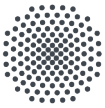


Universität Stuttgart

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Evaluation and Application of Estimated Gaze Depth in Virtual Reality

Tobias Walter

Course of Study:	M.Sc. Informatik
Examiner:	Prof. Dr. Michael Sedlmair
Supervisor:	Sergej Geringer M.Sc., Dr. Kuno Kurzhals, Aimée Sousa Calepso M.Sc., Dr. Guido Reina
Commenced:	December 16, 2021
Completed:	July 18, 2022

Kurzfassung

Eye Tracking Kameras werden zum Standard in neuen Virtual Reality Brillen. Während Evaluation und Bewertung von zweidimensionalen Eye Tracking Daten schon Einsatz in Forschung und Designprozessen finden, ist der Einsatz von dreidimensionaler Blicktiefe weitgehend unerforscht. Üblicherweise wird zur Schätzung der Blicktiefe der Blickstrahl mit einer zweidimensionalen Ebene, z. B. dem Bilschirm, geschnitten. Allerdings setzt dieser Ansatz voraus, dass Abstände der Szene bekannt sind und keine Verdeckung auftritt. In dieser Arbeit wird die Blicktiefe durch das Schneiden der Blickgeraden beider Augen geschätzt. Dies ermöglicht die Verwendung von semi- transparenten Objekten, mit denen ein Benutzer interagieren kann. Einblicke in die Blicktiefe können wertvolle Einsichten in Benutzerverhalten liefern, in verdeckungsreichen Szenen die Frage klären, welches Ziel fokussiert wird und neue Interaktionstechniken ermöglichen. Ziel dieser Arbeit ist es, Blicktiefenschätzung zu evaluieren und neue Verwendungsmöglichkeiten zu erfrorschen. Um eine zuverlässige Schätzung zu erhalten, werden zwei Kalibrierungsprozeduren entwickelt, die auf aktuellen Methoden aufbauen und Modalitäten verglichen, die Einfluss auf die Kalibrierung haben könnten. Die Implementierung wurde in einer Pilotstudie (n=10) verglichen. Die Ergebnisse zeigen, dass Interaktion gut in Distanzen bis zu 1.2 Metern funktioniert, während Objekte, die nur 30 cm vom Benutzer entfernt waren, teilweise als unangenehm empfunden wurden. Außerdem legen die Ergebnisse nahe, dass ein sich bewegendes Kalibrierungsziel zu einer besseren Allzweckkalibrierung führt. Eine sorgfältige Kalibrierung des Raumes, in dem Interaktion verwendet wird, kann daher die Blicktiefenschätzung und Interaktion verbessern.

Abstract

Cameras for eye tracking are becoming standard for modern virtual reality head-mounted displays. While evaluation and interaction of two-dimensional eye tracking data are widely researched and applied, the three-dimensional point of regard is mostly unexplored. Usually, gaze depth is determined using the intersection of a gaze ray with a two-dimensional plane such as the screen. However, this requires complete scene knowledge and an occlusion-free environment. In this work, the estimation is done by intersecting the two eyes' gaze rays. This enables the usage of semi-transparent objects a user can interact with by focusing directly on them or looking through them. Insight into the users' gaze depth could provide valuable data to help understand user behavior, disambiguate targets in high occlusion environments and enable new interaction techniques. The subject of this work is to evaluate the estimation of gaze depth and to explore possible novel use cases. In order to get a stable estimation of the gaze depth, two vergence-based calibration procedures are developed improving on existing research and comparing modalities that could impact calibration quality. The implementations were evaluated in a small-scale pilot (n=10) study. Results show that interaction works well in distances up to 1.2 meters, while very close distances of about 30 cm were perceived as uncomfortable to look at by some users. Results also indicate that a continuously moving calibration target leads to a better all-purpose calibration. Carefully calibrating the space used for interaction can therefore lead to improved gaze depth estimation and interaction.

Contents

1 Introduction **9**

2 Related Work **13**
2.1 Eye Tracking 13
2.2 Depth Estimation 14
2.3 Gaze Interaction 15

3 Concept **17**
3.1 Gaze Depth Estimation 17
3.2 Calibration Procedure 21
3.3 Machine Learning Estimation 22
3.4 Gaze Depth Interaction 23

4 Implementation **25**
4.1 Calibration 25
4.2 Machine Learning Features and Parameters 26
4.3 Interaction 27

5 User Study **29**
5.1 Participants 29
5.2 Apparatus 29
5.3 Procedure 29

6 Results **33**
6.1 Quantitative Results 33
6.2 Subjective Results 36

7 Discussion **39**
7.1 Limitations and Future Work 40

8 Conclusion **43**

Bibliography **45**

A Study Forms **49**
A.1 Consent Form 50
A.2 Questionnaire 55
A.3 Results 59

1 Introduction

The human gaze provides valuable data on how a user interacts with the environment as well as information displayed within that environment. Leveraging this information has already found its way to design processes for example in web development and product design ([Duc02], [WP08]). In two-dimensional eye tracking, interaction has seen a lot of research. While most established procedures that use eye tracking are limited to a two-dimensional plane, e.g. the screen of a smartphone or tablet, the third, spatial dimension could enrich the data and provide more insights. Gaze interaction and eye tracking, in general, are often still limited to a two-dimensional plane, only taking into consideration what the gaze ray hit first. However, in virtual and augmented reality, three-dimensional vision is inherently part of the design. Thus the depth complexity is higher than in two-dimensional environments.

In virtual, augmented or mixed reality environments, virtual and real objects overlap each other, leading to higher occlusion. This can cause ambiguities that could be solved with accurate gaze depth estimation. Another case where gaze ray intersection fails to deliver accurate estimations is when semi-transparent elements are part of a scene. Windows or transparent user interface (UI) elements, as used in Heads up Displays (HUD), can cause this problem. On the other hand, if depth estimation is working properly, these elements of a scene can be turned into interactive objects that can be activated hands-free, enabling new gaze interaction techniques. Therefore, correctly capturing the point of regard (PoR) and leveraging this data could improve and expand the interaction repertoire.

As accurate eye tracking devices become more affordable and a standard in new virtual and augmented reality devices, utilizing the gaze data, including gaze depth, becomes an interesting prospect for interaction and visualization. Starting from simple selection, the possible interactions could also include manipulation of objects using gaze and even locomotion. One method where three-dimensional gaze has found application is foveated (or gaze-contingent) rendering (eg. [RWH+17; WRHS18]). Foveated rendering methods use the gaze data to adapt the image synthesis process to improve performance. The human visual system has limitations where it is not necessary to provide the highest level of detail everywhere. This provides a reduction in computation cost in virtual reality where high refresh rates and low latency are needed.

Gaze depth can be estimated in multiple ways. Usually, it is derived by taking the depth of the object first hit by the users' gaze ray. This way to estimate the point of regard, while easy to calculate, is not suitable for higher occlusion environments and can quickly misinterpret the true gaze depth when eye tracking is inaccurate. This especially happens with thin objects where small inaccuracies lead to a misinterpretation. Accommodation describes the lenses of the eyes adjusting to the fixated object. The curvature of the lens can be measured and used to derive a focus distance. However, the required hardware is bulky and not yet suitable for head-mounted displays (HMD). The movement resulting in the inward rotation of both eyes is called vergence. Another way to calculate the PoR is to intersect the two eyes' gaze rays and use this point as the focus point. The angle in the focus point

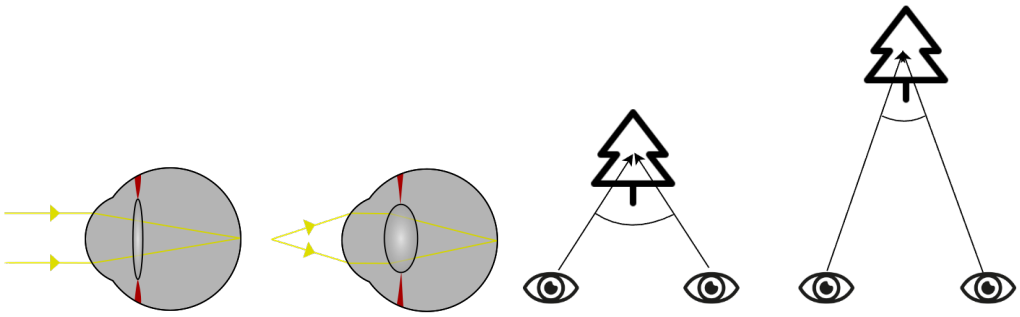


Figure 1.1: Motoric depth cues that can be measured to be used as estimations for gaze depth (accommodation graphic by Silversmith [Sil07]). Left shows the adaptation of the eye lens thickness in order to get a sharp focus (accommodation). Right shows the inward rotation of both eyes (vergence) when looking at different distances.

is the vergence angle. The challenge with this geometric approach is that as distance increases, the angle between the gaze rays in the gaze point and thus the accuracy decreases. A visualization of both of the motoric depth cues can be seen in Figure 1.1. Even modern sub-degree accuracy eye trackers cannot compensate for the exponentially decreasing angle in the gaze point. To compensate for that, the calibration in the space of interest is important, smoothing errors and accounting for differences in users.

Measuring gaze depth requires accurate eye-tracking devices. Different users may face differences in accuracy of eye tracking based on eyesight, visual impairments and other confounding factors [FWT+17]. To achieve accurate eye tracking, HMDs use calibration procedures to adapt the eye tracker to the user. While the calibration on a plane might be enough for traditional eye tracking applications, for gaze depth a calibration that samples the whole depth to be used might be needed [DHG+14]. Calibration procedures inherently pose a problem for usability, requiring a set-up before every usage. The trade-off between calibration time and accuracy, therefore, has to be carefully considered. Additional modalities are the choice of the calibration target and how the targets move through the calibration space. A series of stationary targets that are shown in sequence might be easier to focus on, while a moving target could potentially sample the space more tightly.

The goal of this master thesis is to evaluate gaze depth estimation methods to find suitable applications. The naive geometrical estimation is tested and for more reliable data two calibration procedures are developed. Both procedures use Machine Learning (ML) algorithms such as K-Nearest Neighbors, Support Vector Machine and Multi-Layer Perceptron to regress from the first depth estimate to a second more precise PoR. The features used to train the models are based only on vergence. In that way, we receive a robust estimation that works in high occlusion environments and can be used to activate transparent UI elements.

During the first trials, the following research questions emerged:

- How good is the geometric estimation?
- In which distance is the estimation accurate?
- Is the estimation usable for selection tasks?
- Does a moving target in the calibration impact the calibration quality?

Structure

The master thesis is structured in the following way: In Chapter 2, related work, current research and state of the art are reiterated and developments in eye tracking, gaze depth estimation and gaze interaction are examined. Chapter 3 explains the methods of our approach, the initial idea and early tests that were run in order to evaluate the hardware and the naive approach. Chapter 4 gives insight into the implementation of the gaze depth estimation including the features and parameters for the Machine Learning algorithms. The design of the user study is explained in Chapter 5. Results are presented in Chapter 6, followed by a discussion including possible future work in Chapter 7 and a conclusion in Chapter 8.

2 Related Work

In the last three decades, there has been a lot of work on eye tracking and gaze interaction ([Duc18]). With stereoscopic eye tracking becoming more prevalent, a lot of research has been done concerning eye tracking, perception and interaction. While this work uses VR, some augmented and real-world eye tracking papers will be included for a better understanding of problems and challenges that can arise. The related work chapter will be divided into three sections about eye tracking, gaze depth estimation and a section about gaze interaction.

2.1 Eye Tracking

Eye tracking is the basis for both depth estimation and gaze interaction techniques. However, eye tracking faces a lot of challenges that are tightly coupled to differences in users, devices, calibration and environmental factors. A study by Feit et al. [FWT+17] showed that precision and accuracy between users can vary up to sixfold, while changes in lighting conditions lead to a deviation of 8.3%. Findings by Kuo et al. [KSCC18] support the user dependence of eye tracking accuracy. They also find that amplitudes of fixations between individuals are different from one another, indicating that the customization of parameters via calibration is needed. In order to make data collection, robust filters are used ([HNA+11], [FWT+17], [Špa12]) that serve multiple purposes. Denoising smoothes the signal to remove unwanted jittering and perturbations caused by calibration problems or sensor errors. Other filters can detect events of interest, most commonly fixations and saccades. Fixations are a period of time where the gaze stays in one location or small area, while saccades denote a rapid movement between fixations.

Pfeuffer et al. [PVT+13] developed a calibration procedure using a moving target. Their method detects the user's attention to a calibration target and correlates the eye movement and target trajectory. Data for calibration is only sampled when the user is attending the target.

Filters are not the only technique to analyze gaze during runtime. Schulz et al. [SBBW17] present other analysis and cleansing techniques. Their cleansing function is based on a pipes and filters model that drops data if it is triggered by a signal. In particular, comparing the left and right eye to the combined gaze shows promising results. The effect of the calibration method and eye physiology was the subject of research by Nyström et al. [NAHW12]. Contact lenses, for example, increased the offset between target and estimated gaze direction compared to users without. Glasses, however, did not influence the offset in their research.

Many factors can contribute to the accuracy of eye tracking. Therefore, the collection, filtering and processing of eye tracking data has to be carefully considered. To make the gaze depth interaction technique developed in this work robust for differences in users, we use two calibration methods that allow customization.

2.2 Depth Estimation

One straightforward way to get a gaze depth estimation is to calculate the eyes' vergence based on the normal vectors of the eyes. The closest point you can converge to is estimated to be around 20 cm, while the estimation becomes unreliable at distances over 1.5 meters [MBWK16].

Duchowski et al. [DHG+14] found that there is a difference between depth perception in real and augmented environments. However, Oeney et al. [ORB+20] investigated whether measured 3d gaze depth through eye tracking shows differences in real and augmented reality environments. They found no significant difference in gaze depth between reality and augmented reality. On a similar note, Linton [Lin20] researched if vergence is an important absolute depth cue to help users perceive the distance to objects. They found that vergence might be bad as an absolute depth cue.

A taxonomy by Hirzle et al. [HGG+19] gathers research including various ways of gaze interaction and visualizations. They separate approaches based on three categories and create a design space based on these principles. The categories are: Device Type (VR or AR), Display Type (monocular or stereoscopic) and Gaze Depth Estimation Method (vergence or accommodation).

Pfeiffer et al. [PR14] compared two approaches to gaze depth estimation: on the one hand a geometric approach that uses purely geometric calculations and intersections to derive an estimation. On the other hand, a new approach, using only the observations of the user with knowledge about the world space.

