UNIVERSITY
OF OULU
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Kaarlo Heikkilä, Niilo Pudas, Samuli Ukkola

Comparing Eye Tracking Technologies

Bachelor’s Thesis
Degree Programme in Computer Science and Engineering
November 2023

Heikkilä K., Pudas N., Ukkola S. (2023) Comparing Eye Tracking Technologies.
University of Oulu, Degree Programme in Computer Science and Engineering.
Bachelor's Thesis, 37 p.

ABSTRACT

Eye tracking is a technology that monitors eye movements, and by the data, detects gaze directions and target points. The number of potential use cases and capabilities of such technology are huge, and at the time being, there are at least five commercial VR headsets with built-in eye tracking systems. The aim of this study is to compare the performance of eye tracking technologies with two devices. We compare the eye tracking glasses from SeeTrue Technologies against Varjo Aero's eye tracking system and evaluate, would it be worthwhile to place SeeTrue's eye tracking technology into university's headsets, which have no eye tracking capabilities at all, or is it better to use Varjo's device, whenever eye tracking is needed. Motivated by previous research, we built a physical setup for SeeTrue device and virtual setup for Varjo device, in which the participant is directed to look at a white target dot shown on the black screen. We decided to use a moving target dot and a target dot, which changes its position on the screen, but stays at one place for two seconds. From the scripts that control the target dot position, actual positions of the targets were collected and compared with the gaze target positions that were received by the eye tracking devices. In our study, we used accuracy and precision as measures of performance. According to the results of this study, Varjo performed better, and the results stand for using Varjo's device when eye tracking is needed, instead of placing SeeTrue's system into headsets, which have no eye tracking capabilities. However, both devices were easy to use and highly capable of eye tracking. We recorded the results of mean error in visual angle within five degrees on both devices, and even with the moving target. This study, along with the other studies in the field, gives an idea and methodologies to one kind of performance testing of eye tracking devices.

Keywords: eye tracking, Varjo, SeeTrue, HMD, head-mounted display, gaze detection, virtual reality

Heikkilä K., Pudas N., Ukkola S. (2023) Silmäntunnistusteknologioiden vertailu. Oulun yliopisto, tietotekniikan tutkinto-ohjelma. Kandidaatintyö, 37 s.

TIIVISTELMÄ

Katseenseuranta on teknologia, joka seuraa silmän liikkeitä ja saadun datan perusteella arvioi katseensuuntaa ja kohdetta. Potentiaalisten käyttökohteiden ja mahdollisuuksien määrä on valtava, ja tällä hetkellä markkinoilla on ainakin viisi kaupallista VR-laitetta, joissa on sisäänrakennettu katseenseurantajärjestelmä. Tämän tutkimuksen tarkoituksena on vertailla kahden silmien liikettä tunnistavan ja katsetta seuraavan laitteen toimintakykyä. Tutkimuksessa käytetyt laitteet ovat Varjo Aero ja SeeTrue Technologies:n katseenseurantalasit. Yhtenä motiivina tutkimuksella on tarkoitus arvioida, kannattaisiko yliopiston asentaa SeeTrue:n katseenseurantateknologiaa sellaisiin VR-laseihin, joissa ei ole katseenseurantaa, vai onko parempi käyttää Varjon laitetta, kun tarvitaan katseenseurantamahdollisuutta. Aiempien tutkimusten metodeja hyödyntäen, teimme laboratoriotilaan fyysisen testiympäristön SeeTrue:n laseille ja vastaavan virtuaalisen testiympäristön Varjo Aero:lle. Molemmissa testiympäristöissä osallistujan tuli seurata katseellaan valkoista pistettä mustalla näytöllä. Käytimme liikkuvaa kohdepistettä sekä pistettä, joka vaihtoi paikkaa näytöllä, mutta pysyi aina yhdessä paikassa kahden sekunnin ajan. Kohdenäytöltä tallennettiin kohdepisteen todellinen sijainti eri ajanhetkillä ja sitä verrattiin laitteiden tunnistamaan katseen kohteeseen. Suorituskyvyn mittareina käytimme keskimääräistä virhettä mittaavaa tarkkuutta (engl. accuracy) ja tuloksen toistettavuutta mittaavaa tarkkuutta (engl. precision). Tässä tutkimuksessa Varjo suoriutui paremmin ja tutkimuksen tulokset puoltavat Varjo:n laitteen käyttöä katseenseurantaan, sen sijaan, että SeeTrue:n teknologiaa asennettaisiin VR-laseihin, joissa ei ole vielä katseenseurantaa. Molemmat laitteet olivat kuitenkin helppokäyttöisiä ja kykenivät hyvin silmän liikkeiden ja katseen seuraamiseen. Tämän tutkimuksen menetelmillä saimme keskimääräisen virheen laitteen tunnistaman kohdepisteen ja todellisen kohdepisteen välillä jäämään alle viiteen asteeseen, visuaalisena kulmana mitattuna, ja jopa liikkuvan kohteen kanssa. Tämä tutkimus antaa muiden alaa tutkineiden julkaisujen ohella peruskuvan ja metodeja eräänlaiseen tapaan tutkia katseenseurantajärjestelmien suorituskykyä.

Avainsanat: silmäntunnistus, Varjo, SeeTrue, HMD, päähän puettava näyttö, katseentunnistus, virtuaalitodellisuus

TABLE OF CONTENTS

FOREWORD

We would like to deliver huge thanks to Dr. Evan Center and Prof. Timo Ojala for guiding us through this interesting bachelor’s thesis project.

Oulu, 7.11.2023

Kaarlo Heikkilä,
Niilo Pudas,
Samuli Ukkola

ABBREVIATIONS

CSV	comma-separated values
EOG	electrooculography
FOV	field of view
HMD	head-mounted display
PC	personal compute
RAM	random-access memory
RMS	root-mean-square
VR	virtual reality

1. INTRODUCTION

With the growing prevalence of head-mounted displays (HMD) on the commercial market and the quality and number of features, such as integrated eye tracking, the interest in conducting studies involving these devices has naturally increased. The goal of this study is to compare the performance of two different eye tracking technologies, by an HMD's integrated eye tracking solution and by the purpose-built eye tracking glasses. A specified purpose is to find out if it would make sense to place SeeTrue Technologies' solution into university's HMD's, which are lacking any kind of eye tracking capabilities or is it better to use Varjo HMD's built-in eye tracking system when eye tracking is needed.

1.1. Motivation

This project aims to establish valuable information for studies carried out in the future, by choosing fitting testing equipment and methods in terms of performance and capabilities of today's commercially available eye tracking equipment. This study is limited to the two devices that we are comparing, but the general methodology can be used with different eye tracking solutions.

HMD integrated eye tracking can be an extreme asset for a study that utilizes virtual reality (VR). Adding eye tracking capability to be used in tandem with head position tracking allows more immersive interactions with the virtual environment. This can be very desirable for studies that aim to gather information on the interactions of the user and the environment.

Using VR environments can reduce the cost of creating test setups and allowing for more fine control over the details of the environment since it's completely virtual. Moreover, using a VR environment can make such experiments possible, that would be either too dangerous, or too complicated to be concluded in the real-world.

One possible use case that utilizes eye tracking in virtual space is foveated rendering. Foveated rendering is a rendering technique in which a higher quality image is rendered at the fovea point and the rest of the image can be rendered at a lower resolution to preserve computational power.

Gathering the data from our eyes gives a new dimension to the studies of human behavior. Such data can add up information that is very sensitive and wouldn't be easily noticeable without eye-tracking technologies. By eye tracking, studies can utilize attention allocation as useful information. Data from eye tracking can be more subjective and unconscious for the participant, compared to questionnaire after a test session. Although, it doesn't mean that the eye behavior of the participant would be totally unconscious, as the participant may pay attention to where she/he looks during the test. Also, the data from eye tracking is regular, the sampling rate is high, and the data collecting is done during the test. Certain parts of the data can thus be easily assigned to the corresponding events in the test.

2. RELATED WORK

2.1. Eye tracking in virtual reality

Eye tracking is becoming a part of the VR headsets' features and at the time of this research, at least HTC Vive Pro Eye, Varjo Aero, Varjo VR-3 and Varjo XR-3 have integrated eye tracking components, like eye cameras, on them, and most probably that trend will keep rising. While we were working on this bachelor's thesis, Apple launched their Vision Pro headset, which also includes an eye-tracking system. Eye behavior detection can be used in multiple ways; VR games have nearly unlimited opportunities for it, and eye tracking can be used for studying people's behavior in different contexts, either in the real world, or in a built virtual environment.

There have been previous studies in virtual environments. For example, Clay et al. (2019) assigned a group of people to spend half an hour in a virtual village they had built, and to explore the places [5]. The data about participants' eye behavior was collected, including the target points of gazes and the timestamps of them. After the VR session, participants were asked to answer a set of questions about familiarity of different buildings from the VR village and if they would be able to find their way back to certain buildings. The correlation was detected in the study between the duration of looking at a particular building, the recognition of them and the confidence of finding the way back to the building afterwards.

