82	13.31	2.42	2.52	3.07	2.67	20.50	19.65	21.35	20.50
Condition 8	13.76	13.75	13.56	13.69	2.24	2.23	2.05	2.17	16.04	12.08	11.36	13.16
Condition 9	22.32	21.21	21.25	21.59	2.46	2.18	2.55	2.40	11.26	10.73	12.75	11.58
Condition 10	25.47	19.71	25.89	23.69	3.27	3.29	2.87	3.14	20.65	26.97	26.00	24.54

Chapter 4

Test application

4.1 Introduction

This test application is made to see whether an eye tracker offers added value when going through an instruction set provided by Arkite. To go to the next/previous instruction, there are interactive buttons that can be operated by both eye tracking and the mouse. Test subjects will go through the instruction set both with the eye tracker and the mouse. Afterwards, they fill out a User Experience Questionnaire for both cases. These are compared and from this the decision is made whether the eye tracking has added value or not. For the UEQ 25 people with an age between 20 and 67 are used of which 14 are men and 11 are women. They maintained the same posture throughout the test, their starting position is at the center of the screen, 34.5 cm above the eye tracker and an operating distance of 65 cm.

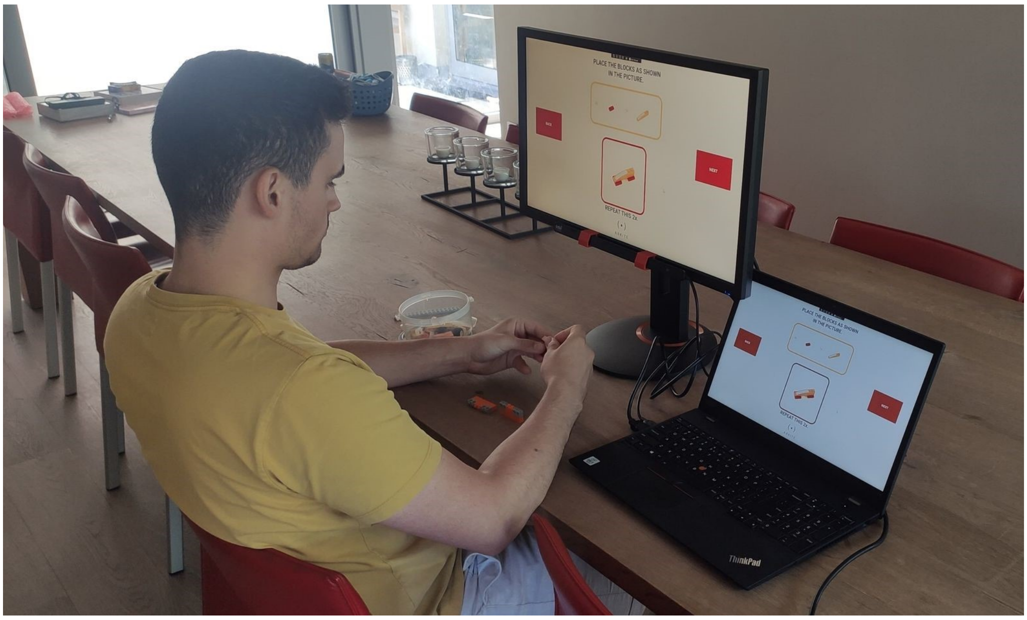


Figure 4.1: Test subject going through the instruction set of the test application.

This test application is developed in C# using Visual Studio. A WPF (windows presentation foundation) is used to create the Windows desktop app. To connect with the Tobii 4L eye tracker, the Tobii stream engine is used as explained earlier. In appendix B screenshots of the test application can be found.

4.2 Test application

When the user opens the application, a home screen is shown. On this screen, the user can click a button to connect with the eye tracker. This button will enter the subscribe phase of the program.

4.2.1 Subscribe

The first step for using the eye tracker is the subscribe function. This function works as follows:

1. Create an API context.
2. Enumerate devices to find connected eye tracker URLs.
3. Create a device context with the API, device URL and license.
4. Calculate the display area with the given screen dimensions.
5. Subscribe to gaze data.

(The code of this function can be found in appendix A.)

To subscribe to the gaze data, the program has to make use of the Tobii stream engine API. In this API, a designated function for subscribing can be found. When calling this function, a device context together with a callback function must be provided. This callback is a function pointer to the OnGazePoint function and will be called with each new gaze data. The OnGazePoint function can be seen in listing 4.1.

```
private static tobii_gaze_point_callback_t newOnGazePoint =
                new tobii_gaze_point_callback_t(OnGazePoint);

private static void OnGazePoint(ref tobii_gaze_point_t gazePoint,
                                IntPtr userData){
    // Check that the data is valid before using it
    if (gazePoint.validity ==
        tobii_validity_t.TOBII_VALIDITY_VALID){
        coordinaat_x = gazePoint.position.x;
        coordinaat_y = gazePoint.position.y;
        timestamp = gazePoint.timestamp_us;
    }
}
```

Listing 4.1: OnGazePoint function.

When the subscribe phase is done, the user can choose to calibrate or to start the test application. The calibration option is highly recommended to improve the experience of the program.

4.2.2 Calibrate

For the calibration step in this application, the decision is made to only use five calibration points. A trade of is made between using the optimal amount of points (eight points) versus the time needed to calibrate. A five point calibration is chosen because the accuracy difference only consisted of 1.08 pixels, but the duration of the calibration is 10.2 seconds faster (eight point calibration duration = 27.2 s and five point calibration duration = 17.0 s).

When calibrating, the operator must first press a button with the mouse to start the calibration process. Once the calibration is started, the first calibration point will appear. A red circle is shown on the screen and will shrink in size during a certain amount of time to get the focus of the test subject's eyes on a particular point. Once the size hits a certain point, the shrinking will stop and the color will change to green. If this happens, the calibration of this certain point will be captured and linked to the given coordinates of the point. When this is all finished, the second calibration point will appear on-screen and go through the same process as mentioned before. This is done for all five calibration points. When the last calibration point is captured, the calibration screen closes and the user is directed back to the home screen of the application.

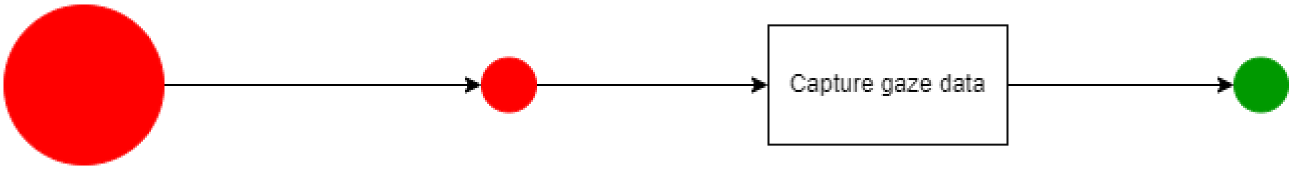


Figure 4.2: Visual process of a calibration point.

