Eye Diagnostics with Virtual Reality

Robin Eriksson, Markus Rahne

2022

Master's Thesis in
Biomedical Engineering

Supervisors: Marcus Nyström, Martin Stridh
Examiner: Leif Sörnmo

Faculty of Engineering LTH
Department of Biomedical Engineering

Abstract

Eye tracking technology has been a great addition to medicine, psychology and behavioural science. Its contribution to diagnosis and the understanding of specific eye diseases should not be understated. However, eye tracking technology is often complex, expensive, immobile and usually requires users that can keep still for longer periods of time. These prerequisites are at times not possible, and a solution for a lighter, more accessible eye tracking system that can give similar results is desirable. In recent years, research and development has created eye trackers small enough to fit inside the cavity of head mounted displays and headsets for virtual reality, creating new possibilities for research and mobility.

The aim of this thesis is to evaluate how close this new technology can replicate existing technology and its ability to diagnose a specific eye disease, nystagmus. Aspects of interest are for example how the two systems differ in ease of use, how they measure data and the data quality. The major research question is to evaluate if the new equipment, a combination of virtual reality and eye tracking, is a viable option to the already established versions of eye tracking systems.

Our thesis has shown that there is indeed potential for using VR headsets equipped with eye tracking technology, but the technology has to improve for it to be viable in a clinical situation. With further technical development and software development for the calibration and test protocol of a truly mobile system, it will complement the stationary systems as a great diagnostic tool.

Acknowledgments

We would like to thank our two supervisors Marcus Nyström and Martin Stridh for their assistance in this thesis work and for helping us with recordings of both systems. Their guidance and expertise when problems arose and their help with the report has been invaluable. We also want to thank the Lund University Humanities Lab for providing a place to work with the equipment. We would also like to thank our family members and friends for supporting us through our studies and the writing of this master's thesis.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The combination of eye tracking and virtual reality is a subject that is very much in its infancy, even more so regarding its potential uses in clinical ophthalmology. The first commercially available combination of virtual reality and eye-tracking premiered in 2014 by Japanese company FOVE and was soon followed by similar solutions developed by already established eye tracking developers such as SensoMotoric Instruments and Tobii. Usually this was done in collaboration with manufacturers of VR-headsets like HTC, where the early models consisted of just an ordinary VR-model with an eye tracker surrounding the lenses. Virtual reality with integrated eye-tracking is a technology on the rise and is set to improve and optimize both virtual reality and eye tracking in ways where they both benefit from each other. Examples include more powerful head mounted displays (HMD) that use foveated rendering to reduce the computational load on the computer by rendering the specific part where the user is currently looking in full resolution and then rendering the parts in the peripheral vision of the user in a lower resolution where fine detail is not as important [1].

Aside from the specific technical improvements that might come from the implementation of eye tracking in an HMD, the advantages are mostly in the form of user interaction and increased developer insights and analytics. Eye tracking makes it possible for developers and researchers to quantify user experience with interaction and gain valuable biometric information [1].

The combination of being able to render a virtual 3D-world and having access to vast amounts of tracking data allows for many interesting applications, one of them being the discovery and diagnosis of various eye conditions, such as nystagmus, which will be the main focus of this thesis. The benefits of doing a clinical diagnosis in a virtual reality environment, includes increased freedom and user availability. The test environment is also more natural and more closely reflects day-to-day experiences as the users can move their head around as they would normally do, compared to a more classical nystagmus test which requires the head to be in a fixed position.

These new technological possibilities gave rise to the overarching setting of this thesis: whether it is possible to use the newly developed VR-powered eye tracking devices as diagnostic tools. Already established eye tracking systems such as the EyeLink 1000 has shown its reliability in capturing accurate eye tracking data and is being used in different scientific projects at the Lund University Humanities Lab for determining diagnostic capabilities of eye tracking systems. If the same capability could be shown for the more portable and affordable systems combining virtual reality and eye tracking, much could be won.

1.2 Goals

As stated in the last paragraph of the background, the overarching goal of this thesis is to explore the possibility of applying some already established procedures and tools for diagnosing nystagmus on a VR-system with eye-tracking. These procedures have previously been developed for the eye tracking system EyeLink 1000. For this purpose, we formulated four principal goals for this thesis:

- Recreate and implement the existing research protocol for the purpose of recording and analyzing nystagmus in software used by the VR headset.
- Record eye movements from participants with and without nystagmus using aforementioned research protocol.
- Evaluate the differences between the stationary eye tracking system EyeLink 1000 and the wearable eye-tracker and VR headset Tobii HTC Vive. Differences that will be evaluated are calibration, precision, accuracy, data quality and effort in terms of processing the acquired data.
- Investigate and examine new possibilities and eventual advantages and disadvantages using head-mounted displays and eye tracking in eye diagnostics.

These four questions will act as the backbone of this thesis and based on their individual results it will be possible to evaluate if the VR headset would be of consideration compared to the EyeLink 1000.

1.3 Scope and limitations

The purpose of this master thesis is to evaluate the usability of a virtual reality headset with eye tracking as a tool for eye diagnostics. The main goals, as stated above, will be the evaluation of the eye condition nystagmus and the quality of the data acquired from the headset. This will be done by creating a 3D test setup replicating a nystagmus test setup from previous work using the game engine Unity. The tests and quality of data will then be compared in order to evaluate and compare the usability, advantages and disadvantages of the headset and its capability as a diagnostic tool.

As we are writing this master thesis during an ongoing pandemic, our ability to do thorough testing on several test subjects has been severely limited, and our test subjects only contained one test subject with nystagmus and 1-3 test subjects without any eye conditions.

1.4 Previous work

The use of eye-tracking in medicine in general and ophthalmology in particular is not as recent as one might think. Early forms of eye tracking were developed as early as the beginning of the 20th century, between the years 1900 and 1920, and was mostly used in research of psychology and reading. The first example of what some would define as a reliable form of eye-tracking appeared in the 1980s, with the development of magnetic search coils. These coils were placed onto the eye with the help of a silicone lens, and by measuring the variation of a magnetic field over the coils, the movement of the eye could be interpreted. Because of the invasive nature of the procedure, it was however determined impractical and measurements could not extend over 30 minutes, due to irritation and discomfort for the subject. Further development lead to the now modern standard of video-based eye-tracking (VOG), and eye tracking as a diagnostic tool has been shown to be promising for diagnosing certain mental illnesses such as schizophrenia, bipolar disorder and several more [2].

One of the newest technologies for eye tracking is to combine virtual reality and eye tracking, a subject of great interest that has been picking up speed during recent years. One compelling piece of work is done by Clay et al. where her team created a simulated world in virtual reality, where the user may roam freely, while their eye movement is captured. Much is focused on the advantages and disadvantages of using this type of wearable eye tracking in tandem with a head-mounted display and showing visual stimuli in a virtual reality. The paper is comprehensible and is a great tool to have as a base for avoiding certain common pitfalls when incorporating eye tracking with virtual reality and how the two technologies merge [3].

In addition to the methods and information presented by Clay et al. [3], information about the specifics of data processing and data quality of eye tracking systems is presented in Lohr et al [4]. It reveals a comprehensive and very welcome case study and comparison of eye tracking incorporated in HMD and regular eye tracking devices like the EyeLink 1000 used in work by Rosengren [5], and presents how well they process the signal data. It presents seven different fields where systems differ and can be compared in regard to their hardware, software and how they gather data. The seven fields are as follows: spatial accuracy, spatial precision, temporal precision, linearity, crosstalk, recalibration and filtering. The results are then compared across the seven categories between the eye tracker with VR and the EyeLink to give a comprehensive list over pros and cons for the two systems and their respective final data quality of eye tracking data [4].

Extensive work in the field of eye tracking and nystagmus has been done by William Rosengren and much of the work done in this thesis is a light continuation of his doctoral dissertation *Characterisation of nystagmus waveforms in eye–tracker signals* [5]. His presented nystagmus test protocol and test setup worked as an excellent principle outline of this thesis, and his algorithm and test platform for recognizing nystagmus signals in data was used extensively as a research and evaluation tool across different platforms of eye trackers.

1.5 Outline of the thesis

The report is divided into the following chapters:

1. **Introduction.** The subject matter is given together with a brief explanation of the existing technology and research. The goals, scope and limitations of the thesis are presented.
2. **Theory and background.** All necessary background information of the subject matter is introduced and explained thoroughly.
3. **Method.** The practical steps taken during the thesis are presented and put into context with the previously presented theory and the research goals.
4. **Results.** The results achieved from the methods are presented.
5. **Discussion.** The results are analyzed and evaluated in relation to the research questions and goals of the thesis.
6. **Conclusion.** The work and results of the thesis are summarized, evaluated and discussed. A brief comment of possible future work is presented and discussed.

2 Theory and Background

2.1 The human eye

The eye is the entry point of the human visual system and is one of the key components of how humans interact with the outer world. The visual system consists of several parts and its main purpose is to detect incoming light and visual stimulation, so the brain can interpret the signal in a correct manner [6].

2.1.1 Anatomy

The human eye is essentially a liquid filled ball with a front part that controls the movement of the eye and projects light into the eye itself. The light then hits the retina of the eye, which is the second part of the eye and is responsible for absorbing the incoming beams of light and transmitting the resulting signal to the brain for further processing. As seen in figure 1 the front part of the eye consists of the cornea, pupil, iris and lens in addition to the muscles and fibers that control the outer and inner movement functions of the eye. The cornea is the outermost layer of the eye. It is transparent, so that light can pass through, and it is connected to the eye's external muscles that moves the eye in its socket. Behind the cornea is the iris, which gives the eye its characteristic color. The iris is also connected to specific muscles, the ciliary muscles and zonular fibers, which determine the diameter of the pupil, the small opening that allows light to pass into the eye. The same muscles that control the iris are also connected to the lens, a crystalline structure that focuses incoming light onto the inner surface of the eye [6].

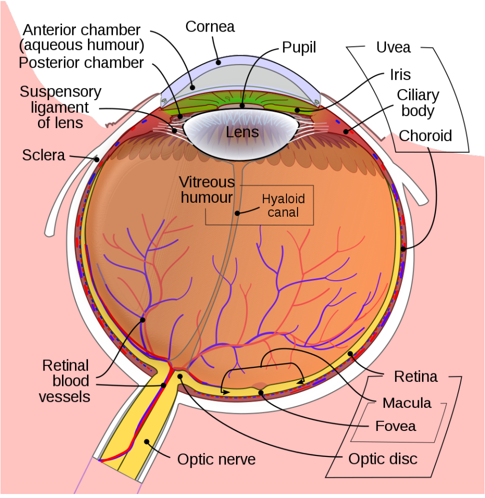


Figure 1: Schematic diagram of the human eye [7].

Incoming light is directed by the lens on to the inner surface of the eye, which is the retina. The retina is littered with specific types of neurons and photoreceptors that detect incoming light. The most notable patch of the retina contains the macula lutea and the fovea centralis, which contains a high concentration of photoreceptors and is the desired target for incoming light. Last there is the optic nerve which channels the signals from the photoreceptors to the brain [6].

The photoreceptors themselves are called cones and rods, resembling their actual appearance. They are each photosensitive, rods more than cones, and they convert light signals into action potentials through the interaction with bipolar and ganglion cells. The axons of the ganglion cells submit the output from the retina and create the first part of the visual pathway, described in figure 2. This first part is the optic nerve which emerge from both eyes and continues towards the back of the brain to meet at the optic chasm where some nerves intermingle, making input from both eyes possible. The signal ends at the visual cortex, where the visual signal is perceived by the brain as a visual image [6].

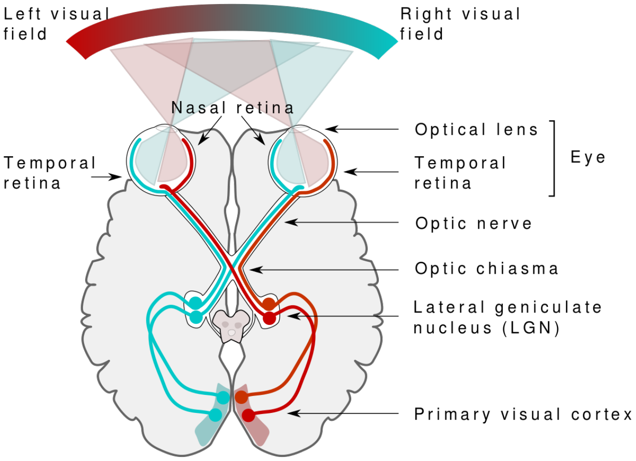


Figure 2: Visual pathway of the human brain [8].

2.1.2 Movements

Movement of the eye is controlled by six skeletal muscles attached to the outside of the eye, displayed in figure 3. The main purpose of these muscles and the movement of the eye is to position the eye in such a way that the incoming light is concentrated on the fovea centralis, the area of the retina that is most densely populated by photoreceptors. To keep the focus point on the fovea, the eye moves freely in its socket, either by slow or fast movements. Slow movements are used when tracking visual objects in the field of view and to counteract head movement. Fast movements, also called saccades, are small and fast movements of the eye where the gaze is snapped quickly to a fixed position. This kind of movement is used when a quick assessment of the visual field is needed, usually when searching for objects in the peripheral vision [6].

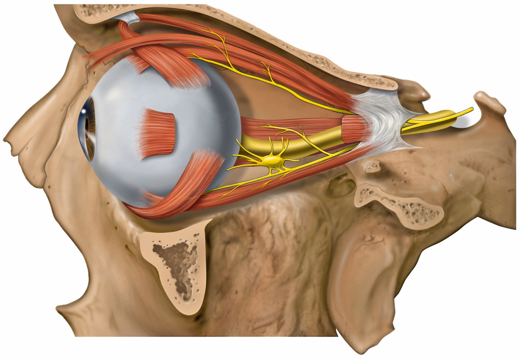


Figure 3: Lateral eye and orbit anatomy with nerves [9].

2.1.3 Nystagmus

Nystagmus is a natural occurance and is characterized by the involuntary movement of either one or both eyes. It is usually an oscillating movement between two set points in the field of view and when involuntary, it can create problems and difficulties when focusing the direction of the gaze on specific points in the visual field. From a clinical point of view, nystagmus is diagnosed and characterized from the oscillation, the direction of gaze and degree of conjugacy, i.e, how much the eyes move in tandem or together. Nystagmus is also diagnosed frequently based on its waveform, amplitude and frequency of its oscillations [10].

Nystagmus is something that can occur for all humans, diagnosis or not. This is called physiological induced nystagmus and is usually the result of rotation, either as an optokinetic response or vestibular movements. Nystagmus from an optokinetic response are induced by fast movements in front of the eyes and vestibular movements are an effect of the vestibular organs of the inner ear in response to movements of the body. The oscillations created by nystagmus themselves usually result in a drift of 3-4 degrees in amplitude and with a frequency of 2-3 Hz. Nystagmus of this type is a response to a moving world, and the visual system tries to keep the observed images of the visual world stable on the retina to maintain clear vision. This is a natural response to rotation or moving objects [10].

Besides naturally induced nystagmus as described above, there exist the field of benign nystagmus, which are different forms of diagnosed nystagmus. These are either congenital nystagmus or manifest latent nystagmus, literally nystagmus present from birth or for the case of manifest latent nystagmus, dormant until certain requirements are met. Both types have different types of waveforms but share similarities in jerkiness and usually horizontal pattern and are both acquired either at birth or at a very early age. They are usually a result of other disorders such as albinism, optic nerve hypoplasia and congenital cataracts, but congenital nystagmus is also present without any ocular or neurological disorders, called idiopathic congenital nystagmus. Patients with congenital nystagmus are able to acquire near-normal vision by developing something called “foveation periods”, where the patients have found either an epoch where the eye is still or looking at an object at specific angles where the nystagmus is suppressed [10].

Infantile idiopathic nystagmus is however the most common type of infantile nystagmus and has a specific type of movement, compared to manifest latent nystagmus. Despite its name of *idiopathic* nystagmus, there are signs towards a genetic fault that is also hereditary. Manifest latent nystagmus on the other hand has a slightly different movement than normal congenital nystagmus and is increased when one eye is covered. It is not uncommon that the intensity of the nystagmus is subdued substantially when both eyes are open, and that patients are able to acquire almost normal vision [11].

Lastly, there is acquired nystagmus, which is usually a result of disease, injury or drug intoxication, opposed to a genetic factor. In general, nystagmus can be traced from disturbances in the mechanisms that usually ensure a steady gaze. These mechanisms are visual fixation, far-right gaze and the vestibulo-ocular reflex, and any disturbance to these systems may show themselves in the form of acquired nystagmus. Fault of the visual system, causing visual loss, is a common precursor to nystagmus because visual fixation on desired objects is no longer possible, causing nystagmus. Diseases affecting the vestibular organ can also cause a certain type of imbalance and nystagmus, closely related to vertigo. Diseases to the cerebellum and the visual cortex are also responsible for several forms of nystagmus. All causes of nystagmus are not investigated or fully researched, but evidence point towards that certain types of nystagmus movements, that are either up-beat or down-beat have been mapped to different types of lesions on parts of the brain like the medulla and vestibulo-cerebellum [10].

2.2 Virtual reality

Virtual Reality is a computer simulated experience that can both be very similar to the real world or completely different. It tricks the brain into a high level of immersion by displaying, in real time, a computer-generated 3D (3-Dimensional) experience through optical lenses placed close to the user's eyes. Its applications include training simulators, education, medical uses and entertainment such as video games, to name a few. Virtual reality has been around in various shapes and forms since 1957 when Sensorama, a theater cabinet multimedia device that offered viewers an interactive experience, was created. The device stimulated the user's senses with a viewing screen for sight, oscillating fans for touch, devices that emitted smells and audio speakers for sound, creating a high sense of immersion [12].

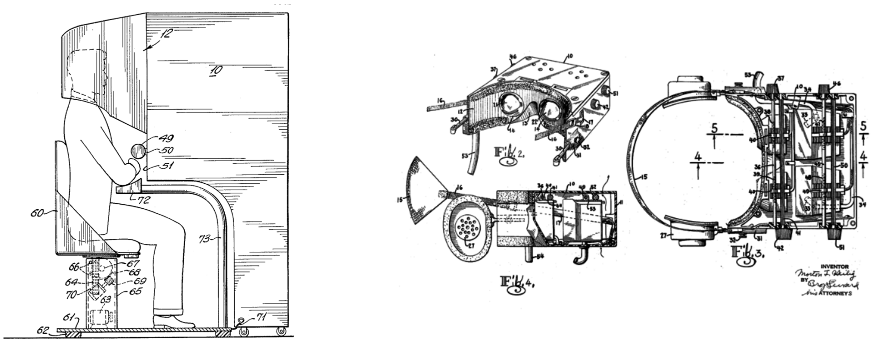


Figure 4: Sensorama cabinet and headset original patent drawings [13][14].

More modern VR systems consists of a wearable headset containing a screen combined with a stereoscopic lens for each eye. Some type of handheld controllers to interact with the world and a tracking system to track the position of the headset and controller's position and movement in 3D space, are also included [12].

In order for virtual reality to be possible, high-precision tracking of where the headset is located and how its position changes is required. This information is used to correctly calculate and render the correct images for each eye respectively when the user moves their head around. The headset itself contains various sensors such as gyroscopes, accelerometers and magnetometers which are very fast and precise at measuring movements in short time spans. However, they suffer from errors due to drift that emerges from integrating noise when being used for longer time spans. In order to combat this, the information from the inertial based sensors are combined with information from the slower and less precise optical tracking systems. This is known as sensor fusion, and it merges information from both the inertial and the optical tracking, compensating for each other's weaknesses to improve overall tracking, stability and precision. Another reason for using multiple tracking techniques is to enable tracking in 6 degrees of freedom. Meaning that both x, y and z room positions and the yaw, pitch and roll rotation of the headset are tracked, as can be seen in figure 5. This kind of freedom tracking is required for a full room-scale experience allowing the user to physically move around in the VR experience and not just look around from a fixed point [15].

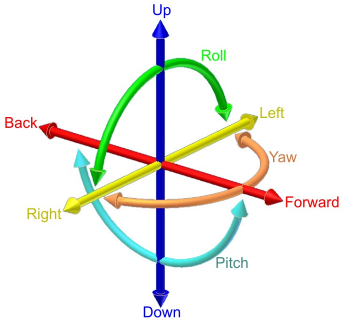


Figure 5: Six degrees of freedom [16].