VR is a great tool for expanding research opportunities, but there are also limitations that such research can face. One thing to consider in research projects, as well as in the final eye tracking applications, is the motion sickness which is caused for some people by the conflict between human senses during the VR sessions. In the previously mentioned research of Clay et al., two out of 31 participants had to stop the session because of motion sickness. Review by Chang et al. (2020) shows that the VR hardware, VR context and human factors are all shown to affect the experience of motion sickness or "VR sickness" in virtual reality [3].

Also, in the research of Chattha et al. (2022), both subjective and objective measurements of motion sickness in virtual reality were collected, and the impact to motion sickness was detected from gender, VR experience, motion sickness experience, experience of 3D games and the type of the virtual environment (pleasant/horror) [4]. In their study, five out of 51 participants had to quit the test after the first phase due to motion sickness.

2.2. Eye tracking metrics

Carter and Luke (2020) have created a very informative paper of the best practices for eye tracking research [2]. They bring up the main terms of eye tracking and eye working principles such as fixation, which is the time window, when the eyes are targeted to the fixation point in a subject's view and the saccade, which means the eyes' target point transition from one to another. Carter and Luke also discuss the different measurements of eye tracking and the possible disadvantages of trying to use too many measurements. The variations of measurements are widely introduced as well as the eye movements and the anatomy behind them.

Comparison research with similar characteristics to our study has been conducted by Pastel et al. (2021) with different hardware [7]. The main metrics compared in the research were gaze accuracy and precision. They had Eye Tracking Glasses 2.0 from SensoMotoric Instruments to represent eye tracking in the real world, and HTC Vive HMD to work with the VR. Pastel et al. had a reconstruction of a real-world space built to VR environment and the research was implemented by assigning 21 young sport students into two groups, and the groups started either with HMD or eye tracking glasses, completed the exercise two times, then switched with the used device, and completed two runs of the exercise again. Between using the different technologies was a five-minute break. In VR and reality, the task was to look at the dots on computer monitor and the accuracy and precision gaze tracking was measured. According to the study, accuracy of the gaze measurement was equivalent with both hardware, but the difference in precision could be detected in favor of the eye tracking glasses. Nonetheless, the paper concludes that the HMD's precision was so close to the precision of the eye tracking glasses, that the VR system can reliably be used in eye tracking purposes.

Stein et al. (2021) have focused on the latencies of different HMDs in their research, in which they compared it among Fove-0, Varjo VR-1 and Vive Pro Eye HMDs [8]. They measured the time it took for the device to detect the eye movement and get it available in the handler software. This measurement was called eye tracker delay. Stein et al. also measured so-called end-to-end latency which meant the time it took from the eye movement to update the HMD's screen with a very simple Unity script. The HMDs were compared using electrooculography (EOG) and Eyelink 1000, which perform with high resolution and low latency. Some variation between the latencies were detected and the eye tracker delays varied from 15ms to 50ms and the end-to-end latencies from 45ms to 80ms.

A book, "Eye Tracking: A comprehensive guide to methods and measures" by Holmqvist et al (2011) gives deep and definite information about human eyes, eye tracking and the ways eye tracking researchers could approach this area of science [6]. The book explains different eye movement types, their qualities, and ways to investigate them. It also gives a good vision of what kind of research has been done and what details should or should not be considered when planning eye tracking research. Holmqvist et al. tell for example, how external disturbance, like typing with keyboard, clicking a mouse, or walking past, can cause error to the measurements. Key measurements, such as accuracy, precision and latency are revealed in the book with clear examples of formulas and tables comparing different ways to do the research.

2.3. Latency Requirements for Foveated Rendering in Virtual Reality

A study by Albert et al. (2017) explores the effect of latency on foveated rendering in VR applications [1]. The study found that eye tracking latency of 80-150ms results in a significant reduction in acceptable foveation, while shorter eye-tracking latencies of 20-40ms do not significantly impact acceptable foveation, suggesting a total system latency of 50-70ms may be tolerated.

3. RESEARCH PLAN

In this section, we describe the design of how the study will be conducted and what hardware and software we will be using in the study. The plan is to measure the performance of the two eye tracking solutions by two different parameters: accuracy and precision. The study will be done by creating a real-world setup and a virtual reality eye tracking testing setup. The two setups will be made as similar as possible to make the performance data comparable.

3.1. Hardware

The project will be conducted using an HMD and a pair of eye tracking glasses. For the HMD we plan to use the Varjo XR-3 HMD and the Varjo Aero (Figure 1) headsets. Both these headsets provide integrated eye tracking solutions via eye facing cameras on the inside of the headset. Both models have a sampling frequency of 200Hz for the eye tracking. Both, XR-3, and Aero have 115 degrees of horizontal field of view (FOW). Varjo Aero has a resolution of 2880 x 2720 pixels per eye. Varjo XR-3 has 2880 x 2720 pixels per eye peripheral resolution and a focus zone of 27 x 27 degrees FOW with 1920 x 1920 pixels per eye.

Figure 1. Varjo Aero Head-Mounted Display (HMD).

For the eye tracking glasses we will be using the eye tracking glasses from SeeTrue Technologies (Figure 2). SeeTrue's eye tracking glasses have two cameras facing to the eyes and one in front. The sampling rate of the SeeTrue's glasses is 100 Hz. SeeTrue has also built-in implementation for collecting the eye tracking data. SeeTrue's glasses have a FOW of 82 degrees horizontally and 58 degrees vertically.

Figure 2. SeeTrue eye tracking glasses.

With both SeeTrue glasses and Varjo HMD, we will be using a university laboratory's computer LENOVO ThinkStation P520. The computer features are as follows: processor is Intel Xenon W-2235 (3.80GHz, 6 cores), the amount of installed physical memory (RAM) on the computer is 32GB, and the operating system is Windows 10 Pro. As a monitor, we use lab's 27" Lenovo ThinkVision 2560 * 1440 (61.40cm * 36.91cm) 60Hz.

3.2. Software

The test environments will be built using Unity3D engine and Python programming language. With Unity we will be using the Varjo Unity XR SDK to allow making the virtual test setup, which is needed for the test and gathering data. For calibration of the Varjo headsets we will be using Varjo Base's calibration tool. For calibration and gathering data from the SeeTrue's eye tracking glasses, we will be using SeeTrue's software, TechPlayer, as it allows us to record and save eye tracking data from the glasses easily in a format of comma-separated values (CSV).

3.3. Test participants

We are using our research project group (three members) as the participants in this research instead of using external resources. That's because the aim of this research is to compare the eye tracking performance between different hardware and thus, we are not focusing on the differences in eye tracking data between the participants. In fact, we think that it is even better for our purposes to use the research group as participants, because we are familiar with the environment, we have some experience in VR, and we can reduce the risk of unnecessary error caused by the participants coming from outside the project. So, in our opinion this is the best way to put the full focus on comparing the eye tracking technologies (hardware).

3.4. Test setup

Since we are comparing a real-life eye tracking solution and a VR eye tracking solution, we need to have a real-life test setup and a virtual reality test setup. These setups should be made as similar as possible to make the data comparable and results of this test valid.

The core idea of the test is that the participant is looking at the screen, on which a white circle is presented against a dark background. The white circle will change place on the screen in regular time intervals for the fixed time of the test.

As both real-life and VR eye tracking solutions can cover a wider visual angle than we need for this test, we will be performing a calibration and normalization for both solutions to make the data more easily comparable.

3.4.1. Real-world eye tracking setup

The real-world eye tracking test setup will consist of a chair, on which the participant will be sitting on, a chinrest, and a screen at a fixed distance away from the participant.

After the participant is seated and the equipment is set up, the output of the glasses will be calibrated by using SeeTrue's TechPlayer application's calibration. After that calibration data for normalizing the gaze point coordinates for the screen is collected by having the participant look sequentially at the two opposite corners of the screen for a short moment. The coordinates in the output of the glasses will be used as calibration data.

After the calibration, the testing application, in which the white circle is shown on different places on the screen, will be run for a few cycles while the participant tracks the circle with the gaze. Figure 3 shows the black screen with one possible place for the target dot, and the target will appear in different positions, but it's shown only in one position at the time. After the test program has been run the data is extracted from the TechPlayer application for the analysis.

Figure 3. Screenshot of the initial testing application.

3.4.2. Virtual reality eye tracking setup

The setup for virtual reality eye tracking will be very similar to the real-life testing setup. Instead of the SeeTrue's eye tracking glasses, the participant will be wearing a Varjo HMD, either the XR-3 or the Aero model, depending on the availability at the time of the testing. The participant will be looking at a virtual reality environment, in which there is a similarly sized dark colored rectangle and a white circle that changes its position inside the rectangle, representing the real-world setup used with SeeTrue's glasses. Figure 4 shows the initial screen presentation in VR. Unity's documentation suggests handling one Unity unit as a meter in real life [9]. According to that, we will physically measure the size of the target screen in the lab and the distance from test participant's eyes to the screen. We will use these measurements to make the virtual screen look the same size as the physical screen from the participant's point of view.