4.2.3 Instruction Set

Back at the home screen, the user has the option to press a button to start the test or press a button to calibrate again. There is also an option to disconnect from the eye tracker. When the "START DEMO" button is pressed, a set of instructions will start to appear. To illustrate an instruction set from Arkite, the instruction of a small Lego build are shown (figure 4.3). The user is provided with all the necessary blocks to start this test. On each instruction page, the blocks that are needed are shown together with the final result of this page. Once the instruction on the certain page is followed, the user can then navigate to the next step or go back one step.

To do this, the user can either click with the mouse on the big red buttons or use their gaze direction to activate these buttons. During the test, after the eye tracker has been activated, a small outline of a circle is shown. This circle represents the gaze direction of the user. This is included to give the user direct feedback to where they are looking to increase the user experience.

Once the user starts looking at one of the buttons, the button will start to change color. When the user looks long enough, the button will eventually turn green. When the button turns green, it will be activated and the next instruction will appear (or the previous). The color transition is added to give the user feedback and let them know that the user is doing it right. During this process, if the user looks outside the button, the transition will reset and the button will become red again. To avoid this from happening when the user accidentally looks outside the button for a fraction of a second or when a small error occurs, an average of the gaze is taken to decide whether the transition has to be reset or not.

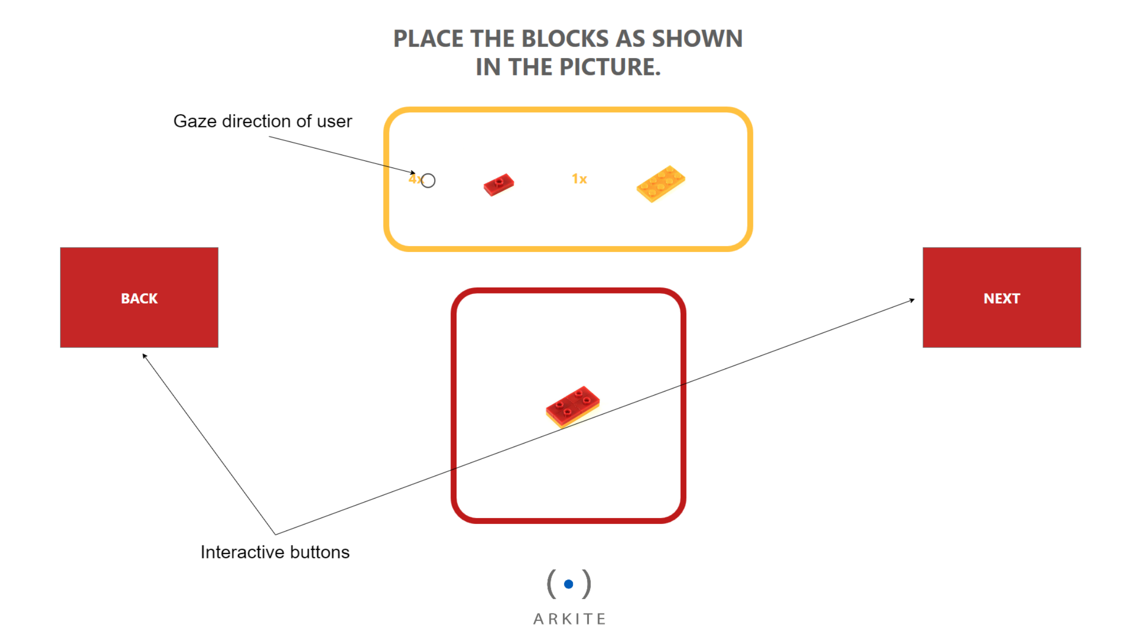


Figure 4.3: Example of instruction screen.

This average is the moving average of the last 150 gaze points obtained. When a new data point is acquired, the oldest data point will be removed from the list and the newest will be added to the end of the list. A visual representation of the moving average can be seen in figure 4.4.

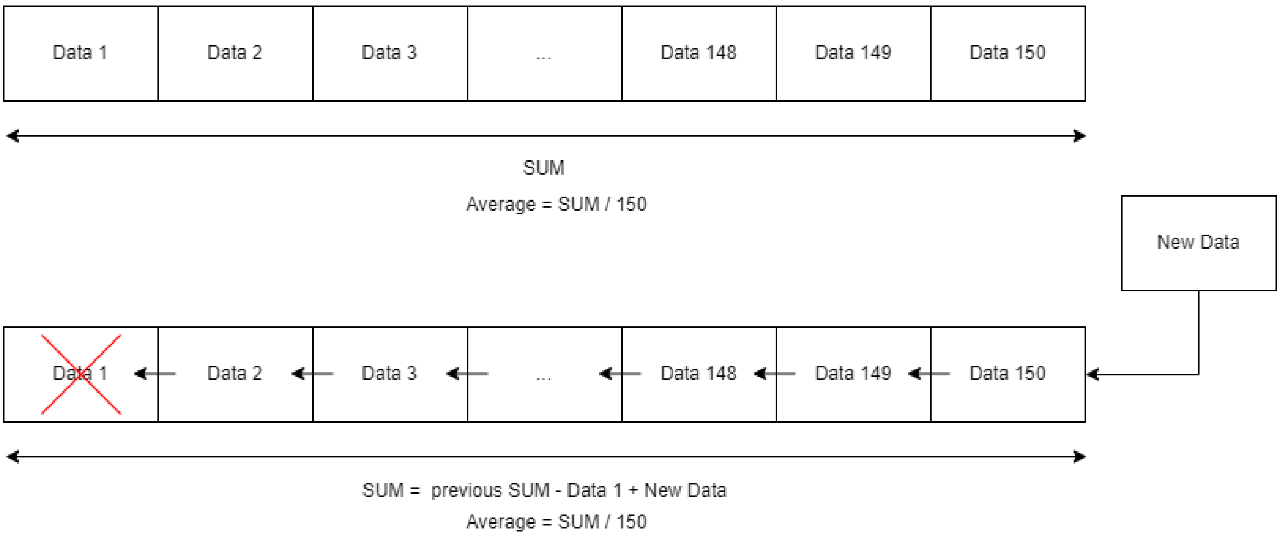


Figure 4.4: Visual representation of moving average.

After each instruction screen is completed, the program will export an Excel file with all the gaze points x and y coordinates recorded during this instruction. This can then be converted into a heatmap to get a visual representation of where the user is looking during a given instruction, as seen in figure 4.5.