For positional tracking, there are two common ways of tracking today. Inside-out tracking which is implemented by having cameras on the headset itself which then takes multiple pictures of the room and surroundings every second. Computer vision algorithms are then used and the difference between each picture is used to calculate the headsets movement and position. The other tracking type used is outside-in tracking which uses some type of external tracking devices, usually cameras or light sources to track the headset's movement. The headset used in this thesis employs outside-in tracking and can be seen in figure 6. [17]

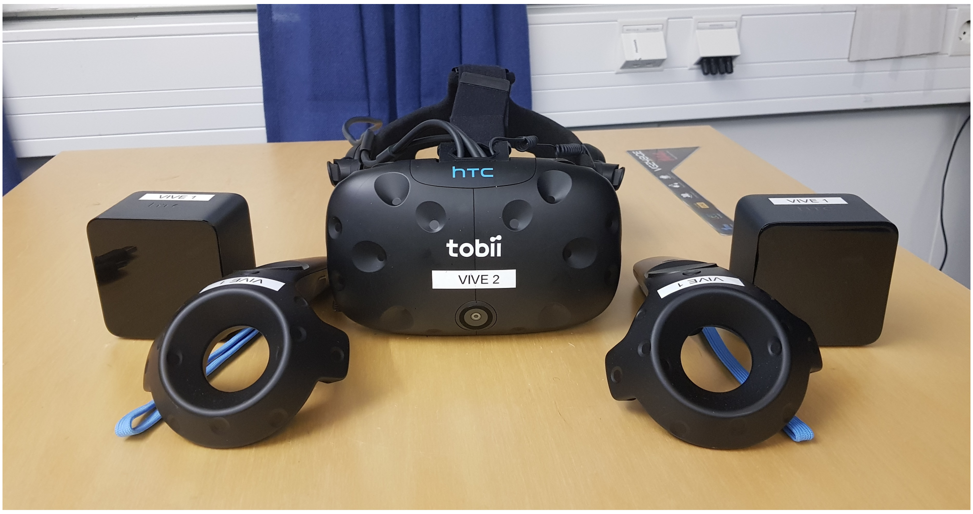


Figure 6: Tobii HTC VIVE Devkit with two controllers and Lighthouse base stations.

The Tobii HTC VIVE Devkit, which is the headset used in this thesis, see figure 6, uses two base stations placed in opposite ends of the room. Each station emit infrared light (IR) in three different phases 60 times per second. The first phase consists of an omnidirectional flash synchronously sent from both stations and this flash tells the headset to start a stopwatch command. The base stations then emit wide-angle two-dimensional laser beams across the room. This is done one axis at a time, left-right, then top-bottom. The headset contains an array of IR photodiodes which can be seen in figure 7, connected to a chip that produce a small electrical current when exposed to IR light. The chip then measures the time difference between the IR flash from the base stations and the time of each photodiode being hit by the laser sweep for each axis. With the help of these (tiny) time differences between the IR flash and swipes, combined with the known position of the IR-diodes on the headset and the angular speed of the IR swipe, the position and orientation of the headset can be calculated to sub millimeter precision. This information is then sent to the computer with other relevant sensory data for further processing [18].

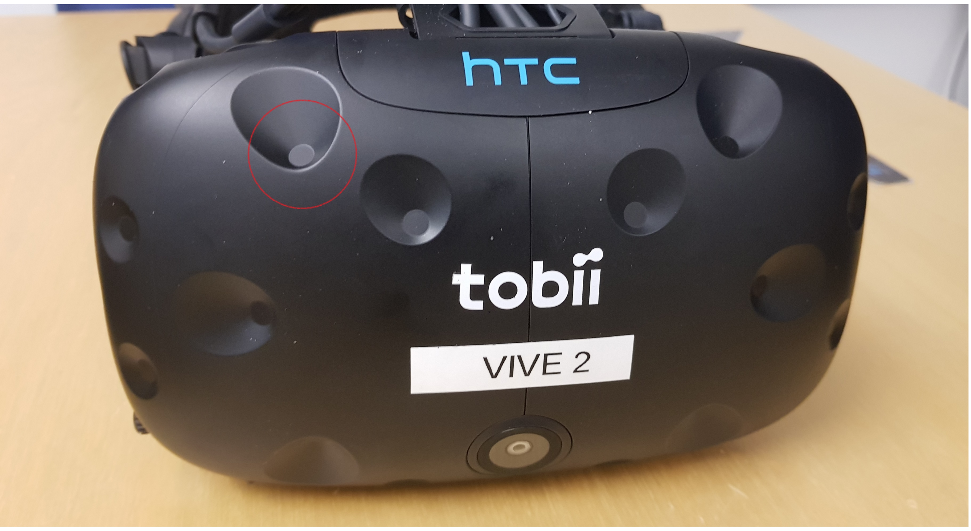


Figure 7: Image of Tobii HTC Vive with one out of several light sensitive photosensors marked with a red circle.

When the position and rotation of the headset is known, two images can be calculated and rendered, one for each eye. These images are then transformed from the flat 2D screens through the headset's lenses into a stereoscopic 3D image with the illusion of depth. This is possible due to the angle and type of lenses used in the headset. The lenses distort the two independent images, one for each eye, into the correct shape for how our eyes would normally experience them in the real world. The user can then move around and interact with this computer-generated world and experience it in a very immersive 3D experience in a way that is not possible with a normal 2D screen [18].

Virtual reality has for a long time been just a dream, depicted in old sci-fi movies and tv-shows as the future, and only recently seen noticeable progress. This is due to the high requirements placed on the hardware and software for a virtual reality experience. If the tracking of the headset is not accurate enough, problems with nausea and headaches known as VR sickness occur. It occurs similarly to motion sickness, e.g., when a person sitting in a car without looking at the road. It emerges when a person experiences physical movement, without perceiving the same movement with their eyes. This difference in perceived and real motion causes motion sickness. In virtual reality the same thing can occur, except the other way around. Instead of motion and it not being perceived, the opposite is true. The user is not moving while the experience in the VR-headset tells the brain that they are. Another cause for VR sickness comes from having too low frame rate when rendering the experience [19].

The screens used in VR applications are of very high pixel density, this is to compensate for the proximity of the screen to the user's eye and to avoid the screen door effect. The screen door effect is a visual artifact caused by the user being able to see the fine lines between each row of pixels, causing it to look like the experience is being viewed through a screen door net, hence the name. This high pixel density requirement and the need to render two images, one for each eye, at a high frame rate puts a very high computational load on the computer used. All of these factors combined have made acceptable VR experiences very hard to achieve. It is not until recently when the required technology has come up to par [20].

2.3 Unity

Unity is a game engine developed by Unity Technologies, it is used to create various applications such as 2D and 3D games, augmented reality and virtual reality. Unity works as an editor containing various libraries and frameworks for different types of applications, making it easy to create new experiences. It uses drag and drop for creating applications, but also allows for C# as a scripting language for creating more advanced experiences and applications.

2.4 Eye tracking

Eye trackers measures eye positions and movements, they are used to conduct research in various different fields such as marketing, research, psychology and medicine. The basic principle of today's modern technologies involves illuminating the eyes with light, taking pictures using cameras and then calculating the position of the eyes from the light reflection in the cornea and the position of the pupil [3].

While several examples of eye tracking have existed throughout the 20th century, only three methods became widely used during the later part of the 20th century and onward. The first and oldest method is based on magnetic sclerical search coils, where a lens containing a thin copper wire that can measure the variation in an electromagnetic field and send signals based on the eye's position. The second method is electrooculapraphy, where electrodes placed around the eye can measure the difference in the corneo-retinal standing potential between the front and back of the eye. The potentials change in such a way that the rotation of the eye in a certain direction either lowers or increases the underlying standing potential and can be used to determine the movement of the eye. Both of these methods have been determined to be rather cumbersome and invasive and have in most regards been solely replaced by video based oculography (VOG), where the eye tracking is based on recorded videos of eyes in motion. This thesis has only used video based eye tracking and will focus the information around that for the rest of this thesis [2].

2.4.1 Estimation of gaze

The central problem of eye tracking lies in finding the gaze, i.e., where the eye is pointing, seen from captured pictures of the eye. In short, this is usually done by locating the eye in each picture, finding the features of the eye and then mapping these features of the eye to the correct position in the world. Finding the features of the eye is a complicated process, but one useful and simple characteristic is to find the darkest pixels in the image in order to locate the pupil. This is one of the first steps involving the detection of the pupil and consists of a series of computational steps that create an image segmentation. One implementation of an image segmentation begins by converting the image into a grayscale image. From the grayscale image, a binary thresholding is implemented, creating a new image with only black and white regions. Lastly, by using an intensity threshold it is possible to find the estimated center of the pupil, based on the center of gravity of the binary image of the eye [2].

After this process is completed and the pupil has been found and highlighted in the original image, it is possible to begin the calculations involved in finding the eye gaze. Estimation of eye gaze is a little different depending on the origin of the eye-tracker, whether it is a head-mounted or a head-free system. With a head-mounted system, the gaze estimation is calculated from the three-dimensional position of the user's head. For a head-free or "remote" system, the solution is a little more technical and uses an infrared light source in order to create a reflection in the cornea. By illuminating the eye with infrared light, a small reflection in the cornea is created, usually called either "glint" or just corneal reflection, CR for short. The corneal reflection is used in gaze estimation because of its property to remain relatively stable while the eye is rotating and because it helps to distinguish between eye rotation and eye translation. This is used with the previously calculated position of the pupil to calculate the point-of-regard (POR) and an estimate of the eye gaze [2].

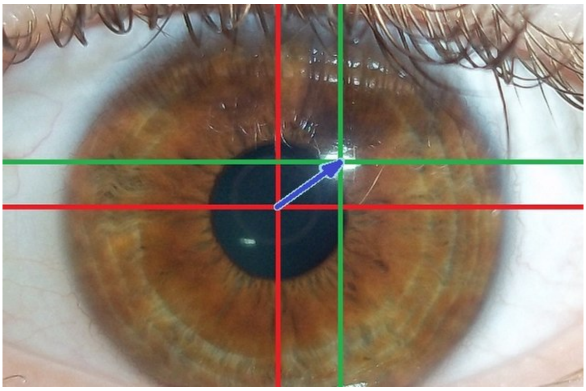


Figure 8: Visible light eye-tracking algorithm vector containing center of pupil and glint [21].

2.4.2 Calibration

The calibration step is in many ways the most important part of successful eye tracking. The goal is to give the features of the eye a location within the world, and without an established coordinate system the acquired data is just measured points in space without a reference point. Calibration is therefore the creation of such a coordinate system and is a required step at the start of every measurement for accurate and optimal data collection. Calibration of an eye tracker is done using a set of visual stimuli with known layout and coordinates, relative to the system in use. The stimuli is usually a set of carefully placed points on the screen that is shown for a short period of time before switching to the next position, i.e., "jumping points". These points are usually placed at nine coordinates, with one in the middle and the rest spread out along the perimeter. From these measurements, it is possible to create so-called mapping functions that defines the relationship between the gaze and the pupil and corneal reflection vector (PCR vector). Mathematically, this is defined as

$$x_s = f(x_e, y_e) \tag{1}$$

$$y_s = f(x_e, y_e) \tag{2}$$

where x_e and y_e are data points from the eye tracker and x_s are estimated coordinates from the screen. These measurements are used in creating a mapping model that places the measured points at approximately correct positions. These mapping models can either be parametric, using polynomial functions, or non-parametric, using a neural network. Using polynomial functions, either linear, quadratic or cubic, the 2D gaze positions are calculated [2]. After calibration, a validation step is taken to quantify the error of the mapping model. This step is similar to calibration but instead of calculating a mapping model, the estimated position from the validation together with the actual position is used to calculate the visual angle between these points, thus delivering a measure of the quality of the calibration [3]. This error is usually calculated using either Root Mean Square Error (RMSE) or determination coefficient [2], and is described further in 2.4.4.

2.4.3 Measurement and event detection

Measurements done by eye trackers are very useful for specific tasks, for example diagnosing specific eye diseases or cognitive functions that closely relates to eye movements. Two specific eye movements are usually detected when investigating eye movements, labeled as fixations and saccades. Fixation is defined as a specific moment when the user keeps their eyes fixed at a position in space and the gaze is almost still. Fixation is determined by its duration, which include a start and stop time, the coordinates of the fixated point and dispersion. The dispersion can be used to calculate the precision of the general measurement and the system at large. The other movement is saccades, which is defined as quick changes of the eye position, i.e., when different positions of fixation changes rapidly. For saccadic movement there is a particular interest in the duration, the amplitude between the two coordinates, i.e., the distance between the two coordinates of the saccadic movement and the speed and acceleration of the eye movement. The properties of the specific parameters of both fixations or saccades are great tools and can be used when detecting and diagnosing certain diseases or cognitive patterns [2].

2.4.4 Accuracy and precision

When talking about eye trackers and their performance, accuracy and precision are two very important metrics. Eye trackers with good accuracy and precision will provide higher quality data, as they are able to more truthfully reflect where the user is looking. For a visual explanation of accuracy and precision, see figure 9.

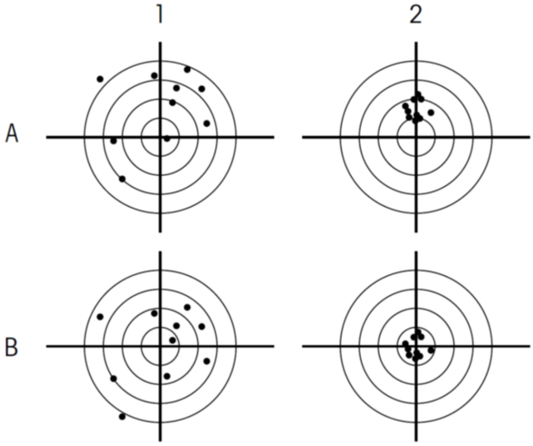


Figure 9: Figure depicting the behavior of accuracy and precision with A1 : Poor accuracy and precision A2 : Poor accuracy good precision B1 : Good accuracy poor precision B2 : Good accuracy good precision.

Accuracy and precision are measured in terms of gaze angles. In figure 10, the (x, y) position represents the gaze position of the subject, whereas $(0, 0)$ represents the current stimuli position. The gaze angle θ is expressed as the deviation in degrees between the two points, with the point of origin determined by the position of the eye and d being the distance from the eye to the stimulus.

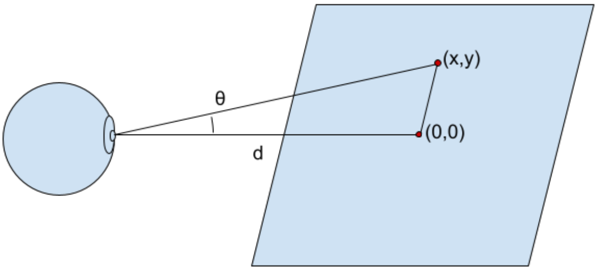


Figure 10: Gaze angle θ, from recorded gaze position and stimuli.

Accuracy is defined as the average gaze angle difference in measured gaze position and the actual position of the stimuli. It is calculated for the horizontal and vertical plane separately and is then combined for both planes with the following equations.

$$\theta_h = \frac{1}{N} \sum_{n=1}^{N} \arctan\left(\frac{|x_n - x_0|}{d}\right) \tag{3}$$

$$\theta_v = \frac{1}{N} \sum_{n=1}^{N} \arctan\left(\frac{|y_n - y_0|}{d}\right) \tag{4}$$

$$\theta_c = \frac{1}{N} \sum_{n=1}^{N} \arctan\left(\frac{\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2}}{d}\right) \tag{5}$$

Where $\theta_h, \theta_v, \theta_c$ is the horizontal, vertical and combined accuracy in degrees of gaze angle. x, y is the measured gaze position, x_0, y_0 is the position of the stimuli and d is the distance from the eye to the stimuli.

Precision is the ability of the eye tracker to reliably reproduce the same gaze point measurement. It measures the variation of the recorded data and can be calculated in various ways, one such being the Root Mean Square (RMS) calculation of successive samples (Holmqvist et al., 2011). Precision is then calculated by taking the Root Mean Square from successive data points in degrees of visual angle θ between each sample, see figure 11 for an image illustrating precision.

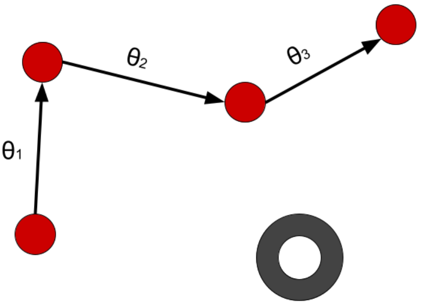


Figure 11: Figure depicting the gaze angle between successive samples for calculation of precision. The red dots represent successive recorded samples from the eye-tracker and the gray circle represent the stimuli.

Precision is calculated for the horizontal and vertical plane separately and is then combined for both planes with the following equations.

$$\phi_h = \sqrt{\frac{1}{N}\sum_{n=1}^{N} {\theta_{n_x}}^2} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (\arctan{(\frac{|x_n - x_{n+1}|}{d}}))^2} \tag{6}$$

$$\phi_v = \sqrt{\frac{1}{N}\sum_{n=1}^{N} {\theta_{n_y}}^2} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (\arctan{(\frac{|y_n - y_{n+1}|}{d}}))^2} \tag{7}$$

$$\phi_c = \sqrt{\frac{1}{N}\sum_{n=1}^{N} {\theta_{n}}^2} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (\arctan{(\frac{\sqrt{(x_n - x_{n+1})^2 + (y_n - y_{n+1})^2}}{d}}))^2} \tag{8}$$

Where ϕ_h, ϕ_v, ϕ_c is the horizontal, vertical and combined precision in degrees of gaze angle between successive samples for the x and y coordinates while N is the total number of samples and d the distance from the eye to the stimuli.

2.4.5 Eyetracking in VR

The equipment used in this thesis, the Tobii HTC VIVE, uses 10 infrared illuminators per eye which can be seen in figure 12, to generate reflective patterns on the corneas of the user's eyes. These reflection patterns, along with other visual data about the subject and images of the user's eyes, are collected by the image sensors by using binocular dark pupil tracking as the principal tracking technique [22]. Dark pupil tracking creates an image where the pupil appears darker than the iris and is easier to track by placing the IR illuminator away from the optical axis, in contrast to bright pupil tracking where the illumination is placed along the optical axis [23]. From the collected data, image processing algorithms then identify relevant features, including the pupil and the corneal reflection patterns. From this, a 3D representation of the user's eyes is created and the gaze direction of the eyes can be calculated [22].

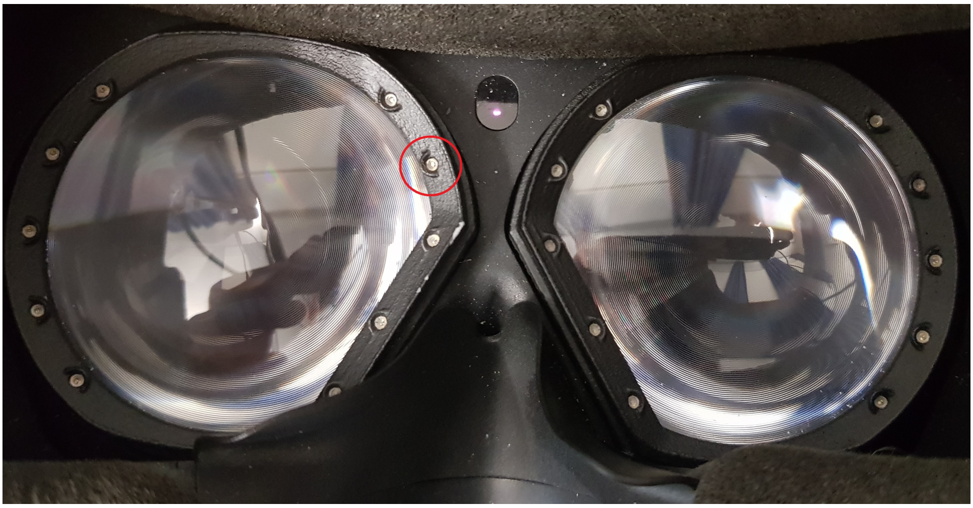


Figure 12: IR Lights in headset, 1 out of 20 illuminators marked with a red circle.

2.5 Nystagmus modeling and quality assessment of data

Nystagmus waveform modeling is a powerful tool for analyzing nystagmus oscillations. By modeling the nystagmus oscillations from a signal containing nystagmus oscillations and then comparing the resulting waveform model to the original signal, one can measure how much noise there is in the system. In a perfect system with a perfect test subject that could perfectly keep their eyes focused except for their nystagmus oscillations on a fixed target, the resulting nystagmus model would be equal to the original signal. By measuring how well the waveform model matches the original signal using the normalized segment error (NSE) one can measure how well the system is able to capture the nystagmus oscillations and how much noise and errors there are in the system. The normalized segment error is described in section 2.5.3, and is computed as the normalized error between the waveform model and the original signal.

The model used for this is based on work done by William Rosengren [5] and is based on a pseudo-stationary assumption of the nystagmus signal. Consider equation 9 with the harmonic sinusoidal model $s[n]$ with H harmonics.

$$s[n] = \sum_{h=1}^{H} = s_h[n] \tag{9}$$

Where

$$s_h[n] = a_h sin[2\pi(f_1 h)n + \phi_h] \tag{10}$$

and, a_h is the amplitude of the h:th harmonic, f_1 is the first harmonic frequency and ϕ_h is the phase of harmonic h, and $n = 0, \ldots$, N-1 .

Estimating the model parameters is done in two steps, first the signal is pre-processed and the harmonic components are extracted which is described in section 2.5.1. The second step is to estimate the model parameters, which is described in section 2.5.2.

2.5.1 Signal pre-processing

The pre-processing is performed in two steps, first the signal is downsampled to 100 Hz, secondly the signal is high passed using a third order Butterworth filter with a cutoff frequency of 2 Hz, removing frequencies lower than the cutoff.