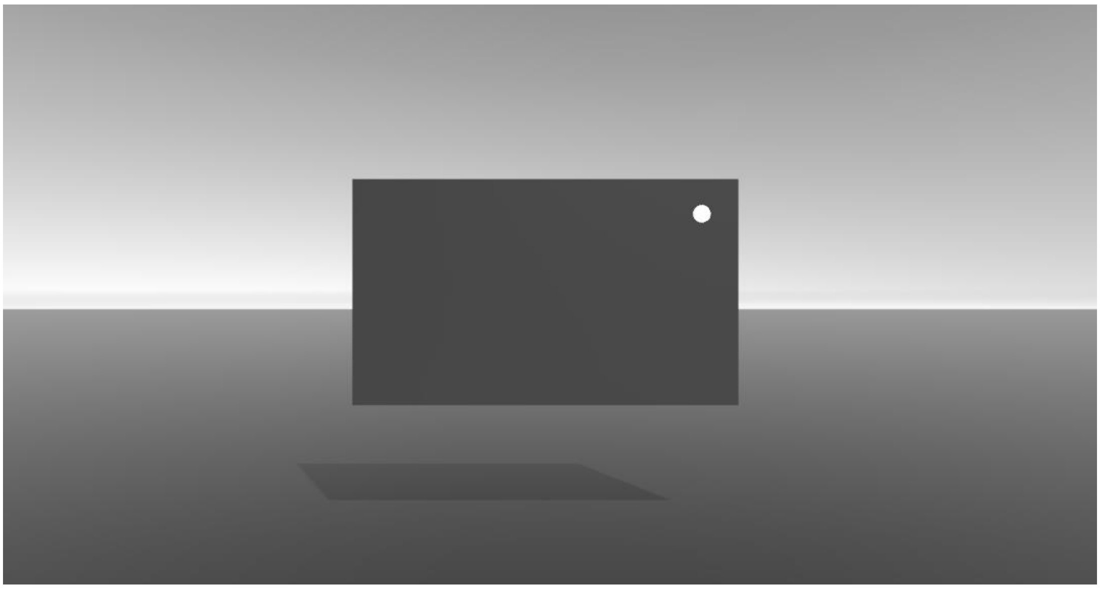

Figure 4. Screenshot of the initial VR testing environment.

Similar with the real-world setup, after the participant is seated and the equipment is ready, the eye tracking will be calibrated, this time using Varjo Base's calibration tool. After the gaze tracking is calibrated, the VR testing environment is loaded. Our own calibration for collecting the bounds of the virtual screen is done in the same way with the SeeTrue's glasses.

3.5. Performance parameters

We will evaluate the performance of the eye tracking solutions by comparing the accuracy and precision recorded during the tests.

Accuracy will be measured by how closely the eye tracking data follows the reference data of the position of the point that is tracked. For calculating accuracy, we will use a formula similar to the one that Holmqvist et al. have introduced in their paper. For each collected data point the difference in visual angle between the target dot and the detected gaze fixation point will be calculated with the horizontal distance from the participant's eyes to the screen and with the 2D vector distance between the target dot and the gaze fixation point on the screen. The principle is demonstrated in Figure 5. The angle difference is calculated with formula (1.1), where d1 is the upright distance from the eye to the screen, d2 is the 2D distance between target dot and eye tracking data dot, and alpha is the visual angle difference between the dots.

$$\alpha = \tan^{-1} \frac{d_2}{d_1} \tag{1.1}$$

The formula above is precisely correct only when the target point on the screen is placed so, that the angle between the participant's gaze and the screen is 90 degrees both horizontally and vertically, i.e., the gaze comes straight towards the screen. When there are target dots around the screen area, the formula could be adjusted a bit. Like Holmqvist et al. brought out, a fixed distance between two dots on the screen corresponds different visual angle difference between them, depending on how far the dots are from the center of the screen and what is the distance between eye and the screen. With reformatting the cosine theorem (1.2), a suitable formula to calculate the precise visual angle between the target dot and the detected gaze fixation point (1.3) can be obtained. The distances from the eye to target point and to measured gaze fixation point need to be calculated with Pythagorean theorem (Figure 6) to get the final formula (1.4).

$$c^2 = a^2 + b^2 - 2ab\cos\alpha \tag{1.2}$$

, where a, b and c are the edges of a triangle and α is the opposite angle of edge c.

$$\alpha = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \tag{1.3}$$

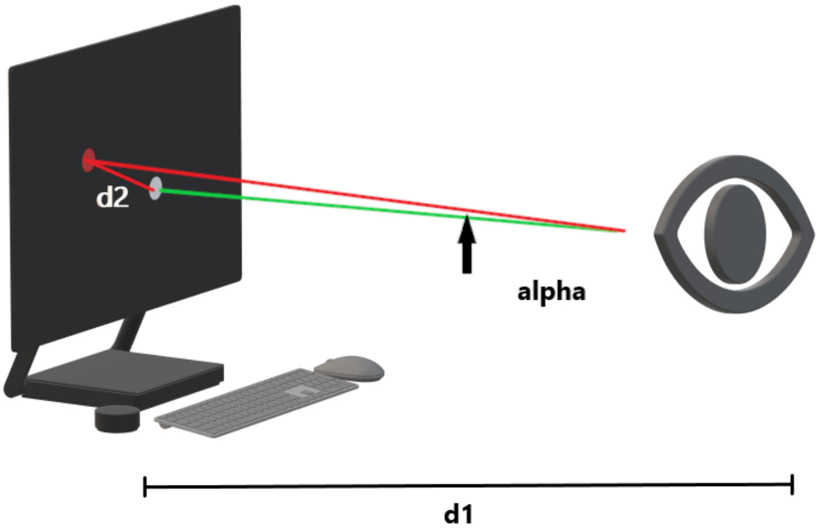

Figure 5. Visual angle difference.

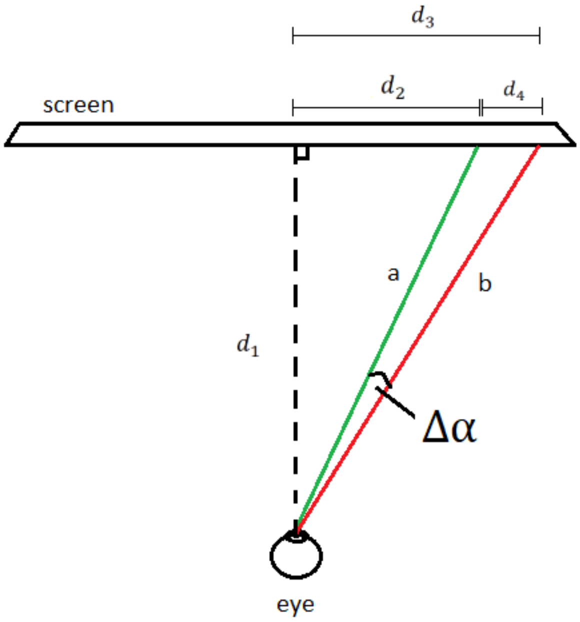

Figure 6. Corrected visual angle difference.

$$\Delta\alpha = \arccos\left(\frac{2d_1^2 + d_2^2 + d_3^2 - d_4^2}{2\sqrt{d_1^2 + d_2^2}\sqrt{d_1^2 + d_3^2}}\right) \tag{1.4}$$

(From formula 1.3: $a^2 = d_1^2 + d_2^2$, $b = d_1^2 + d_3^2$ and $c^2 = d_4^2$)

However, in this study we decided to go with the simple formula (1.1). We know that the furthest target point is 0.25 * screen width and 0.25 * screen height from the center of the screen, which means about 17.91cm calculated with the dimensions of the physical screen. If we think of the worst-case scenario, a fixed distance on the screen corresponds to minimal visual angle, when the direction from the target point to the gaze point is straight away from the center of the screen. By comparing the formulas 1.1 and 1.4 above, we can calculate that if the gaze point is for example 3cm from the furthest target point towards the corner of the screen, the visual angle between the dots differs about 0.062 degrees of what it would be, if the target point was in the center of the screen and the gaze point was same 3cm off. If the gaze point was 6cm off from the target point, the difference in the visual angle would be about 0.141 degrees between the target point being in the furthest point or in the middle of the screen. So, the difference between the formulas 1.1 and 1.4 within our setup (61.40cm * 36.91cm screen and the participant one meter from the screen) is quite small and we are using similar test setups for both devices with the corresponding dimensions on them, and that's why we thought, that the result comparison will be fair and precise enough for the purpose of our study even with the simpler formula (1.1).