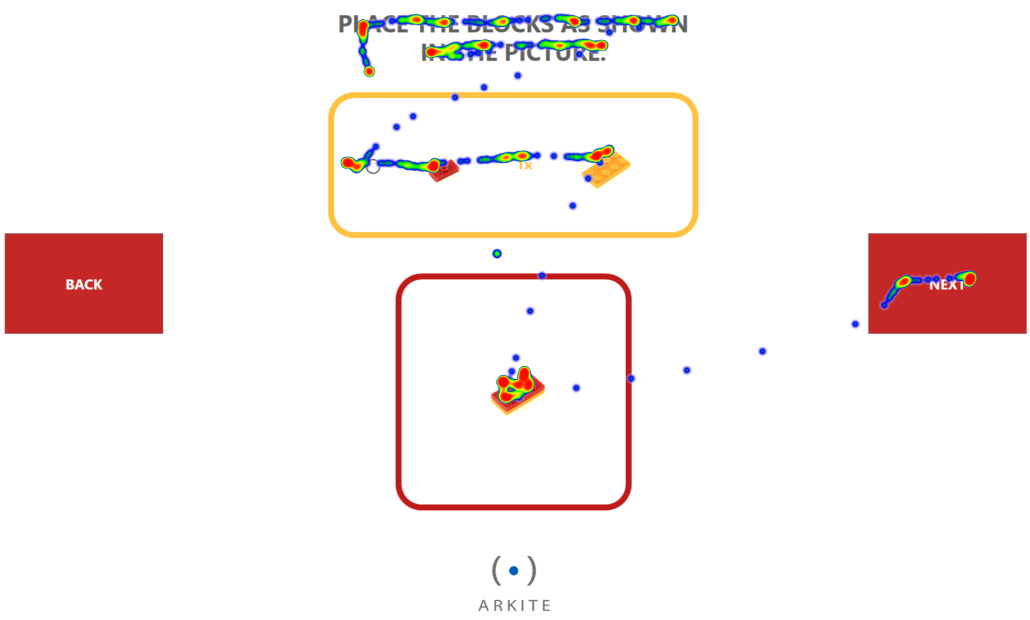


Figure 4.5: Heatmap obtained by data of instruction one.

4.3 User Experience Questionnaire

An important part of this research is to know whether the integration of eye tracking will have a positive impact on the user experience during the completion of the instruction set. To put this to the test, a comparison is made between the eye tracking workflow and the mouse-only workflow. This comparison is made by using an User Experience Questionnaire (UEQ). The UEQ used in this research contains 6 scales with 26 items:

1. Attractiveness: Overall impression of the product. Do users like or dislike the product?
2. Perspicuity: Is it easy to get familiar with the product? Is it easy to learn how to use the product?
3. Efficiency: Can users solve their tasks without unnecessary effort?
4. Dependability: Does the user feel in control of the interaction?
5. Stimulation: Is it exciting and motivating to use the product?
6. Novelty: Is the product innovative and creative? Does the product catch the interest of users?

In figure 4.6 the assumed scale structure of this UEQ and all the items per scale can be seen. The list of questions presented to the test subjects can be seen in appendix C.

4.3.1 UEQ results

After the test subjects have completed the test application in both versions, an analysis is made of the results. This is obtained by filling in the answers of the questionnaire in an Excel sheet that is provided by the author of paper [6]. This Excel tool then automatically calculates all statistics necessary to interpret the results.

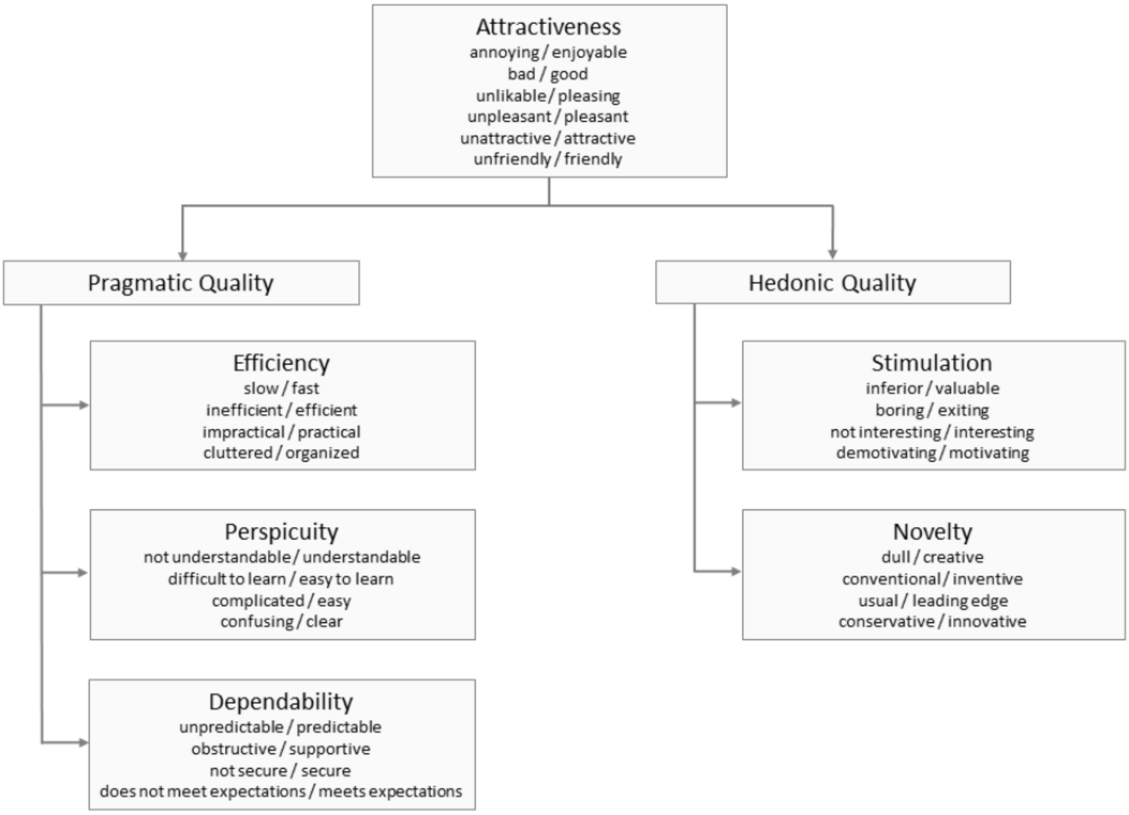


Figure 4.6: Assumed scale structure of the UEQ [6].

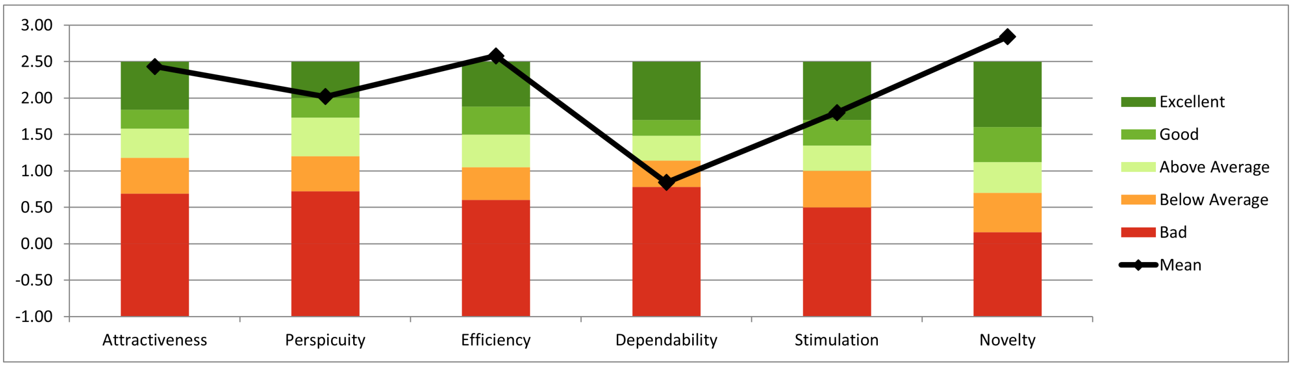


Figure 4.7: Benchmark graph for eye tracking test application.