In order to estimate the harmonic components, $s_h[n]$, each component is extracted from the pre-processed signal. The global first harmonic frequency of the signal, $\hat{F}_1$, is estimated using Welch spectrum estimation with an overlap of 50% and a segment length of 512 samples. The harmonic components $s_h[n]$, are computed as the pre-processed signal filtered through a Kaiser band pass filter, $B_h(f, F)$ with the following design parameters for the first harmonic.

$$B_1(f, F) = \begin{cases} 1, & \text{if } |F - f| \leq f_{w_1} \\ 0, & \text{if } |F - f| > f_{w_1} \end{cases} \tag{11}$$

And,

$$B_h(f, F) = \begin{cases} 1, & \text{if } Fh - (1 + \delta_h) \leq f \leq Fh + (1 + \delta_h) \\ 0, & \text{if } Fh - (2 + \delta_h) \leq f \leq Fh + (2 + \delta_h) \end{cases} \tag{12}$$

for $h > 1$ where $\delta_h = \frac{(h-1)}{2}$ and $f_{w_1} = 1.3$, $f_{w_2} = 2.3$ for this work.

2.5.2 Block model parameter estimation

In general, the frequency of the nystagmus oscillations are not stationary. The signal is therefore divided into short segments of length N_b. When choosing the segment length, there is a trade-off between having the segment length long or short. If it is too short, it may result in poor parameter estimates, and if it is too long, the stationarity assumption of each segment may not be valid. The segment length for this work is set to $N_b = 67$ which is 0.67 seconds and by using a method of overlapping segments the stationarity assumption problem is addressed. Another problem is that the frequency estimate $\hat{F}_h$ is not necessarily representative for all intervals in the recorded signal. If the first harmonic frequency varies more than ±1.3 Hz, the output energy of the affected segments may be severely reduced. In order to remedy this, two additional sets of harmonic components are computed for the frequencies $\hat{F}_0 = \hat{F}_1 - 2.6$ and $\hat{F}_2 = \hat{F}_1 + 2.6$. The frequency estimate for the time interval $\boldsymbol{n}_b = [n_0, \ . \ . \ . \ , n_0 + N_b$ - 1] is determined by maximizing the first harmonic energy,

$$\hat{F}\left[\boldsymbol{n}_b\right] = \text{argmax}_{\hat{F}_i} \left[E\left(\hat{F}_0\right)\left[\boldsymbol{n}_b\right], E\left(\hat{F}_1\right)\left[\boldsymbol{n}_b\right], E\left(\hat{F}_2\right)\left[\boldsymbol{n}_b\right]\right] \tag{13}$$

where,

$$E\left(\hat{F}_i\left[\boldsymbol{n}_b\right]\right) = \sum_{k=n_0}^{n_0+N_b-1} \left|s_1^{(i)}[k]\right|^2 \tag{14}$$

and $s_1^{(i)}\left[\boldsymbol{n}_b\right]$ is the resulting first harmonic signal after $s_p[\boldsymbol{n}_b]$ is filtered through $B_1(f, F)$ and where $F = \hat{F}_i$. The signal however is reconstructed for a time interval $n_c \in [l - c_0, l + c_0]$ where overlap c_0 is computed as

$$c_0 = \frac{N_b}{2\hat{F}\left[\boldsymbol{n}_b\right]} \tag{15}$$

n_0 is the start sample for the interval of length N_b and

$$l = n_0 + \frac{N_b}{2} \tag{16}$$

Every approximate wave is reconstructed separately based on a window around it. The frequency f_h and phase, ϕ_h of each harmonic are estimated according to [24].

$$\hat{f}_h = \underset{f}{\text{argmax}} \left|\sum_{n_b=n_0}^{n_0+N} s_h\left[n_b\right] e^{-j2\pi f n_b}\right| \tag{17}$$

and

$$\hat{\phi}_h = \arctan\left(-\frac{\sum_{n_b=n_0}^{n_0+N} s_h\left[n_b\right]\sin\left(2\pi\hat{f}_h n_b\right)}{\sum_{n_b=n_0}^{n_0+N} s_h\left[n_b\right]\cos\left(2\pi\hat{f}_h n_b\right)}\right) \tag{18}$$

The amplitude, $\hat{a}_h$, for the h:th harmonic is estimated from the analytic signal transformation [25].

$$\hat{a}_h = \frac{1}{2c_0}\sum_{i=l-c_0}^{l+c_0-1}\left|\tilde{s}_h[i]\right| \tag{19}$$

Where $\tilde{s}_h[i]$ is the analytical transformation of s_h.

The signal is not stationary unless $\hat{f}_h = h\hat{f}_1, \forall h$, which is not generally the case. In order to create a stationary model, equation 10 is rewritten as

$$s\left[n_b\right] = \sum_{h=1}^{H} a_h \sin\left[2\pi\left(\hat{f}_1 h\right)n_b + 2\pi\left(\hat{f}_h - \hat{f}_1 h\right)n_b + \hat{\phi}_h\right] \tag{20}$$

where $\hat{f}_h$ is the frequency estimate of harmonic h. The second argument of the sinusoid, $2\pi\left(\hat{f}_h - \hat{f}_1 h\right)n_b$, may be viewed as a phase component. In order for this to be stationary, the index n_b is replaced by a fixed index value, for example the block center index l [24]. The end result is the following model

$$s'\left[n_b\right] = \sum_{h=1}^{H} \hat{a}_h \sin\left[2\pi\left(\hat{f}_1 h\right)n_b + \hat{\phi}'_h\right] \tag{21}$$

where

$$\phi'_h = 2\pi\left(\hat{f}_h - \hat{f}_1 h\right)l + \hat{\phi}_h \tag{22}$$

2.5.3 Normalized segment error (NSE)

As described in section 2.5, the NSE is introduced to measure the normalized error between the original signal and the nystagmus waveform model. The NSE of a segment with length N_s is calculated as

$$NSE_s = \frac{\sum_{n_s=n_0}^{n_0+N_s-1}\left|s'_p\left[n_s\right] - s'\left[n_s\right]\right|^2}{\sum_{n_s=n_0}^{n_0+N_s-1}\left|s'_p\left[n_s\right]\right|^2} \tag{23}$$

where

$$s'_p[m] = s_p[m] - \frac{1}{N_s} \sum_{n_s=n_0}^{n_0+N_s-1} s_p\left[n_s\right] \tag{24}$$

and

$$s'[m] = s[m] - \frac{1}{N_s} \sum_{n_s=n_0}^{n_0+N_s-1} s\left[n_s\right] \tag{25}$$

.

The signals $s_p[n]$ and $s[n]$ denote the pre-processed and reconstructed signals, if $NSE_s > \epsilon$ for some value ϵ the segment error between the original signal and the waveform model is too large, and the segment is rejected. The choice of $\epsilon = 0.18$ in this work is based on the work done by William Rosengren [5] and is chosen by doing *receiver operating characteristics* (ROC) on datasets with known nystagmus waveforms and then picking the ϵ which maximizes the number of segments with true nystagmus waveforms (true positive rate) while minimizing the segments with unwanted waveform modulations (false positive).

3 Method

In the introduction, the four principal goals of the thesis were established, and they are repeated below.

- Recreate and implement existing research protocol for the recording and analysis of nystagmus in software used by the VR headset.
- Record eye movements from participants with and without nystagmus using aforementioned research protocol.
- Evaluate the difference and benefits between the stationary eye tracking system EyeLink 1000 and the wearable eye-tracker and VR headset Tobii HTC Vive. Differences like calibration, precision, accuracy, data quality and effort in terms of processing the acquired data.
- Investigate and examine new possibilities and eventual advantages and disadvantages using head-mounted displays and eye tracking in eye diagnostics.

The overarching goals were, as presented, to find a feasible model and method of downsizing the concept for the EyeLink 1000, and recreate it on the HTC VIVE. The first research question on how to develop and design the existing protocol is presented in section 3.1. The procurement of data and the recording of the test subjects are presented in section 3.2 to 3.4, where the test equipment is shown, the exact testing procedure is presented and how the recording was made. Section 3.5 presents the first part of the signal processing and in section 3.6-3.7 the process of evaluating the acquired data from the eye trackers are presented. The two last research questions concerning the evaluation of the specific systems are closer examined in the discussion, chapter 5.

3.1 Construction of the test protocol

The building phase of the project began with learning and adapting the desired tools for creating the test protocol: the game engine Unity and its scripting language C#. When the tools had been sufficiently mastered, the design process began.

3.1.1 Design choices

From the outset, no particular boundaries were set for recreating a nystagmus test in virtual reality compared to earlier works. Initially, inspiration was taken from work done by Clay et. al. where an entire village had been built and where the user is free to roam around for a predetermined amount of time.

While interesting as a concept, it was determined to take too much effort and time to learn and actually build an entire virtual village. The next idea was to scale down the “roaming” aspect and instead build both a simpler world containing just simple geometric shapes and restrict the user’s movement to being moved along a predetermined path, like in a roller coaster. This idea retained the possibility of creating a test protocol that took advantage of the 3D environment, but created other potential undesired problems, like extra data and coordinates because of the eye tracker being in a 3D-world rather than a 2D-screen, and the potential risk of nausea among users due to VR sickness.

In the end it was decided for the sake of simplicity and, most notably, accuracy across platforms and systems, to focus on the recreation of earlier tests rather than innovation of said test protocols. The main goal thus became to recreate a previous test protocol as closely as possible in use for diagnosing nystagmus, used previously in work done by William Rosengren [5]. These tests are designed as “stationary”, in that they were created with a screen in mind displaying the stimuli and the eye tracker in between the user and the screen capturing the data. As described above, instead of creating a new kind of testing environment, it was decided to recreate a similar test in virtual reality as it is set up in an actual test environment done in the laboratory in order to be able to compare the results.

3.1.2 Creative necessities and parameters

The first priority was to find a way that simulates the way the normal test is carried out, with a screen displaying stimuli and where the participant’s head is placed in a fixed position, not allowing excessive head movement during the recording. This was done in order to replicate the original test setup and to force specific eye movement patterns. For the actual test protocol, the aim was to recreate normal eye movement that is part of daily life: fixation, saccades and smooth pursuit. In addition to the normal eye movements, a section inducing nystagmus was also included, called optokinetic nystagmus. The properties of these test sections are explained in detail in 3.1.3, and they are all present in previous work designed by Rosengren. The following test segments are, in order:

- Calibration
- Fixation
- Saccades
- Smooth pursuit, horizontal and vertical
- Optokinetic nystagmus, horizontal and vertical

Even if the scope of the test in virtual reality were downsized and simplified, Unity as a building tool was still used, primarily for its ease of use, reliability and possibility to change between VR and a non-VR environment with a simple adjustment in the preferences of the program.

By deconstructing the actual test environment to its bare-bones components, it was possible to determine that the test is just composed of two basic parts, (I) a screen in a fixed position that displays (II) a visual stimulus. Both the screen and visual stimulus was easily created by two different geometric shapes, a cube stretched into a wall and a flattened sphere. The sphere was then colored red and placed against the white background of the wall for visibility. For the test in question, it was also necessary to decide whether free head movement should be allowed by the user in virtual reality. It was briefly discussed, but ultimately decided to not allow head movement by the user. This was chosen to more accurately resemble the real test where the user has their head fixed in a specific mount in front of the eye tracker and to make sure the user, when placed in the VR simulation, do not adjust their head to better focus on the visual stimuli. Coincidentally, by making the user experience independent of their head movement it makes this kind of tests available to patients and users that may not be able to place their head in a fixed position, for example young children or patients with Parkinson's disease. The movement was restricted by fixing the in-game camera to the wall object, so wherever the camera is pointing, it points directly into the wall, obscuring the entire field of view.

For the actual test protocol, the design and movement of objects were determined by an outside script that ran at the start of the program. The scripts handled coordinates of the desired positions and duration of each position.

3.1.3 Design of the test protocol

As described earlier, a complete nystagmus test consists of five parts. The most crucial part of any eye tracking procedure is the calibration, in order to get accurate data. Calibration was not possible to implement in the program itself since there was no way to access the calibration settings or protocol directly and was instead solved using the included third-party software from

Tobii, a VR-ready game with eye tracking possibility called Mirrors. The first step of the calibration was to make sure that the participants eyes were in the correct position, which is in the center of each individual lens. A heads-up-display shows when the eyes are in a good position, signaling green. After the eyes were in a good position, the user signals to continue and the process of calibration began. The calibration was used, so the system was adapted to the user's eyes and then accurately portrays where the user was looking. This was done by having the user look at 5 predetermined locations: the middle of the screen and then the corners. No validation of the calibration was performed and after the sequence was finished, Mirrors could be shut down and the nystagmus test protocol in Unity could be started.

Starting the test, the user was presented with some basic information before starting the first part of fixation. The fixation protocol tests the user's ability to keep an object in their gaze for extended periods of time. The red sphere started in the middle of the screen, then changed positions four times, up, down, right and left corner of the screen, in that order. Each segment was 20 seconds long.

Next was saccadic movement. The red sphere started in the middle of the screen and was then moved around the corners of the screen at the same positions as the fixated positions. The whole segment was 90 seconds long, where the sphere stayed in position for 1.5 seconds before being moved to a new random position. The positions were selected randomly, so the participant could not memorize or anticipate the positions. The changing of positions was fast enough to create saccades rather than fixation. In total, the participant changed eye positions 60 times for the duration of the part of the test protocol.

Lastly was the smooth pursuit test, where the participant would follow the sphere with their gaze, now moving continuously across the screen. The smooth pursuit was divided into two parts, one with horizontal movement and the other with vertical movement. The sphere started in the middle and moved between two positions, with a slow increase of speed. The test was 20 seconds long before continuing with the vertical test, which had the same properties as the horizontal one.

The final test deviated from the previous test by incorporating a movement pattern that induced optokinetic nystagmus instead of having the participant follow a visual stimulus. The visual stimulation was an alternating pattern of black and white bars that moved along the screen, either horizontally or vertically. The participants would focus on an unspecified point in the middle of the test, inducing natural nystagmus movement of the eyes. Both parts were 15 seconds long.

3.1.4 Additional fixes

The test protocol was initially developed to be run on a VR platform, but as the thesis progressed it also became apparent that a test protocol was needed for testing of the stationary platform EyeLink 1000 as well in order to be able to compare the VR systems test protocol to some kind of baseline. Fortunately, Unity has an option that can switch seamlessly from running in VR to running on a normal screen. After feedback from users, the pauses between the four different tests were better highlighted using color coding of the sphere, where green indicates a pause and red determines running of the test letting the user briefly rest their eyes between each test.

3.2 Test equipment

Testing was carried out at the Lund University Humanities Lab, Sweden.

3.2.1 VR setup

PC-Hardware
Asus laptop with an Nvidia Geforce GTX 1080
Intel Core i7-8750H CPU 2.2 GHZ 6 cores 12 threads
16 GB RAM

Tobii HTC VIVE Devkit [22]
OLED Display with a Resolution of 2160x1200 (1080×1200 per eye)
Display refresh rate: 90 Hz
Field of view: 110°
Sensors: Accelerometer, Gyroscope, Lighthouse laser–tracking system

Eye-tracking [22]
Number of IR illuminators: 10 per eye
Eye tracking sensors: 1 per eye
Tracking technique: Binocular dark pupil tracking
Gaze data output frequency: (Binocular) 120 Hz
Trackable field of view: 110° (Full HTC Vive field of view)

Software
Unity version 2019.4.19f1 for creation of test routine
Tobii XR SDK API for providing access to core eye tracking data
Mirrors v2 from Tobii for calibration of eye-tracker

3.2.2 EyeLink Setup

PC-Hardware
EyeLink 1000 Plus Host Computer

Screen for presenting stimuli: EIZO FlexScan EV2451 23.8” with (1920x1080p) resolution

EyeLink 1000 Plus [22]
EyeLink 1000 Plus IR Light and Camera

Eye-tracking [22]
Tracking technique: Pupil with Corneal Reflection (CR)
1000 Hz Binocular
Trackable field of view: 60° horizontally, 40° vertically

Software
Host software v. 5.09 and the DevKit 1.11.571.

3.3 Testing and test subjects

In order to acquire data that could be analyzed, a lot of experiments had to be performed. After the test protocol was finalized, early testing among the two authors began in order to test the data performance and get a sense of the properties that could be achieved from the sampled data. After determining that the sampled data was satisfactory, the actual testing could be fully committed to. Because of the ongoing pandemic of 2021 only a small amount of testing subjects could be used, which mostly consisted of the two authors, male between 25-29, one with nystagmus and one without. Thus, we had test subjects that could be considered the control group **CG** and the nystagmus group **NG**.

After several test runs and unfortunate delays, testing of the VR protocol began on Monday 19th April and went on for 5 days, lasting to Friday 23 April. To get plenty of data that also could be considered somewhat diversified, two tests were done every day, one in the morning and one in the afternoon for both the control group and the nystagmus group.

3.3.1 Test setup - VR

The test subject was placed on a chair in the test room, with the test computer in front and the VR base stations placed at each end of the room to give good visibility for the VR-headset. The test subject then placed the headset on their head and made sure that it sat comfortable. When the subject was ready, the test leader started Mirrors in order to calibrate the headsets eye tracker for the test subject’s eyes. The test subject adjusted the placement of the headset and initialized the calibration with one of the VR system’s controllers when ready, and the calibration protocol began. After the calibration was completed, the test subject told the test leader to start the actual nystagmus test protocol. The test leader closed Mirrors and started Unity in order to run the nystagmus test protocol. The test leader told the outlines of the test and told the subject to keep their head still, follow the red sphere with their eyes and try to keep the blinking to a minimum except at the pauses between each section of the test. When the test was finished, the test leader closed the program and the test subject was permitted to remove the VR-headset. If there were additional tests to be done, the roles were switched between the test subject and the test leader.

3.3.2 Test setup - EyeLink 1000

The EyeLink test, also referenced as a “stationary” test, also took place at the Lund University Humanities Lab. At its core, the same type of test protocol was used as in the VR test, but changed slightly, so it would work without VR and instead be displayed on a normal screen. The stationary eye tracker is composed differently than the VR counterpart, both in terms of structure and calibration. The test consisted of one test subject and one test leader that gave instructions and controlled the test and all the equipment.

When the test was about to begin, the test subject sat in a designated chair in front of the screen. The EyeLink eye tracker was located in between the screen and the user. The test subject placed their head on the head support for the eye tracker, and the test leader started the calibration process of the EyeLink 1000. While keeping the head still in the head support, the test leader adjusted the estimated pupil size of the test subject and then initialized the randomized 9 point calibration process. After the EyeLink had been calibrated and correct reference points had been established, the EyeLink initialized a validation to ensure that the EyeLink had been properly calibrated. If everything was deemed ok, the system was good to go and ready to record data. The test leader exchanged the HDMI cable from the computer that displayed the calibration to another computer that displayed the test protocol, while the test subject kept still in the head rest. When everything was correctly plugged in, the EyeLink started to record data and the test protocol was started. For this test, however, only the first section of the otherwise complete nystagmus test was completed, since only the fixation part was interesting for this particular measurement. After the test protocol was concluded, the test leader stopped the EyeLink recording and the test subject was allowed to exit the head rest.

The recording of the EyeLink data was not as thorough as its VR counterpart because of the added complexity of the system and additional requirement of specialized software. Some early test recordings took place in May in order to evaluate the data and its structure in relation to the data measured from the VR-headset. Two measurements were made by **CG** for evaluation purposes and when the appropriate data pipelines were constructed to read and process the collected data, a day of testing was committed to record data from several individuals. Three individuals from **CG** and one from **NG** partook in recordings of the VR system and EyeLink to achieve comparable data from the systems. One individual from **CG** also did 10 separate recordings from the EyeLink in order to have the same amount of recordings as from the VR system so a full comparison between them could be made. Additional recordings from the EyeLink were also done by **NG** in order to have more comparable data to the VR dataset.

3.4 Data recording

Data recording functioned differently for each system and in order to compare the results from both systems against each other, data was first recorded, processed and then converted to have equal structure before evaluating the data with identical analysis algorithms.

Tobii HTC VIVE Devkit

For the VR system, data was sampled at 90 times per second by the eye tracker, which was the same as the frame rate of the headset's display. The eye tracker itself supports up to 120 Hz sampling rate, but when enabled, some timeframe problems arose and duplicates of data were sampled, so the sample rate was instead reverted to 90 Hz. By using the Tobii XR SDK API and a script in Unity, eye tracking data from the eye tracker was obtained each frame update in Unity. This data contain the gaze ray and timestamp of the sampled data. The gaze ray consisted of a normalized three-dimensional vector defining the binocular gaze vector in the direction of the user's gaze and the origin in Cartesian x, y and z coordinates of said vector.