After calculating the visual angles between individual gaze dots and the target dots, the accuracy will be introduced by average visual angle between the target and the gaze point. In a similar way that in the papers of Pastel et al. and Holmqvist et al., precision will be measured by the root-mean-square (RMS) of the errors between target dots and the measured gaze dots.

3.6. Plans for latency measurement

Our research's original idea was to compare accuracy and precision together with latency. Unfortunately, while we were starting our implementation phase, the university's EEG amplifier broke. We would have used the EEG setup to collect EOG data for latency measurements in our project, but due to schedule of the project, it wasn't possible to wait for the replacement and we didn't have any relevant alternatives for the latency measurement reference, so we had to exclude the latency comparison of the eye tracking glasses and the HMD in this research. However, it is described here, how we would have included the latency measurement and comparison in this kind of research.

Latency would have been measured as the time it takes from an occurrence of a saccade to the point when the eye tracking solution has registered the eye movement, and that measurement would have given one more meter for evaluating and comparing the performances. For gathering reference data to help with measuring latency we aimed to use electrooculography (EOG) by attaching electrodes near to the participants' eyes to give us an accurate reading on when a saccade (eye movement to different fixation point) is happening and then comparing the timestamps of when the electrode recognizes a saccade happening and when it is recognized with the used eye tracking device.

The participant would have worn the eye tracking device, and six different electrodes would have placed above the brow, on the side of the eye on the temple, below the eye and on the mastoids. The signals from the muscles that move the eye are distinctive enough that the placement of the electrodes doesn't have to be extremely precise for it to give viable data. The electrode placements are detailed in Figure 7 and in Figure 8.

The electrodes could be placed at least close to the same places with both the SeeTrue's glasses and the Varjo's HMD. Some adjustment might have to be made to accommodate the HMD's bigger frame that might obstruct the placement of the electrodes.

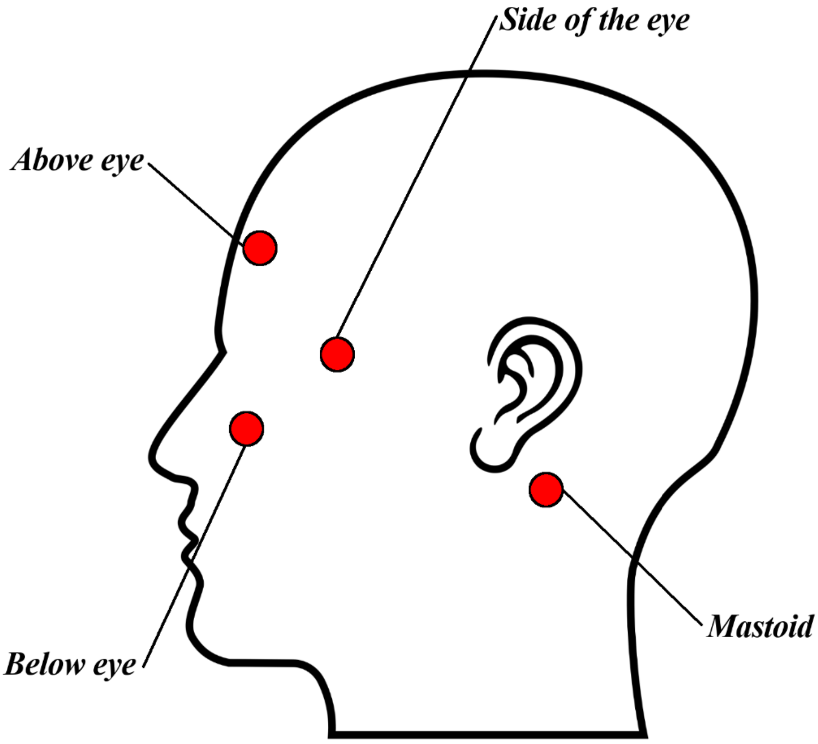

Figure 7. *Diagram of the electrode placement on the left side of the head.*

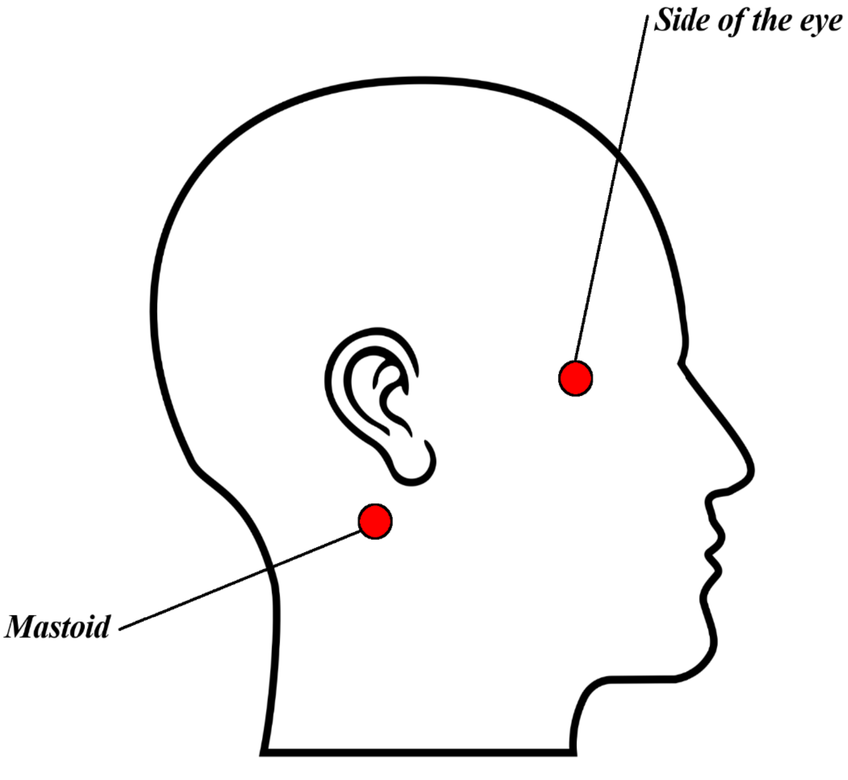

Figure 8. *Diagram of the electrode placement on the right side of the head.*

3.7. Gathering data

Data will be gathered from the Varjo's HMD and SeeTrue's eye tracking glasses. As for accuracy and precision, we will be recording the exact screen space coordinates of the virtual object that is to be tracked.

For gathering data from SeeTrue's eye tracking glasses, we will be using the SeeTrue TechPlayer's recording ability that gives the data, that the glasses track, in CSV-format. From the CSV file we will parse gaze point x, gaze point y, timestamp, system time and eye event data.

As the FOV of the participant will be wider than the portion that the screen, on which the testing application will be shown, we are required to calibrate the coordinates of the gaze point that we get from the glasses and the coordinates of the tracked object.

Getting the screen space coordinates of the tracked object is simple for the real-world testing application, as we can easily access the minimum and maximum x and y values that it can have. The screen space coordinate values of the circle are then normalized to be, according to the screen dimensions, withing the range from 0 to 1,6 horizontally and from 0 to 0,9 vertically, for ease of analyzing (Figure 9).

The SeeTrue's glasses give gaze point coordinates on a range from 0 to 1, where 0 is the minimum limit of the field of view through the glasses, and 1 is the maximum. So, the given coordinates are then multiplied by the dimensions. The gaze point coordinates are calibrated so that (0, 0) responds to the bottom left corner of the screen and (1.6, 0.9) responds to the top right of the screen. This is done by having the participant first look at the bottom left of the screen and then at the top right of the screen and flagging the data as calibration data accordingly.

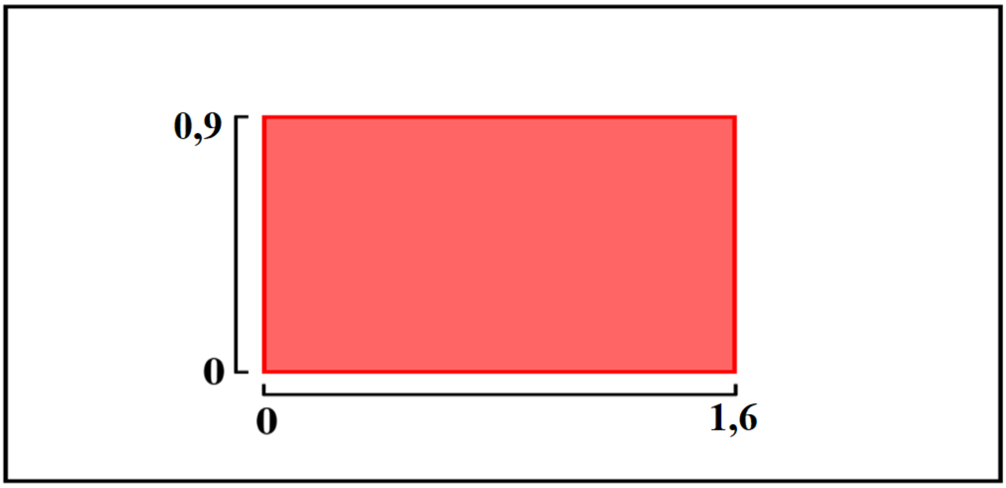

Figure 9. Representation of how the mapping of the gaze point coordinates is done. The screen where the testing application is run is shown in red.