Wang et al. [WPDH12] used a calibration that calculates corrections for measured points on multiple planes using a lagrangian least squares model. The calibration procedure uses a continuously moving point that samples the whole depth of the scene. Mlot et al. [MBWK16] find that using only one plane to interpolate line of sight is not accurate enough for depth estimation and error rises as the point of regard lies further away from the calibration plane. Further, they present an estimation method that uses interpolation, achieving good accuracy in distances between 20 and 40 cm from the user. Weier et al. [WRHS18] use multiple features including vergence, center and variance of the PoR to train a support vector machine. They achieve good accuracies at distances of up to 6 meters, concluding the importance of user-specific calibration. Training larger data sets, as collected in their study, lead to long training times, which could hinder usability. Lee at al. [LSP+17] use vergence-based measures to train a Multi-Layer Perceptron for depth regression.

While the estimation of gaze depth using world knowledge has benefits in terms of accuracy, the training of the model takes a lot of time. Additionally, the intersection of the gaze rays with objects does only work in environments without occluding windows or UI elements. To avoid those limitations, we want to investigate vergence-based estimation methods and compare different Machine Learning algorithms and data collection procedures.

2.3 Gaze Interaction

Gaze offers a unique interaction possibility in that it is inherently hands-free and intuitive. However, the Midas Touch Problem [Jac90] can occur when users are looking at objects. It describes the problem of unintentional commands since every look can be interpreted as interaction. While the hands-free approach would be ideal, often a confirmation action such as a key press can help to avoid the Midas Touch Problem.

Kammerer et al. [KSB08] compared the effectiveness of radial and linear menu layouts for gaze interaction. They found that a semi-circle menu worked best, while finding no evidence that a multi-modal device input (gaze & speech) was superior to unimodal input. The combination of dwell time needed for activation and size of the visual angle were attributed to reducing the effect of the Midas Touch Problem. Another menu navigation technique was developed by Ahn et al. [ASP+21]. In three studies they optimized and developed gaze interaction guidelines. Paulus and Remijn [PR21] tested various dwell times for object selection. Users preferred dwell times of 600 ms, while lower dwell times of 400 ms might be usable for certain systems and more experienced users.

Vidal et al. [VNL14] use vergence to infer attention to change displayed information. When looking at the edge of the screen, the user can switch from virtual world to the real world on demand. Using their design space described before, Hirzle et al. [HGG+19] developed an exemplary technique allowing users to look through a wall to reveal a hidden picture. The technique uses gaze depth estimation to activate the transparency of the wall. A problem they found was that voluntary vergence movement without fixating on an object is relatively hard. As a solution, they provided a "scaffolding" which is a matrix of points the user can look at to make the depth selection easier. The points were color coded to indicate their depths.

A gaze interaction technique presented by Pai et al. [POVK16] uses gaze depth as input for Virtual Reality applications. In a study, they found that users could place a sphere on a target using gaze at a similar precision as mouse users. As a sample use case, they show how a heads-up-display (HUD) could work. The HUD becomes opaque when focusing near and transparent when focusing far to see the main content, in this case, a sports event.

Riegler et al. [RARH20] investigated the impact of dwell times and feedback design for heads-up-displays in automated driving tasks. They find that users prefer longer dwell times for interaction. A circular feedback around the PoR performed better during their study, showing lower mental demand and lower error rates. A prototype by Barz et al. [BKKS21] classifies objects in augmented reality to attach labels to them based on the users' PoR.

The work by Pai et al. [POVK16] and Hirzle et al. [HGG+19] gives the basis for this work, exploring the interaction possibilities of see-through planes and (semi-) transparent objects and possibly providing solutions for high occlusion selection tasks. Tobii, a provider of eye tracking devices and software, offers a "Clean UI" implementation [Tob22], making UI elements transparent until the user looks at them. Gaze depth could take this one step further, allowing the user to look through the UI as well. To achieve this, user-specific calibration as seen in work by Wang et al. [WPDH12] and Weier et al. [WRHS18] are used, finding a trade-off between accuracy and time efficiency.

3 Concept

When interacting in virtual or augmented environments, there is a lot of data, that can be extracted from the human gaze. The prevalence of Eye trackers in HMDs allows using this data in various ways. In particular, the calculation of the users' 3D gaze opens a new interaction space. The initial idea for this work was to have multiple UI elements that potentially overlap. A user would then activate or deactivate them with 3D gaze to adapt the scene to their needs. In high occlusion data, for example, scientific visualizations of molecules, this could be applied to allow the user to dynamically change the displayed information by hiding occluding elements and highlighting important parts of the visualization. Other possible areas are HUDs in cars or in gaming and immersive environments where high depth complexity is present, e.g. sensor data visualization in highly automated production facilities. Hiding UI elements that are not needed could improve the focus on what is important at the moment.

In this chapter, the methods and initial experiments that led to the current design are explained. First the geometrical gaze depth estimation techniques and calculations are presented. The second section is concerned with the developed calibration procedures. Then, based on the data collected during the calibration, the machine learning algorithms to be used are explained. The last section of this chapter describes the interaction technique that allows users to interact via gaze depth.

3.1 Gaze Depth Estimation

3.1.1 Geometrical Gaze Depth Estimation

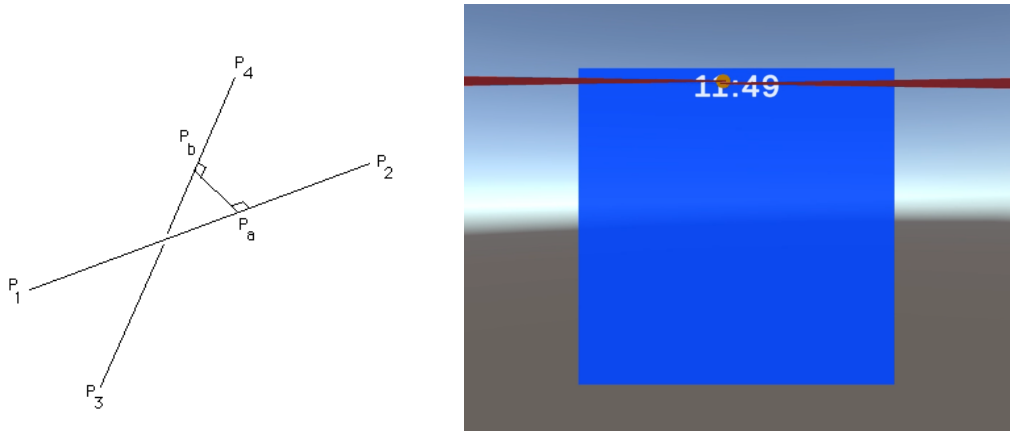
The first step was to implement a simple geometry-based depth estimation. This can be done geometrically using the data we gather with the Varjo eye tracker. To get the PoR, the gaze rays of both eyes are intersected. Since we are in a three-dimensional environment, the rays will not intersect in most cases. To account for that, the closest point between the rays is calculated instead. Figure 3.1a shows an illustration of the gaze rays.

The calculation follows "The shortest line between two lines in 3D" by Bourke [Bou22]. Let $\overline{P_1P_2}$ and $\overline{P_3P_4}$ be the gaze rays for the left and right eye respectively. We are looking for the points P_a and P_b where the distance $\|P_b - P_a\|^2$ is minimal. The fact that the shortest line is perpendicular to both gaze rays gives the equations:

$$\begin{aligned} (P_a - P_b) \cdot (P_2 - P_1) = 0 \\ (P_a - P_b) \cdot (P_4 - P_3) = 0 \end{aligned} \tag{3.1}$$

Rewriting the equations with t, v being the line segments P_a,P_b lie on:

$$\begin{aligned} ((P_1 + t(P_2 - P_1)) - (P_3 + v(P_4 - P_3))) \cdot (P_2 - P_1) = 0 \\ ((P_1 + t(P_2 - P_1)) - (P_3 + v(P_4 - P_3))) \cdot (P_4 - P_3) = 0 \end{aligned} \tag{3.2}$$



(a) **(b)**

Figure 3.1: a) Line intersection in three-dimensional space. Lines do not intersect in most cases. Instead, we search for the closest distance, minimizing $\|P_b - P_a\|^2$ between the gaze rays $\overline{P_1P_2}$ and $\overline{P_3P_4}$, and take the middle point [Bou22]. b) Gaze rays (red) and calculated focus point (yellow) recorded in a Unity scene.

Rearranging these equations again by expanding them yields:

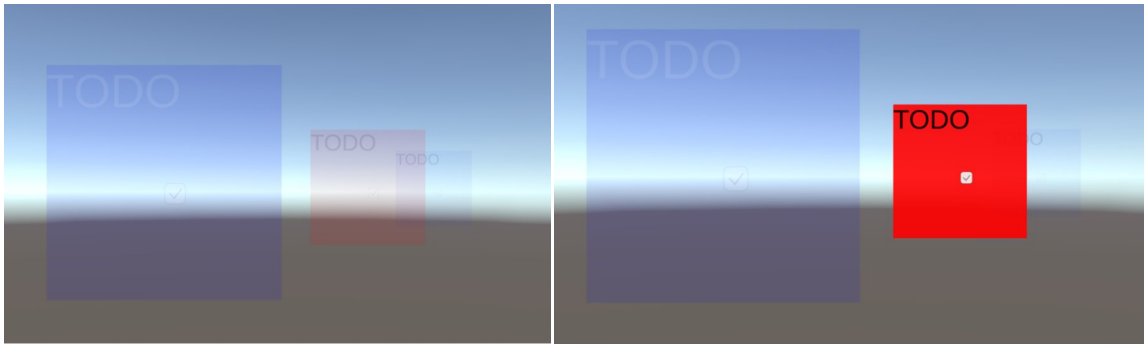
$$\begin{aligned}(P_1 - P_3 + t(P_2 - P_1) + v(P_4 - P_3)) \cdot (P_2 - P_1) = 0 \\ (P_1 - P_3 + t(P_2 - P_1) + v(P_4 - P_3)) \cdot (P_4 - P_3) = 0\end{aligned} \tag{3.3}$$

Simplifying with $\overline{r} = P_2 - P_1$, $\overline{s} = P_4 - P_3$ and $\overline{q} = P_1 - P_3$ results in a system of equations:

$$\begin{aligned}(\overline{q} + t\overline{r} - v\overline{s}) \cdot \overline{s} = \overline{q} \cdot \overline{r} + t(\overline{r} \cdot \overline{r}) - u(\overline{s} \cdot \overline{r}) = 0 \\ (\overline{q} + t\overline{r} - v\overline{s}) \cdot \overline{r} = \overline{q} \cdot \overline{r} + t(\overline{s} \cdot \overline{r}) - u(\overline{s} \cdot \overline{s}) = 0\end{aligned} \tag{3.4}$$

Since we have two variables, t and v, we should receive exactly one solution. If, however, we have no solution, the gaze rays are parallel. The equations can be solved for t and v respectively. Substituting the equation for v in t results in the two final equations 3.5.

$$\begin{aligned}v &= \frac{(\bar{q} \cdot \bar{s})t(\bar{r} \cdot \bar{s})}{(\bar{s} \cdot \bar{s})} \\ t &= \frac{(\bar{q} \cdot \bar{s})(\bar{s} \cdot \bar{r}) - (\bar{q} \cdot \bar{s})(\bar{s} \cdot \bar{r})}{(\bar{r} \cdot \bar{r})(\bar{s} \cdot \bar{s}) - (\bar{r} \cdot \bar{s})(\bar{s} \cdot \bar{r})}\end{aligned} \tag{3.5}$$



(a) (b)

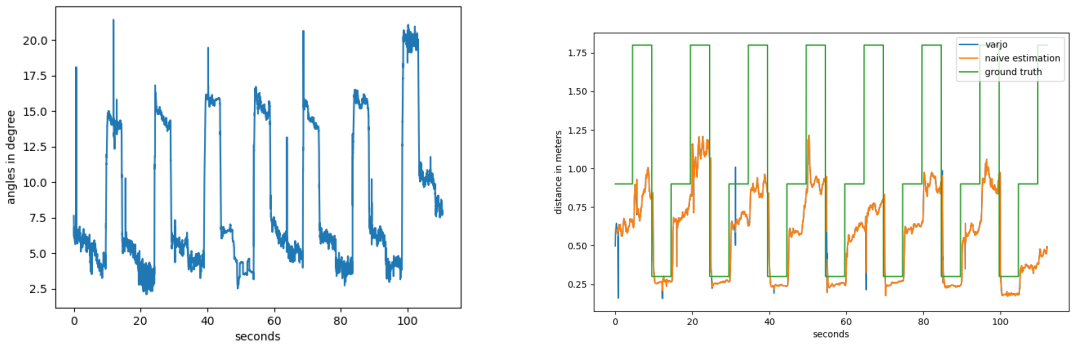
Figure 3.2: Experiment to compare the naive calculation to the values the Varjo API provides. The user has to look at the red, opaque plane.

3.1.2 Depth Experiment

In a simple experiment where a user has to focus on three different depth layers, the depth calculation was compared to the depth estimate, the HMD a Varjo XR-3 [Tec22] provides. Figure 3.2a shows the setup for the experiment. Targets were placed at 30, 90 and 180 cm distance from the user.

Depth layers were set opaque and changed color to red when the user should look at them. The resulting depths can be seen in Figure 3.3b. The naive estimation is nearly the same as the Varjo focus depth, but has some outliers which do not occur in our calculation. And while these estimations are quite accurate at close distances, the differences from the actual object, which the user should focus on, are high at larger distances. Also when focusing on one plane, the focus distance is not always consistent. At some points, there even is a slope from one depth layer to the next, where it is very hard to distinguish the step to the next depth. Sometimes the steps in depth were more clearly visible, other times not so much. An example of this is starting from the 60 seconds mark. There is no distinguishable step in the gaze depth, which makes it hard to see when the user changed the target. At the 80 seconds mark, there is a clear separation, from one depth to the next. Here it is clear when the user switched targets. This ambiguity occurs between the second and the last target, indicating that the difference between 90 and 180 cm is already hard to distinguish. When looking at angles, the same is true as can be seen in Figure 3.3a. It shows that there is a significant drop-off in delta, the change in angle difference if the compared targets are further away. This matches the expected angles. Figure 3.5 shows the exponential drop-off in the angle within the first two meters. The curve is calculated as the angle between the gaze vectors when the gaze rays intersect exactly at the given distance.