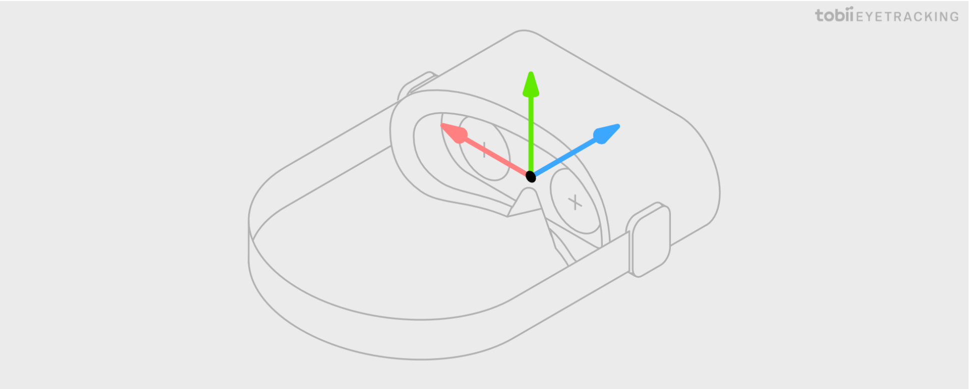


Figure 13: Tobii HTC VIVE Devkit with gaze ray origin source: Tobii AB.

This vector position and direction was then used to calculate where, relative to the nystagmus test surface, the user was currently looking. By calculating the intersection of the gaze ray and the test surface using a built-in function in Unity called ray casting, which takes a vector and its origin as input and then outputs the intersection point of the first collision with another object in the game, coordinates of where the user was looking on the test surface could be obtained.

Since the test surface was locked to the view of the headset, the position of the test surface relative to the game world changed when the user moved their head. In order to get the position of where the user was looking relative to the test surface instead of relative to how the user is moving their head, the following formula was used.

$$(x_r, y_r) = (x_i, y_i) - (x_w, y_w) \tag{26}$$

Where (x_r, y_r) is the x, y coordinates relative to the test surface with the center of the test surface as the origin, (x_i, y_i) is the x, y coordinates for the gaze ray and test surface intersection point and (x_w, y_w) is the x, y coordinates for the world position of the test surface with its center as the origin. As the test surface is always the same fixed distance from the headset, no calculations had to be done for this axis.

This information was then written to a JSON file at each frame update. JSON is a file structure that stores simple data structures. Each entry contains the x, y coordinates of where the user was looking on the test surface, the stimuli positions on the test surface and the timestamp of the sample. The JSON file is structured in a way so that it can be opened and read by a python script using a JSON package later for analysis, see below for example of JSON structure.

```
{’nbr’:0, ’time’:20.369, ’x’:-0.039, ’y’:0.078, ’dotx’:0, ’doty’:0}
{’nbr’:1, ’time’:20.380, ’x’:-0.052, ’y’:0.071, ’dotx’:0, ’doty’:0}
{’nbr’:2, ’time’:20.392, ’x’:-0.054, ’y’:0.061, ’dotx’:0, ’doty’:0}
{’nbr’:3, ’time’:20.403, ’x’:-0.057, ’y’:0.053, ’dotx’:0, ’doty’:0}
{’nbr’:4, ’time’:20.414, ’x’:-0.059, ’y’:0.049, ’dotx’:0, ’doty’:0}
```

Where ’x’, ’y’ are the x, y coordinates relative to the test surface of where the user was looking and ’dotx’, ’doty’ are the x, y coordinates of the stimuli, ’nbr’ was the sample number and ’time’ was the time in seconds since the data sampling started, see figure 14 for flowchart of recording and saving VR data.

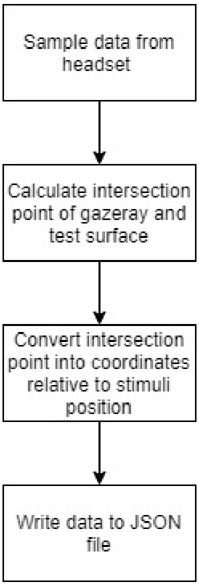

Sample data from headset
Calculate intersection point of gazeray and test surface
Convert intersection point into coordinates relative to stimuli position
Write data to JSON file

Figure 14: VR Data collection Flowchart.

EyeLink 1000

For the EyeLink system, a lot of previous material such as calibration protocol, recording and saving the data in a file already existed and was reused when performing the nystagmus test protocol. The output data structure was however not comparable and the EyeLink data first had to be converted to the same data structure as the VR system, so the same test algorithms could be performed on both systems, see below for original EyeLink output file structure.

```
       TIME    L.X    L.Y    R.X    R.Y
0  230089.0  990.6  557.6  958.3  555.5
1  230090.0  990.2  556.9  959.0  555.3
2  230091.0  991.4  558.5  958.5  556.1
3  230092.0  992.4  558.7  958.9  555.2
4  230093.0  989.2  558.1  959.1  556.8
```

Where TIME is the time in milliseconds since the EyeLink started sampling data, L.X, L.Y is the left eye's x, y estimated gaze pixel value with the origin (0,0) being in the top left corner of the screen displaying the stimuli, R.X, R.Y is the right eye's x, y estimated gaze pixel value, the EyeLink samples data at 1000 Hz.

The first step was to convert all samples of separate gazes for each eye into a single binocular value for both eyes. This was done by taking the average gaze value of both eyes for both x and y values with the following formula.

$$(x, y) = (\frac{R.X + L.X}{2}, \frac{R.Y + L.Y}{2}) \tag{27}$$

With (x, y) being the new binocular value. The data was then structured in the same JSON format as the VR system, see below for structure.

```
{’nbr’: 0, ’time’: 0.0, ’x’: 974.45, ’y’: 556.55}
{’nbr’: 1, ’time’: 1.0, ’x’: 974.60, ’y’: 556.10}
{’nbr’: 2, ’time’: 2.0, ’x’: 974.95, ’y’: 557.30}
{’nbr’: 3, ’time’: 3.0, ’x’: 975.65, ’y’: 556.95}
{’nbr’: 4, ’time’: 4.0, ’x’: 974.15, ’y’: 557.45}
```

The x and y pixel position values were then converted to a new coordinate system with the center of the screen as the origin (0,0). Each value was also converted from pixel values to centimeters for ’x’ and ’y’ with the following formulas.

$$x = (x_p - 960) * P_p \tag{28}$$

$$y = (y_p - 540) * P_p \tag{29}$$

where x, y is the adjusted coordinates x_p, y_p is the original x and y pixel values and $P_p = 0.00275$ is the pixel pitch value, meaning the distance in centimeters between the center of each pixel. The two values 960 and 540 comes from dividing the total resolution of the screen (1920x1080p) by half to get the center point of the new coordinate system in pixels. All of these values were obtained from the technical specifications from the screen that was used. The following JSON data structure was then obtained.

```
{’nbr’: 0, ’time’: 0.0, ’x’: 0.3973, ’y’: -0.4551}
{’nbr’: 1, ’time’: 1.0, ’x’: 0.4015, ’y’: -0.4427}
{’nbr’: 2, ’time’: 2.0, ’x’: 0.4111, ’y’: -0.4757}
{’nbr’: 3, ’time’: 3.0, ’x’: 0.4303, ’y’: -0.4661}
{’nbr’: 4, ’time’: 4.0, ’x’: 0.3891, ’y’: -0.4798}
```

Since there was no way to sync the timestamps for the EyeLink system and the test protocol displaying the stimuli as they were on different computers, the time and position of the stimuli relative to the sampling of the EyeLink data had to be adjusted manually. This was done by creating a list of stimuli positions for each timestamp by knowing the position and for how long each stimulus was presented, and then matching the stimuli positions to the sampled data of the EyeLink. The final JSON data structure was then obtained and saved. See figure 15 for flowchart describing EyeLink system recording of data.

```
{’nbr’: 0, ’time’: 0.0, ’x’: 0.3973, ’y’: -0.4551, ’dotx’:0, ’doty’:0}
{’nbr’: 1, ’time’: 1.0, ’x’: 0.4015, ’y’: -0.4427, ’dotx’:0, ’doty’:0}
{’nbr’: 2, ’time’: 2.0, ’x’: 0.4111, ’y’: -0.4757, ’dotx’:0, ’doty’:0}
{’nbr’: 3, ’time’: 3.0, ’x’: 0.4303, ’y’: -0.4661, ’dotx’:0, ’doty’:0}
{’nbr’: 4, ’time’: 4.0, ’x’: 0.3891, ’y’: -0.4798, ’dotx’:0, ’doty’:0}
```

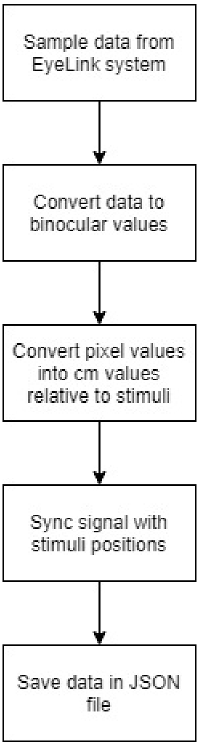

Sample data from EyeLink system
Convert data to binocular values
Convert pixel values into cm values relative to stimuli
Sync signal with stimuli positions
Save data in JSON file

Figure 15: EyeLink Data collection Flowchart.

3.5 Data analysis

3.5.1 Preprocessing and data quality

Before evaluating the collected data, it needed to be examined and preprocessed to ensure that the data quality was at an acceptable level. The preprocessing aims to reduce or remove unsatisfactory data emanating from human errors, such as the user accidentally moving the headset slightly after calibration was performed, or by repeatedly blinking or closing their eyes during a test sequence. Which in turn affects the feature detection and gaze calculations, causing errors and blink artifacts in the signal, see figure 16.

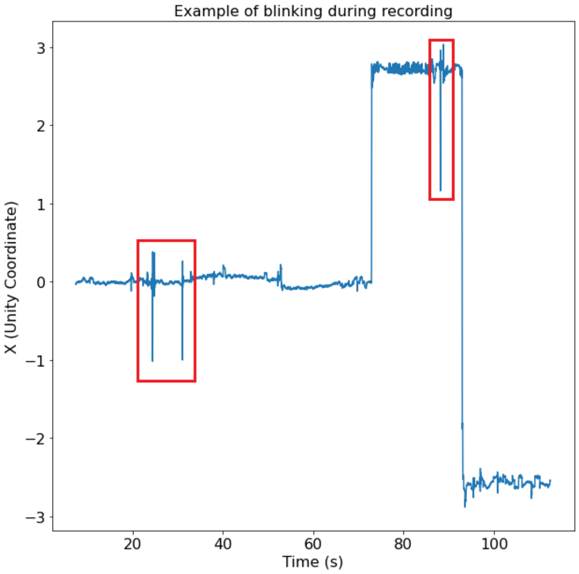


Figure 16: Example of blink artifacts (marked with red squares) in the x-component of an eye tracking signal during a sequence of fixations from the VR system.

Since all the evaluations of data and system quality were performed on the fixation part of the nystagmus test for both systems, the data needed to be preprocessed. Before further analysis, the first and last 2 seconds of data for each separate fixation were removed. This was done to ensure that the user had time to move and focus their vision between each newly presented stimuli before any data was considered for analysis.

In the next step, each data point was evaluated and removed if it was further than a fixed distance from the current stimuli position, see equation 30. This was done to remove outliers from, e.g., when the user was losing focus for a short time or from eye blinks, see figure 16.

$$\sqrt{(x-x_0)^2+(y-y_0)^2} > d \tag{30}$$

Where the coordinates x, y represents the position of where the user is looking, d is the maximum distance from the stimulus allowed for keeping the data point and x_0, y_0 are at each time the position of the stimuli. For the VR system, d = 1 (Unity coordinates), and for the EyeLink system, d = 4 (cm). The end result is that the entire fixation test was divided into five parts, one for each stimuli position during the fixation test, and most outliers were removed from the data, for visual explanation see figure 17.

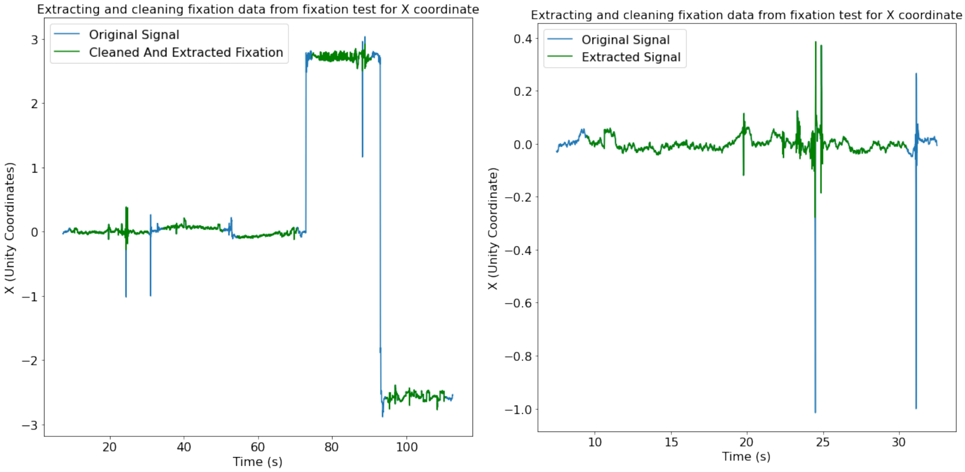


Figure 17: The left image displays one whole fixation test for the x-coordinate divided into 5 fixation parts displayed in green. The right image displays a zoomed in removal of outliers from the first fixation of the same test.

3.5.2 Calibration validation of VR headset

As mentioned earlier, one of the most important steps of a functioning eye-tracking system is that it gives accurate data that correctly represents where the user is looking. As explained in section 2.4.2, this is done by a thorough calibration before any tests are committed.

In order to evaluate the calibration of an eye tracking system, there needs to be a way to examine and evaluate the measurements it produces. This was best achieved by making a test subject with a stable eye gaze focus on a fixed point for an extended period of time. Thus, to determine the accuracy of the measurements, it is important to find circumstances when the user's gaze is stable. For this reason, and to achieve the best measurements, data during the fixation part of the test and from the control group, i.e., from test persons without nystagmus, were selected for accuracy measurements.

In order to verify that the calibration was working as promised, a way to determine the accuracy of the measurements compared to the placement of the visual stimuli, was needed. High precision was also necessary to determine that the user could control their gaze in order to determine the accuracy. A measurement with good accuracy but with bad precision is indistinguishable from a system with bad calibration or a user that is unable to control their gaze, for example a user with nystagmus. A system, however, with good precision but bad accuracy would be a telltale sign that the calibration does not work as advertised and that the calibration needs to be improved in order to give better results.

For the measurement to be as accurate as possible, the processed data was used instead of the raw data. This was done in order to minimize the chance of random noise disturbing the data. Each measured point has an offset error from the desired coordinate. The desired coordinate was placed at the same location as the visual stimulation and in order to quantify the error, the Root Mean Square Error (RMSE) was calculated for all the measured points for each of the five fixation points. The RMSE is calculated by subtracting the target value, in this case the true coordinate, y_n by the measured value $\hat{y}_n$. The result is squared to remove any negative results, and the mean of all the measured values is calculated. Lastly, the square root of the mean is calculated, and the result is the RMSE, eqn (31).

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2} \tag{31}$$

This procedure was repeated for each of the ten recordings of the control group **CG**, giving a total of 10 measures of RMSE for each of the five fixation points. To determine an acceptable RMSE for the measurements, it was noted that each focus point and visual stimuli in the Nystagmus program is a red dot with radius of 0.15, relative to the coordinate system. It is reasonable to determine that each measurement with a RMSE over 0.15 is likely to have been measured outside the desired gaze point.

In total, 50 measures each of RMSE were achieved for measurements along the x-axis and y-axis. The results are then presented relative to the actual point of reference, where it could visually determine if there are any points in the fixation where the accuracy is better or worse than average. Finally, a percentage of all the measurements that had a lower RMSE than 0.15 was calculated in order to have a basic quantified number of the accuracy of the RMSE and whether the calibration was working or not.

3.6 Eye-tracking validation across systems

After both systems had been calibrated and data was recorded with the nystagmus test protocol and then preprocessed following the steps in chapter 3.5, the data quality of each system could be evaluated and compared. Accuracy and precision in degrees of visual angle for both systems were calculated and evaluated on the fixation test in the nystagmus test protocol for the data from the control group without nystagmus.

The reasoning behind only using the data from **CG** was the same as when performing the calibration evaluation, stable fixations were needed in order to as accurately as possible measure each system's accuracy and precision.

After the fixation data was extracted from the nystagmus test, the data was further divided into 5 parts, one for each stimuli position. Extraction of the fixation data and dividing it into parts for each stimuli position were done with a script in Python which went through each data point from the recorded nystagmus test. Each data point was then evaluated and divided into parts depending on the values of its stimuli position.

Horizontal, vertical and combined accuracy and precision were then calculated as an average for all 10 recorded tests from both systems for all 5 stimuli positions combined and for each separate stimuli position respectively using equations 3 - 8. Finally, the average horizontal, vertical and combined accuracy and precision across all tests were calculated. See figure 18 below for a flowchart describing the process of calculating accuracy and precision from one nystagmus recording from **CG**.

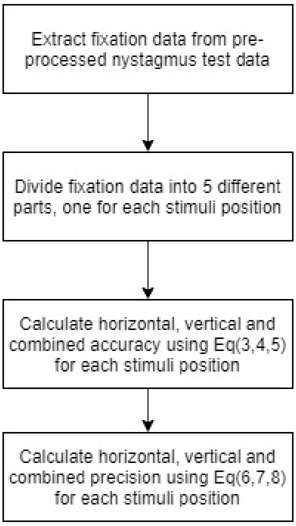

Extract fixation data from pre-processed nystagmus test data
Divide fixation data into 5 different parts, one for each stimuli position
Calculate horizontal, vertical and combined accuracy using Eq(3,4,5) for each stimuli position
Calculate horizontal, vertical and combined precision using Eq(6,7,8) for each stimuli position

Figure 18: Eye-tracking validation flowchart depicting the steps done to calculate accuracy and precision metrics for one test.

3.7 Nystagmus modeling and quality assessment of data

When performing the nystagmus modeling for both systems, only the preprocessed data from group **NG** was used. The algorithm is a method that can be used to model the nystagmus waveform from the input signal, and using this method on data which does not contain a nystagmus waveform would not yield any results as there are no nystagmus oscillations in the signal to model. For the same reasons only the horizontal fixation test data was used since group **NG** subjects nystagmus is mostly only active in the x-axis. And as aforementioned if the nystagmus modeling is performed on a signal without or with low nystagmus oscillations the resulting modeled waveform signal is zero or mostly zero which does not yield any results.

The VR data was first up-sampled to 1000 Hz by interpolating the original signal and then resampling it at 1000 Hz, so that both systems had the same sampling rate before using it as input in the waveform algorithm. The following steps were then performed for all five recordings for both systems for group **NG**.

Each fixation test was divided into five parts, one for each stimuli location, and the signal was preprocessed following the steps in section 2.5.1.

When the preprocessing was completed, each fixation had to be centered around zero since the absolute coordinate location of each stimulus differs and the resulting waveform model when estimating the original signal ignores offsets and is always centered around zero. The centering was performed by calculating the median for the original signal and then subtracting the calculated median from the signal, see figure 19 for an example of centered and non-centered signal.

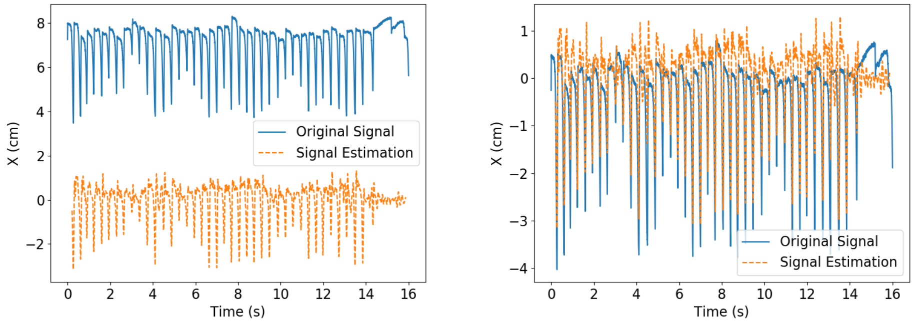


Figure 19: Before and after centering the data around zero by calculating and subtracting the median for the signal for a fixation part on the EyeLink system.

After each fixation point had been preprocessed and centered, the nystagmus waveform model was estimated using the nystagmus waveform modeling algorithm with the following steps. First, the signal was downsampled from 1000 Hz to 100 Hz to reduce model calculation times. The signal was then highpass filtered using a third order Butterworth filter with a cutoff frequency of 2 Hz, removing frequencies below the cutoff.

After the signal was downsampled and filtered it was divided into shorter segments, nystagmus oscillations frequencies are generally not stationary over longer periods of time. The signal was therefore divided into shorter overlapping segments of time and the sinusoidal harmonics were calculated for each separate segment instead of the whole signal. When deciding on the length of the segments, there is a tradeoff to consider. If the segment length is too short, it may result in poor harmonic parameter estimates, and if it is too long, the stationary assumption of each segment may not be valid. For this work, the segment length was set to 0.67 seconds, being the same used by William Rosengren [5] in his work.