With the HMD we will be gathering the screen space coordinates of the gaze point, system time and eye event data. The gaze point coordinates that we can get from casting a ray within the virtual environment, from where the eyes are fixated at, is represented as a three-dimensional vector inside the virtual environment.

This vector first needs to be mapped to a point on the two-dimensional projection that is displayed with the HMD to the user. Unity3D game engine provides methods that we can use to make these conversions. We can apply the same procedure of mapping the virtual three-dimensional vector to two-dimensional point on the screen to the tracked object to get its coordinates.

4. IMPLEMENTATION

4.1. Preparing the setup

Before we executed the data collecting session, we prepared the lab room for it. We collected data from the environment, like computer specs and display size. We placed a chinrest to the end of the lab's desk and tested that it can be tighten well and adjusted for each participant. Then we discussed the proper distance between the participant's eyes and the display to show the target program on. SeeTrue's glasses have quite wide rims around the eyes, so the final choice was to set the display one meter horizontally from the participant's eyes, so that the participant could see the whole screen easily. Figure 10 and Figure 11 show the setup in the lab environment.

We checked that the cables from the computer to Varjo HMD and to the SeeTrue's glasses were not tight and were free to move, not pulling participant's head in any direction. After that, we made sure every required software is ready to use and seemingly works as it should. SeeTrue's TechPlayer was opened, and one test run was executed to see that the calibration and saving of the output file work and the participant can follow the targets in the screen with ease. Varjo Base and Unity were opened and tested to communicate with the connected HMD. The target Python program for SeeTrue's glasses was run once to see that it works well.

We decided together the data storage paths on computer and the naming rules of the output data files, because there were two members of our group using the computer during the tests, and we wanted to make sure that the data files are correctly stored and can be easily identified when doing the analysis based on them. We also went through the usage of all needed software and the flow of the data collecting session verbally. We decided to take turns as a participant and to have three sequential runs per participant with short breaks between them, first with SeeTrue's glasses for all participants and then the same again for the Varjo Aero. We also decided to run each run for about 20 seconds to have a decent amount of data to analyze. That meant about 10 target dots on the screen to look at for every run. With a 200 Hz sampling rate that means roughly 4000 data points per run and 12 000 data points per device (SeeTrue / Varjo Aero) for every participant.

Figure 10. Test setup (1/2).

Figure 11. Test setup (2/2).

The size of the virtual screen was initially set up by the physical dimensions of the screen we were using, but we did some finetuning with the values by visually evaluating and comparing the sizes of virtual and physical screens and the visual angle when looking at the borders of the screens. The visual evaluation was done in the final setup and participant's head was on the chinrest when looking at the screen. Also, the white dot size on the screen was initialized with values from the physical setup, but it needed adjustment by visual evaluation as well.

4.2. Target scripts

The program for displaying the target object for the participant with SeeTrue glasses was written using Python programming language, and it showed white dots on black background. The screen borders calibration was done in SeeTrue's TechPlayer by just asking a participant to look at the opposite corners of the screen and recording the data from that. After the calibration, the Python program was run, and it showed white dots one at a time sequentially in four different coordinates. The dot places were ¼ * screen width, and ¼ * screen height towards the center of the screen from the four screen corners. Figure 12 demonstrates all the distinct screen positions of the target dots, but please notice that in the test session there was only one of the dots visible for the participant at a time. Dots were shown for two seconds in one place before changing to the next position. These four positions are well inside the screen borders and in participant's view, but they are still clearly separate from each other, so that the points can be easily detected from the output data.

The script for dots shown on the virtual screen was written using C# (c sharp) programming language and it works completely the same way as the Python program described above. The dot positions were equally calculated from the screen corners, the duration of certain dot was again two seconds, and the only difference was that the script showed the dots on the virtual screen made in Unity.

Both scripts saved a CSV file as an output where a new line on every dot movement was stored, including the system time and the dot position on the screen.

Figure 12. Possible positions of the target dots on screen.

4.3. Running the test sessions

4.3.1. First data collection

One of the participants sat down to chair to the end a table and wore the SeeTrue eye tracking glasses. The glasses were tightened by the band they have behind the head. The participant laid his head on the chinrest tightened to the table and took a stable comfortable position facing to the target screen. The member who was operating on computer (operator) asked if the participant was ready and when so, started the calibration from SeeTrue's TechPlayer. We used the TechPlayer's built-in calibration where red dots were shown on the screen, and it changed its position around the screen for a moment (about 15 – 25 seconds). The calibration of the glasses was executed again before each run.

When glasses were calibrated, operator asked the participant to look steadily at the bottom left corner of the target screen and then recorded a short data set (for about 5 – 10 seconds). Then again, the operator asked the participant to look at the top right corner of the screen and recorded another short data set there. These two calibration data sets were used to determine the borders of the screen in the eye tracking data. After the calibration points, the operator started the data recording again from SeeTrue's TechPlayer and started our Python program to show white dots on a black screen for the participant. In the program, the white dot stays in each position for two seconds at a time, and thus the operator checked that the program showed at least 10 dot positions to the participant to ensure that we have 20 seconds of recorded data from the run. After that, the operator exited the program and stopped the recording.

Between the test runs, the operator named the files and stored them in the target folder. During that time the participant could rest his eyes and relax for about 40 – 60 seconds. When the operator and the participant were ready, the new run started with the SeeTrue glasses' calibration.

After all the three participants had completed the session of three test runs with SeeTrue's glasses, we switched to the Varjo Aero. The physical test setup remained the same through all the sessions. Again, the participant took comfortable and steady position on the chair and put chin on the chinrest while wearing the Varjo Aero HMD. Every test run started by calibrating the HMD, and it was performed by operator clicking the calibration button from the Varjo Base software. After the calibration, the Unity scene was launched to HMD and the test run started. First, a white dot in the precise bottom left corner of the virtual screen was shown for a short period of time (5 – 10 seconds), after which, a dot in top right corner was shown for about the same time. After that the test itself started and the white dot changed its place between the four different positions. Like in the case of SeeTrue's glasses, the operator made sure that the data was gathered for at least 20 seconds before stopping the script and the eye tracking recording. The participant had about 30 – 60 seconds break between the runs and we did three runs in a row for a participant before changing the participant.

4.3.2. Second data collection

After the first data collecting session and after analyzing the data, we decided to arrange still another session to gather another dataset to analyze and compare whether the results remain similar with the new data. Also, we got feedback that it would be interesting to see eye tracking measurements from tracing different kinds of targets and we added a new implementation for both SeeTrue's glasses and Varjo Aero HMD. Now we had a script, in which the white dot was moving "continuously" around the screen through fixed dots and this case could be compared to the data from the first collection run. The whole path of the dot is shown in Figure 13.

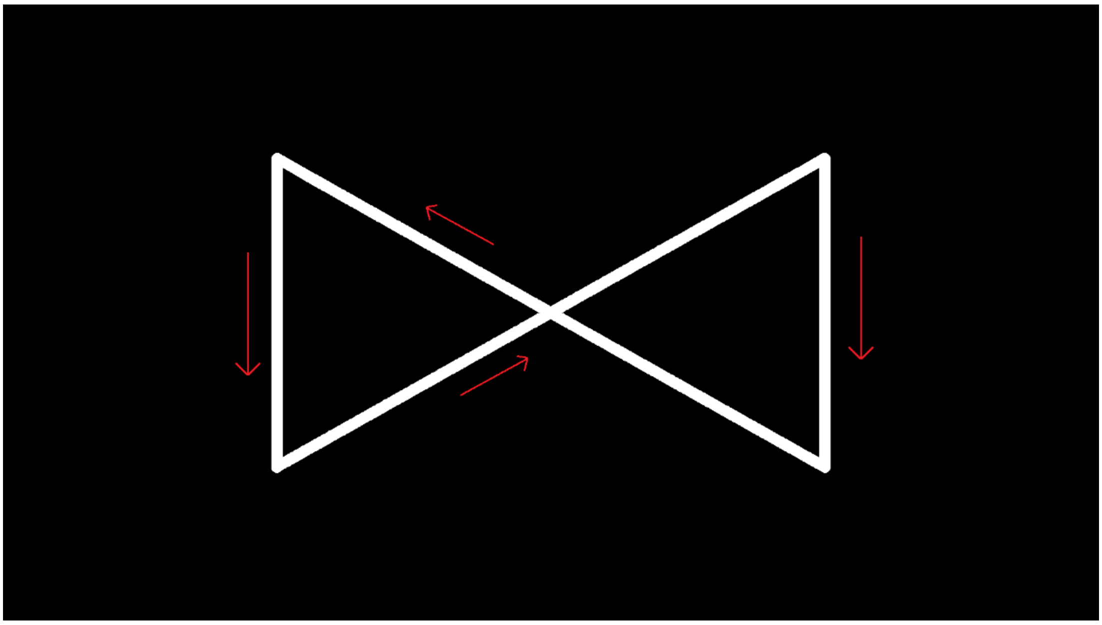

Figure 13. Path of the moving white dot.

4.4. Exporting eye tracking data from SeeTrue's glasses

For the implementation of the study, we started with looking for an efficient and appropriate way to export data from different systems. As described in the design section, in the case of SeeTrue's glasses, exporting the data was very straightforward and when testing, we got a fine CSV file with a great amount of data. The x and y coordinates of the gaze target point as well as the system time were the data columns, which we were most interested in, as the accuracy and precision were the main measurements of this research, and the latency couldn't unfortunately be involved. Earlier mentioned timestamp that the SeeTrue's software provides, is calculated in milliseconds starting from the beginning of camera recording, so it was more relevant for us to use system time as timestamps to be able to synchronize the data between different sources. We collected the complete data set from the SeeTrue's glasses and parsed it in the analysis phase.

4.5. Exporting eye tracking data from Varjo HMD

According to our original plan we would have used both Varjo XR-3 and Varjo Aero HMDs to compare against the SeeTrue's glasses and against each other, but during the implementation phase of our project, we faced lab space and time schedule related issues with Varjo XR-3 as there were other studies around it going as well. Furthermore, we couldn't find significant differences between the HMDs' eye tracking implementation in terms of accuracy and precision from the Varjo's site, and we were in planning phase using the same Unity plugin for the both HMDs, so we decided to leave out the XR-3 model and gather the HMD data only from the Varjo Aero for this research. The same research setup and software should be completely compatible with Varjo XR-3.

With Varjo HMD, all data could be gathered via Unity. Varjo has provided an open-source project called "VarjoUnityXRPlugin" for the features of their HMDs. We added that to Unity and modified the eye tracking sample (C# script) of the project to export the needed eye tracking data via a CSV file. Given libraries provide everything needed to record the timestamp of datapoint and the coordinates of the gaze target point. From the HMD's eye tracking data, we saved the frame number and capture time (in nanoseconds relative to Varjo system epoch) in case it would be necessary to do closer synchronization or inspection of certain data points. Then, we saved the logging time in Unity system time, a Boolean value indicating the calibration state of the HMD and the fixation points' coordinates, which were given as Vector3 object which represents three-dimensional coordinates (x, y, z), but for this study's purposes it was enough for us to use the x and y coordinates.

For the latency inspection, the timestamps could be read by the capture time (nanoseconds), converting it to comparable form, like Unix timestamp or datetime, but now as we focus on accuracy and precision in this research, the Unity system datetime was accurate enough.

4.6. Analyzing the data

For the data analysis, we wrote a Python program, which reads CSV files and categorizes them by the file name by the device and the test case. There were still some unnecessary columns in the files such as frame numbers, so we parsed the files to contain only x and y coordinates and the timestamp with millisecond accuracy. Next, we normalized the coordinates of both gaze point data and the target dot data. In the planning phase we were discussing normalizing both axes from 0 to 1, but eventually we decided to calibrate the x coordinates from 0 to 1.6 and the y coordinates from 0 to 0.9 corresponding to the dimensions of the screen. The edge coordinates of the screen were obtained by separate coordinate data in the case of SeeTrue's glasses, and by the main data and calibration flags in case of Varjo HMD. As the physical setup was measurable, and the virtual setup was implemented to be highly similar with the physical setup, the physical screen dimensions were measured from the lab's monitor, and they were used to get the rate of the distance between two dots on the screen and the distance between the participant and the screen.

Accuracy was measured by formula 1.1 after calculating the mean distance and visual angle between the dots. The precision we got by calculating the root-mean-square of the distances, i.e., taking a square root of the arithmetic mean of squared angles or distances in other form. The validation of our algorithms was done by picking random data points from the data sets and checking manually, that the program returns correct result. Lastly, the data vas visualized.

5. RESULTS

5.1. Characteristic of the data

In the first data gathering, we collected a data of 20 470 data points from the SeeTrue's glasses and 59 810 data points from the Varjo Aero HMD, making it total of 80 280 data points excluding the SeeTrue's calibration points, which were used to outline the area of the screen. In this set, there were some individual outliers, where the device was not able to detect the gaze target. The proportion of the outliers was about 0.3 percent of the data of the first collection, with both devices (71 out of 20 470 for SeeTrue and 171 out of 59 810 for Varjo). Due to the small number of outliers, and the lack of logic between them, they were just removed from the data.

We collected a total number of 99 845 data points of the participants looking at the moving target points in the second data gathering session (16 619 measurements from the SeeTrue, and 83 226 from the Varjo HMD). This time, there occurred 119 failed data points in SeeTrue data, which was about 0,7 percent of the data, and 483 failures in Varjo data, which was about 0,6 percent of the data. Again, the outliers were removed from the data before analysis. It is noteworthy that these numbers include the number of Varjo data points for calibration, which is equal to about one third of the total Varjo data points. So, the data amount for Varjo is roughly twice as much as for SeeTrue, due to the difference in sampling rates.

5.2. Stable dot in fixed positions

From the first data set, where we measured the accuracy and precision of the systems' gaze fixation point data collecting in reference to the target point on the screen, which was changing its position between the four fixed positions described earlier. The collected visualization of the collected fixation points is shown in Figure 14 below. The target points are in (0.4, 0.225), (0.4, 0.675), (1.2, 0.225) and (1.2, 0.675), which are visualized in Figure 15. There is some noise in SeeTrue data, in the y axis and the eye movement between the dots can be seen from the middle of the images, but the target of 4 dots can be seen from both images.

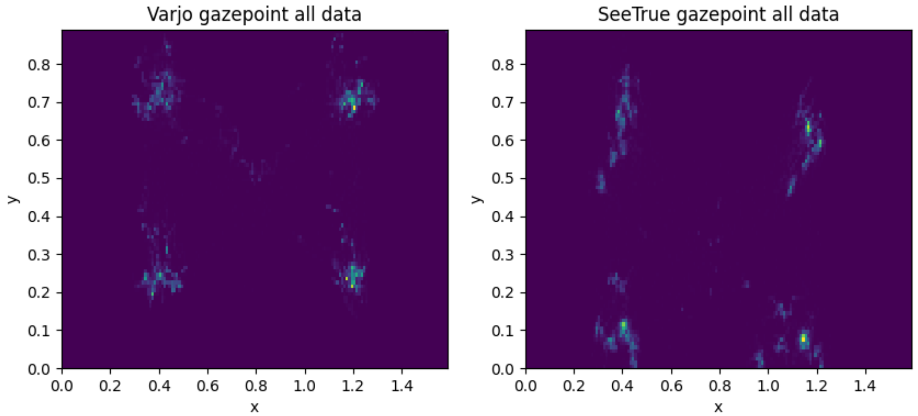

Figure 14. The distribution of the gaze target points on both devices (0ms threshold).

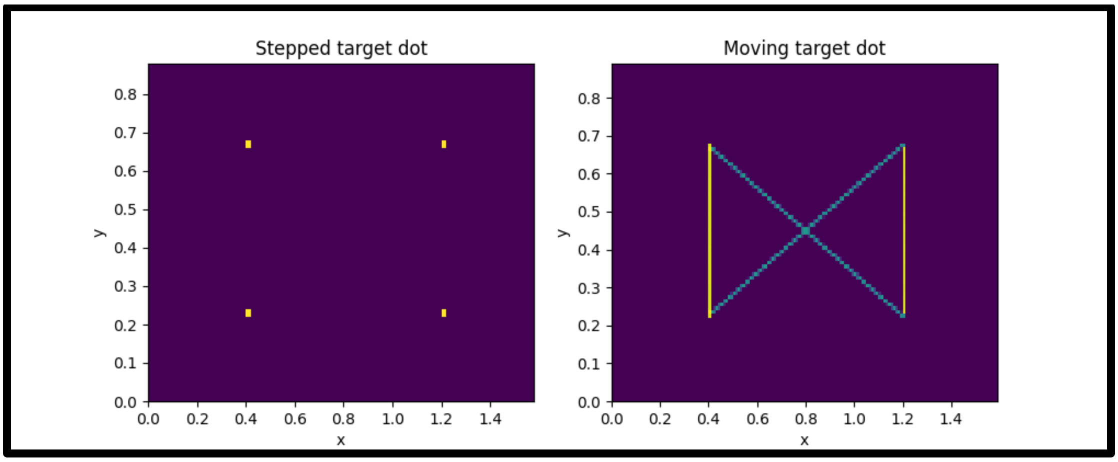

Figure 15. Objectively correct target locations for the stepped and moving target dots.