Another notable observation was that successive experiment runs varied quite heavily. Another run of the test described before can be seen in Figure 3.4. The first focus period in this run is heavily flickering, and when changing the target, the difference in the estimated depth is nearly indistinguishable. The accuracy within the first 50 cm is still acceptable. We attribute this to differences in how the HMD was worn and the quality of the device calibration. However, it is not clear what the true cause is.



(a) **(b)**

Figure 3.3: a) The angles in the focus point over time in the experiment. b) The corresponding calculated depths by Varjo (blue), our calculation (orange) and the distance of the object the user should look at (green).

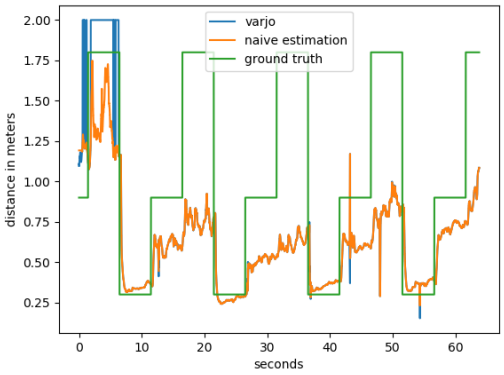


Figure 3.4: A run of the depth estimation experiment with bad calibration.

All these developments resulted in a nearly usable implementation at least in closer areas. A problem that persisted was that looking through layers did not always work and sometimes the gaze point stuttered and flickered between layers.

A common approach to combat flickering is to use fixations or dwell times. Fixations are calculated as averages over a time period to smooth out areas where the user's gaze is dwelling. A good dwell time is considered between 0.2 and 0.8 seconds ([BKKS21; PR21; SG00]).

However, the resulting interaction was not yet satisfactory. Partially because the depth tracking only worked at close distances, and also because calibration and correctly wearing the HMD seemed to affect accuracy. The conclusion was to implement an additional depth calibration that should account for differences in users and current calibration quality. When looking at related work (eg. [LSP+17; WPDH12; WRHS18]), there are two basic approaches to capture the gaze data needed for such a calibration. One approach is similar to regular device eye tracking calibrations, using static points that a user should focus on. The points could lie in multiple depths and in different locations in the field of view. Moreover, there is also a continuous approach where a point is moving through

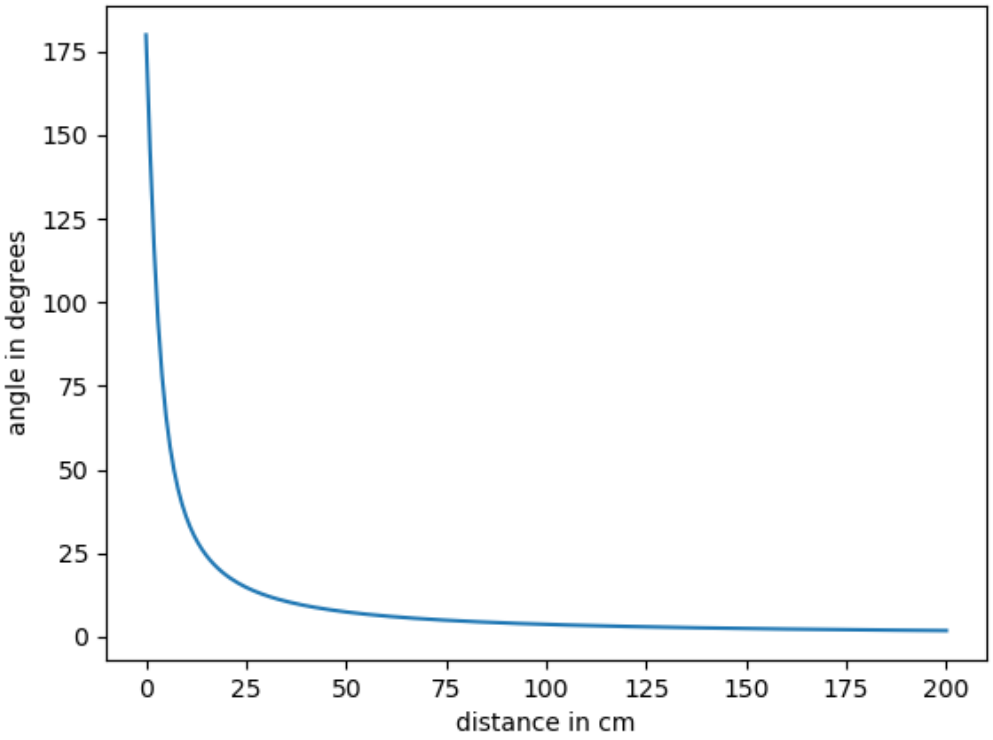


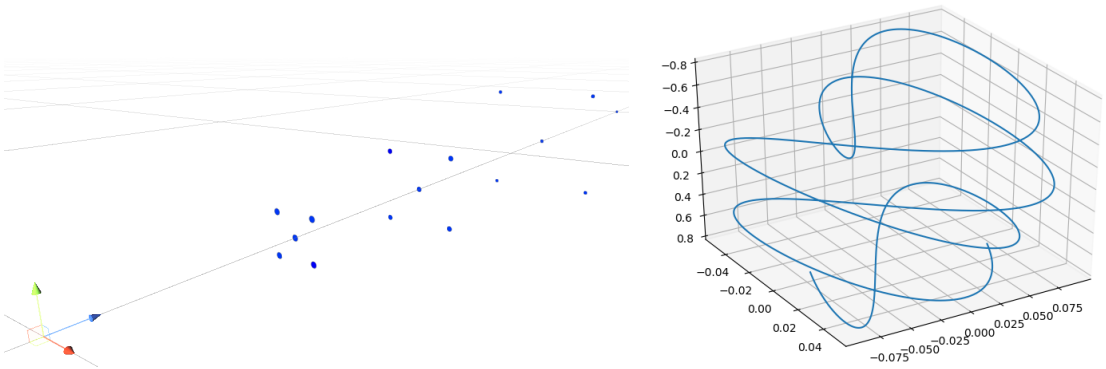
Figure 3.5: A visualization that shows how vergence angle changes with fixation distance. The angle in the first two meters changes exponentially. Here we calculated the angle between the gaze rays in an idealized setting. The interpupillary distance is 6.5 cm and gaze rays intersect at the fixation point.

the sampling space. The data collected during calibration is then used to calculate some models to correct systematic inaccuracies. In this work, we train machine learning methods to estimate the depth of the object a user looks at based on gaze depth. Other modalities include the filters that are used before training, the number of samples, and the machine learning or estimation model.

3.2 Calibration Procedure

Since the raw vergence-based estimate struggles with inaccuracy and differences between users and HMD set-up differences are expected, we developed two calibration procedures to gather training data for machine learning estimators. The estimators can overcome the non-linearity of the gaze depth problem and allow to personalize the estimation for each user. Ideally, the estimators also help to increase accuracy in larger distances.

The two calibration procedures we want to compare are, first, a calibration that is similar to regular eye tracking calibration, in that stationary points are the targets that the user has to focus on. Since the estimation of depth should be improved, the points lie on multiple depth planes. We call this



(a) **(b)**

Figure 3.6: a) Five calibration points in each of the three planes for the stationary "PC" calibration procedure b) The curve used in the second "Wang" calibration. Wang et al. [WPDH12] proposed a calibration using a similar curve.

calibration point cloud (PC) calibration. Figure 3.6a shows the positions of the calibration points. We decided to use four different planes, the first three having five points each and the last one having just one point. The first three should resemble the space where interaction should take place, while the fourth plane could be used to look through all interface elements. The planes were placed at distances that were later used in the study: 0.3, 0.6, 1.2 and 2 meters from the user. The 16 targets are then shuffled and during calibration, each target is shown once for a duration of 3 seconds.

The second procedure uses a curve on which one single target moves along. The curve is similar to the one used by Wang et al. [WPDH12]. We refer to this calibration as 'Wang calibration'. A picture of the calibration curve can be seen in Figure 3.6b. Coordinates of the point $s(t)$ at time step t in seconds are given by the equations 3.6. The parameters are given in Table 3.7. A_x,A_y, A_z denote amplitude, f_x, f_y, f_z frequency in Hz and ϕ_x, ϕ_y, ϕ_z phase angle in radians.

For our calibration, the parameters were adapted in such a way that the curve covers the whole depth from 0.3 to 2 meters. We found that it was not too hard, to follow the point, despite the larger depth covered by the curve. The curve was traversed in 65 seconds.

$$
\begin{aligned}
s_x(t) &= A_x \cos\left(2\pi f_x t + \phi_x\right) \\
s_y(t) &= A_y \sin\left(2\pi f_y t + \phi_y\right) \\
s_z(t) &= A_z \cos\left(2\pi f_z t + \phi_z\right)
\end{aligned}
\tag{3.6}
$$

A_x : 9 cm	f_x : 0.081 Hz	ϕ_x : 0°
A_y : 5 cm	f_y : 0.127 Hz	ϕ_y : 0°
A_z : 80 cm	f_z : 0.025 Hz	ϕ_z : 57°

(3.7)

3.3 Machine Learning Estimation

Machine learning algorithms can be categorized into supervised and unsupervised learning algorithms [Bis06; HTFF09]. In supervised learning, each training sample is assigned a target vector, while unsupervised learning methods do not come with a predefined set of target values. The goal here is to discover underlying structures to group the data or find estimates for density. Problems where the goal is to assign a finite number of labels to the data are called classification. If the output is continuous, it is called a regression problem. In the case of depth estimation, we have a

supervised learning task. The problem of estimating a user's gaze depth, however, can be solved as either a regression or classification problem. If only predefined depth layers are used, classification could simplify training and improve accuracy. However, if continuous depth estimation is needed, e.g. in an unknown environment, regression might be needed. In the case of the Wang calibration, the target values are continuous. In this work, we treat the problem as a regression problem to generalize from the learned samples and allow interaction in the whole space.

Theory and explanations of the algorithms are adapted from Bishop [Bis06] and Hastie et al. [HTFF09].

K Nearest Neighbor Regression

K Nearest Neighbors (KNN) minimizes Euclidean Distance to the k nearest observed samples. The values of the closest neighbors are then averaged. The problem with KNN is that the whole dataset has to be saved and searched every time an estimation is done. While efficient tree algorithms can solve the searching part, large data sets still pose a problem.

Support Vector Regression

Support Vector Machines (SVM) fit separating hyperplanes between classes. A hyperplane should separate classes, maximizing the margin between the plane and the points of both classes. The regular SVM classifier only applies to two classes. However multi-class SVM can be applied by using multiple hyperplanes, separating the problem into multiple binary tasks.

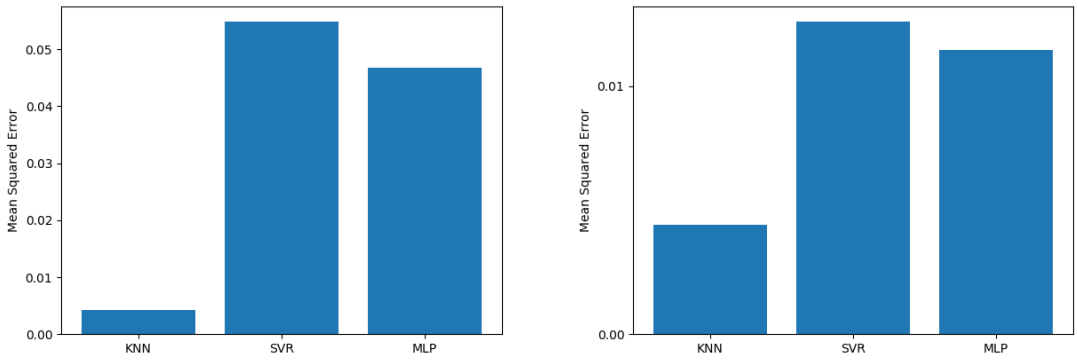
Multi-Layer Perceptron

A Multi-Layer Perceptron is a feed-forward neural network using one or more hidden layers. It tries to find a function mapping from input to output space. It is capable of finding a non-linear function approximation. However, it is susceptible to the impact of random weight initialization. MLPs also require hyper-parameter tuning, including the size and number of hidden layers and iterations.

A first test was run to identify a good starting point for the study. Data was gathered during a calibration run, described in Section 3.2 on one user. The algorithms were trained on 80% of the data gathered and tested on the remaining 20%. The results are depicted in Figure 3.7. Here we can see that the KNN regression worked best with a mean squared error (MSE) below 0.01 m. The errors of SVM and MLP in the PC calibration are more than four times higher than for KNN, while only doubled in the Wang calibration.

3.4 Gaze Depth Interaction

The basic idea to allow interaction is to move a spherical hitbox along the gaze ray, from the user to the gaze point, at the calculated gaze depth. Similar interaction techniques were used before, using a controller to navigate the ray, and control sticks to change the depth (eg. [BPC19; GB06]). Figure 3.8 shows an illustration of the PoR placed at the calculated depth from the user. The first interactions we want to try are simple selection tasks. When the user looks directly at the object it is altered in such a way that new information is displayed. For selection in occlusion disambiguation, either the target or the occluding object changes its opacity so that it is easy to distinguish from one another.



(a) **(b)**

Figure 3.7: Mean squared error for the three algorithms trained and tested on PC (a) and Wang (b) calibration data of one user.

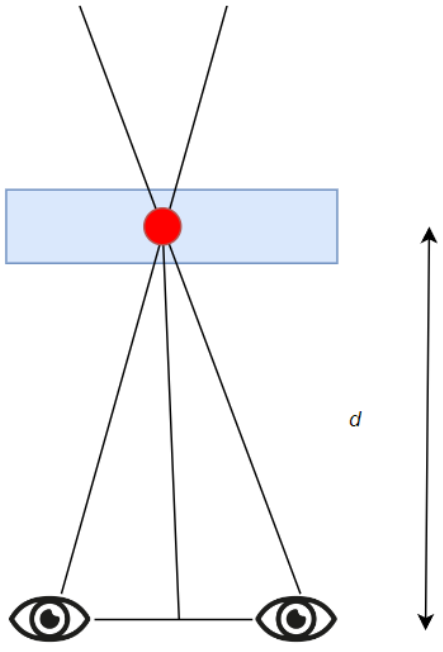


Figure 3.8: The red fixation point is placed at the calculated depth d from the user. Collision of the hit boxes of the gaze point (red) and the object (blue) allows to trigger interaction.