When the nystagmus waveform model had been calculated it was observed that the modeled signal timestamps did not match the original signal timestamps exactly and the total length of the two signals did not match, so before performing the calculations for the normalized segment error the two signals had to be synchronized. The signals were synchronized with a script in Python going through each timestamp in both the estimated and the original signal and if both signals contained the same timestamp it was added to a new list with the corresponding values from both signals, see figure 20 below for example of unsynchronized and synchronized signals.

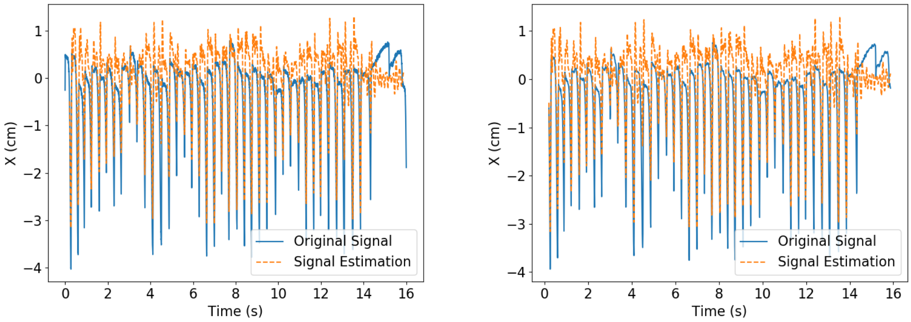


Figure 20: Before and after syncing the data around zero for central fixation point.

It was also observed that the calculations to remove the offset by calculating the median for the original signal and subtracting it from the signal worked well in some cases and not so well in other cases, depending on the structure of the signal. When the signal was evenly distributed, this approach worked well, but for more erratic signals with less symmetry it did not perform as well. To combat this, an extra step to match the two signals was implemented after syncing the signals. By taking the original signal and shifting it up or down a short distance for a number of different positions, 200 positions with an increment of ± 0.01 in this case. And then by calculating the squared average of the error between the original and the estimated signal, a more optimized offset could be calculated by minimizing the error between the two signals. This was then used to adjust the offset of the original signal to better match it to the estimated one, see figure 21 for a visual explanation.

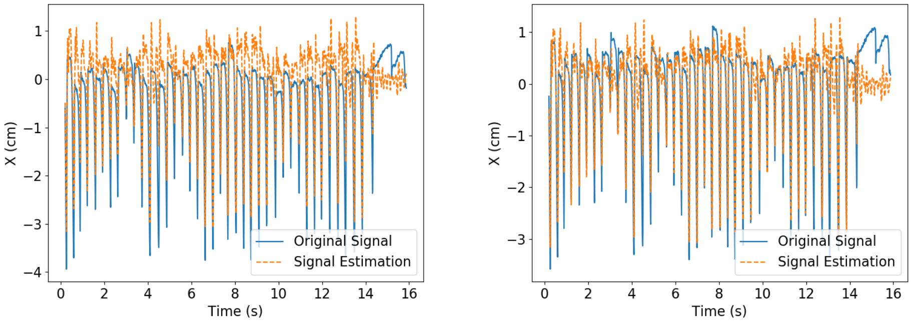


Figure 21: Before and after syncing signals by performing the optimizing offset adjustments.

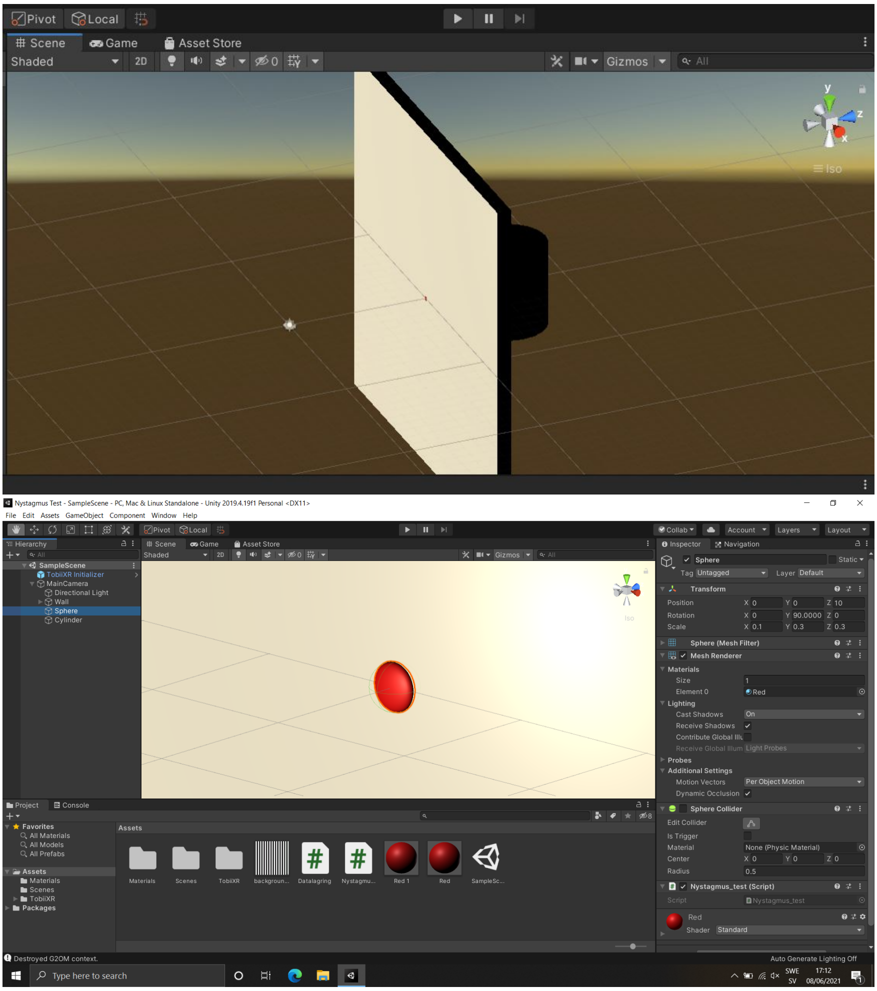
Once the signals had been synchronized and the timestamps matched, the NSE was calculated and plotted with equation 23 and a segment length of 20 samples (200ms) for each test and fixation location for both systems was used. The average amount of accepted samples was then calculated and compared for each stimuli location for all test on both systems.

4 Results

The results follow much of the same structure as described in the method above. The design and finished work of the test protocol, described in detail in section 3.1, are briefly presented in section 4.1 to give an overview of how the test protocol looked like. Likewise, in subsection 4.2.1, a selection of the raw data are presented to give insight into how the original signal looked like before any pre-processing. The remainder of section 4.2 shows much of the data captured from both the Vive and the EyeLink and the gradual steps taken to clean up the signal, as described above in section 3.5.1. Both sections 4.3 and 4.4 lay the foundation of the evaluative part discussed at length in chapter 5 and seeks to give enough insight to answer the two most pressing questions of the thesis, whether the Vive has adequate accuracy to be used as a diagnostic tool.

4.1 Test Protocol

The final version of the test protocol includes all four specified test sections: fixation, saccades, smooth pursuit and optokinetic nystagmus. Five seconds of pause were included between each segment, so the user could rest their eyes between each segment. Green color of the stimuli indicates a pause in the system. The test was built in Unity using a sphere, a large cube stretched into a wall and a cylinder for performing the OKN test. Figure 22 shows the test components as displayed in the Unity build platform and figure 23 shows a scene from test section 1-3 and figure 24 a scene with horizontal and vertical OKN.



Sphere Static
Tag Untagged Layer Default

Transform

	X	Y	Z
Position	0	0	10
Rotation	0	90.0000	0
Scale	0.1	0.3	0.3

Sphere (Mesh Filter)

Mesh Renderer

Materials

Size 1

Element 0 Red

Lighting

Cast Shadows On

Receive Shadows ✓

Contribute Global Illu

Receive Global Illum Light Probes

Probes

Additional Settings

Motion Vectors Per Object Motion

Dynamic Occlusion ✓

Sphere Collider

Edit Collider

Is Trigger

Material None (Physic Material)

Center X 0 Y 0 Z 0

Radius 0.5

Nystagmus_test (Script)

Script Nystagmus_test

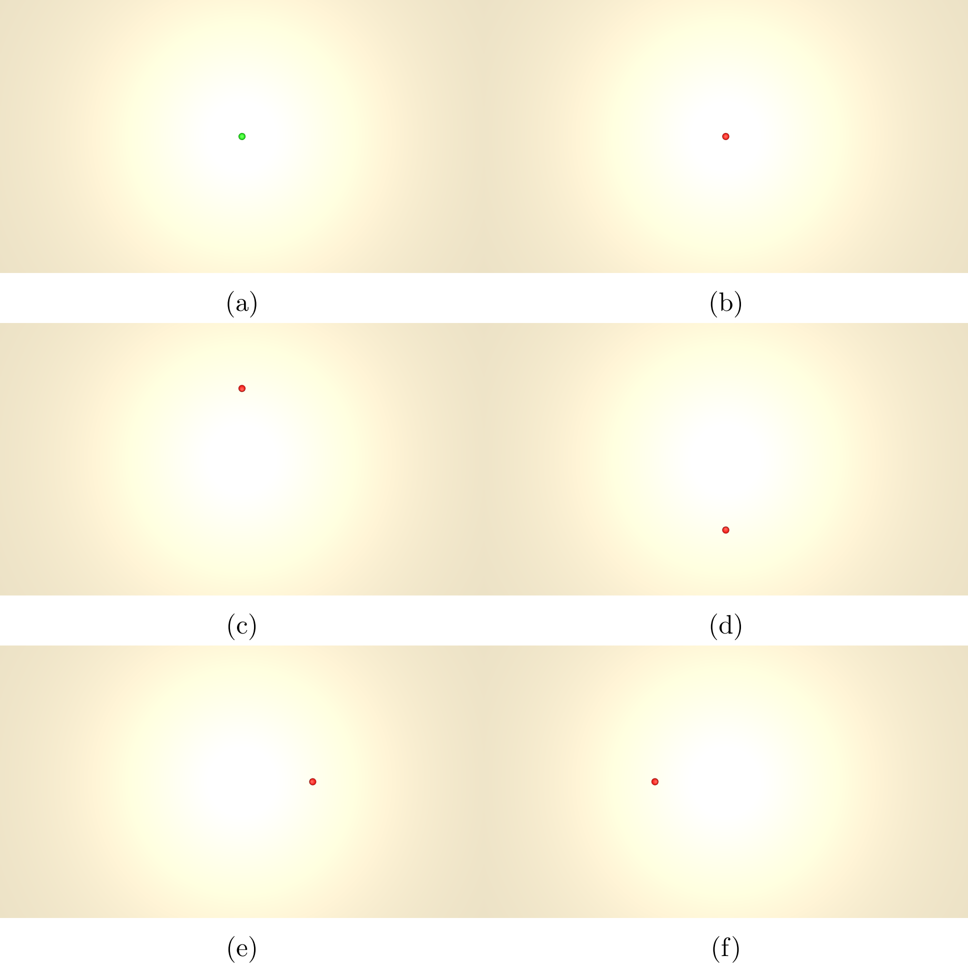
Red

Shader Standard

Destroyed G2OM context.

Auto Generate Lighting Off

Figure 22: Example of the test protocol components and the interface of the Unity real-time development platform. The upper picture shows all components that comprise the testing protocol; the sphere, the wall and the cylinder. The lower picture presents the entire interface of the Unity development platform with a close up of the visual stimuli.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 23: Pictures of the test protocol in action. From left to right, the picture a displays the stimuli with the color green, indicating pause in the test. The test started by being positioned in the middle of the screen and continued by placing the stimuli in the following order: up, down, right, left. The positions of the stimuli were the same for the fixations and saccades. For vertical smooth pursuit, the stimuli moved between the upper and lower position, shown in image c and d. For horizontal smooth pursuit, the stimuli moved between the left and right position, shown in image e and f.

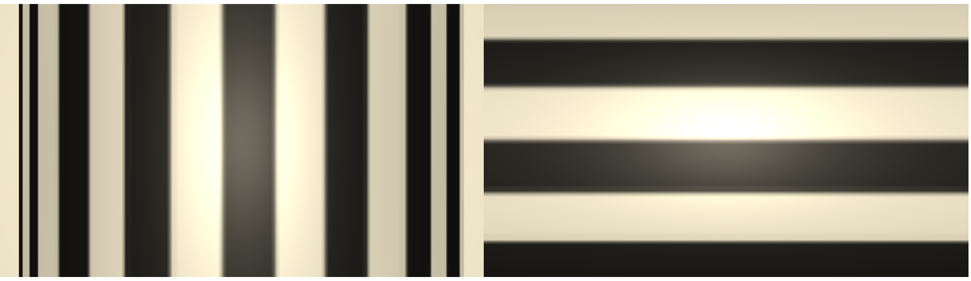


Figure 24: Screen caption of the horizontal and vertical example of the optokinetic nystagmus.

4.2 Recorded Data

Presented below in the following graphs are the collected data from both the VR system and the EyeLink 1000 system. The results are both collected from the raw data and from the filtered results with minimized outliers. Out of size consideration, only one recording from **CG** and **NG** for each system was chosen for presentation of the test results in its entirety.

4.2.1 Tobii HTC Vive Devkit

Raw data - Control group

Presented below in figures 25 and 26 is the raw data signal from **CG** for the x and y coordinates over time from the VR system, where each separate plot contains test data from a test in the VR test protocol. As can be seen in the different plots, the data contains different disturbances, such as several instances of involuntary blinking and small movements of the user etc. This data is in need of some filtering in order to get a signal that is easier to analyze and interpret.

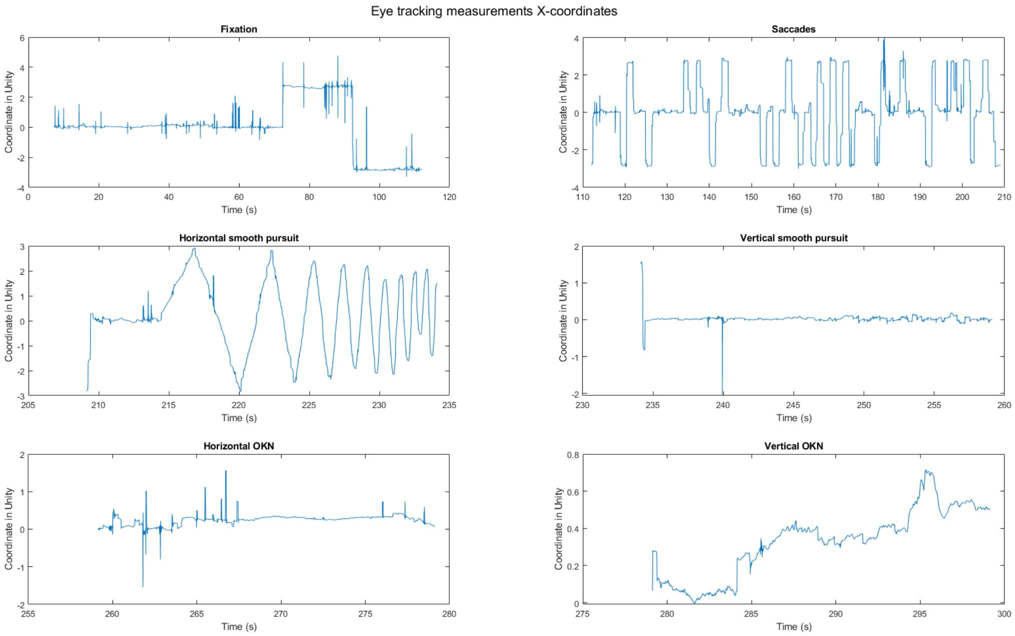


Figure 25: Measurements of gaze position in x-coordinates for different sections of the VR nystagmus test protocol. Test 9 for **CG**, recorded on the morning of 23rd April on the VR system.

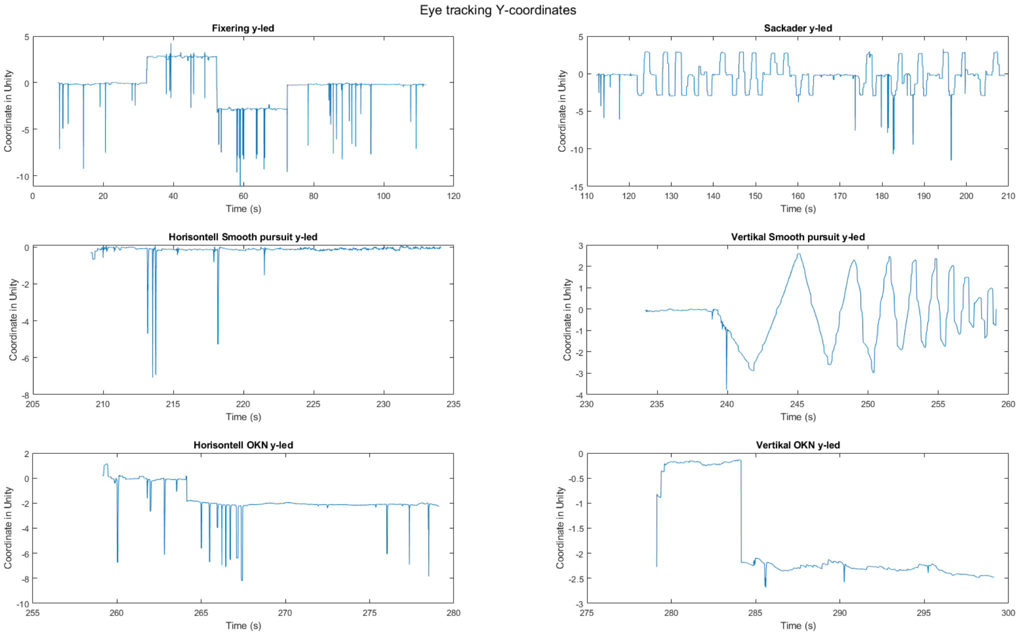


Figure 26: Measurements of gaze position in y-coordinates for different sections of the VR nystagmus test protocol. Test 9 for **CG**, recorded on the morning of 23rd April on the VR system.

Presented below in figure 27 is the unfiltered scatter plot for the fixation test in the VR test protocol for **CG**, observe the unfiltered points far from the visual stimuli which is due to different disturbances, such as instances of involuntary blinking and small movements of the headset or user.

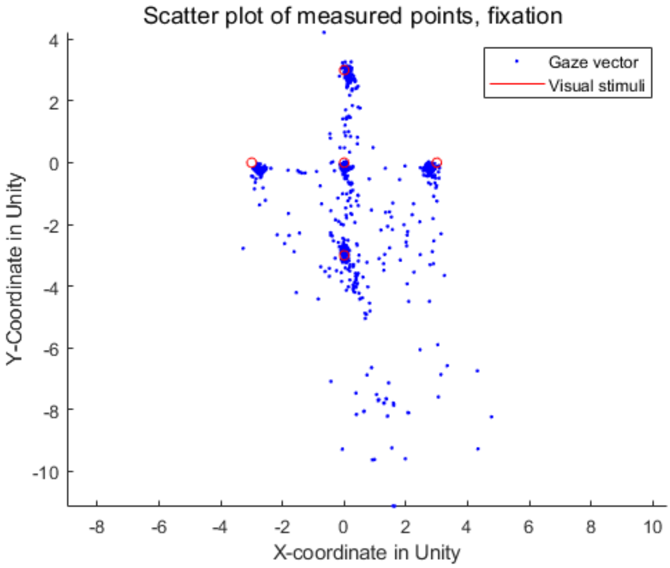


Figure 27: Scatter plot of gaze points in a 2D coordinate system, for the fixation part. The blue dots are captured points, and the red rings are the circumference of the visual stimuli. Test 9 for **CG**, recorded on the morning of 23rd April on the VR system.

Raw data - Nystagmus group

Presented below in figures 28 and 29 is the raw data signal from **NG** of the x and y coordinates over time from the VR system, where each separate plot contains test data from the different tests in the VR test protocol. Here the nystagmus oscillations can clearly be seen in each test for the x-coordinates in figure 28, and then to a much lower degree for the y-coordinates in figure 29. The lower degree of nystagmus oscillations in the horizontal plane compared to the vertical plane is due to **NG** nystagmus waveform characteristics, which is mostly active in the horizontal plane.