The corresponding values to Figure 14 are shown in Figure 16 below, in numerical form. As introduced, accuracy is calculated as a mean visual angle difference between the gaze point and the target dot position, and precision as an RMS of the differences between gaze points and target points in degrees of visual angle. In Figures 14 and 16, all data points are linked to the last reference point at capture time, so the gaze points during the eye movement to the new dot position from the previous one is recorded as a data of the new dot position.

	Varjo Aero	**SeeTrue glasses**
	Difference in normalized coordinates	
Mean x	0.20750	0.15537
Mean y	0.14963	0.20225
	Difference in millimeters	
Mean x	79.62866	59.62165
Mean y	61.36349	82.94655
	Difference in degrees of visual angle	
Mean x	4.35858	3.35220
Mean y	3.47976	4.70721
Combined accuracy	6.22457	6.21188
Combined precision	9.19010	8.36307

Figure 16. Numeric results from the first data set (0ms threshold).

In comparison with the results above, we wanted to reduce the eye movement's effect to accuracy and precision, by adding a threshold to give the eye time to move to the next spot, before starting to count the measurements for the changed target dot. The interval of the dot being 2 seconds set some limitations to not lose too much data due to the threshold, especially with the 100 Hz sampling rate of SeeTrue's eye tracking glasses, but we tried with threshold values of 150 milliseconds and 500 milliseconds, just to see the effect. The threshold time increased accuracy and precision, like demonstrated in Figure 17 below. In Figure 18, there is the visualization, like in Figure 14, but with the time threshold value of 500 milliseconds, and some improvement with the noise is visible.

After removing the data of 500 milliseconds from the beginning of each new dot position, there were total number of 14 528 data points left for SeeTrue's glasses and 25 874 for Varjo's HMD.

	Varjo Aero	**SeeTrue glasses**
Difference in normalized coordinates		
Mean x (150ms)	0.16585	0.13816
Mean y (150ms)	0.10780	0.18739
Mean x (500ms)	0.03880	0.12109
Mean y (500ms)	0.04970	0.17125
Difference in millimeters		
Mean x (150ms)	63.64583	53.01806
Mean y (150ms)	44.20974	76.85009
Mean x (500ms)	14.88923	46.46875
Mean y (500ms)	20.38382	70.23161
Difference in degrees of visual angle		
Mean x (150ms)	3.49140	2.98525
Mean y (150ms)	2.51315	4.36446
Mean x (500ms)	0.85051	2.61902
Mean y (500ms)	1.16643	3.99138
Accuracy (150ms)	4.78388	5.67353
Precision (150ms)	7.13965	7.58482
Accuracy (500ms)	1.59546	5.12671
Precision (500ms)	2.08489	6.76307

Figure 17. Numeric results from the first data set (150ms and 500ms thresholds).

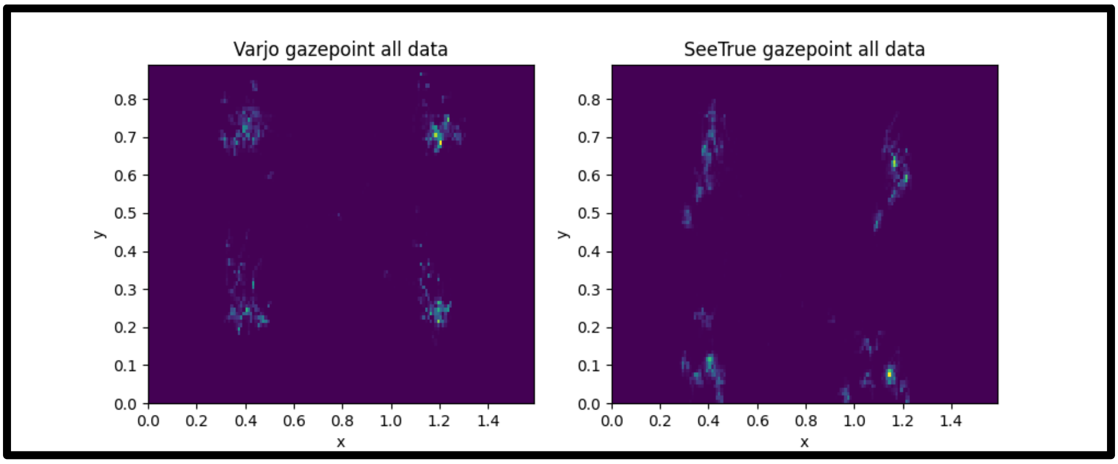

Figure 18. The distribution of the gaze target points on both devices (500ms threshold).

5.3. Moving target dot

In the second data set, we had data from the participants following the moving dots with their eyes. The route of the moving dot in the used script can be seen in Figure 13. And the visualization of the correct target dot route can also be seen in Figure 15. In the following Figure 19, there is a visualization of all gaze points from all four sessions of all the participants and for both Varjo Aero and SeeTrue's glasses. Figure 20 shows the numerical data from the moving target dot in the second data set.

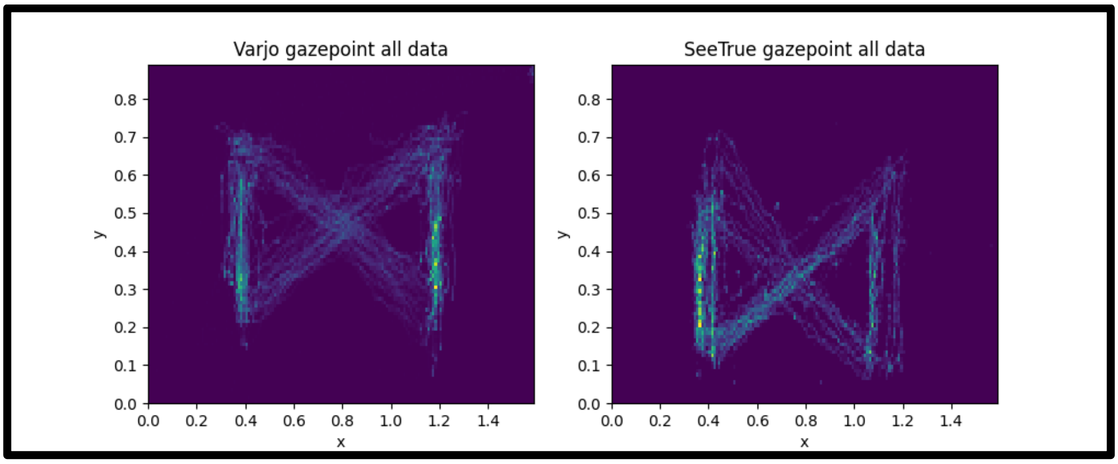

Figure 19. The distribution of the gaze target points on both devices with the moving dot.

	Varjo Aero	**SeeTrue glasses**
Normalized coordinates		
Mean x	0.05141	0.09096
Mean y	0.04684	0.13405
Millimeters		
Mean x	19.72946	34.90752
Mean y	19.21035	54.97387
Degrees of visual angle		
Mean x	1.11243	1.99298
Mean y	1.09882	3.14113
Accuracy combined	1.73587	3.88964
Precision combined	3.16248	4.73968

Figure 20. Numeric results from the moving target dot in the second data set.

6. DISCUSSION

In this study, two eye tracking devices were compared against each other, in terms of accuracy and precision of capturing the user's eye gaze target positions. The used devices were Varjo Aero HMD, which has built-in eye tracking system, and SeeTrue Technologies' eye tracking glasses, which have no virtual environment, but which detect user's eye gaze positions in physical world with the cameras. One of the main motivations of this study was to evaluate whether it would be reasonable to place SeeTrue's technology into the university's HMDs with no eye tracking system in them or is it better to use Varjo's device for eye tracking purposes.

The physical lab setup was set into university's premises, and it contained a table, with a chinrest attached to it for the participant, computer and 27" monitor in one meter from the participant's face. With Varjo Aero, the same physical setup was used, but instead of showing the visual targets on the physical screen, they were showed on a virtual screen shown in HDM screen. The scale similarity between the setups was implemented by using Unity metrics and physical measurements, and by doing visual evaluation by the participants.

The three members of our project group were also participants in this study. We arranged two data gathering sessions, where all participants were using both headsets and we recorded the eye tracking data from the devices, while the participant followed white target dots on black background with his eyes. The dots were shown either on the physical screen or in Varjo case, on the virtual screen shown in the HMD. In the first session, the script was showing the dot changing systematically between four fixed positions, showing the dot for two seconds in each spot before changing to the next one. In the second session, we modified the script regarding the received feedback, so that the dot was moving constantly between the four fixed positions. This way, we received the data with the moving target, in comparison.