4 Implementation

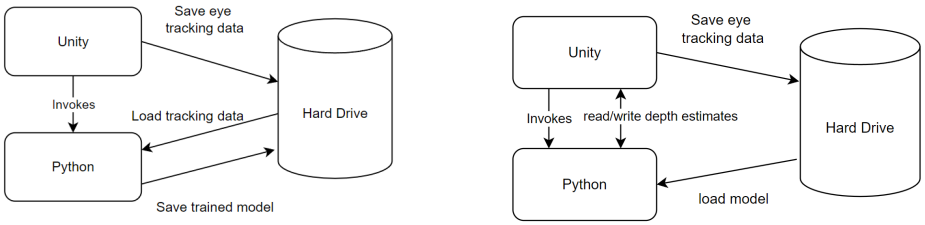
The hardware we use is the Varjo XR-3 [Tec22] which comes with two eye tracking cameras that sample at up to 200Hz. The XR-3 also provides a gaze depth estimation alongside usual measures such as gaze rays for each eye, head and eye positions and a focus stability measure that indicates how well the user focuses on an object. It is also equipped with two low latency 12 megapixel cameras for video pass-through and LiDAR for augmented reality applications. The HMD measures and adapts the lenses to the user's interpupillary distance (IPD). IPDs between 57 mm and 73 mm are supported. The HMD filters the data stream using a "standardfilter" which smoothes the output. Data can be accessed via a Unity XR SDK.

The application was written in C# using the Unity Engine version 2021. Unity provides many useful VR frameworks and libraries such as the XR plug-in which allows for tracking and moving of the user with the HMD. The three ML algorithms were trained in python using the Scikit-learn library [PVG+11]. The Python for Unity plugin provides an API to invoke Python scripts from the C# script and allows Python scripts to interact with the Unity environment.

4.1 Calibration

For both calibration procedures, the implementation looks similar. When the calibration is started, the recording of the eye tracking data, as well as the calibration task, are started. The following data is captured during the recording:

- **Time** The three time measures that are recorded are the timestamp in nanoseconds when the data was captured, the timestamp in milliseconds when the data was logged and a unique frame number.
- **Position and Rotation** For the HMD and both eyes, a position vector is recorded. For the HMD an additional rotation vector is recorded, that indicates the three-dimensional orientation.
- **Forward** A normal vector that indicates the direction of the gaze is captured for both eyes and the combined gaze.
- **Status** The status indicates whether the eye tracking works properly for both eyes and the combined gaze. It signals "INVALID" when an eye can not be captured, if the user is blinking, if eye tracking is not enabled or if the device is not calibrated.
- **Focus Stability** The Focus Stability is a value between 0.0 and 1.0 that specifies how stable the user's focus is. 0.0 means not stable and 1.0 means stable.
- **Target** This value saves the active study or calibration target at the moment of recording.



(a) **(b)**

Figure 4.1: a) Communication of the application framework in the calibration. The Python script is invoked as soon as calibration is finished. b) For interaction, each frame the Python script is invoked to predict the gaze depth.

After the procedure is finished, the training of the machine learning algorithm is started. The Python script loads the data from the CSV file, filters and trains the model. A schematic can be seen in Figure 4.1a. During calibration, up to 10000 samples are recorded that then can be used for training.

4.2 Machine Learning Features and Parameters

Since the goal of this work is to get a purely vergence-based estimation, features such as mean and variance of points around the PoR like Weier et al. [WRHS18] used cannot be utilized here. Instead, the features are the geometrical calculation described before, and the averaged estimated depth of last the 20 gaze depth samples. This provides stability to the algorithm since a combination of the average and current value indicates how big changes in depth in recent times were. Additionally, it smoothes out outliers and flickering to some degree. For training, samples are first filtered. Only samples that have both eyes recorded are used, . In that case, the Varjo API gives the value $VALID$ for both eyes and the eye tracking availability in general. Varjo also provides a "focus stability" measure which indicates how stable the users' focus is. Here, 0.0 indicates unstable and 1.0 indicates a stable focus. For the training of the machine learning algorithms, only samples with 0.9 or higher stability are used. That achieved good results when training and testing on the calibration data of one person in an 80:20 split as, described in Chapter 3.

For KNN, k was set to 5. To optimize search times, the KNN regressor uses a tree-based data structure. For SVM the radial basis function (rbf) kernel was used and the regularization parameter C was set to 1.0 and epsilon to 0.2, which is a parameter that indicates how much error is penalized in the loss function. Regularization allows for some degree of miss classification. Well-chosen regularization parameters tightly fit onto the data but allow the model to generalize. The multi-layer perceptron uses a ReLu activation function and an adam solver. A hidden layer with 250 units worked well. Training of the three models was fast enough to be completed during runtime. Only the training of the multi-layer perceptron with 1000 iterations and 250 hidden units slowed down the program to a perceivable stutter.

4.3 Interaction

To receive the gaze depth estimates, when evoked, the python script reads the gaze depth and the calculated fixation period from the unity environment and predicts the output using the model, that was saved during calibration. The eye tracking data is recorded to a CSV file, the same way as before.

The fixation point as well as other items that should be interactive need a hitbox. For easier use hitboxes slightly larger than the object were chosen. A picture of the colliders can be seen in Figure 4.2. When a collision is detected, the interaction can be triggered. Colliders of user interface elements were set to 10cm in depth. The gaze point's hitbox radius r scales by distance d to the user according to Equation 4.1, accounting for the loss of accuracy in higher depths.

$$r = 1 + d * 10 \tag{4.1}$$

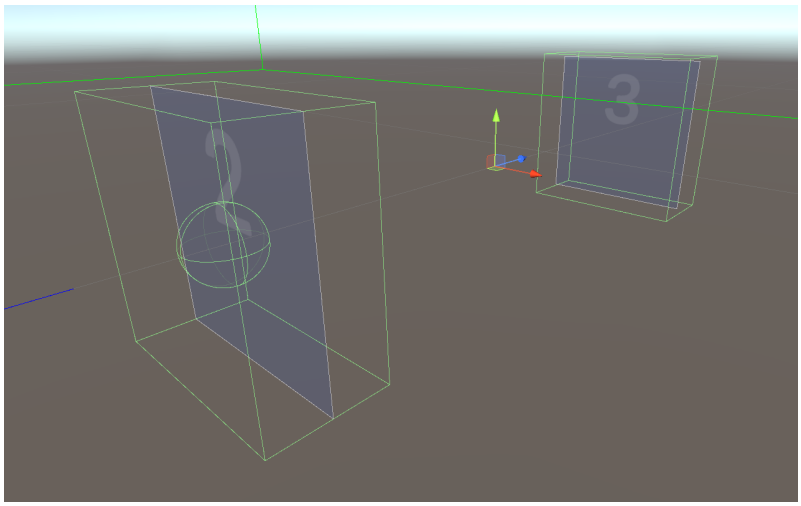


Figure 4.2: The colliders of the interactive objects. The spherical collider is at the position of the focus point.

For the first interaction we implemented simple highlighting when the correct plane is focused. An alpha value of 0.1 for transparent objects made the object visible, but transparent enough to look through to focus on another object, while 0.8 is reasonably opaque and at the same time allows to see the outline of objects behind. A comparison of alpha = 0.1 and alpha = 0.8 can be seen in Figure 4.3. If a collision is detected between the fixation point's collider and the target, it changes its opacity within 0.3 seconds. This value is based on the lower bound of dwell times for gaze interaction reported by Paulus and Remjin [PR21]. The fixation on objects takes between 180 and 330 ms in visual search tasks [Ray09]. In that way, we receive a responsive interaction that leaves some time for errors until the highest opacity value is reached.

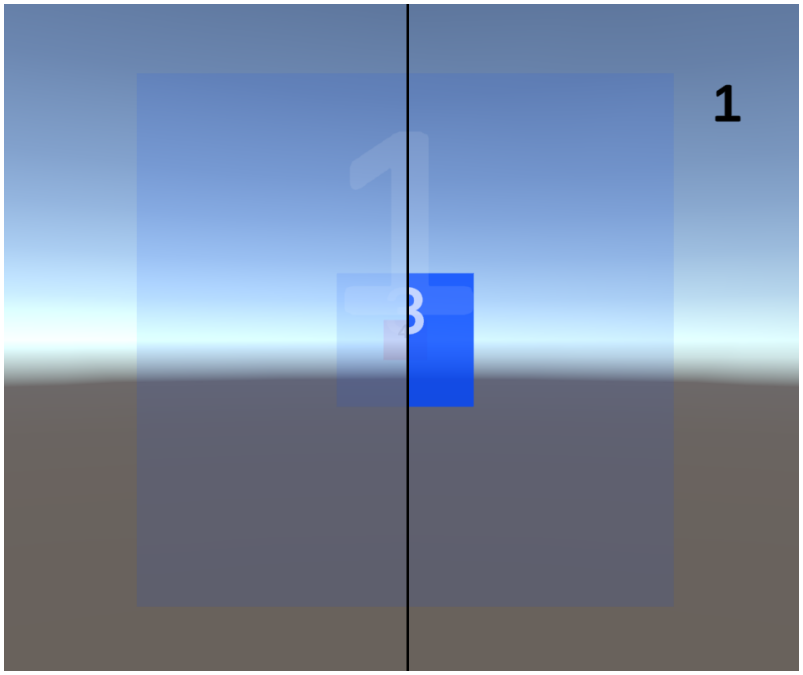


Figure 4.3: Occluded object with an alpha value of 0.1 (left) and 0.8 (right). The outlines of the last object can still be seen in both modes.

5 User Study

5.1 Participants

To validate the findings of the early experiments and to asses how the accuracy and usability of the implementation fares, a pilot study was conducted. Ten (four female and six male) participants aged 21 to 37 were recruited. The mean age was 26 with a standard deviation of 4.5. All participants had an academic background. Most of them were students of computer science or related fields. Since eye tracking was not working with glasses because the reflections of the infrared light inferred with the device, two short-sighted participants completed the study without their glasses. One participant wore contact lenses without negative effects on eye tracking. One participant claimed to have no depth perception.

5.2 Apparatus

The HMD that was used is the Varjo XR-3 [Tec22]. It is equipped with two 1920x1920p Displays achieving a 115° Horizontal field of view. The displays have a refresh rate of 90 Hz. Eye Tracking can be recorded at up to 200 Hz, with sub-degree accuracy. It is a relatively heavy HMD at nearly 980 g (594g HMD + 386g headband). Tracking is done with Steam VR 2.0 and 4 basis stations. The computer that was used had the following specifications: NVIDIA GeForce RTX 3090 GPU, AMD Ryzen 9 5950X 16-Core CPU, 128 GB RAM and running Windows 10 Enterprise. The study setup is depicted in Figure 5.1.

5.3 Procedure

First, participants were asked to read and sign the consent form (see Appendix A.1). Then participants went through the Varjo XR-3 device calibration with five calibration points. A device calibration is performed for the HMD eye tracker to customize for the user, including eye tracking accuracy and interpupillary distance. The Unity scene was started and one of our depth calibration procedures as described in Chapter 3.2 was completed. The calibration each participant started with was randomized to eliminate training effects in the second run.

Each participant then performed a series of six tasks. Tasks were set in a scene with two or three semitransparent, colored squares with numbers one through four on them. The targets were placed at distances of 30, 90, 120 and 200 centimeters from the user. The objective was to look directly at the plane with the number on it that was displayed on the top right of the view. As an additional audio cue, a sound was played to indicate the change. The order of the squares to be focused was randomized. In each task, each square was set as a target two times for a period of five seconds. The

Figure 5.1: PC and HMD set up for the study.

first three tasks displayed two target depth layers while the last three displayed three target layers. At default, the targets had an alpha value of 0.1 and became activated when directly fixating them. When activated, the alpha value changed to 0.8 within 0.3 seconds. A Unity scene showing such a task can be seen in Figure 5.2. The transparency of the target at alpha 0.8 was enough to recognize the outlines of the targets behind.

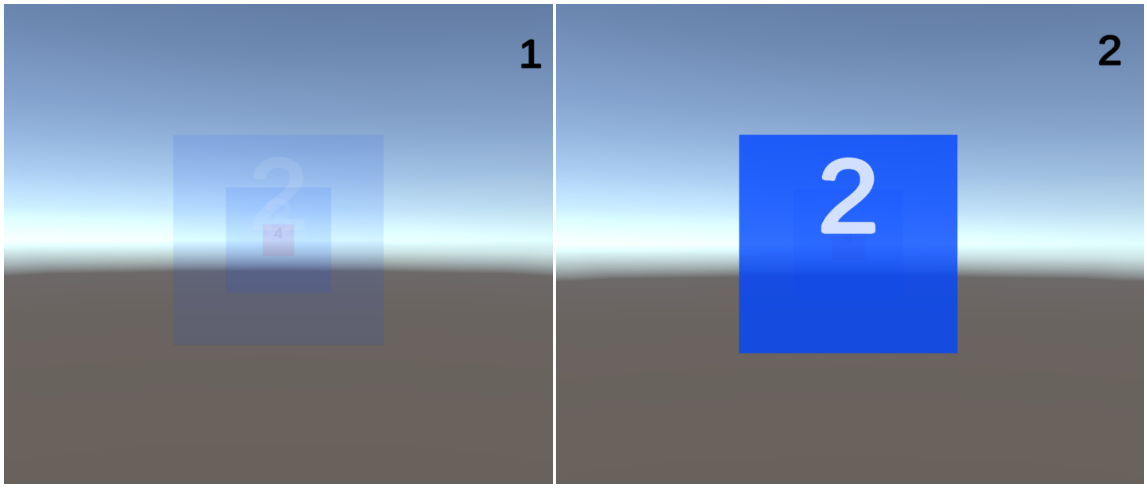
After completing the first round, participants were asked to fill in a digital questionnaire. Questionnaires were created in Google Forms and can be seen in Appendix A.2. The questionnaires were derived from the NASA task load index [Har06] using a seven-point scale while ignoring the temporal demand component. The last four questions concerned the difficulty to activate the different depth levels. A second round of the study was conducted with the other calibration method.

We decided to do the questionnaire outside of VR to eliminate errors when putting on the HMD, and to perform a second, fresh device calibration for each of our calibration procedures. The calibration data was used to train the KNN regression since it was the best in earlier tests (see Section 3.3). Five participants did the PC calibration first and the other five did the Wang calibration first. Participants were recorded, capturing the data that the Varjo API provides to a CSV file.