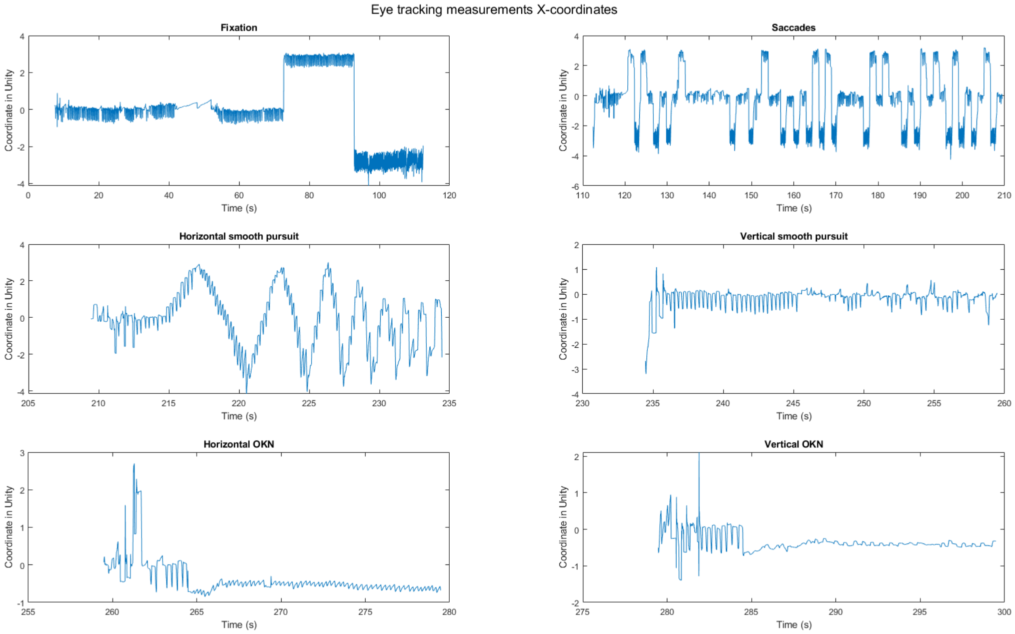


Figure 28: Measurements of gaze position in x-coordinates for different sections of the nystagmus test protocol. Test 8 for **NG**, recorded in the afternoon of 22nd April on the VR system.

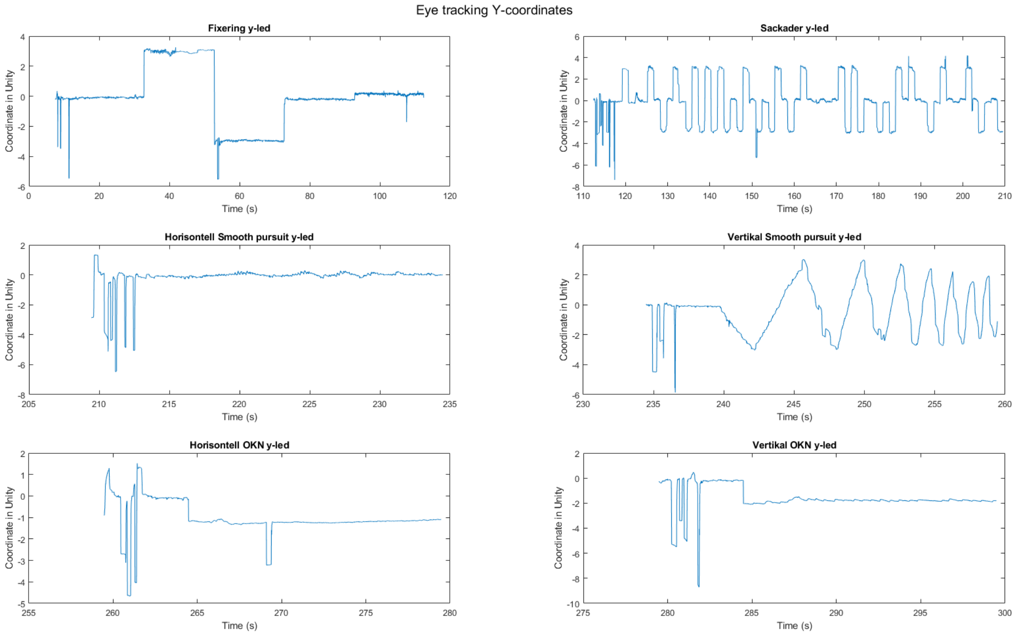


Figure 29: Measurements of gaze position in y-coordinates for different sections of the nystagmus test protocol. Test 8 for **NG**, recorded in the afternoon of 22nd April on the VR system.

Presented below in figure 30 is the unfiltered scatter plot for the fixation part in the VR test protocol for **NG**, here the nystagmus oscillations can clearly be seen in the horizontal spread of the data for each fixation point.

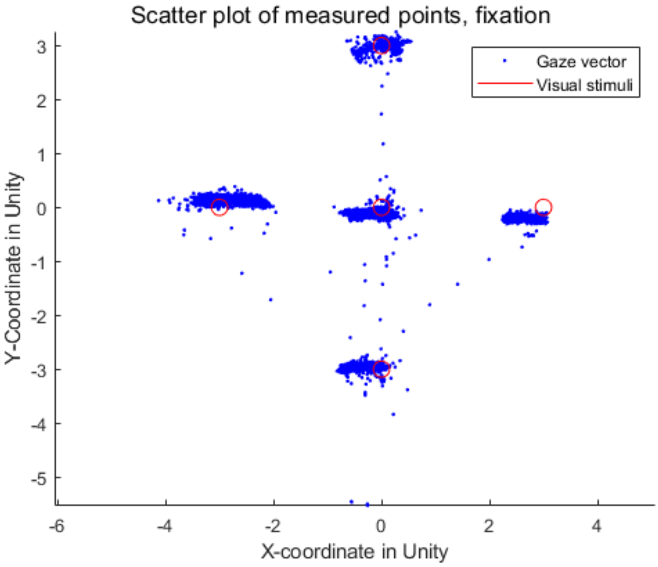


Figure 30: Scatter plot of gaze points in a 2D coordinate system, for the fixation part. The blue dots are captured points, and the red rings are the circumference of the visual stimuli. Test 8 for **NG**, recorded in the afternoon of 22nd April on the VR system.

Filtered fixation signal - Control Group

Presented below, in figure 31, is the filtered data signal from **CG** as the x coordinate over time and as a scatter plot of the data from the fixation part of the test. The data has now been filtered to remove as many disturbances and outliers as possible from the signal to allow for better processing.

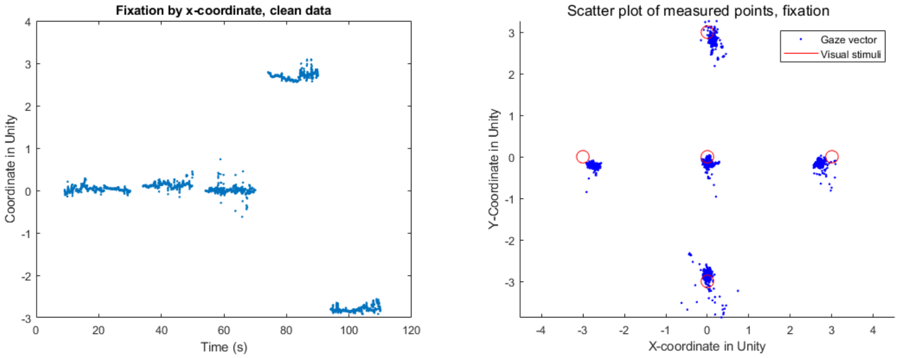


Figure 31: X-coordinates over time and scatter plot of gaze points in a 2D coordinate system, for the fixation part. Blue dots are captured points and red ring are the circumference of the visual stimuli. Test 9 for **CG**, recorded on the morning of 23rd April on the VR system, with recorded data on the left and scatter plot on the right.

Filtered fixation signal - Nystagmus Group

Presented below in figure 32 is the filtered data signal from **NG** as the x coordinate over time and as a scatter plot of the data from the fixation part of the test. Notice here that the filtering has retained the nystagmus oscillations of the signal even after being processed.

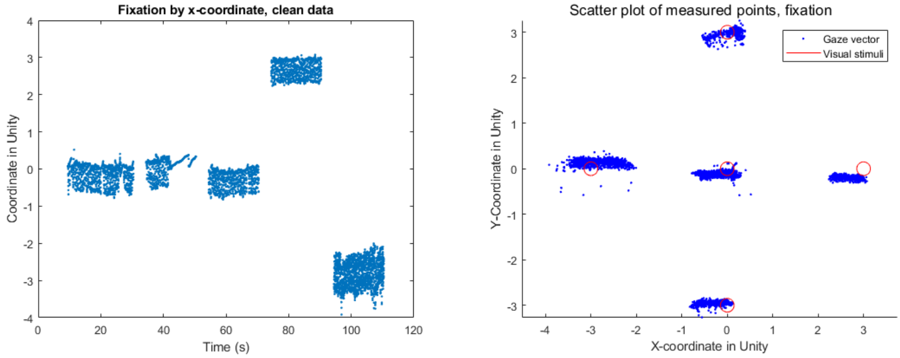


Figure 32: X-coordinates over time and scatter plot of gaze points in a 2D coordinate system, for the fixation part. Blue dots are captured points and red ring are the circumference of the visual stimuli. Test 8 for **NG**, recorded in the afternoon of 22nd April on the VR system with x-coordinates on the left and scatter plot on the right.

4.2.2 EyeLink 1000

Presented below in figure 33 and 34 is the filtered x-coordinates over time and scatter plot of the fixation test for the EyeLink system for both **CG** and **NG**. Notice here the lower amount of spread around the stimuli compared to the VR system, especially for data collected on **CG**. This data conveys the superior accuracy and precision of the EyeLink system as compared to the VR system.

Filtered fixation signal - Control Group

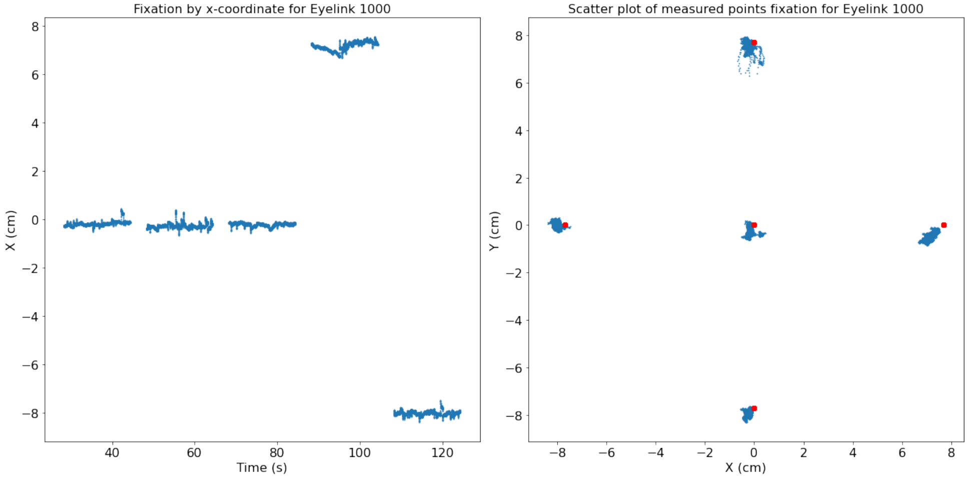


Figure 33: X-coordinates over time and scatter plot of gaze points in a 2D coordinate system, for the fixation part. Blue dots are captured points and red dots are the visual stimuli from the EyeLink system for **CG**.

Filtered fixation signal - Nystagmus Group

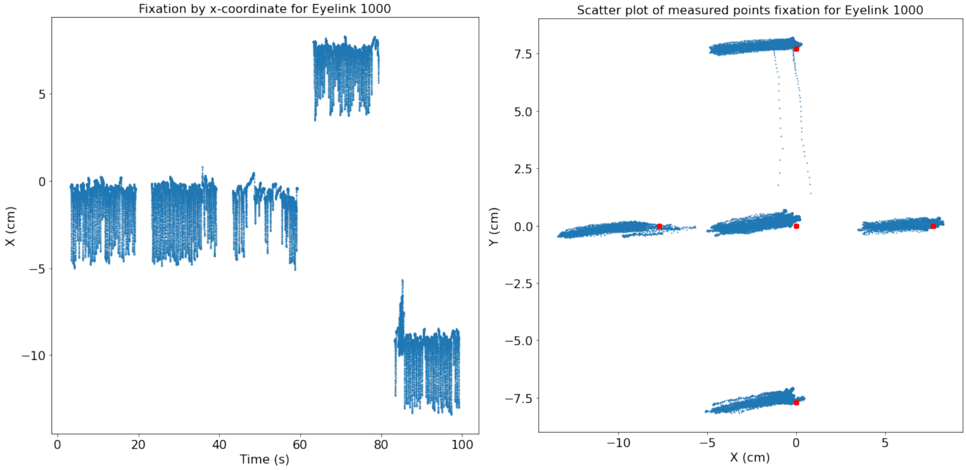


Figure 34: X-coordinates over time and scatter plot of gaze points in a 2D coordinate system, for the fixation part. Blue dots are captured points and red dots are the visual stimuli from the EyeLink system for **NG**.

4.3 Data Analysis

4.3.1 Validation test scores of the Tobii HTC Vive Devkit

Shown below in figure 35 and figure 36, are the filtered fixation signals with high RMSE, i.e., an example of a bad result, and a fixation signal with low RMSE, an example of a good result from recordings of **CG**. When calculating the RMSE of all fixation measurements from **CG**, 62 % of all gaze positions in the x-coordinate are within the visual stimuli and 56 % of the y-coordinates are within the visual stimuli.

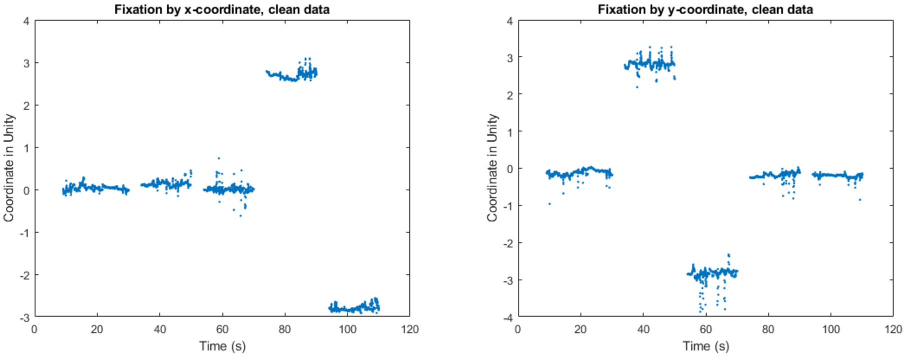


Figure 35: Fixation from test 9, captured on morning 23rd April of **CG** with high RMSE.

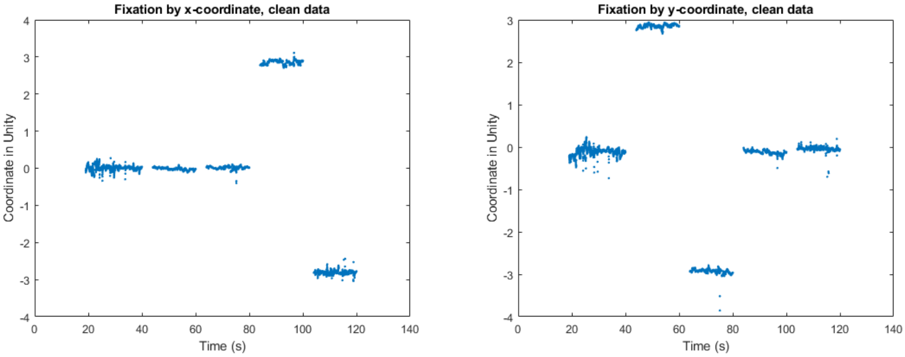


Figure 36: Fixation from test 8, captured on afternoon 22nd April of **CG** with low RMSE.

Presented below in table 1 and 2 are the calculated RMSE of the x-coordinates and y-coordinates for each position in the 10 fixation measurements, along with mean and median RMSE of every fixation position. It gives a quantitative measure of how close the measured data comes to a "perfect" score, which is 0. A perfect score of zero would mean that the measured eye tracker signal is exact on the visual stimuli in the test protocol, and the larger number means a greater drift from where the visual stimuli were located and where the eye tracker measured the gaze. This is visually presented in figure 37 where the average RMSE of all fixation measurements are visualized against the size of the visual stimuli of the test protocol, giving a visual measure whether the measured gaze vector was on average measured inside the visual stimuli or not.

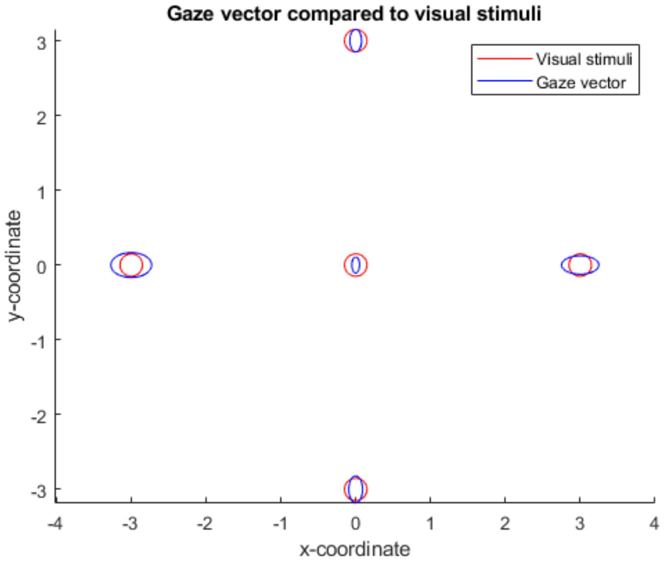


Figure 37: Comparison between the size of the visual stimuli and the average measured gaze vector from the different fixations from **CG**. The blue ellipses symbolize where the gaze vector was measured on average for all five fixation periods, and the red circles are the circumference of the visual stimuli as portrayed in the test protocol.

Test (x)	Center	Up	Down	Right	Left
1	0.0302	0.0755	0.0606	0.2997	0.4396
2	0.0167	0.02847	0.0966	0.2511	0.3211
3	0.0450	0.0530	0.0651	0.2189	0.2957
4	0.0298	0.0475	0.1212	0.4211	0.2474
5	0.0951	0.1743	0.0834	0.2322	0.0688
6	0.0639	0.1172	0.0515	0.2961	0.1229
7	0.0776	0.0391	0.2126	0.0475	0.5195
8	0.0528	0.0357	0.0398	0.1550	0.2000
9	0.0660	0.1459	0.0758	0.3228	0.2082
10	0.0401	0.0650	0.1131	0.2459	0.3003
Mean	0.0518	0.0782	0.0920	0.2491	0.2724
Median	0.0490	0.0591	0.0797	0.2486	0.2716

Table 1: Root-mean-square error in the x-coordinate of five fixated positions from the nystagmus test protocol from **CG**, performed with the Tobii HTC Vive Devkit.

Test (y)	Center	Up	Down	Right	Left
1	0.0756	0.0443	0.1816	0.1373	0.0445
2	0.0717	0.1085	0.1482	0.0623	0.2474
3	0.0702	0.1481	0.2349	0.0746	0.0683
4	0.0759	0.1305	0.2059	0.1967	0.1041
5	0.1967	0.1043	0.1663	0.0825	0.1635
6	0.0933	0.1724	0.1506	0.1150	0.1763
7	0.0856	0.1654	0.1882	0.0912	0.3703
8	0.1284	0.1518	0.0886	0.1288	0.0581
9	0.1307	0.2415	0.2091	0.1953	0.2062
10	0.1423	0.2439	0.2256	0.1441	0.2220
Mean	0.1071	0.1511	0.1799	0.1228	0.1661
Median	0.0895	0.1500	0.1850	0.1219	0.1619

Table 2: Root-mean-square error in the y-coordinate of five fixated positions from the nystagmus test protocol from **CG**, performed with the Tobii HTC Vive Devkit.