The result of this study brings out, like previous studies such as Clay et al. (2019) or Pastel et al. (2021), that the accuracy and precision of available eye tracking devices is generally on a very good level, and it has great potential to be utilized in various applications. In the first dataset, which we collected in our study, Varjo Aero detected the gaze point initially at average of 79.63 mm off from the real target point horizontally and on average of 61.36 mm of vertically, on the label on meter away from the participant's eyes. This means the average error of 4.36 degrees horizontally and 3.48 degrees vertically, in the visual angle. However, when we kept the 500-millisecond pauses in measuring, each time when the dot changed its place, so that we didn't measure the gaze points, when the eye was moving between the targets, Varjo performed with the average horizontal offset of 0.85 degrees (14.89 mm on the screen), and average vertical offset of 1.17 degrees (20.38 mm on the screen). The corresponding values for SeeTrue's device in the first dataset were 3.35 degrees (59.62 mm on the screen) off horizontally and 4.71 degrees (82.95 mm on the screen) off vertically on average, when there were no pauses in measuring between the dots. With the pauses, the average result was 2.62 degrees (46.47 mm) off horizontally, and 3.99 degrees (70.23 mm) off vertically.

As we got suggested to record the data also with the moving dot, our initial thought was that the accuracy and precision will suffer, when the target is moving. Following the moving target dot could be identified as smooth pursuit, like Holmqvist et al. have it introduced, giving an example of watching a flying bird in the sky. Anyways, neither the movement of the target dot on the screen, nor the eye movement is completely continuous, and, in our study, there are the for points, where the dot changes its movement direction over 90 degrees relatively often, so it was not surprising, that due to human eye anatomy and participant's reaction time, the movement of the target dot affected to the results.

Although, the effects of the new script were not radical, it is visible from the results, that in the Varjo case, the moving target causes a bit worse score, especially in precision, which is logical, as the gaze is trying to keep up with the dot movement and is necessarily slightly behind the dot position. The higher sampling rate of the device is of course helping in this, from the hardware side. On the other hand, we managed to end up with even better accuracy and precision scores with moving dot, than with the stepping dot, by SeeTrue's glasses, in our study. The results from the second data gathering nonetheless correlate with the results of the first set. Varjo Aero had a performance of 1.74 degrees in accuracy and 3.16 degrees in precision combined from all four runs of all participants, with the moving target. On the other hand, SeeTrue had a combined accuracy of 3.89 degrees and precision of 4.74 degrees. In the other form, the average offset on the screen level of one meter from participant's eyes, was 19.73 mm horizontally and 19.21 mm vertically with Varjo Aero, and 34.91 mm horizontally and 54.97 mm vertically with SeeTrue's glasses.

Varjo has announced the accuracy of Aero HMD to be "sub-degree" and SeeTrue Technologies has provided information of 1.35-degree accuracy and 0.1-degree precision of the eye tracking glasses. From the results of our study, we believe that the promised accuracies and precisions could be reached, at least within optimal circumstances. One simple way could be to measure accuracy and precision when the participant is just looking at a single fixed point, and no intended eye movements would be involved. Another way to implement the measurement with more stable target could be very similar to our setup, but with just gathering more data, so that higher time threshold value could be used at every dot movement, and the measurements would thus include just the time windows, when the participant is statistically looking at the target dot. Also, some universal offset can be seen especially in vertical axis in SeeTrue data both with the stepped dot in first dataset and with the moving target dot. We couldn't find a rational reason for it, but if fixed, the results of the SeeTrue's eye tracking glasses would be improved.

7. CONCLUSION

Some things to consider in future studies, that we noticed during this project, were that there could be the stable dot target involved in the study, because it would have fewer distracting factors than moving dot or dot, which is changing its place frequently. So, the accuracy and precision measurements would be most probably best from the stable target test, and it could be used to compare, that the range is somehow similar with the performance values given by the manufacturers. Also, it would give good additional insight and comparison points on how much eye movement really affects accuracy and precision. Calibration could be investigated more, especially when having devices for reality and for virtual reality, like in our study, to make sure that there is no noticeable offset due to the calibration. The calibration of the device, has not much, but the different given calibration techniques could be compared (1-dot, 5-dot, 10-dot etc.). We found it very important to visually evaluate the correctness of the device calibration by the test operator, by asking the subject look at certain points right after the device calibration, when there was the visualization of subject's gaze point visible with both Varjo HMD and SeeTrue's glasses. When calibrating the target screen area with any devices and when trying to simulate real setup in virtual reality, the calibration methods could be modified to improve the comparability. We used two dots in the opposite corners of the screen area to do the comparison, but the effect of using 5-dot comparison (one in all corners and one middle of the screen) could be tested, for example.

The latency measurement would be a good addition to this kind of study, with these devices from Varjo and SeeTrue Technologies, or with any other devices to compare. It would give one more dimension to detect the capabilities of the devices, and by that, the best hardware solutions for different study or business purposes could be evaluated.

Of course, the data could be gathered a lot more to increase the reliability of the results, and like said, we could have used bigger data sets to raise the duration of measurement pauses, when the target dot was changing its position in the stepped target dot script, which would have given ability to analyze devices' accuracy and precision with even more stable gaze. Also, one approach to consider in the future comparison, would be to match the sampling rate of the devices, if the default sampling rates are not equal, like in this case.

This study, along with the previous studies in the field, gives some guidelines and ideas of how to implement further studies, in which utilizing eye tracking devices. The book by Holmqvist et al. is recommendable and a great basement to start gaining information about eye tracking and how to measure it. Also, we believe that the results of this study motivate to find more usage for such devices, as the efficiency of eye tracking is on a very good level at the time being. One main question of this study in terms of the practical outcomes, was that should University of Oulu place SeeTrue Technologies' eye tracking system into its headsets, which have currently no eye tracking capabilities or should University just use Varjo devices whenever the eye tracking is needed.

Varjo Aero and SeeTrue's glasses were both performing somehow on similar level, each of them winning one another at some measurements, but overall, the Varjo Aero's eye tracking resulted to have better accuracy and precision, in terms of our test methodology. Anyhow, achieving high reliability in this kind of comparison is challenging, due to the combination of the physical setup and the setup in virtual environment. As we were able to use only Varjo Aero for the final comparison, another potential test setup in the future would be Using Varjo XR-3, which has inside-out cameras and tracking capabilities. In that way, the comparison with SeeTrue glasses could be made in a single physical setup equally for each device. It would reduce some of the differences in comparison, but at the same time Varjo's inside out tracking system would be involved as a new factor. The other way around, if it was anyhow financially possible, the SeeTrue's eye tracking system could be placed into a headset which has no eye tracking capabilities. Then it could be tested against Varjo Aero in an equal virtual test setup.

8. REFERENCES

[1] Albert, R., Patney, A., Luebke, D. & Kim, J. 2017. Latency Requirements for Foveated Rendering in Virtual Reality. ACM Trans. Appl. Percept. 14, 4, Article 25 (September 2017), 13 pages. https://doi.org/10.1145/3127589

[2] Carter, B. T., & Luke, S. G. (2020). Best practices in eye tracking research. International Journal of Psychophysiology, 155, 49-62.

[3] Chang, E., Kim, H. T., & Yoo, B. (2020). Virtual reality sickness: a review of causes and measurements. International Journal of Human–Computer Interaction, 36(17), 1658-1682.

[4] Chattha, U. A., Janjua, U. I., Anwar, F., Madni, T. M., Cheema, M. F., & Janjua, S. I. (2020). Motion sickness in virtual reality: An empirical evaluation. IEEE Access, 8, 130486-130499.

[5] Clay, V., König, P. & König, S. (2019). Eye Tracking in Virtual Reality. Journal of Eye Movement Research, 12(1), 1–18.

[6] Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). Eye tracking: A comprehensive guide to methods and measures. OUP Oxford.

[7] Pastel, S., Chen, C., Martin, L., Naujoks, M., Petri, K. & Witte, K. (2021). Comparison of gaze accuracy and precision in real-world and virtual reality. Virtual Reality, 25:175–189 https://doi.org/10.1007/s10055-020-00449-3

[8] Stein, N., Niehorster, D. C., Watson, T., Steinicke, F., Rifai, K., Wahl, S., & Lappe, M. (2021). A comparison of eye tracking latencies among several commercial head-mounted displays. i-Perception, 12(1), 1–16. https://doi.org/10.1177/2041669520983338

[9] Unity documentation. Believable visuals: preparing assets. Read 17.4.2023. https://docs.unity3d.com/2020.2/Documentation/Manual/BestPracticeMakingBelievableVisuals1.html

9. APPENDICES

Appendix 1. Work distribution within the group.

STUDENT SUMMARY					
First name	**Last Name**	**Hours**	**% of project total**	**% of nominal total***	**Tasks**
Kaarlo	Heikkilä	226	38.31%	104.63%	Reading research papers, writing the thesis, collecting, and analyzing the data, managing the project.
Niilo	Pudas	193	32.71%	89.35%	Reading research papers, writing the thesis, using Unity writing the analysis tool, collecting, and analyzing the data.
Samuli	Ukkola	171	28.98%	79.17%	Reading research papers, writing the thesis, writing the target dot script, collecting, and analyzing the data.

*(8 cp a' 27 hours ~ 216 hours per student)