HMD and surfaces were disinfected after every participant.

When looking at related work and the early experiments, the following hypotheses were formulated:

H1: *Closer objects are easier to focus on and activate than objects farther away.*
The experiment in Chapter 3 shows that farther away targets are harder to focus on because the estimation suffers from the inaccuracy of the vergence-based estimation. We expect this problem to persist through the calibrated estimation.



(a) **(b)**

Figure 5.2: a) One Scene of the Study shows three transparent depth planes and the target indication in the top right. b) If the plane is activated, by directly looking at it, it becomes opaque.

H2: *A Moving target (Wang) calibration is more fatiguing than a stationary target (PC) calibration.* Both prolonged fixations and a large number of long saccades can cause fatigue ([HCRB20]). The long continuous pursuit of the moving target could therefore cause more fatigue than the changing stationary targets.

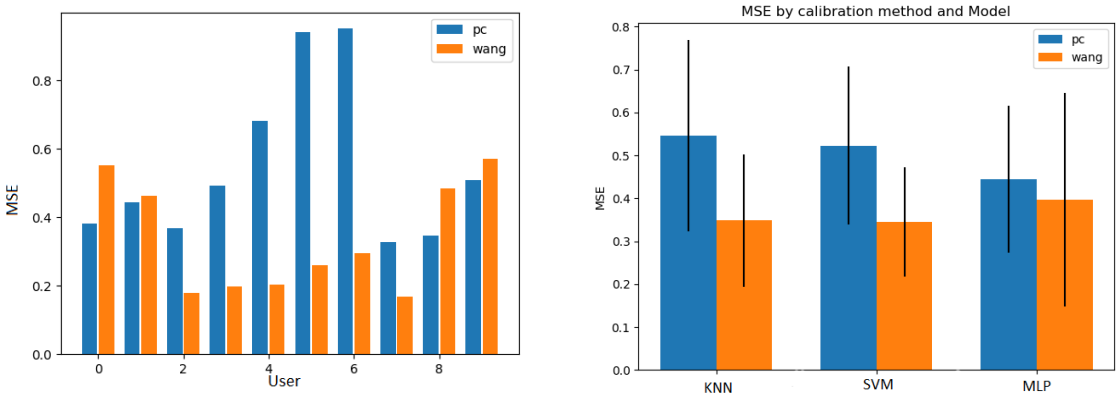
H3: *There are differences in accuracy between scenes with more or fewer depth levels (occlusion).* The targets, occluded by one or more other objects might be harder to focus on. The semi-transparent objects could infer with the user's gaze and cause the Midas Touch problem of unintentionally selecting the wrong object.

6 Results

In this chapter, we analyze the data gathered in the study. All of the ten participants completed the two sessions. There were no cases of motion sickness or nausea. Two Participants reported uncomfortable strain on their eyes. Another two participants commented on the closest targets being too close. They had to squint to focus on that plane, causing higher fatigue. In the following sections, the quantitative results based on eye tracking data, the subjective task load as well as the verbal comments are analyzed.

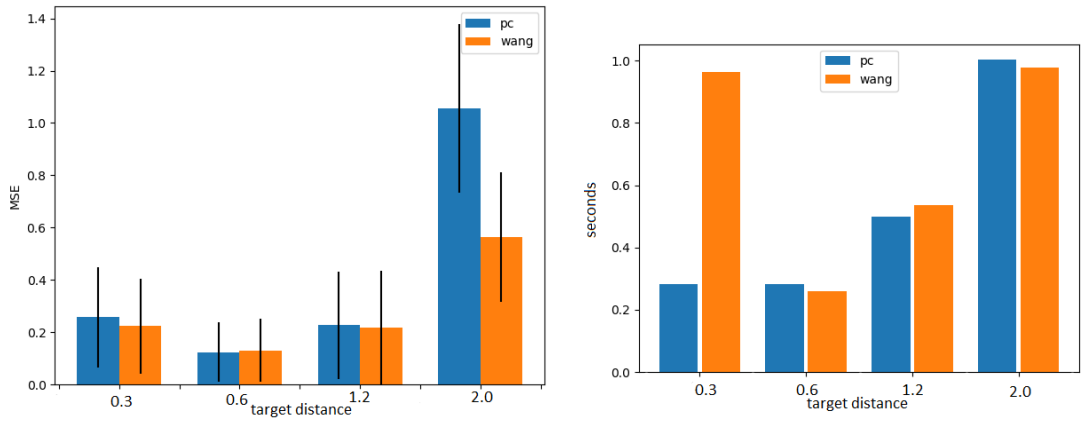
6.1 Quantitative Results

First, we look at how the users performed in the study. In particular, we look at the mean squared error between the calculated depth estimate and the target plane the user should focus on. The MSEs in meters for all users over all six tasks when using the KNN regressor are 0.546 (standard deviation (SD) 0.22) for the PC calibration and 0.348 (SD 0.15) for the Wang calibration, indicating that the Wang calibration leads to better estimations. When looking at each user individually in Figure 6.1a, most participants had lower MSE during the Wang calibration part. However, there are participants that scored better with the point cloud calibration.



(a) **(b)**

Figure 6.1: a) Here, the MSE for each user and both calibration procedures are plotted. Error bars show the deviation of squared errors. b) The MSE over the whole study for each of the ML algorithms and the calibration methods. Error bars indicate the standard deviation of users.



(a) **(b)**

Figure 6.2: a) MSE for each of target depths. Error bars indicate the standard deviation of users. b) Time until first activation of the target. A target counts as activated if the fixation point is close enough to trigger a collision of the hitboxes.

The error was generally higher when the last target should be focused. This is consistent with the earlier experiments. The MSE in meters can be seen in Figure 6.2a. The errors for the first and third target (at 0.3 and 1.2 m) are nearly the same, while the second target (at 0.6 m) lead to the smallest error.

Figure 6.3 shows the errors in each task. A look at the MSE for each task reveals that participants got lower MSE in each successive task. While the Wang calibration achieves lower MSEs, the first task in the Wang calibration is on par with the three target tasks of the PC calibration. The targets in the first task are also the targets with the largest distance between them, while the targets in task three are the closest together.

Another measure indicating how responsive the design is to look at how long a participant took to activate the target after they got the cue. The average time until the first activation occurred can be seen in Figure 6.2b. The calculation takes the size of the hitboxes into account. Participants could activate the first two layers the fastest with the PC calibration. Layers three and four took more time to activate. Using the Wang calibration, the first and last layers took the same time.

The three algorithms' mean squared error for both calibration methods can be seen in Figure 6.1b. Contrary to the initial test, all the algorithms perform similarly with more user data. The SVM (PC: MSE = 0.52, SD = 0.18, Wang: MSE = 0.35, SD = 0.13) performs slightly better than the KNN (PC: MSE = 0.55, SD = 0.22, Wang: MSE = 0.35, SD = 0.15). However, the difference is less than 2 cm for the point cloud calibration and less than 5 cm for the Wang calibration. The MLP performs the best with the point cloud calibration but the worst with the Wang calibration (PC: MSE = 0.44, SD = 0.17, Wang: MSE = 0.4, SD = 0.24).

There is a large difference between participants in how many samples remain after filtering. Some users had around 10000 samples for training, while others had as low as 300. A plot of the MSE and the number of samples used for training the KNN algorithm can be seen in Figure 6.4.

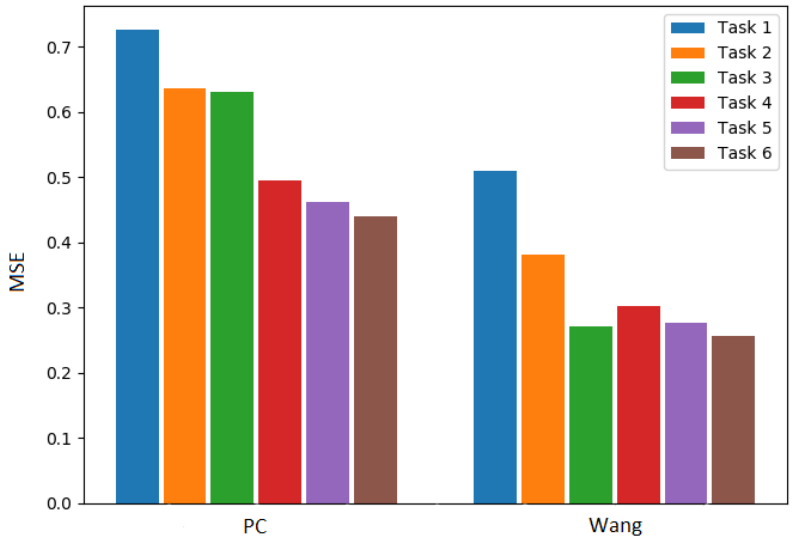


Figure 6.3: MSE in each task and both calibration procedures. The numbers represent the number of the task during the study. The first three tasks used two depth layers to display targets, the latter used three depths.

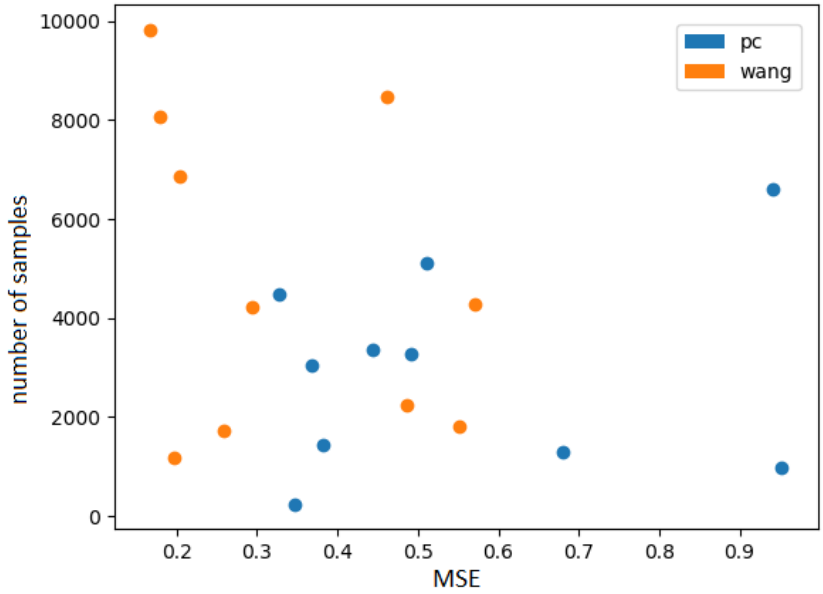


Figure 6.4: MSE and number of samples of each user for both calibration procedures.

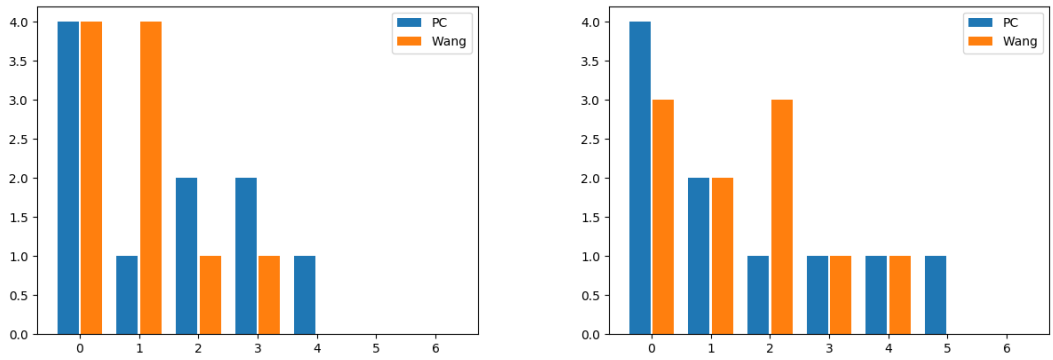


Figure 6.5: Mental (left) and physical (right) demand results after using the corresponding calibration procedures. Bars show how many participants reported the value. One means 'very low' demand. Seven means 'very high' demand.

6.2 Subjective Results

The subjective task load indicates that users perceived the PC calibration part of the study to be slightly more mentally demanding. This can be seen in Figure 6.5, showing that the mean of the mental demand is 2.3 (SD 1.5) for the point cloud calibration and only 1.9 (SD 1.0) for the Wang calibration. In the physical demand, PC achieves an average value of 2.6 (SD 1.8) and Wang 2.5 (SD 1.4). The results of the physical demand question can be seen in Figure 6.5. The other measures from the NASA TLX were similar for both parts of the study (see Appendix A.3). Only the frustration question in the PC calibration was rated higher at a mean value of 3.6 (SD 1.9) and only 2.4 (SD 1.7) for the Wang calibration.

Users, independent of the type of calibration they started with, felt that the second round was more mentally and physically fatiguing. This is expected since continuous concentrated looking and wearing the HMD can cause fatigue.

The subjective difficulty to activate the different depth layers is shown in Figure 6.6 and Figure 6.7. For both of our calibration methods, the first layer was activated the easiest (PC: mean = 2.3, SD = 1.2, Wang: M = 1.9, SD = 1.2), followed by the second (PC: M = 2.4, SD = 1.9, Wang: M = 2.8, SD = 1.9), third (PC: M = 3.7, SD = 2.1, Wang: M = 4.7, SD = 1.8) and fourth layer (PC: M = 5.3 , SD = 2.2, Wang: M = 4.9, SD = 2.3). The Figures show how the difficulty changes with each layer. However, there were users which found activating the last layers easier than the first, while other users said that layers two and three were easiest.

In verbal comments, one user said that the closest plane was uncomfortable to look at. They felt they had to squint to focus the plane which was too fatiguing for regular use. Another user supported this comment, voicing similar concerns. One participant felt that activating layers three and four was nearly impossible. Another participant said that they were not sure if they learned to interact with the activation technique or if they found a spot to look at to activate the target.

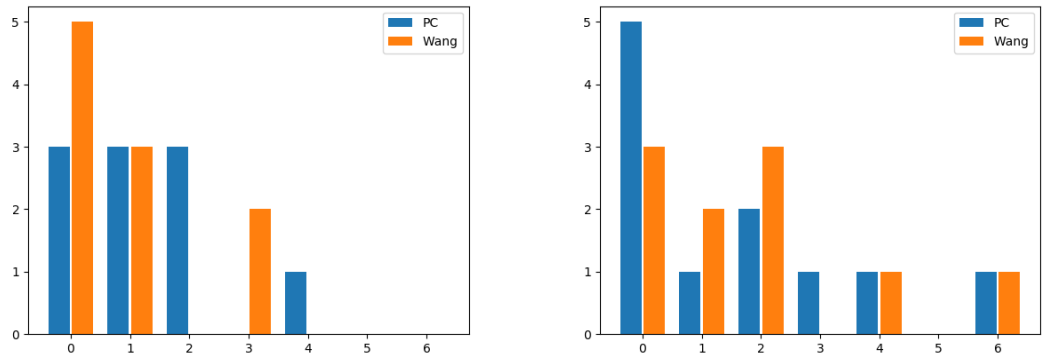


Figure 6.6: Subjective difficulty to activate layer one (left) and two (right). One means 'not at all' difficult. Seven means 'very difficult'.