4.4 Accuracy and Precision

Presented below in tables 3 and 4 are the calculated accuracy and precision values for each test and the average value for all tests from both systems. In table 5 and 6 the calculated average precision and accuracy values for each stimuli position over all tests for **CG** are presented. Accuracy measures on average how far away each gaze point is from the presented stimuli, while precision measures the average distance between each gaze point. A perfect system with a user that could perfectly control their gaze would get a perfect score of 0 on all tests. When comparing the average values of both systems, the EyeLink system outperforms the VR system in every metric and stimuli position. Worth noting in Table 5 is the spread in combined average accuracy for the different stimuli positions for the VR system, with the center position scoring much better on average than the other stimuli positions, especially the right and left stimuli positions.

```
HA = Horizontal Accuracy
VA = Vertical Accuracy
CA = Combined Accuracy
HP = Horizontal Precision
VP = Vertical Precision
CP = Combined Precision
```

Test	HA	VA	CA	HP	VP	CP
1	0.7513	0.6548	1.0951	0.0681	0.1239	0.1413
2	0.9507	0.4741	1.1948	0.1275	0.166	0.2093
3	0.8919	0.6757	1.2286	0.1814	0.321	0.3686
4	0.6677	0.5744	1.0517	0.1869	0.2711	0.3292
5	0.6357	0.7013	1.0601	0.2036	0.2689	0.3372
6	0.671	0.6682	1.0809	0.1636	0.2705	0.316
7	0.4989	0.5989	0.8861	0.1308	0.1697	0.2142
8	0.8869	0.9041	1.3407	0.2253	0.2026	0.3029
9	0.7702	0.9916	1.3673	0.2108	0.3009	0.3672
10	0.8136	1.0186	1.3915	0.1842	0.282	0.3367
Average	0.7538	0.7262	1.1697	0.1682	0.2377	0.2923

Table 3: Horizontal, vertical and combined accuracy and precision for each test for the Tobii HTC Vive Devkit for **CG**

Test	HA	VA	CA	HP	VP	CP
1	0.1735	0.3127	0.396	0.0216	0.0227	0.0313
2	0.362	0.2608	0.4738	0.0209	0.0227	0.0309
3	0.1813	0.2376	0.3325	0.0208	0.0225	0.0306
4	0.1379	0.1848	0.2559	0.0207	0.0226	0.0306
5	0.119	0.2799	0.3254	0.0207	0.0222	0.0304
6	0.1025	0.1227	0.1771	0.0209	0.0217	0.0301
7	0.1266	0.2689	0.3116	0.0208	0.0219	0.0302
8	0.293	0.2421	0.3993	0.0199	0.0209	0.0289
9	0.1276	0.1506	0.2162	0.0202	0.021	0.0292
10	0.1531	0.186	0.2696	0.0199	0.0213	0.0291
Average	0.1776	0.2246	0.3157	0.0207	0.0219	0.0301

Table 4: Horizontal, vertical and combined accuracy and precision for the EyeLink 1000 for **CG**

Stimuli position	HA	VA	CA	HP	VP	CP
Center	0.2534	0.4808	0.5938	0.1328	0.2278	0.2653
Up	0.4077	0.8035	0.9666	0.081	0.1448	0.1668
Down	0.4382	0.956	1.0998	0.1871	0.2692	0.3304
Right	1.3826	0.6132	1.5722	0.1384	0.1755	0.2266
Left	1.4837	0.8808	1.8445	0.2073	0.2442	0.323

Table 5: Average horizontal, vertical and combined accuracy and precision for each stimuli position Tobii HTC Vive Devkit for **CG**

Stimuli position	HA	VA	CA	HP	VP	CP
Center	0.0962	0.2293	0.2617	0.0202	0.023	0.0306
Up	0.2008	0.2096	0.3335	0.0193	0.0167	0.0255
Down	0.1137	0.1994	0.2492	0.0211	0.0247	0.0325
Right	0.283	0.264	0.4073	0.0192	0.0222	0.0293
Left	0.195	0.2216	0.3279	0.0233	0.0222	0.0322

Table 6: Horizontal, vertical and combined accuracy and precision for each stimuli position for the EyeLink 1000 for **CG**

4.5 Nystagmus modeling and quality assessment of data

Presented below are results from the nystagmus modeling and quality assessment of data from **NG** for both systems. Only a few graphs are shown for illustrative purposes.

Tobii HTC Vive Devkit

Presented in figure 38, 39 are two examples of a good and a bad nystagmus model estimations with high and low average amount of accepted NSE segments from data collected during the fixation part of the test protocol. In Table 7 the average amount of accepted NSE segments for each test and stimuli position are presented.

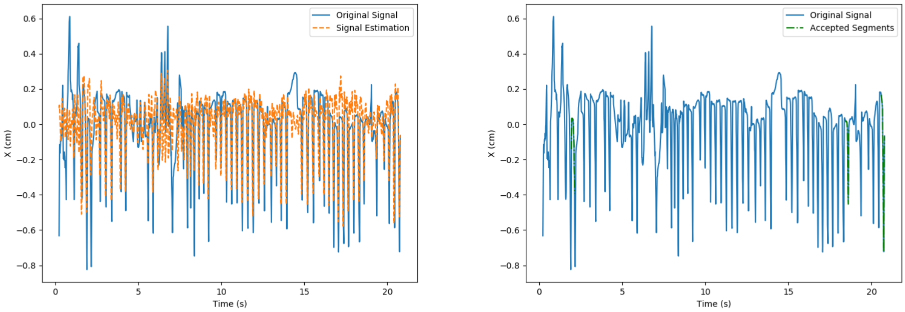


Figure 38: Example of a bad nystagmus model estimation from fixation data for the VR system.

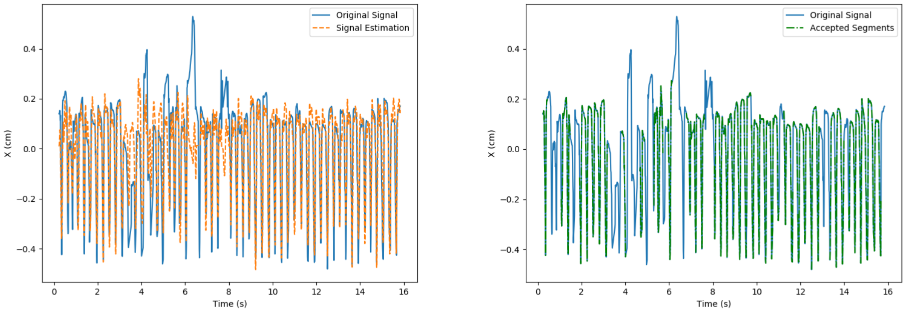


Figure 39: Example of a good nystagmus model estimation from fixation data for the VR system.

Stimuli position	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Center	0.3535	0.0312	0.17	0.3235	0.4412	0.2639
Up	0.0	0.026	0.0274	0.0137	0.0274	0.0189
Down	0.0133	0.013	0.026	0.0822	0.04	0.0349
Right	0.0	0.6757	0.0519	0.0779	0.16	0.1931
Left	0.4638	0.7703	0.8406	0.6757	0.8919	0.7284
						0.2478

Table 7: Average proportional amount of accepted NSE segments, with a score of 1 being equal to all segments accepted for each test and stimuli position for the VR system

EyeLink 1000

Presented in figure 40, 41 are two examples of a good and a bad nystagmus model estimations with high and low average amount of accepted NSE segments. In Table 7 the average amount of accepted NSE segments for each test and stimuli position are presented.

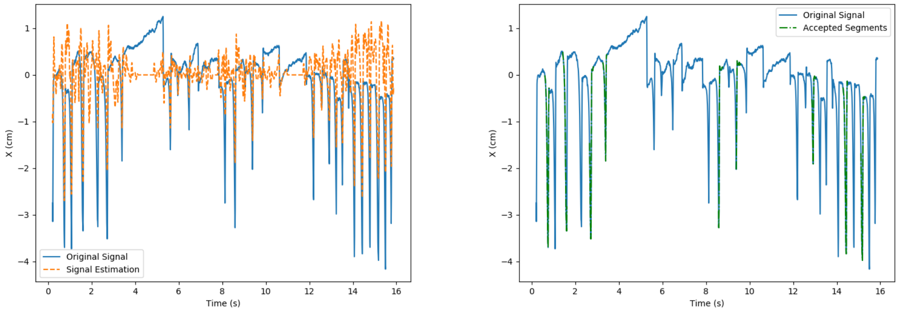


Figure 40: Example of a bad nystagmus model estimation from fixation data for the EyeLink system.

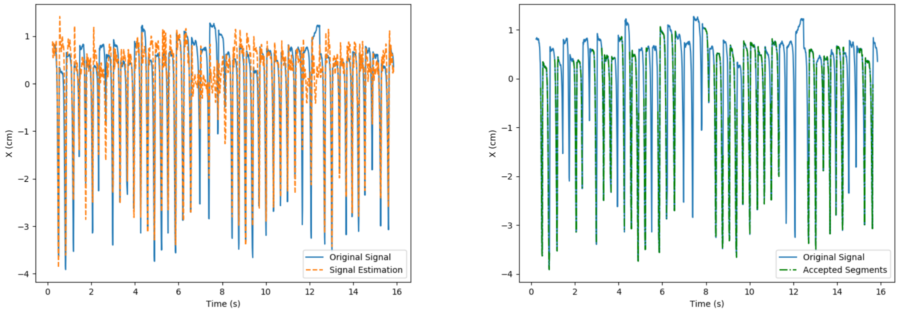


Figure 41: Example of a good nystagmus model estimation from fixation data for the EyeLink system.

Stimuli position	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Center	0.5068	0.4933	0.5067	0.0154	0.2133	0.3471
Up	0.3836	0.5333	0.4533	0.1081	0.2206	0.3398
Down	0.1169	0.0	0.2727	0.3649	0.0667	0.1642
Right	0.5068	0.0779	0.1194	0.0923	0.2208	0.2035
Left	0.4545	0.5067	0.4026	0.3867	0.6	0.4701
						0.3049

Table 8: Average proportional amount of accepted NSE segments, with a score of 1 being equal to all segments accepted for each test and stimuli position for the EyeLink system

5 Discussion

To recap things, we set out to create a new kind of testing protocol, based on earlier work, most notably done by William Rosengren in his doctoral thesis *Characterisation of nystagmus waveforms in eye–tracker signals.* It was desired to see if it was possible to "shrink down" the whole test procedure and make it fit into a more affordable and most notably, more portable device like a VR-headset. To evaluate this procedure, four principal questions were posed, one concerning the development of the program, one regarding the recording of test subjects and two of the evaluation of the actual test results.

Looking back at the actual method and the results, most of the questions were met with satisfaction. A satisfactory translation of the original test designed for the EyeLink were made for VR in mind and the recording of test subjects were accomplished with satisfying results, and is discussed at length in section 5.1 and 5.2 respectively. For the actual answer whether the Vive has the desirable results to be a viable method for testing clinical eye movements, the results in 4.4-4.5 are evaluated in particular and the findings are presented in 5.4-5.6.

5.1 Design and construction of test protocol

From the outset, the central goal of this project was to recreate the previously used nystagmus test, mostly used by William Rosengren. This was to be done in Unity as it is readily-made software, easy to convert into a VR environment and the only engine capable of implementing with eye tracking. Unity worked great as a base platform to learn about game development and creating virtual environments and while being both accessible and easy to use, learning a new programming language and a new kind of program is always challenging. Just learning the interface proved to be a challenge, as it is quite complex. Unity is heralded for its ease of use and its ability to create and render objects on the fly, while also giving the opportunity to try out possible design and solutions immediately without having to write any code or to render in advance. It is not without its faults however, and a proper game engine still has to be advanced enough to create large projects and with that comes large and cluttered menus which takes time to learn.

5.1.1 Learning process

The first part of the process was to learn and get comfortable with the game engine Unity and its development kit. Unity is a widely available and popular game engine with a big community and a wide library of interactive courses to get started with developing and creating game worlds in Unity. During this process of learning, a lot of different design ideas for the nystagmus test were also tested and created, both as a learning tool but also as a quick test protocol to check if they were viable solutions. Creating with Unity is usually set up in two different parts, learning to create worlds using the building tool, i.e. the Unity real-time 3D development platform, and deciding what is supposed to happen in the scene with the help of scripts and code. In the building tool, the developer can create and place the desired objects into the world of the game or program. These objects can be manipulated in terms of size, orientation and appearance. The second part of learning and creation is to create and code scripts that interact with the world that has been created. The scripts are written in C# and the programming language as such is pretty rudimentary but still possesses a learning curve in order to create the scenes that are desired. For the sake of learning, a lot of very basic geometrical shapes and structures were created, like platforms, walls, and singular shapes like spheres, cubes and cylinders. These could then be manipulated using scripts to be moved at different speeds, they could interchange position, change color etc. C# is pretty similar to other object-oriented programming languages, but with every unfamiliar programming language comes the hurdle of learning the syntax, which took a couple of days to figure out and get comfortable with.

The learning process at large was quite straight forward because of the excellent courses provided by Unity from their website and community. It still took a couple of days of fooling around in the menus to get comfortable with the software, and then some extra days to get comfortable with C# as a method for controlling the objects in Unity. Much was learned from the community, at least when it came to scripting the actual events that decided the positions of the test.

5.1.2 Design process

The main goal was to recreate a similar test used in previous studies, but with the additional tools and possibilities given by Unity and its 3D world. Together with the inspiration from tests done by Clay et. al [3], experimentation with different approaches to create the test protocol was performed. It was tempting to create a fully realized 3D world, to fully take advantage of the possibility to investigate how nystagmus works in a real life scenario. This was however quickly scrapped because of the amount of work that was needed to just build a small environment that was believable, easily navigable and included inconspicuous visual stimuli that the test person was able to track with their eyes. The wish to take advantage of the possibilities of a 3D world in virtual reality still remained and while learning more about scripting and creating moving objects in Unity, very early types of possible nystagmus test protocol were created as well. In an early attempt to learn continuous movement of objects and scripted movement of the camera, a test protocol was created with a sphere moving in a vertical circular movement with the player camera circled around the sphere. This created the effect of a more vivid smooth pursuit and was considered for further development. While promising, it was unsure how the addition of a third vector, measuring the depth of where the test person was looking, would impact the final recorded result from the eye tracker. It would also be hard to do equal comparisons when doing recordings on the stationary EyeLink since the experience would differ greatly between experience of the protocol in VR as compared to playback from a 2D screen, as were the case for the EyeLink.

In the end, with several options for design of the test, it was decided to mimic the previous work done by William Rosengren, as it was the simplest and most comparable design. Uninspired perhaps, but as the thesis progressed and the research questions materialized, it became more apparent that a test protocol that could be comparable across systems and easily applied to the evaluation algorithms already developed, was a necessity. The protocol was implemented as close to the original as possible within the confines of Unity, where the issues of implementation mostly came from creating a 2D workspace in a 3D world. Creating the first three sections were no big problem since it only needed a background and some visual stimuli, but the OKN needed some extra innovation. It was not possible to display the OKN pattern on the same "wall object" as the previous sections, but a different object was need to obscure the other test objects and display the OKN-pattern. This was achieved by putting the striped pattern on a cylinder that rotated, thus creating the desired OKN effect.

When running the test protocol in VR, the entire viewpoint of the test subject was obscured by the test and the visual stimuli was located at the edges of the visual field, but still easy to focus on. Over the course of the different tests sessions, test subjects noted that the visual stimuli drifted to slightly different positions. Nothing was changed in Unity and could not be traced in the results either. While it is possible that these anomalies are just imagination, the more likely explanation is that the positions of the visual stimuli changed ever so slightly because of the changes in the position of the VR headset or changes in the calibration of the entire VR headset. This might have changed the position of the camera in Unity relative to the objects active in the test protocol.

Changing the aspect from VR to the EyeLink system, where the stimuli was displayed on a 2D screen, changed some aspects of the test. First and foremost, the change of perspective from VR to screen also changed where the visual stimuli was located. When running the test protocol on the EyeLink the visual stimuli was more centered than they appear in VR. It was discussed to change the coordinates for the stimuli but even with the more centered stimuli it still gave the desired effect and comparable results for both **CG** and **NG**. The only noticeable difference was for **NG** where the more centered stimuli did not trigger the more erratic form of nystagmus. Another difference between the two methods of measurement became the coordinates of the two systems. The VR-system has the advantage of being an integrated system, and the results was all in the Unity coordinate system which made it very simple to check for validation, accuracy and precision. The EyeLink however gave results in pixels, making the process of achieving results a bit more time-consuming and arduous.

5.2 Test procedure and recording of data

Recording data for evaluating the two eye tracking systems was one of the central objectives of this thesis. Unfortunately, because of the COVID-19 pandemic, extra precautions were necessary. Work and planned recordings could be abrupted at just a day's notice. Suffice to say, the recordings did not go as smoothly as was initially hoped for.

The recording of VR data was the initial focus and played a two-fold role, both in regard to actually collecting data from a VR-headset doing a test protocol for the detection of nystagmus, but also as a starting point of where the thesis needed to be focused further. Since the VR-system is the untested one, it was motivated to do extra testing sessions, so the results were both plenty and separated in time. All in all, 10 different recordings were taken at 5 different dates, which gave plenty of material to work with. Concerning the actual testing procedure, the procedure done for VR was the easiest one, because it could be done without any outside help and could be accomplished while simultaneously continuing the writing and research of the thesis. There were some concerns regarding how much of an impact the lack of an integrated calibration would have on the end result, but it was decided to turn into a research question instead because nothing in particular could be done about the lack of integrated calibration.

When the recording of the VR data had been wrapped up, the initial steps of evaluation and data processing began. More importantly, the next step of the recording process would be decided. While one of the thesis main purposes was the evaluation and comparison of VR eye tracking and stationary eye tracking, it was not decided until rather late how this comparison would be actually implemented.

Unlike the tests in VR, the test procedure for the EyeLink was not as straightforward. Because of the different setup structure, a couple of test recordings of the system were needed before any more commitment to a particular evaluation method would be decided. This also had the implication of the need for additional help from the supervisor because of the added complexity of the EyeLink system and that specific software was needed for both the calibration and the recording. The recording of the EyeLink could therefore not be accomplished as easily or as regularly as for VR.

When the first test results were acquired, it contained strange anomalies, with good results along the x-axis but a shift along the y-axis. Since the calibration is much more thorough and also gives a quantitative validation, it was extra confusing to not receive good data for the system that was supposed to be the benchmark. After consulting with the supervisors, it was detected that the calibration and the test protocol had differently lit backgrounds, the calibration had a gray background while the test protocol was bright white. This difference in light created a slight difference in pupil size of the test participant, which in turn affected the final result. The background for the calibration was changed to a brighter color and the difference disappeared, and we received accurate results again.

For evaluation purposes, the participant of the control group did 10 separate recordings of fixation sessions in order to compare to the results of the 10 fixations from the VR recordings. This was needed to properly compare the precision and accuracy of the eye trackers performance in both the EyeLink and VR headset. In addition to these measurements, there was also the possibility to record additional measurements from two individuals that were assigned to the control group, giving more “normal” data to be compared across the systems.

Lastly were the issue of gathering recordings of the nystagmus group. Because of the COVID-19 pandemic, there was no possibility to have a wide selection of candidates or do widespread testing, either with candidates with nystagmus or those without. With at least one participant diagnosed with nystagmus it became possible to do the recordings and answer the research questions, however with a somewhat lack of diversity in the results. When doing the actual recordings the VR headset proved no difficulty, the calibration worked as intended and showed no issues. The EyeLink proved to be something else entirely. Because of the greater precision and more thorough calibration, the automatic calibration could not initialize because the participant was unable to focus steadily on the calibration point, which incidentally is the very symptom of nystagmus. Because of this, the calibration had to be done manually by recording sections of measurements for each nine normal calibration points and then manually assigning values to the recorded data. Recording the data was therefore no issue at all, but actually knowing where the test subject was looking became a whole different story. Fortunately, in the end, the nystagmus recordings of the EyeLink showed that it was not necessary to know where the test subject was looking, and the important data instead was the shape of the eye tracking signal and whether it conformed to what a nystagmus signal should look like.

Recording the data was the most time-consuming element of the thesis. Not because it was difficult, but mostly because the surrounding circumstances of doing something that consistently relies on either other participants, equipment or premises during an ongoing a pandemic. It became a frequent, unreliable and frustrating problem, and at least four times we had to postpone important recordings with just a day's notice because of suspected exposure to the Coronavirus. This halted the progress of the work done for the thesis and became a rather frustrating element. With frequent stops, especially at the end of writing this thesis, made the work rather slow and sluggish. Secondly, the pandemic prevented us from doing a bigger experiment study than was carried out. One of the original research goals was to record eye movements from several individuals, both with and without nystagmus, in order to create a larger pool of study and to receive more diverse and reliable results. While desired, the extra steps and precautions that would be needed to have extra test participants was simply not possible within the time frame that was given for this thesis. Thus, the bulk of the recordings and the data that this thesis relies on comes solely from this thesis's two authors. While the diversity of data is limited, it was circumvented by recording a lot of data over a longer time period in order to receive good results. Some extra data from two extra participants was also recorded to compare to the larger data sets already recorded.

5.3 Calibration validation of Tobii HTC Vive Devkit

When development started for the test protocol, the question regarding the calibration method quickly arose. In earlier works, both by Rosengren and Clay, the calibration is an integral part of achieving desired results for the eye tracker and both have their own solution of solving the specific issue of a good calibration. Early on, the possibility of creating a calibration model from scratch was discussed, but was however prohibited because Tobii hides that data. To access that kind of data, additional analytical tools needed to be purchased, and this was simply not possible for the scope of this thesis. Because of the lack of access to the calibration protocol for the HTC Vive it was difficult to determine how the system actually works and if it would be necessary to make any changes for different cases of eye conditions. Having the calibration locked to a second program also greatly reduces the ease of use and usability in a real test setting. The headset that was used is a sort of prototype with a standard HTC Vive with built-in Tobii eye trackers, meaning the HTC Vive was not created with the specific purpose of having eye trackers from the start. Newer headsets with built-in eye trackers such as the HTC Vive Pro Eye have the calibration protocol built into the eye tracker software instead, making it easier to access and use.

In order to use the HTC Vive in an actual clinical setting, direct access to the calibration would be optimal, since different eye conditions might need different kinds of calibrations. Nystagmus is known for being one such condition and has proved to be very difficult to calibrate accurately, at least with a precise system like the EyeLink, where the normal calibration requires the user to fix their gaze on a number of calibration points. This is difficult for someone with nystagmus, as described in earlier sections, but research done by Rosengren in his paper *"A Robust Method for Calibration of Eye Tracking Data Recorded during Nystagmus"* shows that it is possible to create an automatic calibration protocol for test subjects with nystagmus as well. This however requires that the chosen eye tracking system allows the calibration protocol to be modified and that access to the calibration protocol is possible. This was however nothing that the VR test setup allowed, creating somewhat of an obstacle for further development.