Figure 4.7 shows that the eye tracker version scores excellent on all scales except the dependability scale. This can be explained because this is a new way of interacting with the user interface. This does not yet feel familiar for most users, and therefore the score on the dependability scale will be below average. It can be seen that if the test users complete the test more than once, the score of the dependability scale will increase. This will happen because the user gets more familiar with the eye tracking workflow.

The next step in the analysis is to compare the user experience of the eye tracker to the user experience of the mouse-only version. To do this, the same UEQ was executed for both workflows. Each test subject must complete one instruction set with the eye tracking integration and one instruction set with only the mouse. To compare them both, a graph is made with the data generated by the provided Excel file. As shown in figure 4.8, the eye tracking version scores better for all scales, with an exception of Dependability, than the mouse-only version.

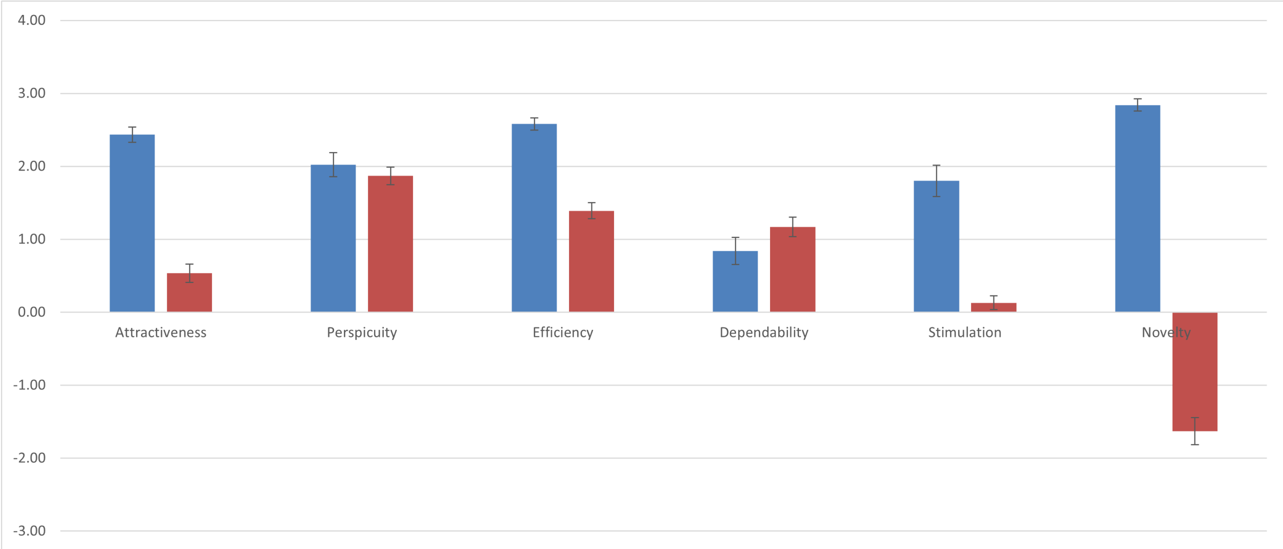


Figure 4.8: Comparison of eye tracker version and mouse-only version.

4.4 Time difference

To test whether using an eye tracker would be an improvement to the instruction set, 25 test subjects completed the test with and without the eye tracker. The test consisted of the same Lego build as mentioned before. They first completed the instruction set by only using the mouse. The test with the eye tracker was tested a few days later to rule out memorization of the Lego set. The results of these test can be seen in table 4.1.

Table 4.1: Time difference between eye tracker version and mouse-only version

Test subject	time eye tracker version (s)	time mouse-only version (s)	Difference (s)
Test subject 1	89	118	29
Test subject 2	95	142	47
Test subject 3	131	143	12
Test subject 4	103	132	29
Test subject 5	109	127	18
Test subject 6	108	132	24
Test subject 7	132	145	13
Test subject 8	152	143	-9
Test subject 9	92	134	42
Test subject 10	125	139	14
Test subject 11	121	118	-3
Test subject 12	136	134	-2
Test subject 13	135	119	-16
Test subject 14	141	144	3
Test subject 15	143	155	12
Test subject 16	125	152	27
Test subject 17	150	147	-3
Test subject 18	95	155	60
Test subject 19	99	133	34
Test subject 20	112	156	44
Test subject 21	143	130	-13
Test subject 22	159	156	-3
Test subject 23	100	151	51
Test subject 24	105	121	16
Test subject 25	130	121	-9
Average time	121.2	137.88	16.68

From this table can be concluded that the average time difference is positive, meaning that the eye tracker version is quicker. Eight of the 25 test subjects took longer to complete the eye tracker version, with a maximum of 16 seconds. This could happen because these users were either more accustomed to using a mouse. During testing, a lot of people were struggling with attaching some Lego blocks. This could also affect the duration of the tests. Overall, a positive difference in time can be seen from the eye tracking approach, with an average time difference of +16.68 seconds.

4.5 Optimization

To further optimize the experience with the eye tracking in these instruction sets, the head position of the operator can be useful. The Tobii stream engine API offers a function to retrieve this data. A position of the head relative to the center of the tracker can be obtained. During previous tests, it can be seen that there is an optimal working area for the head position to be in. In this area, the highest accuracy and precision of gaze direction will be obtained. So a further improvement could be to keep track of the head position and notify the user when their head position is outside the working area.

Chapter 5

Conclusion

The aim of this thesis was to research on the optimal setup for eye tracking and to optimize the instruction sets of Arkite with an eye tracker. For this, several tests have been established such as amount of calibration points, luminous intensity, horizontal, vertical and operating distance, head rotation and different types of glasses and lenses. These results were used to create an environment that was used during the test application. This test application was run both with eye tracking and mouse-only. These two cases were compared with each other by means of an User Experience Questionnaire (UEQ).

The results of the tests show that the highest accuracy and precision is achieved with an eight point calibration and when no direct sunlight falls on the eye tracker or eyes of the test person. The perfect working area for the Tobii 4L eye tracker is within ten cm to the left and to the right of the origin on the x-axis. The most optimal position is on the origin. For the y-axis, the working area is between 24.5 cm to 44.5 cm above the eye tracker. The most optimal position is 34.5 cm above the eye tracker. The operating distance (z-axis) will be optimal at 65 cm to a maximum of 95 cm away from the tracker. For the optimal head rotation, the operator can have a head rotation of max 15° in both directions. For the vertical head rotation, the operator can have a downwards angle of max 30° and an upwards angle of max 5°. The results also show that when the operator is wearing glasses, it is most optimal to also wear the same glasses during calibration. This will result in the best accuracy and precision. Reflective sunglasses are not recommended due to their low accuracy and precision.

The eye tracker version scores excellent on all scales of the UEQ, except on the dependability scale. This can be explained because this is a new way of interacting with the user interface and does not yet feel familiar for most users. When comparing the eye tracker version and the mouse-only version, it can be seen that the eye tracker version scores better for all scales, with an exception of dependability.

As a final reflection, it can be stated that eye tracking definitely is an overall improvement. It will speed up the hole process and make it easier for the operators to keep using their hands while navigating through all the different instructions. The only downside to this approach will be the learning curve. This will always be larger than just using a mouse. When the operator has never used this form of navigating before, it can be slower than using a mouse-only version.

Bibliography

[1] "Tobii 4l product sheet." https://www.tobii.com/siteassets/tobii-tech/4L-product-sheet.pdf?v=1. Accessed: 2022-03-21.

[2] "Dark and bright pupil tracking." https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/what-is-dark-and-bright-pupil-tracking/. Accessed: 2022-07-18.

[3] Wartenberg, C and Holmqvist, K, "Daily newspaper layout – designers' predictions of readers' visual behaviour - a case study," *Lund University Cognitive Science*, vol. 126, 2005.

[4] D. Salvucci and J. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on eye tracking research applications*, ETRA '00, pp. 71–78, ACM, 2000.

[5] S. Eraslan, Y. Yesilada, and S. Harper, "Eye tracking scanpath analysis techniques on web pages: A survey, evaluation and comparison," *Journal of Eye Movement Research*, vol. 9, Dec. 2015.

[6] Schrepp ,M, "User Experience Questionnaire Handbook Version 2," 2016.

[7] Carter, Benjamin T. and Luke, Steven G., "Best practices in eye tracking research," *International Journal of Psychophysiology*, vol. 155, pp. 49–62, 2020.

[8] C. Klein and U. Ettinger, *Eye Movement Research: An Introduction to Its Scientific Foundations and Applications*. Studies in neuroscience, psychology and behavioral economics, Cham: Springer International Publishing AG, 2019.

[9] B. Smolka, K. Czubin, J. Hardeberg, K. Plataniotis, M. Szczepanski, and K. Wojciechowski, "Towards automatic redeye effect removal," *Pattern recognition letters*, vol. 24, no. 11, pp. 1767–1785, 2003.

[10] I. Rehman, N. Mahabadi, M. Motlagh, and T. Ali, "Anatomy, Head and Neck, Eye Fovea," *StatPearls*, 2021.

[11] K. Rayner, "The 35th sir frederick bartlett lecture: Eye movements and attention in reading, scene perception, and visual search," *Quarterly journal of experimental psychology (2006)*, vol. 62, no. 8, pp. 1457–1506, 2009.

[12] "Welcome to tobii tech." https://tech.tobii.com/. Accessed: 2022-02-17.

[13] "Eye trackers and eye tracking hardware - sr research." https://www.sr-research.com/hardware/. Accessed: 2022-02-17.

[14] K. Harezlak and P. Kasprowski, “Application of eye tracking in medicine: A survey, research issues and challenges,” *Computerized medical imaging and graphics*, vol. 65, pp. 176–190, 2017.

[15] R. Romanowski, M. Lewicki, W. Ciechomski, and E. Baron, *Managing Economic Innovations - methods and instruments*. Poznań: Bogucki, 2019.

[16] F. Merkle, D. Kurtovic, C. Starck, C. Pawelke, S. Gierig, and V. Falk, “Evaluation of attention, perception, and stress levels of clinical cardiovascular perfusionists during cardiac operations: a pilot study,” *Perfusion*, vol. 34, no. 7, pp. 544–551, 2019.

[17] “Use eye tracking to cut training time in half — find out how.” https://www.tobiipro.com/applications/industry-human-performance/case-studies/global-car-parts-giant-denso-halves-training-time-thanks-to-eye-tracking/. Accessed: 2022-02-14.

[18] Nrulatifa, H and Hartanto, R and Wibirama, S, “Spontaneous Gaze-Based Application Using 2D Correlation and Velocity Threshold Identification,” *5th International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 75–80, 2021.

[19] R. S. Hessels, C. Kemner, C. van den Boomen, and I. T. C. Hooge, “The area-of-interest problem in eyetracking research: A noise-robust solution for face and sparse stimuli,” *Behavior research methods*, vol. 48, no. 4, pp. 1694–1712, 2016.

[20] A. T. Duchowski, “Gaze-based interaction: A 30 year retrospective,” *Computers graphics*, vol. 73, pp. 59–69, 2018.

[21] D. Noton and L. Stark, “Eye movements and visual perception,” *Scientific American*, vol. 224, no. 6, pp. 34–43, 1971.

[22] “An introduction to the second edition of the cambridge handbook of expertise and expert performance: Its development, organization, and content,” 2018.

[23] A. T. Duchowski, N. Cournia, and H. Murphy, “Gaze-contingent displays: A review,” *Cyberpsychology behavior*, vol. 7, no. 6, pp. 621–634, 2004.

[24] “Stream engine - tobii developer.” https://developer.tobii.com/product-integration/stream-engine/?cv=1session-id=8be5bdd8bf804028b0d71335ab32321f. Accessed: 2022-07-18.

[25] “Accuracy and precision test method for remote eye trackers.” https://www.tobiipro.com/siteassets/tobii-pro/learn-and-support/use/what-affects-the-performance-of-an-eye-tracker/tobii-test-specifications-accuracy-and-precision-test-method.pdf/?v=2.1.1. Accessed: 2022-07-18.

Appendix A

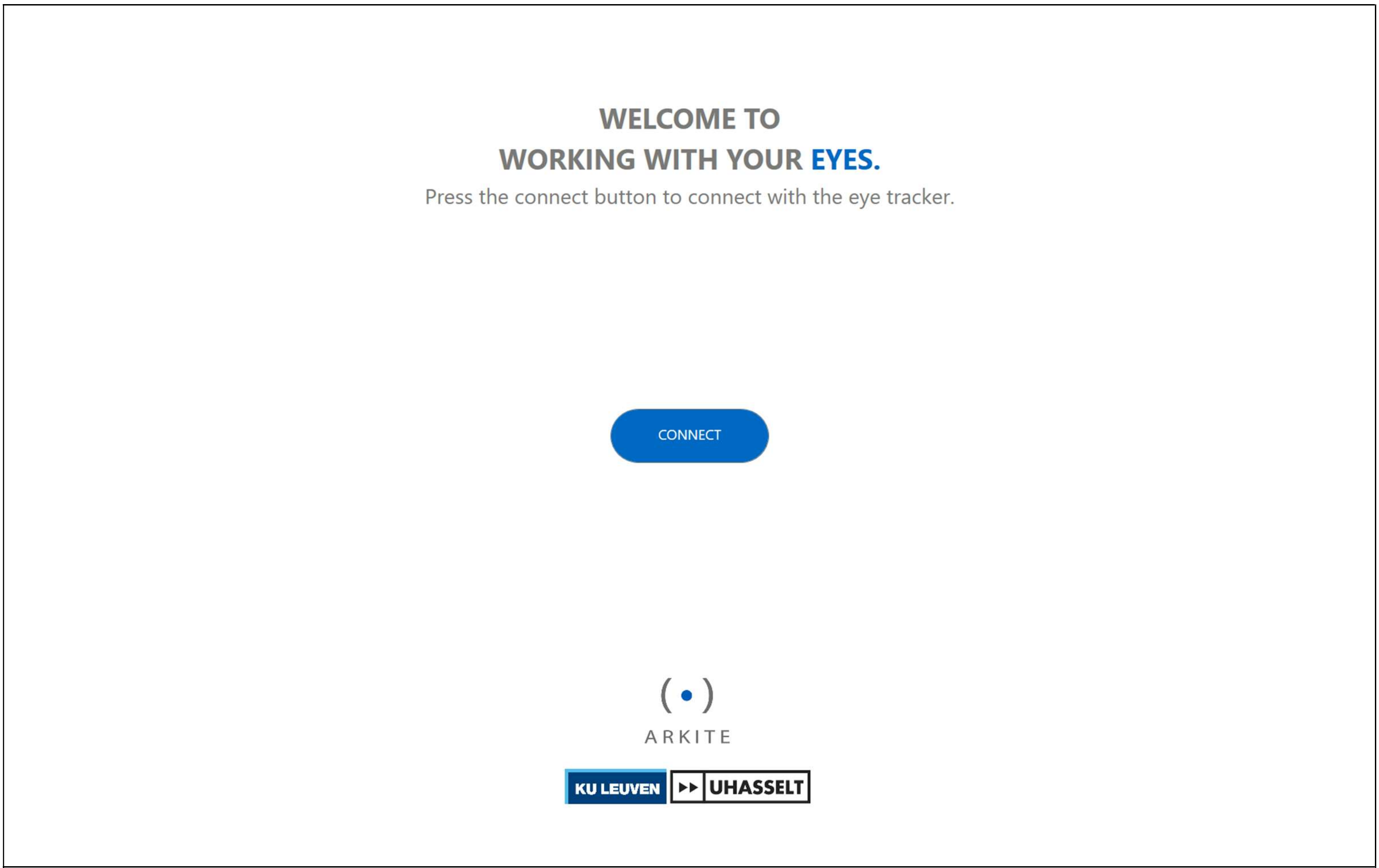
Appendix - Code subscribe function

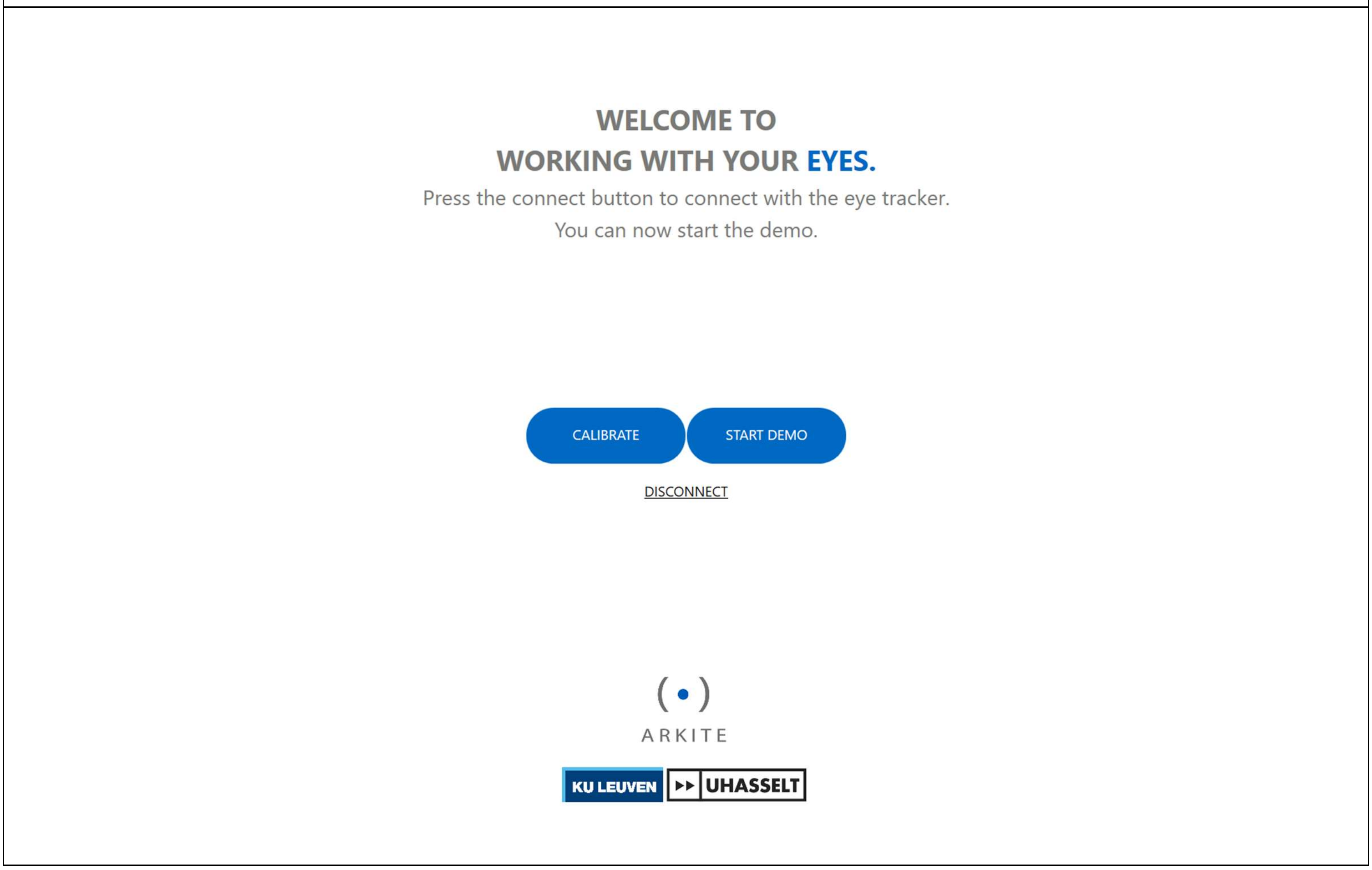
```
public static IntPtr CreateTrackerWithLicense(IntPtr api, string url, string licenseFileName){
    var licenseResults = new List<tobii_license_validation_result_t>();
    var licenseContents = File.ReadAllText(licenseFileName, Encoding.Unicode);
    var resultCode = Interop.tobii_device_create_ex(
        api,
        url,
        Interop.tobii_field_of_use_t.TOBII_FIELD_OF_USE_STORE_OR_TRANSFER_FALSE,
        new[] { licenseContents },
        licenseResults,
        out var device);

    Debug.Assert(resultCode == tobii_error_t.TOBII_ERROR_NO_ERROR);
    return device;
}
```

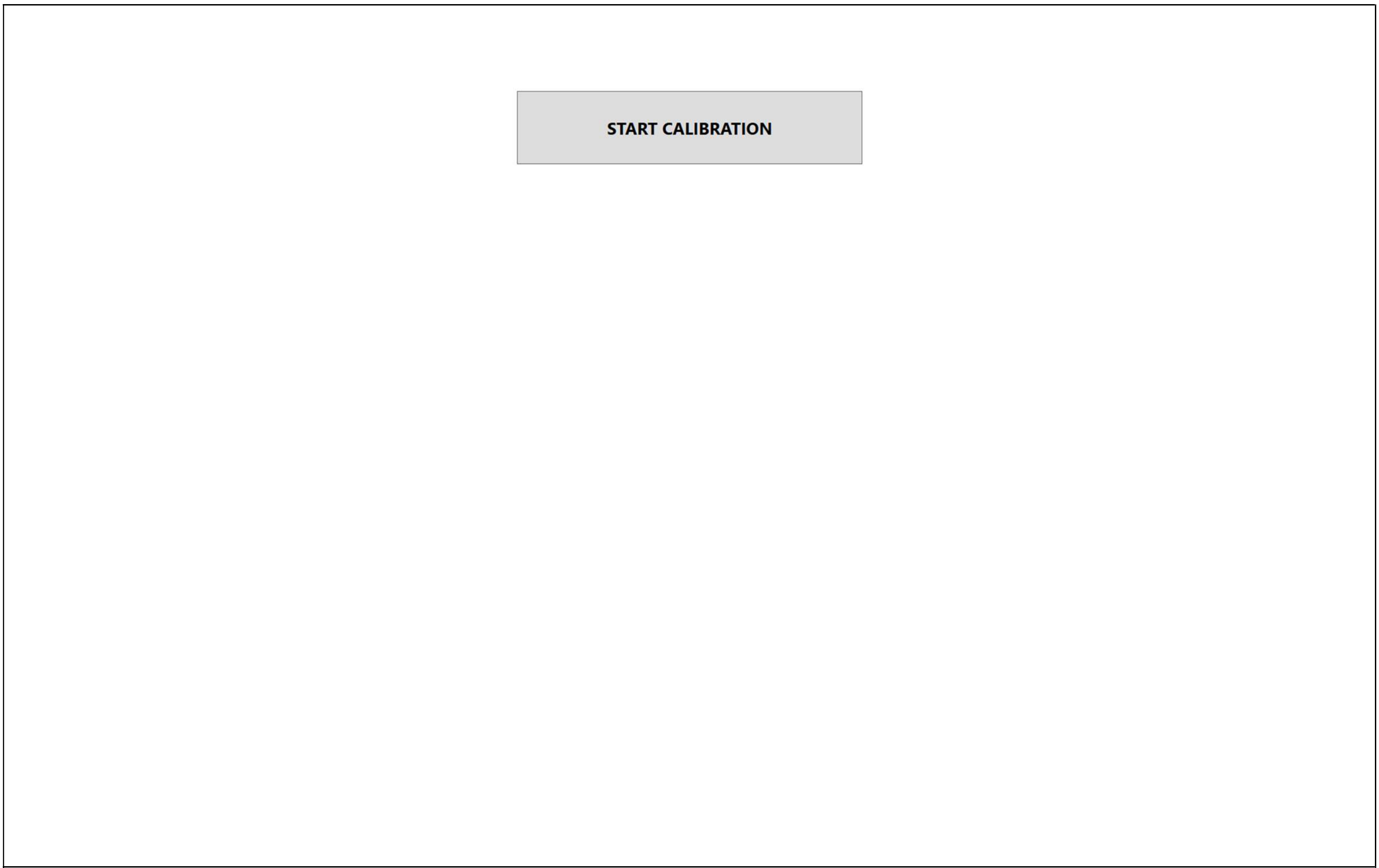
```
public static Tuple<IntPtr, IntPtr> subscribe(){

    // Create API context
    IntPtr apiContext;
    tobii_error_t result = Interop.tobii_api_create(out apiContext, null);
    Debug.Assert(result == tobii_error_t.TOBII_ERROR_NO_ERROR);

    // Enumerate devices to find connected eye trackers
    List<string> urls;
    result = Interop.tobii_enumerate_local_device_urls(apiContext, out urls);
    Debug.Assert(result == tobii_error_t.TOBII_ERROR_NO_ERROR);
    if (urls.Count == 0)
    {
        Console.WriteLine("Error: No device found");
        return null;
    }

    string licensePath = pathOfLicense
    IntPtr deviceContext;
    deviceContext = CreateTrackerWithLicense(apiContext,urls[0],licensePath);
    Debug.Assert(result == tobii_error_t.TOBII_ERROR_NO_ERROR);

    // Calcualte display area
    // MONITOR SCREEN SIZE
    float width_mm = (float)535.0;
    float height_mm = (float)300.0;
    float offset_x_mm = (float)0.0;

    result = Interop.tobii_calculate_display_area_basic(deviceContext,
   width_mm, height_mm, offset_x_mm, ref geometry, out displayArea);
    Debug.Assert(result == tobii_error_t.TOBII_ERROR_NO_ERROR);

    // Set display area
    result = Interop.tobii_set_display_area(deviceContext, ref displayArea);
    Debug.Assert(result == tobii_error_t.TOBII_ERROR_NO_ERROR);

    // Subscribe to gaze data
    result = Interop.tobii_gaze_point_subscribe(deviceContext,newOnGazePoint);
    Debug.Assert(result == tobii_error_t.TOBII_ERROR_NO_ERROR);

    return Tuple.Create(deviceContext, apiContext);
}
```

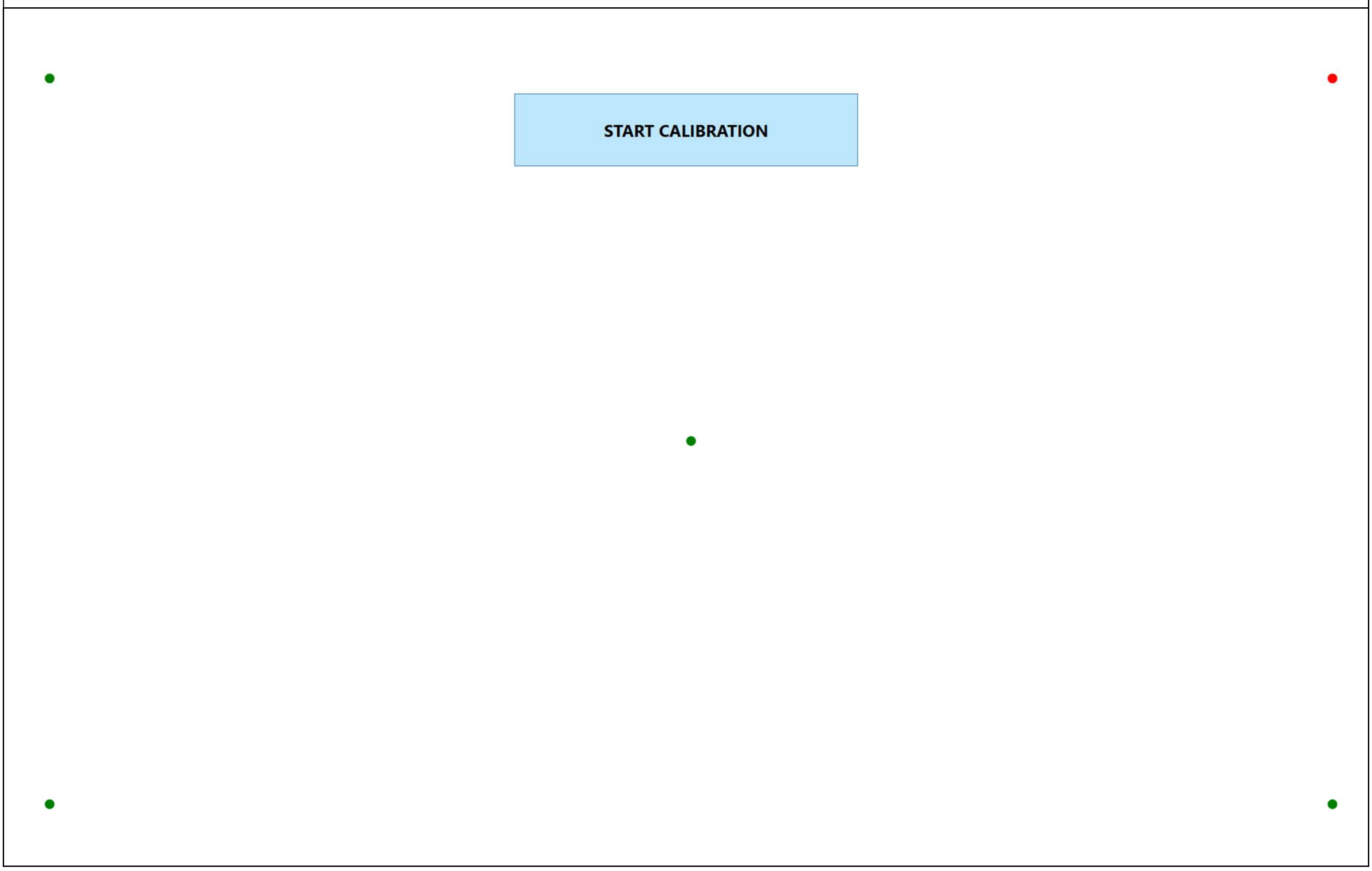
Appendix B

Appendix - Test Application All Screens


WELCOME TO
WORKING WITH YOUR EYES.
Press the connect button to connect with the eye tracker.
CONNECT
ARKITE
KU LEUVEN
UHASSELT


WELCOME TO
WORKING WITH YOUR EYES.
Press the connect button to connect with the eye tracker.
You can now start the demo.
CALIBRATE
START DEMO
DISCONNECT
ARKITE
KU LEUVEN
UHASSELT


START CALIBRATION


START CALIBRATION

LOOK AT THE CENTER PICTURE
TO START.

START DEMO

ARKITE

KU LEUVEN UHASSELT

PLACE THE BLOCKS AS SHOWN
IN THE PICTURE.

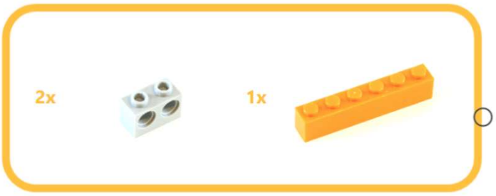
4x  1x 

BACK

NEXT

ARKITE

PLACE THE BLOCKS AS SHOWN IN THE PICTURE.

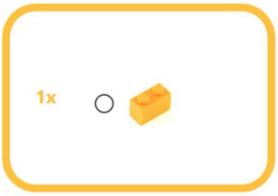


BACK

NEXT

ARKITE

PLACE THE BLOCKS AS SHOWN IN THE PICTURE.



BACK

NEXT

REPEAT LAST 2 STEPS.

ARKITE

PLACE THE BLOCKS AS SHOWN IN THE PICTURE.

4x  1x 

BACK

NEXT

REPEAT THIS 2x.

ARKITE

PLACE THE BLOCKS AS SHOWN IN THE PICTURE.

2x  2x 

BACK

NEXT

REPEAT THIS 2x.

ARKITE

PLACE THE BLOCKS AS SHOWN IN THE PICTURE.

2x 

BACK

NEXT

ARKITE

PLACE THE BLOCKS AS SHOWN IN THE PICTURE.

2x 

BACK

NEXT

ARKITE

Appendix C

Appendix - User Experience Questionnaire in Dutch and English

Maak dan nu uw evaluatie.

Voor de beoordeling van het product, vragen we u de onderstaande vragenlijst in te vullen. De vragenlijst bestaat uit twee tegengestelde eigenschappen die van toepassing zijn op het product. De rondjes staan voor verschillende gradaties. U kunt uw beoordeling geven door het rondje, die het meest uw indruk weerspiegelt, aan te vinken.

<u>Voorbeeld:</u>

aantrekkelijk	○	⊗	○	○	○	○	○	onaantrekkelijk

Dit antwoord zou betekenen dat u het product beoordeelt als meer aantrekkelijk dan onaantrekkelijk.

Graag uw eerst ingeving invullen. Wacht niet te lang met invullen om te voorkomen dat u gaat twijfelen over uw eerste ingeving.

Soms bent u misschien niet helemaal zeker van uw antwoord of u vindt de eigenschap niet volledig van toepassing, kruis dan toch een rondje aan.

Het is uw mening die telt. Let op: er is geen goed of fout antwoord!

Gelieve beoordeel het product nu door het aanvinken van een rondje per regel.

	1	2	3	4	5	6	7		
onplezierig	○	○	○	○	○	○	○	plezierig	1
onbegrijpelijk	○	○	○	○	○	○	○	begrijpelijk	2
creatief	○	○	○	○	○	○	○	saai	3
makkelijk te leren	○	○	○	○	○	○	○	moeilijk te leren	4
waardevol	○	○	○	○	○	○	○	inferieur	5
vervelend	○	○	○	○	○	○	○	spannend	6
oninteressant	○	○	○	○	○	○	○	interessant	7
onvoorspelbaar	○	○	○	○	○	○	○	voorspelbaar	8
snel	○	○	○	○	○	○	○	langzaam	9
origineel	○	○	○	○	○	○	○	conventioneel	10
belemmerend	○	○	○	○	○	○	○	ondersteunend	11
goed	○	○	○	○	○	○	○	slecht	12
complex	○	○	○	○	○	○	○	eenvoudig	13
afstotend	○	○	○	○	○	○	○	aantrekkelijk	14
gebruikelijk	○	○	○	○	○	○	○	nieuw	15
onaangenaam	○	○	○	○	○	○	○	aangenaam	16
vertrouwd	○	○	○	○	○	○	○	niet vertrouwd	17
motiverend	○	○	○	○	○	○	○	demotiverend	18
volgens verwachtingen	○	○	○	○	○	○	○	niet volgens verwachtingen	19
inefficiënt	○	○	○	○	○	○	○	efficiënt	20
overzichtelijk	○	○	○	○	○	○	○	verwarrend	21
onpraktisch	○	○	○	○	○	○	○	praktisch	22
ordelijk	○	○	○	○	○	○	○	rommelig	23
aantrekkelijk	○	○	○	○	○	○	○	onaantrekkelijk	24
aardig	○	○	○	○	○	○	○	onaardig	25
conservatief	○	○	○	○	○	○	○	innovatief	26

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

<u>Example:</u>

attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don’t think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26