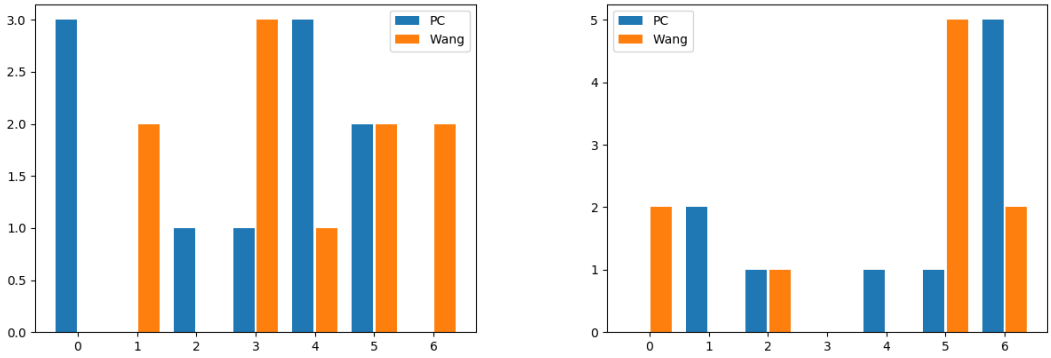


Figure 6.7: Subjective difficulty to activate layer three (left) and four (right). One means 'not at all' difficult. Seven means 'very difficult'.

7 Discussion

In the results we have seen that the Wang calibration results in a better estimation model, while also being less frustrating. The time it took users to activate the closest layer was slower with the Wang calibration than with the PC calibration. Since this was not reflected in the difficulty to activate the first layer, the Wang calibration provides an overall better calibration and interaction.

The fact that users achieve lower MSE in every successive task can have multiple reasons. Either the increased depth complexity does not impact the accuracy, or the learning effect counteracts the occlusion and users learn where to look. The tasks are also ordered by the distance between the targets. The first task only uses targets one and four, which are 1.7 meters apart. Part of the error could be explained by the larger distance the user has to adapt to. Another possibility is that the additional objects act as depth guidance showing the user, where they are looking. If a wrong plane gets activated, the participant might try harder to avoid those mistakes. It was discovered before that users can learn to converge their eyes on purpose (e.g. [HGG+19]). However, we did not expect the learning effects to appear so quickly.

When using the Wang calibration users take longer to activate the target in the first layer. This might stem from the low amount of samples gathered at a very close distance from the user during the Wang calibration. The number of samples collected in each depth is not distributed equally in the Wang calibration. This is the case since the curve reaches the nearest point to the user only once. The samples gathered might not be enough to correctly estimate the closest plane, resulting in a slower activation time. In the PC calibration, this is only the case for the last layer since there are five targets in every other depth layer during calibration.

The reason why layers one or four appear easier for some users might be the sampling space. If there are no more samples beyond the target, there is a hard edge in the sampling space where the algorithm would otherwise find samples to compare to. Depending on the algorithm generalization, beyond these samples yields different results.

When looking to compare our results to related work, it is quite hard to interpret the results of other studies, in terms of errors and distance to the user. Additionally in this study, we gather calibration data and then test on a completely new set of scenes, which is closer to an 'In the Wild' experiment. Weier et al. [WRHS18] show multiple results as part of their work, including methods proposed by Wang et al. [WPDH12]. In the first two meters, which is the interval tested in our study, the method proposed by Weier et al. achieves an MSE below 0.025. Their reported MSE over all scenes is closer to our results at around 0.2 m. While we cannot improve on the remarkable results of their combined feature set, our proposed method performs similar to the vergence-based method by Wang et al.. The method used by Lee et al. [LSP+17] using a multi-layer perceptron, ranges from errors of 0.2m to over 1 m based on user averaging at 0.42 m. However, it is hard to say, how the accuracy changed with distance.

Looking at the hypotheses formulated before, we can summarize some preliminary results.

H1: *Closer objects are easier to focus on and activate than objects farther away.*
While we can generally say that this is true, there were discrepancies between users. Most participants found it easiest and achieved the best accuracy focusing on closer targets. However, some users preferred layers in between. This may be a result of the first layer being too close to comfortably focus on. The angle one's eyes have to converge to is larger. More muscle movement is needed to rotate the eyes when looking at the close object. This also depends on the user's physiology to some degree. Larger IPD results in a wider angle at the same distance compared to a smaller IPD.

H2: *A moving target (Wang) calibration is more fatiguing than a stationary target (PC) calibration.*
Users did not report more fatigue after the Wang calibration part of the study. This could be the result of multiple causes. On the one hand, the Wang calibration produced a more accurate model, possibly resulting in better interaction and thus less stress and insecurity. The frustration measure of the NASA TLX supports this hypothesis. Participants reported a mean value of 3.6 (SD 1.9) for the PC calibration and only 2.4 (SD 1.7) after the Wang calibration. On the other hand, because the order is random, the targets shown during PC calibration sometimes jump from the closest to the farthest plane, making the vergence movement harder than the continuous movement. This, combined with the short time to focus on them (3 seconds), could be another problem.

H3: *There are differences in accuracy between scenes with more or fewer depth levels (occlusion).*
While we can see that there are differences, the results show lower errors in scenes with three targets than in scenes with just two targets. There might be learning effects, that enable users to more efficiently focus on the right spots in every successive task. It is also possible that the additional objects provide a guidance, indicating when a participant looks at a wrong layer. However, we also see a higher MSE when trying to focus on targets farther back which tend to be occluded by other targets. The effect of occlusion can therefore not be pinned down exactly. To investigate this effect further, more focused data is needed.

7.1 Limitations and Future Work

The proposed design as well as the presented study have some limitations. First, due to the limited number of participants tested in this study, we did not test the statistical significance of the results. The participants tested in our study were in a limited range of age and most of them were students or researchers with varying degrees of experience in VR. More precisely, most of them were students of computer science or related fields. Therefore the findings might not generalize well to the general population. Another limitation is that targets used during the PC calibration were shuffled randomly. A predefined set of targets would lead to more uniformity between users. Since we restricted ourselves to vergence-based estimation only, the feature space was limited. Further exploring the feature selection, e.g. taking velocity or acceleration into account, might provide additional benefits.

There is a large difference in how many samples were used for training for each user. While this did not seem to correlate with the achieved accuracy, a controlled calibration that reaches a fixed number of valid samples for each user could improve the estimation. This would result in some users spending more time in calibration, which would results in other problems e.g. fatigue.

The hitboxes were chosen based on the experiment described in Chapter 3. While these hitbox choices worked well, a more careful choice, for example based on the expected error at the target depth, might improve the interaction. The sizes and scaling yielded good results in terms of responsiveness. However, carefully testing the colliders could lead to a better user experience, and also allow for more objects to be placed closer together. Additionally, experimenting with how tightly the objects can be placed in a given depth range could lead to interesting applications. Finally, a comparative study that investigates how a gaze depth-based interaction or selection technique compares to conventional techniques e.g. selection with a controller could highlight strengths and weaknesses.

8 Conclusion

In this Master thesis, we presented a vergence-based gaze depth estimation method using the eye trackers of the Varjo XR-3. Data used to train the machine learning estimators was gathered in two different calibration procedures. An interaction technique was implemented and the estimation and interaction were tested in a user study. We compared different modalities in calibration, training and interaction methods. Results show that interaction with occluded objects via our proposed method works well within 1.2 meters distance from the user.
Users learned fast where to look and how to activate the different interactive objects. Our gaze depth estimation performs at least on par with previous vergence-based methods in terms of accuracy in an interactive environment. However, the developed calibration procedures take less time and training can be completed during runtime, offering fast customization similar to conventional HMD eye tracking calibrations. The continuous Wang calibration outperforms the stationary PC calibration. The continuous sampling of the interactive space has shown to result in a more responsive and accurate model. The choice of machine learning algorithm comes down to requirements. KNN has no training time but it has drawbacks in terms of prediction time and requires saving the whole training set, the SVM and MLP need more time in training but offer faster predictions and less space complexity.

Sampling the interaction space continuously with a moving target during calibration showed improved accuracy compared to stationary calibration targets. We did not expect that participants would feel less fatigue and frustration when having to focus on a moving target. However, the resulting calibration being more responsive in the actual study tasks probably plays a role in their assessment. As we expected, there were large differences between users in terms of accuracy and also in target preferences. Closer targets were easier to interact with and reliably activated and deactivated based on the user's gaze for most users, while some users reported the last layer being the easiest to activate.

There are many use-cases that are possible with the proposed estimation technique. Improving the accuracy of gaze depth estimation can improve the capabilities of gaze contingent rendering. More accurate tracking also improves the visualization of 3D gaze data, which makes it more usable as a tool similar to 2D eye tracking. The activation of the semitransparent objects worked well and could provide an interesting tool for the design of interactive environments. The vergence-based approach also allows to use this technique not only in VR but in any environment, where stereoscopic eye tracking is available. Turning any transparent surface like mirrors or windows in virtual and augmented environments into UI elements could prove useful, when hands-free interaction is required. HUDs in everyday AR applications such as driving and gaming could be equipped with the proposed interaction technique to allow the user to look through an otherwise opaque UI element.

Bibliography

[ASP+21] S. Ahn, S. Santosa, M. Parent, D. Wigdor, T. Grossman, M. Giordano. "StickyPie: A Gaze-Based, Scale-Invariant Marking Menu Optimized for AR/VR". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–16. DOI: 10.1145/3411764.3445297 (cit. on p. 15).

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738 (cit. on pp. 22, 23).

[BKKS21] M. Barz, S. Kapp, J. Kuhn, D. Sonntag. "Automatic Recognition and Augmentation of Attended Objects in Real-time using Eye Tracking and a Head-mounted Display". In: *ACM Symposium on Eye Tracking Research and Applications*. 2021, pp. 1–4. DOI: 10.1145/3450341.3458766 (cit. on pp. 15, 20).

[Bou22] P. Bourke. *Points, Lines, and Planes*. Website. 2022. URL: http://paulbourke.net/geometry/pointlineplane/ (cit. on pp. 17, 18).

[BPC19] M. Baloup, T. Pietrzak, G. Casiez. "RayCursor". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12. DOI: 10.1145/3290605.3300331 (cit. on p. 23).

[DHG+14] A. T. Duchowski, D. H. House, J. Gestring, R. Congdon, L. Świrski, N. A. Dodgson, K. Krejtz, I. Krejtz. "Comparing estimated gaze depth in virtual and physical environments". In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. 2014, pp. 103–110. DOI: 10.1145/2578153.2578168 (cit. on pp. 10, 14).

[Duc02] A. T. Duchowski. "A breadth-first survey of eye-tracking applications". In: *Behavior Research Methods, Instruments, & Computers* 34.4 (2002), pp. 455–470. DOI: 10.3758/bf03195475 (cit. on p. 9).

[Duc18] A. T. Duchowski. "Gaze-based interaction: A 30 year retrospective". In: *Computers & Graphics* 73 (2018), pp. 59–69. DOI: 10.1016/j.cag.2018.04.002 (cit. on p. 13).

[FWT+17] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, M. R. Morris. "Toward Everyday Gaze Input". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 1118–1130. DOI: 10.1145/3025453.3025599 (cit. on pp. 10, 13).

[GB06] T. Grossman, R. Balakrishnan. "The design and evaluation of selection techniques for 3D volumetric displays". In: *Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06*. 2006, pp. 3–12. DOI: 10.1145/1166253.1166257 (cit. on p. 23).

[Har06] S. G. Hart. "Nasa-Task Load Index (NASATLX) 20 Years Later". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50.9 (2006), pp. 904–908. DOI: 10.1177/154193120605000909 (cit. on p. 30).

[HCRB20] T. Hirzle, M. Cordts, E. Rukzio, A. Bulling. "A Survey of Digital Eye Strain in Gaze-Based Interactive Systems". In: *ACM Symposium on Eye Tracking Research and Applications*. 2020, pp. 1–12. DOI: 10.1145/3379155.3391313 (cit. on p. 31).

[HGG+19] T. Hirzle, J. Gugenheimer, F. Geiselhart, A. Bulling, E. Rukzio. "A Design Space for Gaze Interaction on Head-mounted Displays". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12. DOI: 10.1145/3290605.3300855 (cit. on pp. 14, 15, 39).

[HNA+11] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, J. van de Weijer. *Eye Tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011. ISBN: 9780191625428. URL: https://books.google.de/books?id=5rIDPV1EoLUC (cit. on p. 13).

[HTFF09] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. New York: Springer, 2009. ISBN: 978-0-387-84858-7 (cit. on pp. 22, 23).

[Jac90] R. J. K. Jacob. "What you look at is what you get: eye movement-based interaction techniques". In: *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people CHI 90*. 1990, pp. 11–18. DOI: 10.1145/97243.97246 (cit. on p. 15).

[KSB08] Y. Kammerer, K. Scheiter, W. Beinhauer. "Looking my way through the menu". In: *Proceedings of the 2008 symposium on Eye tracking research and applications - ETRA 08*. 2008, pp. 213–220. DOI: 10.1145/1344471.1344522 (cit. on p. 15).

[KSCC18] T.-S. Kuo, K.-T. Shih, S.-L. Chung, H. H. Chen. "Depth from Gaze". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 2910–2914. DOI: 10.1109/icip.2018.8451156 (cit. on p. 13).

[Lin20] P. Linton. "Does vision extract absolute distance from vergence?" In: *Attention, Perception, and Psychophysics* 82.6 (2020), pp. 3176–3195. DOI: 10.3758/s13414-020-02006-1 (cit. on p. 14).

[LSP+17] Y. Lee, C. Shin, A. Plopski, Y. Itoh, T. Piumsomboon, A. Dey, G. Lee, S. Kim, M. Billinghurst. "Estimating Gaze Depth Using Multi-Layer Perceptron". In: *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*. 2017, pp. 26–29. DOI: 10.1109/isuvr.2017.13 (cit. on pp. 14, 20, 39).