The issue of calibration was not an established research question from the start. Instead, it evolved from the need to further explain the difference between the two eye tracking systems and an issue that might have a major impact on the end result. From a scientific standpoint, the calibration done for the VR, lacks several key components. Most notably, it only conducts a five-point calibration and no validation is required, so it is never presented or quantified what the system deems as an acceptable accuracy. One might also discuss how accurate a five-point calibration is compared to the otherwise standard nine-point calibration. One interesting difference in benefit to the VR-system is that it calibrates just fine with a user that has nystagmus, something that the EyeLink struggles with. This in turn might add additional questions regarding its ability to calibrate properly and as to how accurate the measurements actually are. It might however just be the result of a lower sampling rate that makes it less sensitive to eye fluttering compared to the EyeLink.

The way validation works in contemporary systems is that measured results are compared to the true value, and that value difference becomes the error of the system. If the calibration was perfect and the user was physically able to keep their gaze completely still, then the error would be zero. This is of course impossible, but every system has a threshold of what an acceptable error is. If it exceeds that threshold, the validation has failed and the system or user needs to be corrected in some way.

Without a validation for the VR system, we had to construct one ourselves in order to quantify how effective the calibration done by Mirrors actually was. Thus, we had to do a lot of calculations of the root-mean-square error of the data collected from the control group. A properly recorded measurements shall give results that both have good accuracy and good precision on target. Bad accuracy but good precision gives signs of bad calibration, while the opposite indicates some fault of the user. Because of this, data from the nystagmus group was discarded because it produces neither good accuracy nor good precision.

Usually the magnitude of error is expressed in degrees, but for this particular case when the raw data was given in exact Unity coordinates and the size of the visual stimuli is known, it was deemed unnecessary to convert the result into degrees and then calculate the root-mean-square error. The decision to keep the data in Unity coordinates also made it more understandable whether the measurements are considered accurate or not at a quick glance.

Examining figure 35 and 36 it is hard to determine the reliability of the results just by looking at the graphs. Some results with a high RMSE have a noisier signal, but is mostly due to the user and not the system in itself. A clearer picture is achieved, by looking at the actual quantified results displayed in table 1 and 2 and the visualization of this data in figure 37. The visual stimuli of the test protocol has a diameter of 0.3 coordinate units, which means that the visual stimuli has a radius of 0.15 from the desired optimal gaze point, the center of the visual stimuli and its circumference. The most basic validation test was to calculate how many of the 50 fixation points for x and y-coordinates could be determined to have been measured inside the visual stimuli, i.e., having a RMSE lower than 0.15. Unfortunately, the results were not very good with just 62 % of the x-coordinates measured inside the stimuli and 56 % of the y-coordinates, which hardly can be deemed an accurate result by any means. This does not paint the whole picture, however, and by looking at the RMSE for each individual fixation, we see a clearer picture of the situation. In table 1 RMSE for the x-coordinates are presented, and for each individual fixation point of the 10 tests, there exist a clear divide between the three first fixation points. They are still located in the middle, where the x-coordinate remains very much still and the test subject changes their gaze vertically. These results remain under the threshold of 0.15, both regarding the mean and the median, which seem to indicate that initially the calibration works great for horizontal changes.

This is however abrupted when the fixation continues with fixation parts of left and right. The RMSE takes a sharp turn for the worse, and the error is above the threshold for almost all the 20 measurements. It receives a large, but comparable mean and median, indicating that the results are consistently outside the visual stimuli with a large margin. Doing the same comparison across the y-coordinates, the results are more consistent across the different fixations and measurements, with a mean that is close to or just outside the threshold of 0.15. It still is measured outside the visual stimuli, but not as much as for the left and right measurements of the x-coordinate.

Obviously, the accuracy of the measurements are lacking but is it to be considered a fault of the system or fault of the user? Usually, for measurements to be considered a user error, both accuracy and precision has to be bad. However, a similar mean and median also show some indication that the measurements are centered together, just not at the desired point of focus, and this indicates bad accuracy but good precision. The data indicates that the calibration is excellent at finding the center point and measuring it with good accuracy, but that it deteriorates after each move of the gaze. Moving the gaze away from the vertical centerline destroys the values and the accuracy jumps from 0.09 to 0.25, showing signs that the calibration was unable to create an accurate model along the x-axis. Measurements along the y-coordinate do not contain similar extreme jumps, but also indicate that the system has trouble calculating an accurate model for the y-axis as well.

These results point towards that the further away from origin the measurements are captured, the more error you get. There are also some indications that the calibration deteriorates for each measurement. This is shown both in the tables and figure, showing an increased chance of recording data points outside the visual stimuli for every new fixation point. This has also been claimed by Clay et. al as *"In our experience, the precision of eye tracking slowly deteriorates due to drifts, e.g., slight slips of the headset on the subject's head during the experiment in VR"* and our research show some signs of similar results.

All in all, the calibration for HTC Vive Devkit leaves a lot to be desired. Sidestepping the obvious hurdle of the need to have different programs for calibration and test protocol, it does not even achieve good results if the user is not supposed to just focus on stimuli positioned at the middle of the screen. Thus, for further research it is advisable to create a system that has calibration, validation and the test protocol all in the same program.

5.4 Accuracy and Precision

As can be seen in Table 3 and Table 4 the EyeLink system greatly outperforms the VR system in both accuracy and precision which was to be expected. What is more interesting is the spread of scores for the different stimuli positions that can be seen in Table 5 and Table 6. The EyeLink system has a considerably more even spread of scores over the different stimuli positions, while the VR system varies greatly, with the right and left accuracy being much worse than the accuracy for the center stimuli position. The same pattern also exists in the EyeLink scores, but to a much lower degree. One reason for this is, as discussed earlier, how the two test setups differ between the two systems. Even though the same test was used on both systems, the relative gaze angles to look at the different stimuli positions were not identical. This comes from the translation from a 3D experience in VR to a 2D experience being displayed on a screen not being correctly scaled. This leads to the required gaze angle to look at the different stimuli position, was lower in the EyeLink system compared to the VR system causing the VR system to perform worse for these large gaze angle positions, as eye-trackers generally perform worse for large gaze angles. This is something that was not realized until the end of the thesis and to avoid this problem, two different tests should have been created. One for the VR system and one for the EyeLink system, where each stimuli positions were placed to make sure the gaze angle required to look at them would be identical for all positions. For now, it would be a more fair comparison to look at the accuracy for only the center stimuli position for both systems, and the VR system would then have an average accuracy of 0.59 degrees and the EyeLink an average of 0.26 degrees. Another reason for the VR system spread of scores for the different positions could be due to how it is designed. The VR system is built with gaming and entertainment in mind, and having high accuracy in fields with large gaze angles simply is not very useful. Because in a VR experience, the user would turn their head in the position where they are looking instead of only turning their gaze to the side and keeping their head fixed.

The precision scores for both systems performed as expected, since precision measures the average distance between each sample and the EyeLink system samples at 1000 Hz, compared to the VR system's 90 Hz. It makes sense that the EyeLink system's precision is approximately 10 times better than the VR system.

When considering improvements for the accuracy and precision testing that was performed, a few things comes to mind. First, the two test systems stimuli positions should have been measured in gaze angles instead of absolute positions to ensure the translation between the 3D and 2D test would have been equal. Second, the stimuli positions that were tested only include the ones existing in the nystagmus test protocol, for a more general test setting more positions should have been tested for more gaze angles. And as always, having more test subjects and test samples would have helped to generalize the test results.

5.5 Nystagmus modeling and quality assessment of data

When modeling the signals for the two systems it was noticed that the VR system got a relatively good score for the horizontally placed stimuli, being the center, left and right stimuli while giving close to zero for the top and bottom stimuli positions, which can be seen in Table 7. This likely due to two different reasons. For starters, the author with nystagmus that the tests were performed on experiences larger nystagmus stimulation for specific eye positions. With the leftmost position being the strongest such stimulation position, leading to the nystagmus oscillations being larger and more uniform. This in turn made the nystagmus modeling perform better. The reason this is not reproduced in the EyeLink tests to the same degree was due to the fact that the relative gaze angles needed to look at the stimuli were lower and therefore did not stimulate the nystagmus oscillations as much. The fact that nystagmus behaves differently for different gaze angles was something that was not considered when choosing the stimuli positions for the VR test protocol, and the issues it presented were not realized until the end of the thesis. The second reason for the VR system's poor score for the top and bottom stimuli positions was a combination of the VR eye-tracking being less accurate when measuring further away from the center position, combined with the fact that the nystagmus oscillations also were weaker, due to not being as active with these gaze angles. This made it hard for the nystagmus modeling algorithm to accurately reproduce the nystagmus oscillations when creating the model. This is due to the fact that the algorithm looks for patterns and uniformity when estimating the model and when the nystagmus oscillations are weak combined with inaccurate and noisy data, leads to poor nystagmus waveform model estimations, which shows itself in the low scores for these specific positions.

The EyeLink scores which can be seen in Table 8 are much more even over the different stimuli positions, indicating that the accuracy and precision of the system varies less with varying gaze angles, which could also be seen in the accuracy and precision calculations. A few outliers exist in the EyeLink data which is more likely due to the data being non-optimal as can be seen in figure 40, depicting a poor signal remodeling due to an irregular signal. The reasons that the EyeLink average accepted segments (30%) is as close as it is to the VR average accepted segments (24%) has more likely to do with the difference in gaze angles required to focus on the stimuli and the low amounts of test performed. It would be more indicative to look at only the score of the central stimuli for both systems, since the gaze angle for these stimuli is identical in both tests, giving a 26.5% average for the VR system and a 34.7% average for the EyeLink system.

A more optimal way of doing the nystagmus modeling and quality assessment of data would have been to make sure that the tests were identical, both in distances between subject and stimuli positions and gaze angles between each stimuli position, to ensure equality between the two systems for testing. Many more tests on different people should also have been gathered in order to reduce the effects of outliers and to increase the diversity of the testing.

5.6 Comparison of the EyeLink 1000 and Tobii HTC Vive Devkit

After much work, what have actually been achieved? What pros and cons can actually be stated between a stationary system like the EyeLink 1000 and the Tobii HTC Vive Devkit? When looking at the accuracy and precision scores for both systems, the EyeLink system outperforms the VR system with a large margin. This performance however does not extend to the same degree when looking at how each system performed with the nystagmus modeling and quality assessment of data, where the EyeLink system still outperformed the VR system, but to a much lower degree compared to the accuracy and precision tests. What can be said about this is that it matters on what type of research and results one is after. For eye tracking tasks which require high precision and accuracy such as evaluation of reading text the EyeLink system is the way to go, but for tasks where high precision and accuracy is of less importance such as tracking larger objects or sweeping movements the VR system performs to a comparable degree.

The Vive has several benefactors when it comes to ease of use, portability, cost efficiency. If it was anyone's first dabble into the world of eye tracking, the low entry point for use, both in terms of virtual reality and eye tracking, makes it a solid candidate for a preferred system. Because of its consumer oriented background, it feels like most of the software and surrounding technology was made for ease of use in mind. It is likely developed to spread the use of eye tracking for all purposes, and this is evident both in terms of the origin of the VR headset, based around the consumer variant of HTC Vive and the interest of Tobii, the manufacturer of the eye tracking technology. The technology behind it all is incredibly advanced, but the software that is required to use and extract data is quite straightforward. It is all very well documented on Tobii's web page, and it did not take many minutes before you have created your first interactive world that can respond to eye gaze. Learning to create the desired world to put your user in has a slightly higher entry barrier, but within VR development it is very much encouraged to use the open game engine Unity, which has a tremendous amount of learning tutorials and a large community. The eye tracking data that was received was also easy to understand, as it was given in coordinates relative to the coordinate system used in Unity, thus making the evaluation of where the user was looking easy to understand and evaluate. All in all, using and developing for the HTC Vive was a great experience, and the project could move along quick and without larger hurdles. Of course, this ease of use comes at a price, and that is most notably shown in the quality of the measurements acquired. The eye tracker for the Vive is capable of 120 Hz but is usually capped at the HMD's screen's frame rate of 90 Hz for compatibility reasons, putting the sampling rate at a clear disadvantage of more robust systems. It is never disclosed how the calibration of the eye tracker works or how it is performed, and the dev kit gives no possibility to create a customized calibration, which in turn creates an uncertainty of how good the measurements actually are.

On the other spectrum, we have the EyeLink 1000, a big, expensive system with lots of bells and whistles. The HTC Vive is to be considered a toy in comparison to the EyeLink, and it is not far from the truth. The advanced optics and metrics of the EyeLink include a sampling rate of 1000 Hz, customizable calibration and individual output results from each eye of the test participant. The measurements can be monitored in real time along with the footage of the eyes and where the pupil and glint are located. The ease of use of the system is however non-existent in comparison to the Vive. Much of the calibration and the capturing of data needs additional coding and program knowledge and since the EyeLink is not linked to the visual stimuli, unless specifically told so, the outputs are given as positional data of pixels in relation to the calibration, which might be harder to interpret.

Side by side, the EyeLink 1000 and the Tobii HTC Vive Devkit serve two entirely different markets. With the specs of the Vive, it is obvious that the intention of its development and manufacturing were never scientific research, but a more fun addition to interact with the world in virtual reality. For example, when the main focus of the included games and the coding tutorial are focused on making big blocks light up when looking at them, it is quite clear that the focus is not pinpoint accuracy but more about achieving a good enough result for its intended purpose. The EyeLink is a scientific tool, and this is evident in all of its design. It is heavy, complex, expensive and takes an effort to learn how to use and get the most out of all its features, but when supreme data quality is of essence, you cannot go wrong using it. The HTC Vive has potential but is in need of hardware upgrade, especially of the eye tracker side, which will drive up the price. It is not in need of something extreme though, possibly just upgrade the lenses to a refresh rate of 100 Hz in order to drive both the screens and eye tracker at a sampling rate of 100 Hz which would make it easier to scale to the EyeLink.

Finally, the Vive is a fantastic tool for learning and doing research when the accuracy is not of importance. It works well enough when the study of the signal in itself is the goal, but its usability as a scientific tool, with good data quality and repeatability of results, tops off fairly quickly. The EyeLink on the other hand is like any other professional tool, it takes a ton of time to learn properly but when that hurdle has been passed, it will serve you well for a long time.

6 Conclusions

A virtual reality test environment was successfully created for the Tobii HTC Vive Devkit using the game engine Unity, containing a nystagmus test protocol comparable to a previous research protocol created by William Rosengren, who used the EyeLink 1000 system. Eye movements were recorded for both systems, and several tests were performed to evaluate and compare the data quality between both systems. The evaluation showed that the performance of the VR system was lower than the EyeLink system to a degree, but it was still possible to clearly see the nystagmus waveforms in the recorded data. It was also possible to accurately reproduce nystagmus waveform models from the recorded eye movements. When comparing the advantages and disadvantages of using the VR system compared to the EyeLink system, the VR system proved to be very easy to use and gave good results for simple tasks. It was possible to create an entirely new test setup from scratch and use it to record data without previous experience, either with the game engine Unity or virtual reality. As we did not implement anything specific for the EyeLink system but instead relied on the assistance from supervisors and using the same implemented protocols as for the VR-system, it is hard to say anything on the comparable ease of use of this system. In conclusion, based on the limited scope and data included in this thesis. When choosing between the two systems for performing some type of diagnostic test depends entirely on the nature of the test itself. For some tests such as reading a text, where accuracy and precision are very important metrics, the EyeLink system will outperform the VR system. For other tests, such as visually tracking a moving object, the lower data quality of the VR system might be enough for a correct diagnosis.

6.1 Future work

Since this thesis was written during an ongoing pandemic, it was not possible to test more than 2-3 subjects. Having more test subjects with diagnosed nystagmus would have been preferable, as nystagmus behaves differently from person to person. It would have been beneficial to have the ability to test the system on different variations, as well as having more test data for a more thorough analysis. Additionally, it would have been interesting to increase the number of fixation points to for example 9 points, which is the standard amount for calibration.

During this thesis, we only compared the VR system to another system in order to get a baseline in performance. It would however have been interesting to examine how good the eye tracking actually has to be in order to evaluate nystagmus and other eye conditions to an acceptable degree.

From the start of the thesis, one of the goals were to examine new possibilities for different type of conditions and diagnostics. After the thesis started, however, it was decided to only focus on the nystagmus evaluation for time frame reasons and as we needed some way to evaluate and compare the results. There exists, however, an endless amount of new possibilities that could be examined.

References

[1] S. Rogers. ”Seven Reasons Why Eye-tracking Will Fundamentally Change VR”. *Forbes.* 2019-02-05. `https://www.forbes.com/sites/solrogers/2019/02/05/seven-reasons-why-eye-tracking-will-fundamentally-change-vr/?sh=3b34a2053459`

[2] K. Harezlak, P. Kasprowski “Application of eye tracking in medicine: A survey, research issues and challenges” *Computerized Medical Imaging and Graphics*, 65:176–190, 2016.

[3] Clay, V., König, P. & König, S. Eye Tracking in Virtual Reality. *Journal of Eye Movement Research*, 12(1):3. 2019.

[4] Lohr, D. J., Friedman, L., Komogortsev, O. V. Evaluating the Data Quality of Eye Tracking Signals From a Virtual Reality System: Case Study Using SMI's Eye-Tracking HTC Vive. *arXiv e-prints*, art. arXiv:1912.02083v1, Dec 2019.

[5] W. Rosengren. *Characterisation of nystagmus waveforms in eye-tracker signals.* Department of Biomedical Engineering, Lund University. 2020.

[6] E. P. Widmeier, H. Raff, K. T. Strang. *Vander's Human Physiology: The Mechanisms of Body Function.* New York: McGraw-Hill, Thirteenth Edition, 2014.

[7] Wikimedia Commons, *”Schematic diagram of the human eye”*. 2007. `https://commons.wikimedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg`

[8] Wikimedia Commons, *”Human visual pathway”*. 2015 `https://commons.wikimedia.org/wiki/File:Human_visual_pathway.svg`

[9] Wikimedia Commons, *”Lateral orbit nerves”*. 2006 `https://commons.wikimedia.org/wiki/File:Lateral_orbit_nerves.jpg`

[10] R.V. Abadi, “Mechanisms underlying nystagmus,” *Journal of the Royal Society of Medicine*, 95(5):231–234, 2002.

[11] N. Hussain, “Diagnosis, assessment and management of nystagmus in childhood,” *Paediatrics and Child Health*, 26(1):31–36, 2016.

[12] Virtual reality `https://en.wikipedia.org/wiki/Virtual_reality`

[13] Wikimedia Commons, *”Sensorama patent”*. 1961. `https://commons.wikimedia.org/wiki/File:Sensorama_patent_fig5.png`

[14] Wikimedia Commons, *"Sensorama headset patent"*. 1960. https://commons.wikimedia.org/wiki/File:Sensorama_morton_heilig_patent.png

[15] VR positional tracking Sensor Fusion https://en.wikipedia.org/wiki/VR_positional_tracking#Sensor_Fusion

[16] Wikimedia Commons, *"Six degrees of freedom"*. 2015. https://commons.wikimedia.org/wiki/File:6DOF.svg

[17] VR positional tracking Inside-out Tracking https://en.wikipedia.org/wiki/VR_positional_tracking#Inside-out_Tracking

[18] T. English, "VR Headsets Work through a Combination of Different Tracking Technologies" *Interesting Engineering*, 2020-05-09. https://interestingengineering.com/vr-headsets-work-through-a-combination-of-different-tracking-technologies

[19] Virtual reality sickness https://en.wikipedia.org/wiki/Virtual_reality_sickness

[20] Screen-door effect *Screen door effect* https://en.wikipedia.org/wiki/Screen-door_effect

[21] Wikimedia Commons, *"Visible light eye-tracking algorithm"*. 2015. https://commons.wikimedia.org/wiki/File:Visible_light_eye-tracking_algorithm.jpg

[22] Tobii Pro VR Integration – based on HTC Vive Development Kit Description https://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-vr-integration-product-description.pdf/?v=1.7

[23] Y. Durna, F. Ari. "Design of a Binocular Pupil and Gaze Point Detection System Utilizing High Definition Images" *Applied Sciences*, 7(5):498, 2017.

[24] M. Stridh, D. Husser, A. Bollmann, and L. Sörnmo, "Waveform characteriza-tion of atrial fibrillation using phase information," *IEEE Transactions on Biomed- ical Engineering*, vol. 56, no. 4, pp. 1081–1089, 2009.

[25] L. Sörnmo and P. Laguna, *Bioelectrical signal processing in cardiac and neurological applications,* vol. 8. Academic Press, 2005.

[26] D.Heaney, "How VR Positional Tracking Systems Work" *Upload*, 2019-04-29, `https://uploadvr.com/how-vr-tracking-works/`

[27] B. Poetker "The Very Real History of Virtual Reality (+A Look Ahead)" *G2*
`https://www.g2.com/articles/history-of-virtual-reality#:~:text=Virtual%20reality%20technology%20was%20invented,1987%20by%20researcher%20Jaron%20Lanier.`

[28] VR positional tracking Outside-in Tracking
`https://en.wikipedia.org/wiki/VR_positional_tracking#Outside-in_Tracking`