[MBWK16] E. G. Mlot, H. Bahmani, S. Wahl, E. Kasneci. "3D Gaze Estimation using Eye Vergence". In: *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies*. 2016, pp. 125–131. DOI: 10.5220/0005821201250131 (cit. on p. 14).

[NAHW12] M. Nyström, R. Andersson, K. Holmqvist, J. van de Weijer. "The influence of calibration method and eye physiology on eyetracking data quality". In: *Behavior Research Methods* 45.1 (2012), pp. 272–288. DOI: 10.3758/s13428-012-0247-4 (cit. on p. 13).

[ORB+20] S. Oney, N. Rodrigues, M. Becher, T. Ertl, G. Reina, M. Sedlmair, D. Weiskopf. "Evaluation of Gaze Depth Estimation from Eye Tracking in Augmented Reality". In: *ACM Symposium on Eye Tracking Research and Applications*. 2020, pp. 1–5. DOI: 10.1145/3379156.3391835 (cit. on p. 14).

[POVK16] Y. S. Pai, B. Outram, N. Vontin, K. Kunze. "Transparent Reality". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 2016, pp. 171–172. DOI: 10.1145/2984751.2984754 (cit. on p. 15).

[PR14] T. Pfeiffer, P. Renner. "EyeSee3D". In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. 2014, pp. 195–202. DOI: 10.1145/2578153.2578183 (cit. on p. 14).

[PR21] Y. T. Paulus, G. B. Remijn. "Usability of various dwell times for eye-gaze-based object selection with eye tracking". In: *Displays* 67 (2021), p. 101997. DOI: 10.1016/j.displa.2021.101997 (cit. on pp. 15, 20, 27).

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 25).

[PVT+13] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, H. Gellersen. "Pursuit calibration". In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 2013, pp. 261–270. DOI: 10.1145/2501988.2501998 (cit. on p. 13).

[RARH20] A. Riegler, B. Aksoy, A. Riener, C. Holzmann. "Gaze-based Interaction with Windshield Displays for Automated Driving: Impact of Dwell Time and Feedback Design on Task Performance and Subjective Workload". In: *12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 2020, pp. 151–160. DOI: 10.1145/3409120.3410654 (cit. on p. 15).

[Ray09] K. Rayner. "The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search". In: *Quarterly Journal of Experimental Psychology* 62.8 (2009), pp. 1457–1506. DOI: 10.1080/17470210902816461 (cit. on p. 27).

[RWH+17] T. Roth, M. Weier, A. Hinkenjann, Y. Li, P. Slusallek. "A Quality-Centered Analysis of Eye Tracking Data in Foveated Rendering". In: *Journal of Eye Movement Research* 10.5 (2017). DOI: 10.16910/jemr.10.5.2 (cit. on p. 9).

[SBBW17] C. Schulz, M. Burch, F. Beck, D. Weiskopf. "Visual Data Cleansing of Low-Level Eye-Tracking Data". In: *Eye Tracking and Visualization*. Springer International Publishing, 2017, pp. 199–216. DOI: 10.1007/978-3-319-47024-5_12 (cit. on p. 13).

[SG00] D. D. Salvucci, J. H. Goldberg. "Identifying fixations and saccades in eye-tracking protocols". In: *Proceedings of the symposium on Eye tracking research and applications - ETRA 00*. 2000, pp. 71–78. DOI: 10.1145/355017.355028 (cit. on p. 20).

[Sil07] E. Silversmith. *Focus in an eye*. 2007. URL: https://commons.wikimedia.org/wiki/File:Focus_in_an_eye.svg (cit. on p. 10).

[Špa12] O. Špakov. "Comparison of eye movement filters used in HCI". In: *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12*. 2012. DOI: 10.1145/2168556.2168616 (cit. on p. 13).

[Tec22] V. Technologies. *Varjo*. 2022. URL: https://varjo.com/ (cit. on pp. 19, 25, 29).

[Tob22] Tobii. *Clean UI*. Website. 2022. URL: https://developer.tobii.com/pc-gaming/design-guidelines/explored-features/clean-ui/ (cit. on p. 15).

[VNL14] M. Vidal, D. H. Nguyen, K. Lyons. “Looking at or through?” In: *Proceedings of the 2014 ACM International Symposium on Wearable Computers*. 2014, pp. 87–90. DOI: 10.1145/2634317.2634344 (cit. on p. 15).

[WP08] M. Wedel, R. Pieters. “A Review of Eye-Tracking Research in Marketing”. In: *Review of Marketing Research*. Emerald Group Publishing Limited, 2008, pp. 123–147. DOI: 10.1108/s1548-6435(2008)0000004009 (cit. on p. 9).

[WPDH12] R. I. Wang, B. Pelfrey, A. T. Duchowski, D. H. House. “Online Gaze Disparity via Bioncular Eye Tracking on Stereoscopic Displays”. In: *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. 2012, pp. 184–191. DOI: 10.1109/3dimpvt.2012.37 (cit. on pp. 14, 15, 20, 22, 39).

[WRHS18] M. Weier, T. Roth, A. Hinkenjann, P. Slusallek. “Predicting the gaze depth in head-mounted displays using multiple feature regression”. In: *Proceedings of the 2018 ACM Symposium on Eye Tracking Research and Applications*. 2018, pp. 1–9. DOI: 10.1145/3204493.3204547 (cit. on pp. 9, 14, 15, 20, 26, 39).

All links were last followed on July 15, 2022.

A Study Forms

A.1 Consent Form

University of Stuttgart
Germany

University of Stuttgart • VISUS • Allmandring 19 • 70569 Stuttgart • Germany

Visualization Research Center (VISUS)

Directors
Prof. Dr. Thomas Ertl
Prof. Dr. Daniel Weiskopf

Contact

Sergej Geringer

Allmandring 19
70569 Stuttgart
Germany
T 0711 685-88630
Sergej.Geringer@visus.uni-stuttgart.de

Kuno Kurzhals

Allmandring 19
70569 Stuttgart
Germany
T 0711 685-88609
Kuno.Kurzhals@visus.uni-stuttgart.de

www.visus.uni-stuttgart.de
www.twitter.com/vis_visus

28.05.2022

Information and declaration of consent for the participation in the research study "Evaluation and Application of Estimated Gaze Depth in Virtual Reality"

Dear prospective participant,

We would like to invite you to participate in the following study. In particular, we analyze the perception and processing of information, as well as the user interaction, using different visualization techniques in a virtual reality environment. In the current study, we are investigating the use of estimated gaze depth in virtual reality applications. We are interested in the accuracy of gaze depth estimation and the possible use cases regarding visualization and interaction.

Study procedure

1. Calibration of the Varjo XR-3.
2. You will perform a series of tasks, where you have to look at highlighted objects in virtual space, starting with Gaze Depth calibration. You are asked to perform several navigation tasks.
3. You will be asked to evaluate your experience after each task.
4. After the first series is done, you are asked to fill in a questionnaire.
5. Steps 2 – 4 will be repeated with another calibration method.
6. After the second series is done, you are asked to fill in another questionnaire.

We collect the following data during the study:

- Your head movement in the virtual scene.
- Your gaze behavior, i.e., what you look at in the virtual reality environment.
- Your controller and gesture input.
- The time needed to complete individual task.

Please note:

- Because the Eye-Tracker records every eye movement, gaze data that you do not want to disclose can be recorded.
- No images or videos of either the pupils or the face will be stored.

Bank
Baden-Württembergische Bank
Stuttgart – BW-Bank

IBAN
DE51 6005 0101 7871 5216 87

SWIFT/BIC
SOLADEST600

Umsatzsteuer-IdNr.
DE147794196

University of Stuttgart
Germany

Visualization Research
Center (VISUS)

General Conditions of Participations

- You have no visual impairment (short or long-sightedness not included).
- You are 18 years of age or older.

Risks

- Increased physical strain due to the weight of the AR/VR headset and gesture-based interaction
- Use of a VR headset can sometimes cause motion sickness with symptoms such as nausea. Please inform the research staff if you feel unwell at any time during the study.
- When using an AR headset, users are sometimes heavily distracted by virtual objects, increasing the risk of tripping over, or colliding with real objects. Research staff will always be present during the study to assist participants and watch their surroundings.

University of Stuttgart
Germany

Visualization Research Center (VISUS)

Privacy Information (Article 13 GDPR) regarding the collection of data in the study "Evaluation and Application of Estimated Gaze Depth in Virtual Reality" of the Visualization Research Center of the University of Stuttgart (VISUS)

Responsible body under data protection laws

University of Stuttgart
Keplerstraße 7
70174 Stuttgart
Germany
Phone: +49 711/685-0
E-Mail: poststelle@uni-stuttgart.de

Data protection officer
University of Stuttgart
Data protection officer
Breitscheidstr. 2
70174 Stuttgart
Tel: +49 711 685-83687
Fax: +49 711 685-83688
E-Mail: datenschutz@uni-stuttgart.de

Legal Basis

1. Conduction of the survey as part of a research project

   Art. 6 Paragraph. 1 lit. e in conjunction with Art. 6 Paragraph. 3 General Data Protection Regulation (GDPR) and in conjunction with §13 Abs.1 Landesdatenschutzgesetz Baden-Württemberg.

   Art. 6 Paragraph. 1 lit. c GDPR in conjunction with §70, §75 Landeshaushaltsordnung Baden-Württemberg.

   Art. 17 Paragraph. 3 lit. d GDPR

2. Optional Agreement to further usage Art. 6 Paragraph. 1 lit. a GDPR

Data Recipients

The datasets collected during the study are available only in anonymous form that cannot be linked to a specific person. The recorded data is processed and evaluated statistically to be published in scientific journals or conference proceedings.

- Evaluated research data: Worldwide readers / users of scientific publications.
- Raw data within a repository: Users that have been permitted to use the data within the university and the provider of the repository within the university. For reviewing processes of scientific publications, the raw data could be passed on to the reviewers and the publisher (but is then subject to ethics guidelines and confidentiality requirements).

- The data above can potentially also be processed outside the EU in countries, where there are no comparable data protection laws. This can mean potential restrictions of your rights.
- For receiving your compensation in cash, you must sign a receipt and provide your address and full name. The internal accounting of the university will process this receipt.
- Based on policies the university archive must be consulted before deletion of data. The archive then decides on whether or not to keep the data.

Duration of the Storage Period

All research data is stored until 10 years after the completion of the research project. Potentially, the concerned data will be transferred to the respective university archive, which can store it indefinitely.

Your rights

No associations between data and the participant's identity are stored. It is not possible to delete a participant's data after the study has ended and the results of the data evaluation have been published in a scientific journal or conference proceedings, since by doing so, published results could no longer be reproduced and are potentially invalidated.

Your participation in this study is voluntary. By giving your informed consent you are under no obligations. You may revoke your consent at any time without any legal consequences. You may abort the study at any time without giving reasons. Doing so will not result in any legal consequences for you. If you decide to abort the study, you may no longer be entitled to receive your compensation.

You have the right to complain to the supervisory authority, should you be of the opinion that the processing of the personal data relating to you breaches legal regulations.

The competent supervisory authority is the State Data Protection and Freedom of Information Officer of Baden-Württemberg: Landesbeauftragte für den Datenschutz und die Informationsfreiheit Baden-Württemberg

University of Stuttgart
Germany

Visualization Research Center (VISUS)

Declaration of informed consent

- I have read or have been read to the preceding explanation and understood it.
- I have been informed about the study by the research staff and all my questions have been answered to my satisfaction.
- I have been informed about possible risks associated with the use of VR/AR headsets.
- I volunteer to participate in this study, and I am aware of the fact that I can discontinue my participation at any time.
- The agreement required for participation is entirely voluntary. Not participating does not result in any kind of disadvantage.
- I was given sufficient time to make a decision about participating in the study.
- I have read the privacy information and agree to it.
- I have been informed that the obtained data is saved and processed on computers that are connected to the internet.
- I have been informed that the obtained data cannot be deleted after my participation in the study is concluded.
- I have received a copy of the information sheets.

_____
Last name, first name

_____
Place, date, and signature of participant

_____
Date and signature of experimenter

Thank you for your participation!

A.2 Questionnaire

Are you?

- ◯ Female
- ◯ Male
- ◯ Other

How old are You?

Meine Antwort

Do you have any visual impairment?

- ☐ No
- ☐ Sonstiges:

How mentally demanding was the task? *

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

How physically demanding was the task? *

	1	2	3	4	5	6	7	
Very Low	◯	◯	◯	◯	◯	◯	◯	Very High

How successful were you in accomplishing what you were asked to do? *

	1	2	3	4	5	6	7	
Failure	◯	◯	◯	◯	◯	◯	◯	Perfect

How hard did you have to work to accomplishyour level of performance? *

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

How insecure, discouraged, irritated, stressed, and annoyed were you? *

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

How difficult was it to activate depth layer 1? *

	1	2	3	4	5	6	7	
Not at all	○	○	○	○	○	○	○	Very Difficult

How difficult was it to activate depth layer 2? *

	1	2	3	4	5	6	7	
Not at all	○	○	○	○	○	○	○	Very Difficult

How difficult was it to activate depth layer 3? *

	1	2	3	4	5	6	7	
Not at all	○	○	○	○	○	○	○	Very Difficult

How difficult was it to activate depth layer 4? *

	1	2	3	4	5	6	7	
Not at all	○	○	○	○	○	○	○	Very Difficult

How mentally demanding was the task? *

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

How physically demanding was the task? *

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

How successful were you in accomplishing what you were asked to do? *

	1	2	3	4	5	6	7	
Failure	○	○	○	○	○	○	○	Perfect

How hard did you have to work to accomplishyour level of performance? *

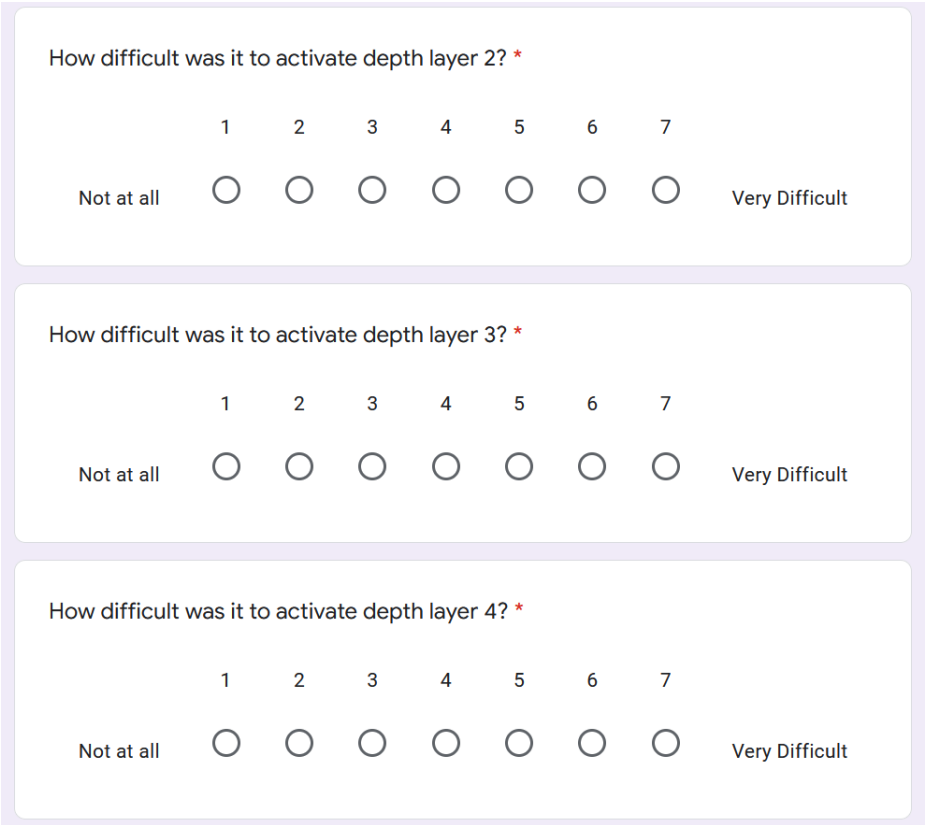
	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

How insecure, discouraged, irritated, stressed, and annoyed were you? *

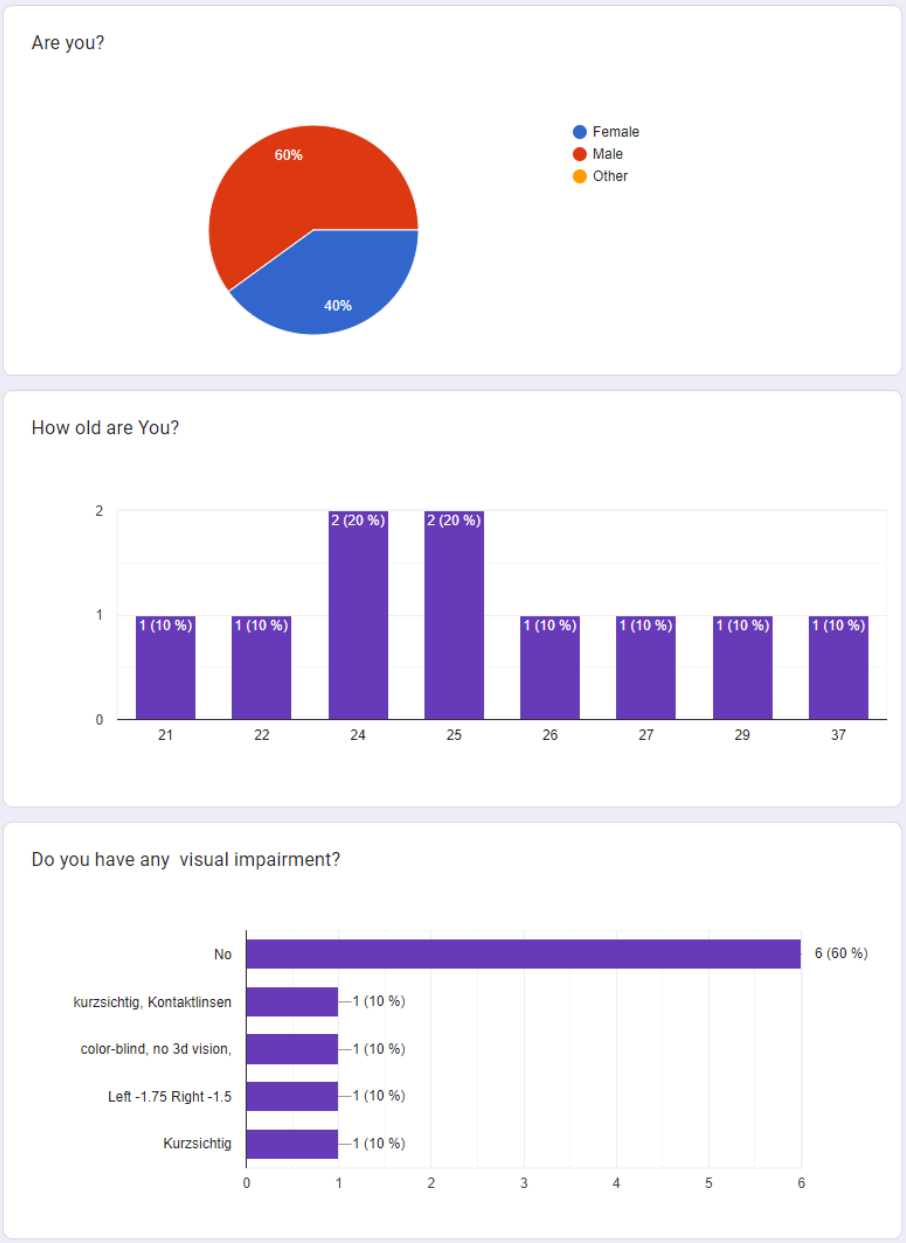
	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

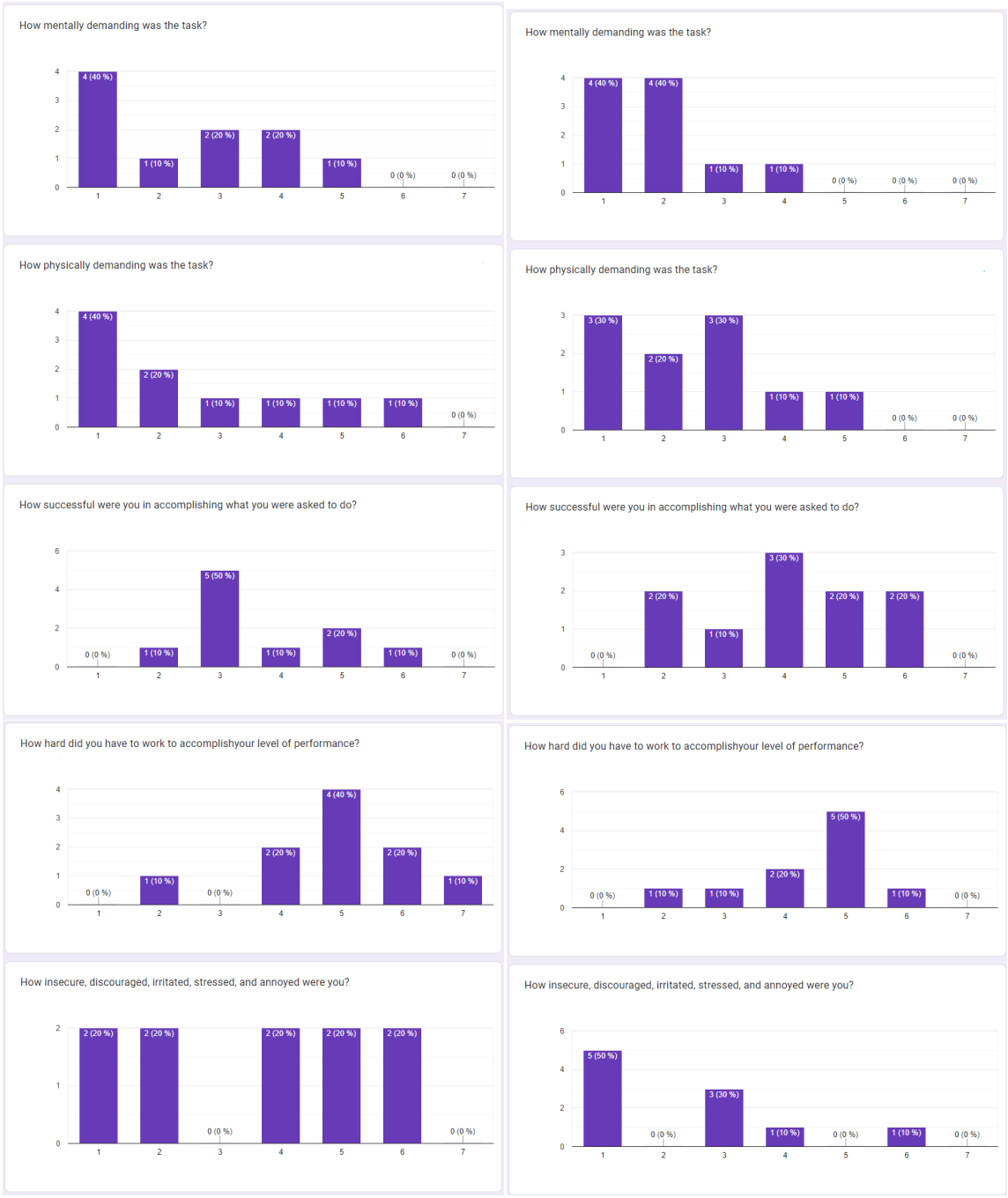
How difficult was it to activate depth layer 1? *

	1	2	3	4	5	6	7	
Not at all	○	○	○	○	○	○	○	Very Difficult

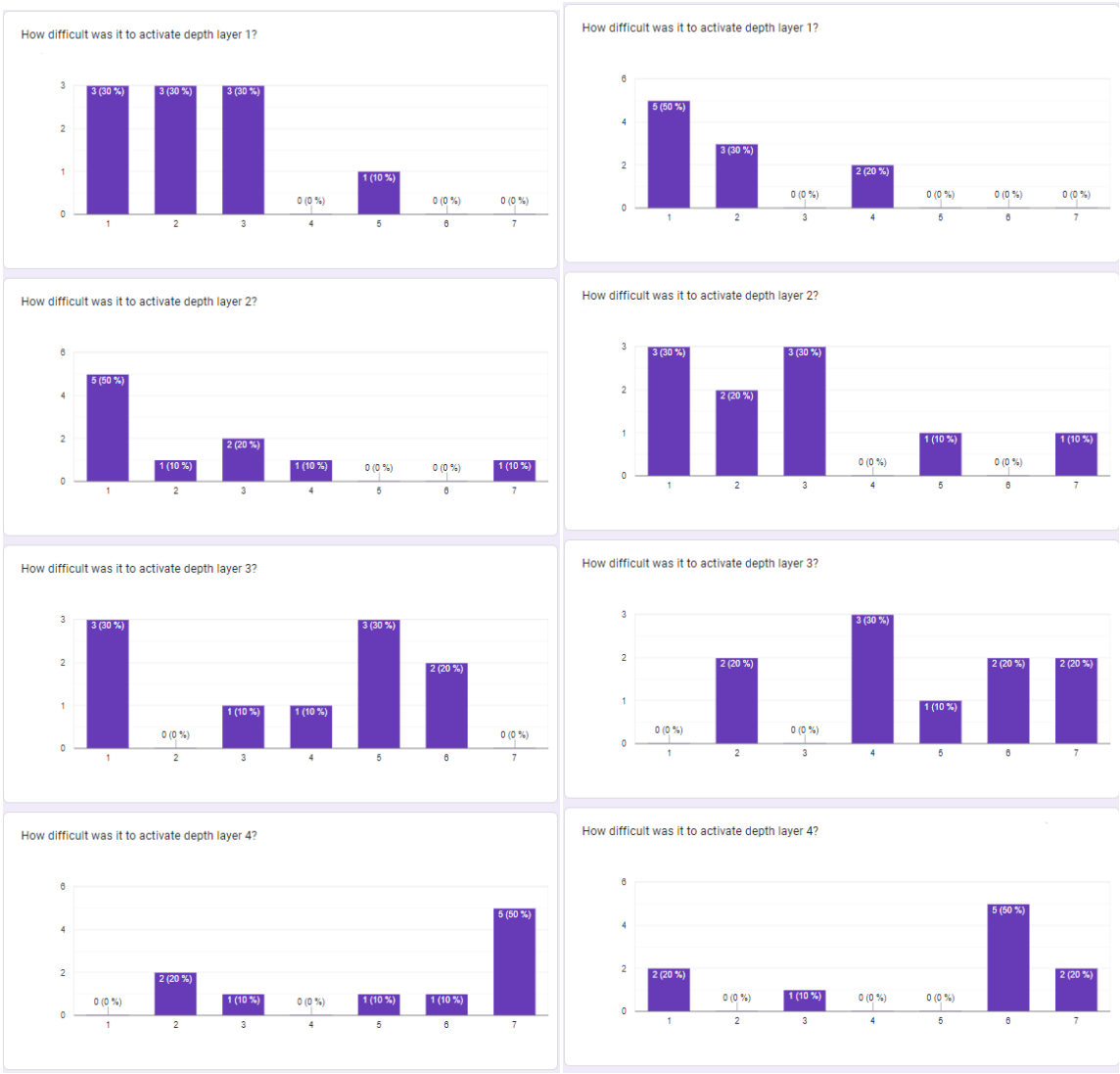

How difficult was it to activate depth layer 2? *
1 2 3 4 5 6 7
Not at all
Very Difficult
How difficult was it to activate depth layer 3? *
1 2 3 4 5 6 7
Not at all
Very Difficult
How difficult was it to activate depth layer 4? *
1 2 3 4 5 6 7
Not at all
Very Difficult

A.3 Results


Are you?
60%
40%
Female
Male
Other
How old are You?
2
1
0
2 (20 %)
2 (20 %)
1 (10 %)
1 (10 %)
1 (10 %)
1 (10 %)
1 (10 %)
1 (10 %)
1 (10 %)
21
22
24
25
26
27
29
37
Do you have any visual impairment?
No
kurzsichtig, Kontaktlinsen
color-blind, no 3d vision,
Left -1.75 Right -1.5
Kurzsichtig
6 (60 %)
1 (10 %)
1 (10 %)
1 (10 %)
1 (10 %)
0
1
2
3
4
5
6



Left: results after PC calibration. Right: Questionnaire results after Wang calibration.



Left: results after PC calibration. Right: Questionnaire results after Wang calibration.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature