Utrecht University

Rotation Methods for 360° Videos in Virtual Reality - A Comparative Study

Leo Zeches

Supervisor: Dr. Wolfgang Hürst

Second Examiner: Dr. Jacco Bikker

April 25, 2022

Acknowledgements

First and foremost, I would like to thank supervisor, Dr. Hürst, for his exceptional support and constant feedback. I would not have been able to complete this thesis without it.

Secondly, I would also like to thank my second examiner, Dr. Bikker, for the time and effort put into the analysis and grading of my work.

Thirdly, I would like to thank my family, friends and colleagues, who supported me throughout the past months.

Thesis advisor: Dr. Wolfgang Hürst *Leo Zeches*

Rotation Methods for 360° Videos in Virtual Reality - A Comparative Study

ABSTRACT

Advances in VR technology enabled the development and distribution of 360° videos. Also called panoramic videos, this type of media is shot from multiple camera angles, forming a complete surround experience. Paired with a HMD device, this creates a fully immersive experience.

For the majority of 360° video players, view point changes in order to explore all viewing directions are often exclusively performed by head movements.

While this design helps greatly with immersion, it also creates some practical problems. To explore the content of a video in all viewing directions, a viewer has to rotate their whole body along the yaw axis. This is especially problematic if the viewer wishes to remain stationary, for example on a non-rotating chair. In this case, the whole video content can only be accessed by performing extreme head movements and straining neck rotations.

This work focuses on implementing alternative rotation methods which allow a viewer to fully navigate the content of a video while remaining in a stationary position and while avoiding straining head rotations. To this end, three distinct rotation methods have been created, each being a representative for a basic interaction concept. A GUI method, which implements additional *graphical user interface* elements, a thumbstick method, utilising a *physical VR controller* as input, and a dragging method, which uses *gestures* for video rotation. These methods are then analysed and compared against each other in a user study, focusing both on viewer performance as well as subjective feedback.

The results show that while there is no complete elimination of head movement, nor significant difference of it between the three rotation methods, every rotation method allows a viewer to navigate the video content from a stationary position. Participant feedback suggests that the Thumbstick method was best suited for this task. These results suggest that implementing the Thumbstick method into 360° video players provides viewers with a powerful alternative for video navigation. The additional inclusion of the Dragging method provides viewers with multiple options for rotation which accommodates different preferences.

This thesis is split into two parts. The first part summarizes the main scientific results of this work in the form of a self contained paper in the IEEE format.

The second part is the appendix, which contains all of the additional information, including an extended literature review (A), implementation details, (B) study design (C), additional results (D) and a final conclusion and suggestions for future work (E).

Contents

Chapter 1

Scientific Paper

This section contains the main results of this thesis in the form of a scientific paper. The paper is self-contained, serving as a full report in the IEEE paper format. Additional information regarding the thesis, results, and conclusions, are reported in the Appendix.

Rotation Methods for 360° Videos in Virtual Reality - A Comparative Study

Leo Zeches
Information and Computing Sciences
Utrecht University
Utrecht, Netherlands
Email: l.f.zeches@students.uu.nl

***Abstract*—360° videos are an important sub-area in virtual reality (VR) entertainment. Yet, current 360° video players for head mounted displays (HMD) generally employ a standard 2D video player control scheme, and users are forced to often uncomfortable head rotations when they want to explore content in all viewing directions.**

The goal of this work was to find out if additional rotation methods allow viewers to explore the full content of a 360° video in an enjoyable way while remaining in a stationary position. To this end, three distinct rotation methods were evaluated: one method with a graphical user interface, one using the VR controller thumbstick, and one based on gesture.

The rotation methods were tested in a user study and compared against each other. Both viewer behaviour and user feedback were analysed.

The results of the study show that there were no significant differences in head movement between the rotation methods. Every method allows the viewers to take in the full content of the video. However, there were significant differences in viewer behaviour in terms of rotation speed, number of rotations and rotation amount. Additionally, subjective feedback has shown that the Thumbstick method was the preferred method, due to the high level of precision and the overall ease-of-use. Finally, we concluded that the inclusion of both the Thumbstick method and the Dragging method is a worthwhile addition to 360° video players.

1. Introduction

Digital video as a medium and area of human-computer interaction has benefited a lot from recent developments in camera and virtual reality (VR) technology. One of these technologies is 360° videos. This type of video is shot from multiple camera perspectives in all directions, creating a full 360° view of the surroundings. The video can then be watched either on a 2D desktop or using a virtual reality setup, for example a head-mounted display (HMD). Traditional 2D videos have established and well documented controls to interact with and manipulate the video. Most of these controls have been established early on [1] and have been continuously refined and extended over the years [2]. When the video player device changes, the controls tend to adapt as well, such as on the first generations of the iPod [3], and with the introduction and development of the modern smartphone [4][5].

Yet, certain interactions that are essential to explore the full content of a 360° video are often not supported by modern 360° video players.

360° videos have the three spatial dimensions of yaw, pitch and roll. This allows a viewer to orientate themselves and rotate their point of view. On 2D desktops, this is usually done by dragging the mouse, for example in [6]. For HMD devices, this navigation is generally done exclusively with head movements. Additional video controls, like timeline interaction, are directly translated from 2D desktop applications, for example in the Youtube video player for 360° videos. This means that the control schemes for 360° video players do not take advantage of the unique aspects of VR and HMD devices.

Digital videos are generally considered to be consumed in a passive, *lean-back* fashion [7]. However, different situations require fast and efficient interaction, for example quickly skimming the content of a video to assert its relevance or rewatching interesting or complicated parts of a video. For 360° videos, the need for interaction is even greater. Viewers have to navigate a larger area with potentially multiple points of interest located around them. When these situations arise, conflicts can appear. Quickly skimming the content of a 360° video means analyzing a larger amount of content, often translated to a 2D plane using thumbnails [8]. If this method proves insufficient, additional orientation via head movement is required. This involves turning around quickly, leading to strain, fatigue and discomfort due to the weight of modern HMD devices [9][10].

In addition to the lack of different navigation controls, viewers often find themselves in a seated or stationary position when consuming digital videos, including 360° videos. This fact exacerbates the problem, since a large portion of the 360° video content becomes inaccessible. If a viewer wants to look behind them, they have to physically change position, or rotate their head and neck to an extreme and uncomfortable degree, possibly for an extended period of time, depending on the type of video [11].

In order to tackle the aforementioned problems, this study focuses on viewers could manipulate the viewing direction of 360° videos while in a seated position and without the need for uncomfortable head rotations. We implement and compare different control methods for navigating and manipulating the viewing direction. The goal is to create insight into additional control methods for 360° videos which are adapted to the VR context and can extend the traditional video controls.

2. Related Work

2.1. User Behaviour and Performance in VR

The context of VR has unique properties and aspects which will define how a user will act and perform in this type of environment.

One focus of this research is viewer behaviour in 360° videos. Watching a 360° video can be done either by using traditional desktop devices [12] or HMD devices [13][14][15]. Movement patterns are being extracted by analyzing the amount of movement along the two spatial dimensions of yaw (horizontal) and pitch (vertical). Results show that different types of video encourage a different amount of exploration and spatial navigation. For example, in a city tour or travel video, viewers often want to actively explore the environment around them. Other types of videos, such as sports videos, encourage following a single point of focus which often moves around quickly on the screen. For both HMD and desktop devices, the main movement is along the yaw axis [13][15]. In comparison, movement along the pitch axis is not as extreme and slower in speed. Additionally, these studies noted the existence of an initial "exploration phase" [12][13][16]. Most of the movement happens towards the beginning of the video when a viewer explores the environment for the first time. This demonstrates that for most videos, viewers will want to explore the full content, even if prolonged spatial navigation is more relevant for certain types of videos (like city tours compared to other types.

This desire for exploration creates some problems. According to Katz [7], videos are generally consumed in a passive, *lean-back* fashion. In contrast, head-controlled HMD devices need movement and physical rotation in order to explore the full content of the video. This creates problems in the shape of physical discomfort, as outlined by Wille et al. [9], and Kim & Shin [10]. HMD devices cause significantly higher neck muscle activation and perceived discomfort than traditional desktop devices, especially over prolonged use.

Additionally, this perceived discomfort inhibits exploration and increases the fear of missing content. In a study from 2018 [11], participants were either on a fixed chair or had the ability to partially or fully rotate themselves on a swivel-chair. The fixed condition lead to higher discomfort, less exploration and overall less enjoyment than the swivel-chair condition.

These findings point towards a need for exploration of the full 360° video content. This need, combined with the limited exploration and physical discomfort experienced by a viewer in a seated position, motivates the need to research alternative approaches for spatial navigation than pure head movements.

2.2. Interaction in Virtual Reality

Interactions in virtual reality are *reality based interactions* [17], often trying to emulate real-life human gestures and movements in a virtual context. These are most often realised using head movement, gestures and input controllers.

Specifically in VR, interaction at a distance is often necessary, since locomotion capabilities tend to be limited. This has been solved using the popular raycasting method [18], where a laser pointer is being projected from the user's hand to select objects at a distance. This selection becomes less efficient the smaller and further away the target is. To this end, solutions like *Raycursor* [19], *IntenSelect* [20] or the *crossing paradigm* [21][22] have been implemented. These interaction methods are all based on the raycasting idea, adapting them to be more selective and giving more control to the user.

Another issue that arises when using either raycasting or hand gestures (for example explored in [23]) is the physical strain, which can be experienced when performing these interactions with mid-air hand positions. This aspect, also known as *gorilla arm effect*, can inhibit users from fully engaging with VR applications with this type of interaction [24]. Another problem for interaction is the so-called *Heisenberg effect* [25]. This phenomenon describes the accuracy problem when pressing buttons on a controller which are also used for selection. The button press itself can move the selection ray, leading to errors, especially for small targets.

The different types of interaction come with their own benefits and drawbacks and they are very use-case dependant. In this work, we focus on evaluating three basic interaction concepts: Graphical user interfaces (GUI), physical buttons, and gestures. The goal is to evaluate which of these interaction standards provide a functional solution in this context and can extend or even replace the simple head and body rotation method.

2.3. 360° Video Manipulation in VR

360° videos in VR have a combination of different media aspects [26] and can be interacted with in different ways. Studies focusing on viewer behaviour in VR [27][28][29] show that viewers tend to engage in a more passive, lean-back behaviour [7], most likely stemming from their TV viewing habits. While a high level of video control can be appreciated, especially for videos with fast moving points of interest (for example sports videos), too much interaction is often not desired and can also act as a distraction.

Different control methods allow the user to engage with and explore the content in a unique way. The paper by Lilija, Pohl and Hornbaek [30] adopts the concept of direct interaction [31], creating a fully explorative 3D animation video where users can interact with objects and jump to any point in the animation by directly manipulating them. Another approach [32] proposes gamifying 360° videos in VR by adding human characters with the goal of creating a more engaging and immersive experience, moving towards a *lean-forward* behaviour [7].

Methods to improve interaction with 360° videos in VR often adapt traditional video control schemes to the new context, taking the advantages and disadvantages of VR into account. To this end, time and spatial navigation are decoupled into head movement and hand movement [33]. This leads to the commonly found control scheme in applications like Youtube VR, where the tracked head movement controls the orientation while the hand movement and controller input interact with the video UI, which, in turn, is often solely focused on time-based navitation (e.g. play/pause, fast forward, or time skimming).

Other interaction methods try adapting and even combine these control methods. In the paper by Sargunam et al. [34], head movement is amplified in order to allow for a stationary viewing experience, avoiding extreme and uncomfortable head movements. Head movement can also be used for selection [35] or interaction in combination with eye tracking [36] when interacting with a traditional video UI.

Different gesture controls have also been explored in the context of 360° videos in VR. A study by Rovelo et al. [37] asked participants come up with their own gestures to interact with videos inside of a CAVE setup, which is a dome structure with the video projected onto it from the inside. The authors found that viewers prefer linear, dynamic and coarse gestures when controlling a video. Again, they observed that these gestures were strongly influenced by previous desktop application interactions and mental models. One such example of a previously internalized mental model is the left/right motion for navigating time, as used in traditional 2D horizontal timelines.

A different aspect of video interaction is the visual feedback in form of thumbnails, which are used for orientation during time scrubbing. For 360° videos, this visual feedback becomes distorted, since the spherical video has to be projected onto a 2D plane. This problem can be addressed by using rounded thumbnails displaying the current view point [8], although this still requires rotation and orientation to view the whole video content and is not as efficient for search tasks. A different method by Lo et al. [38] works by creating route tapestries for city walking and virtual tour videos. The content on either side of the moving camera is stitched together and displayed over the timeline during interaction, creating two continuous orthographic-perspective projections acting as large thumbnails. This has been shown to be faster and more accurate for search tasks than equirectangular thumbnails.

In our work, we focus on the most common, best performing video interaction methods and compare them. Raycast, gestures and controller input will be compared in order to find the best interaction methods to control 360° video navigation, while keeping their disadvantages in mind.

In order to verify how spatial navigation can best be supported in the context of 360° videos, we focus on comparing the most common interaction concepts with each other. While each of them provides a well-established way to interact with content in both 2D and 3D contexts, it is unclear which of those is most suited for 360° video navigation in a stationary position.

3. Research Aim

The research aim of this paper is to introduce different methods for spatial navigation (rotation) in 360° videos, using a HMD device in a stationary position. The goal is to find a control scheme that can be seamlessly integrated into modern video players. This new control scheme will not solely rely on head movement, but also incorporate other interaction methods like UI elements or gestures. The goal is to provide a rotation method that encourages exploration in a stationary position while minimizing discomfort from extreme head movements.

We summarize this goal via the following research aim:

Which interaction concepts for 360° videos allow for a complete exploration of the content in a comfortable way while maintaining a stationary position?

We further specify this aim via the following research question by focusing on three basic interaction methods and defining 60° as our threshold for an uncomfortable head rotation:

Can a 360° video be navigated completely with GUIs, physical buttons, or gestures from a stationary position, without moving the head more than 60° along the yaw dimension?

Another important aspect is differences in viewer behaviour between rotation methods. Different interaction methods will illicit different approaches from a viewer based on previous experiences and personal preference, like number of rotations, preferred direction, and amount of rotation. Furthermore, in addition to the quantitative analysis on head rotation and viewer behaviour, subjective user experience is also of high relevance and will be evaluated as well.

Because all interaction methods will provide viewers with the opportunity to change their viewing direction, we expect uncomfortable head rotations above 60° to be kept to a minimum. Because all of the interaction concepts are well-established in 2D environments and to some degree have proven their benefits (as well as possible disadvantages) in 3D and VR settings, we cannot make a prediction about which one may be best for 360° video navigation. We formulate these assumptions in the following hypotheses:

- H1: The different rotation methods allow viewers to explore the full 360° video content in a stationary position.
- H2: Viewers will spend an appropriate amount of time for each view section using the rotation.
- H3: Using the different rotation methods, viewers will avoid head movements above 60° in the yaw direction, avoiding neck strain.
- H4: There is no significant difference in head movement between the three rotation methods.
- H5: Individual performance and viewing behaviour will not change significantly between rotation methods.

4. Methodology

The following section outlines the general experiment design, including the method and implementation (4.1), the experiment tasks (4.2), the study procedure (4.3), as well as the participant details and recruitment (4.4).

4.1. Implementation

To test the outlined hypotheses, a custom video player was created. This player was built using Unity [39] version 2019.4.35f1 using the XR Interaction Toolkit [40] version 0.10.0. The scripts were written in C on Microsoft Visual Studio 2019, and the application was developed specifically for the Oculus Quest HMD [41].

The video content is projected as a 2D, equilateral texture onto a sphere object surrounding the fixed camera position. The camera rotation is mapped to the HMD rotation using the XR toolkit. This creates a basic 360° video player. Additional video player controls like play/pause, time skimming, and skipping have been disabled to get a more uniform study design and shorter study duration. The user interface elements for these functions, including a timeline, are still implemented and displayed, to create a better representation of a standard video player. The timeline also helps viewers to gauge the overall video length and the current time frame in the video. These UI elements (displayed in figure 1) are displayed at a fixed distance from the camera and are rotating with the camera rotation in order to be visible at all times. This "following of the head movement" is done with the goal to eliminate head movement while looking back and forth between video content and the UI, as well as to facilitate interaction with the GUI rotation method (4.1.1).

The program gathers data, such as head movement and rotation speed (outlined in detail in chapter 5) in the background during the run time. This data is compiled and sent to a Google form, from which it can be processed and analysed further. Additionally, three questionnaires are provided graphically inside of the program throughout the experiment in order to quantify qualitative data and capture participant feedback. The results of these questionnaires are also compiled and sent to a similar, separate Google form.

In addition to the basic video player, three distinct rotation methods have been implemented. These methods work by rotating the video sphere around the camera, enabling a separate rotation to the camera rotation. This means that a viewer can rotate the video using the novel rotation methods while still using head movements to look around separately.

4.1.1. Method 1: GUI. For this method, two graphical buttons are displayed alongside the timeline interface. Interacting with these buttons rotates the video sphere in the left or right direction. This interaction is performed by pointing and aiming a ray at the graphical button and pressing the physical trigger button on the controller (raycast interaction). While the button is pressed, the video rotates in the given direction. Keeping the button pressed over a period of time also increases the rotation speed every 0.5 seconds by 70%, up to a maximum.

The GUI elements follow the viewer head movement in order to be in view at all times and easily accessible during head rotations. During a button press, the GUI following is disabled, in order to enable independent head movement while rotating the video. After the rotation has finished, the GUI elements move back into the center of the field of view.

4.1.2. Method 2: Thumbstick. For this method, the physical thumbstick on the right VR controller is mapped to the video sphere rotation. Tilting the stick towards the left or right rotates the video sphere in the corresponding direction. The rotation speed is dependant on the amount of tilt, allowing for small, incremental rotation changes. Tilting the thumbstick all the way rotates the sphere at the fixed maximum speed.

4.1.3. Method 3: Dragging. For this method, the orientation of the right VR controller is mapped to the orientation of the video sphere directly. Pressing and holding the so-called "Grip" button on the controller starts the dragging process. The video sphere mimics the rotation of the controller on the yaw axis, allowing the viewer to directly manipulate the sphere via gesturing, similar to swiping left and right on a touch screen. Letting go of the button fixes the sphere back in place, allowing for the controller to be moved back to the initial position and start the dragging again. This enables the viewer to rotate the sphere completely using multiple, smaller gestures. The speed of the sphere rotation is directly linked to the speed of the gesture.

4.2. Experiment Tasks

The aim of the experiment is to verify which of the three interaction methods works best for exploring 360° videos. To simulate such an exploration, users must fulfil certain tasks that simulate a situation in which the rotation method needs to be fully utilised. While these tasks do not

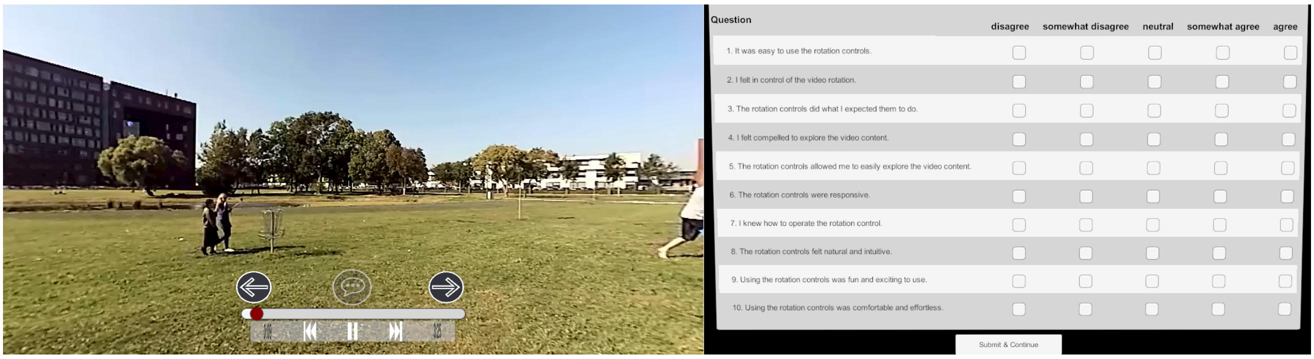

Figure 1: Screenshot of the GUI method, including basic video UI and rotation buttons (right), and of the questionnaire (left).

correspond directly to a common, real life use case, they do require the participant to rotate the video sphere efficiently over the duration of the video in order to successfully complete the task. This requires the participants to fully engage with the rotation methods and provides opportunity to gather data providing the necessary insight. It also gives the participants a full impression of the rotation method and allows them to provide well-informed feedback during the questionnaires.

The tasks given to the experiment participants are defined as *search tasks*. The goal of such a task is to search, identify, and count a specified item in the video. These items appear multiple times during the video duration and are specific to each video. For example, one such task would be ”Count the number of orange cats.” during a video inside of a cat cafe. Each video has a unique search task. The videos have been chosen to be similar in length and to have enough content appropriate for the purpose of the search tasks. The participants are required to rotate the video sphere using the given rotation method in order to access the whole content of the video and to be able to find and count all of the items.

Without the ability to pause or rewind the video, these search tasks can be considered quite difficult. However, the actual accuracy of the item counting is not the focus of this study. The tasks are simply a way of engaging the participants to use rotation methods to their full extend. The data analysis focuses on the behaviour data captured in the background as well as the participant feedback. Based on this data and feedback, conclusions can be drawn on the validity of a rotation method in the broader context of 360° video viewing behaviour.

Different videos contain varying numbers of different items and each video has its own unique search task. Considering the differences in these individual tasks, the difficulties of the tasks are not uniform. To this end, the distribution and order of the search tasks are balanced over the number of participants and the rotation methods, allowing the data between the rotation methods and between the participants to be comparable.

4.3. Procedure

Over the duration of the study, each participant will explore the three different rotation methods. Each rotation method is paired up with and tested on three different videos and corresponding search tasks, resulting in a total of 9 tasks per participant. Having three tasks per rotation method allows for a better average while keeping the overall experiment duration to a reasonable limit. Both the pairing of the tasks and the rotation methods are balanced, as are the individual task order and the rotation method order.

Before each rotation method is tested, a short tutorial is provided, allowing the participant to familiarise themselves with the Oculus HMD, the controller and the rotation method. Before each search task, the item in question is described. After the video has been played, a prompt allows the participant to choose between one of three options to answer the search task. For example, the answer options for the task ”Count the number of orange cats in the video.” would be ”Less than 3; Between 3 and 5; More than 5”. The reason to have a multiple-choice answer instead of a concrete number is to make it clear that an exact answer is likely impossible and to evoke a more realistic exploration behaviour.

After three search tasks have been performed, a questionnaire is provided for each rotation method. The questionnaires are in the form of a 5 point Likert scale and aim to gather feedback on the rotation method. After each rotation method has been explored and all the search tasks and questionnaires have been answered, the experiment is finished. A short interview with the participant is conducted in order to gather additional, qualitative feedback, such as remarks on the study procedure and rotation methods.

Participants were allowed to take breaks between videos and rotation methods if they desired to do so. The total duration of the experiment without any breaks is roughly 25 minutes.

4.4. Participants

The study was performed in person in order to remove possible uncontrolled outside factors and different study setups influencing the results. This in-person nature also allowed for a less structured post-experiment interview, allowing participants to express their subjective feelings on the rotation methods more freely. 26 participants signed up to participate in the study. 15 participants were male, 11 female, with an age range between 23 and 33 years old (mean $= 27.35 \pm 2.23$).

Due of the Covid-19 pandemic during the period of this work (November 2021 - April 2022), it was deemed not feasible to recruit participants from public areas. Instead, the participants were approached in private, and the study was performed in person. The study equipment was cleaned after every use, and the guidelines regarding private gatherings and general sanitary measures in place during the time of the study were respected.

5. Data Gathering

In order to answer the research question (see chapter 3) and verify the related hypotheses, the rotation methods need to be evaluated. The data for this evaluation can be divided into two categories, *quantitative* data (5.1) and *qualitative* data (5.2). The data is gathered in the background and compiled between search tasks. The compiled data is then sent to a private Google form, which has entry fields for each type of data, along with a unique participant ID.

5.1. Quantitative Data

Quantitative data is gathered to compare viewer behaviour and performance for the three rotation methods. Four main types of data are gathered:

1) Rotation speed
2) Individual rotations
3) Head movement
4) View time distribution

For *rotation speed*, both maximum rotation speed and average rotation speed are measured for each individual task. These reflect how fast the rotations were performed for each rotation method, measured in degrees per second. Speed is one aspect of the implementation that cannot be made uniform between the rotation methods. Analysing the rotation speed will give insight into how the speed implementation translates into the use case.

For *individual rotations*, both the number of individual rotations (average) as well as the amount of degrees rotated (maximum and average) for each individual task are analysed. In combination, these data points give insight into viewer behaviour and performance. By comparing how often and for how much the rotation methods are used, clear use case patterns can emerge. Having distinct use case patterns can give additional information about how to adapt and improve the rotation methods and how to implement other control methods in the future.

Additionally, the distribution of left versus right rotations are compared between rotation methods, both in absolute numbers and in percentage of total rotations. This data can give additional insight into viewer behaviour related to the rotation method.

For *head movement*, both maximum head movement (in degrees) and time above 60° (in seconds for both left and right direction) are measured. These measurements are taken only on the yaw axis. Measuring head movement gives crucial insight into how the rotation methods are used to avoid straining head movement. In general, high maximum head movement and high time above 60° are deemed to be non-optimal for a rotation method, since this can indicate that extreme head movement was preferable to using the rotation method. Head movement can also be used to analyze general viewer behaviour outside of the context of rotation methods.

For *view time distribution*, the orientation along the yaw axis is divided into 8 equal sections of 45° each (see figure 2). The average time that the center viewpoint is pointed into each section is measured, both in seconds and as a percentage over the whole task duration. This gives insight into what part(s) of the video the viewer is looking at the most, and how the view time is distributed over the sections. If a rotation method reveals that some sections have no view time, then this could point to the fact that this rotation method hinders a viewer to engage with the entirety of the video content.

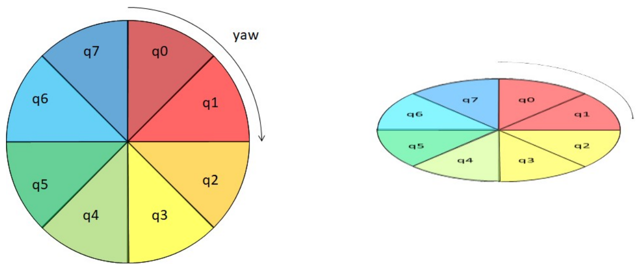

Figure 2: Division of the yaw axis, left side from a top down perspective, right side angled forward. The initial center position is between q_7 and q_0.

5.2. Qualitative Data

The qualitative data is quantified using a 5 point Likert scale and presented to the participants in the form of a questionnaire. This questionnaire has 10 questions relating to general enjoyment, ease-of-use, physical discomfort, etc.

Evaluating this qualitative data helps with understanding subjective impressions and personal preferences of the participants. These results are generally not reflected in a purely quantitative evaluation, but are essential when comparing different VR controls in a user study. This data is also compiled and added into a separate Google form from where it can be analysed later on.

In addition to the questionnaire, an informal interview is performed after the experiment, in which the participants have the possibility to voice less structured opinions and to freely express their remarks and give more feedback on the rotation methods.

6. Results

The following section outlines the statistical results of the analysis from the evaluation. One participant had to drop out of the evaluation due to excessive cybersickness, resulting in 25 valid experiment runs.

6.1. Quantitative Results

For the quantitative data, each rotation method (GUI; Thumbstick; Dragging) was tested in three consecutive runs, resulting in $N = 75$ data points per condition. For each variable, the distribution was analysed. Data points that were classified as extreme outliers were removed. A data point is classified as an extreme outlier if it is either more than 3 interquartile ranges below the first quartile or 3 interquartile ranges above the third quartile. Data is noted as mean ± standard deviation, unless specified otherwise.

6.1.1. Rotation Speed. For the maximum rotation (in degrees per second deg/s, $N = 66$), a Friedman test revealed statistical differences in maximum rotation speed between the three rotation methods; $\chi^2 = 84.273, p < 0.001$. Post-hoc analysis revealed significant differences between GUI (87.76 ± 35.35 deg/s) and Thumbstick (331.82 ± 47.06 deg/s; $p < .001$), as well as between GUI and Dragging (324.09 ± 187.34 deg/s; $p < .001$). There was no significant difference between Thumbstick and Dragging ($p = .172$).

For the average rotation speed (in deg/s; $N = 75$), a one-way repeated measure ANOVA test was performed, revealing significant differences in average rotation speed ($F(1.567, 115.985) = 56.49$; $p < .001$; partial $\eta^2 = .433$). Post-hoc analysis revealed significant differences between GUI (36.38 ± 11.91 deg/s) and Thumbstick (98.88 ± 58.34 deg/s); difference of 62.50, 95% CI, 46.47 to 78.53 deg/s, $p < .001$; as well as between GUI and Dragging (85.34 ± 37.28 deg/s); difference of 48.56, 95% CI, 38.29 to 59.64 deg/s, $p < .001$. There was no significant difference between Thumbstick and Dragging ($p = .200$).

6.1.2. Individual Rotations. The individual number of rotations, maximum and average rotation amount are analysed, as well as left / right rotations.

A one-way repeated measures ANOVA test ($N = 75$) revealed significant differences in number of individual rotations; $F(2, 142) = 104.371$, $p < .001$, partial $\eta^2 = .595$. Post-hoc analysis revealed significant differences between GUI (39.43 ± 21.46) and Thumbstick (137.99 ± 69.26); $p < .001$, between GUI and Dragging (104.75 ± 55.38); $p < .001$, as well as Thumbstick and Dragging; $p = .002$.

For the maximum rotation amount (in degrees deg, $N = 63$), a Friedman test was performed and revealed significant differences; $\chi^2 = 84.554$, $p < .001$. Post-hoc analysis revealed significant differences between GUI (199.87 ± 91.67 deg) and Thumbstick (310.55 ± 175.50 deg); $p = .009$, between GUI and Dragging (76.95 ± 28.66 deg); $p < .001$, and Thumbstick and Dragging; $p < .001$.

For average rotation amount (in degrees deg, $N = 63$), a one-way repeated measures ANOVA revealed significant differences between the rotation methods; $F(1.707, 105.855) = 31.901$, $p < .001$, partial $\eta^2 = .340$. Post-hoc analysis revealed significant differences between GUI (50.52 ± 23.44 deg) and Thumbstick (34.55 ± 16.85 deg); GUI and Dragging (29.07 ± 6.14 deg); as well as Thumbstick and Dragging;

For rotation direction, one-way repeated measures ANOVA was performed to analyze differences between rotation methods for both left (L) and right (R) rotations.

The differences between L and R rotations were analysed (in %, negative means $R > L$). ANOVA revealed differences between GUI ($-58.19\% \pm 93.43$) and Thumbstick ($-14.15\% \pm 72.20$) difference of 44.04%, 95% CI, 12.26 to 75.82, $p = .003$, GUI and Dragging ($17.51\% \pm 63.90$); difference of 75.70%, 95% CI, 46.20 to 105.20, $p < .001$) , as well as between Thumbstick and Dragging ; difference of 31.66%, 95% CI, 6.05 to 57.27 deg, $p = .010$.

6.1.3. Head Movement. For head movement, both maximum head rotation (in degrees deg) and time spent with a head rotation greater than 60° (in seconds s) are analysed.

For maximum head rotation, a Friedman test ($N = 75$) revealed no significant differences between the three rotation methods; $p = .257$. GUI had a mean of 89.98 ± 40.98 deg, Thumbstick had a mean of 73.23 ± 94.94 deg, and Dragging had a mean of 79.69 ± 33.99 deg.

For time spent above 60°, a Friedman test was performed, $N = 75$, $\chi^2 = 5.610$, $p = .060$. The differences in time spent above 60° were not statistically significant. GUI measured a mean of $13.58 \pm 21.05s$, Thumbstick measured $5.04 \pm 8.62s$, and Dragging measured $9.82 \pm 16.17s$.

In terms of frequency (times was rotated above 60° compared to keeping the head rotated towards the front), a chi-square test found no significant differences between the rotation methods, $p = .158$. GUI had $56/75 = 74.7\%$ cases larger than 0 seconds. Thumbstick method had $44/75 = 58.7\%$ cases > 0. Dragging method had $51/75 = 68.0\%$ > 0.

6.1.4. View Direction. The view direction is analysed by comparing the time spent looking at the different video sections of the video sphere. The time is described as viewtime over task duration (in percentage) and compared for the

8 different sections $q_0; ...; q_7$ (outlined in figure 3). One-way repeated measures ANOVA was performed ($N = 75$) to check for significant differences between the rotation methods for each section.

GUI was deemed significantly different between both Thumbstick and Dragging for the sections q_0 ($p = .016$ and $p = .005$ respectively); q_2 ($p = .049$ and $p = .007$); q_3 ($p < .001$ and $p < .001$); q_4 ($p < .001$ and $p < .001$); q_5 ($p < .001$ and $p < .001$); q_6 ($p < .001$ and $p < .001$); and q_7 ($p < .001$ and $p < .001$). There were no significant differences between any rotation method pair for section q_1.

There was no significant difference between Thumbstick and Dragging for any section.

6.2. Qualitative Results

The answers to the questionnaires (illustrated in figure 4) are analysed and compared for each rotation method using Friedman's test and pairwise comparison with Bonferroni correction.

The Thumbstick method was rated significantly higher than both GUI and Dragging for *ease-of-use* (Q1); *expectation fulfillment* (Q3); *exploration* (Q5); *responsiveness* (Q6); *intuitiveness* (Q8) and *fun* (Q9).

For *level of control* (Q2) and *encouragement of exploration* (Q4), Thumbstick was rated significantly higher than GUI. There was no significant difference between Thumbstick and Dragging, nor between GUI and Dragging.

There was no significant difference in reported *knowledge on how to operate the controls* (Q7) between the three rotation methods.

Finally, Dragging was rated to be significantly more *exhausting and demanding* (Q10) than both GUI and Dragging.

6.3. Effects on Cybersickness

Of the 26 participants, an unexpected high number of 14 (53.85%) reported moderate to severe cybersickness during the experiment procedure. To analyze the effect of other independent variables on this phenomenon, different tests have been performed to establish a possible link.

A chi-squared test for association was performed to determine the effect of gender on cybersickness. The test determined no statistical significant difference between experienced cybersickness between males and females ($\chi^2 = 2.735$, $p = .098$).

A Cochran-Armitage test was performed to determine the possible trend between level of VR experience and cybersickness. The test determined no statistical significant trend, $p = .130$.

A binomial logistic regression was performed to determine the predictive factor of age on cybersickness. The resulting model was not statistically significant, $\chi^2 = .326$, $p = .568$. The model only explained 1.7% of variance in cybersickness (Nagelkerke R^2).

7. Discussion

After analysing the quantitative and qualitative data, along with the subjective feedback from the participants, clear differences between the rotation methods have emerged. In the following, we discuss the findings in the proper context in order to draw conclusions for possible implementation and design guidelines.

7.1. Qualitative Analysis

Overall, the three rotation methods allowed the participants to view the full 360° content of the videos from a stationary position, as demonstrated by the view time distribution. Although there are significant differences in terms of their individual distribution, it can be stated that overall, the three methods achieve their intended goal of allowing for full video rotation, meaning the first hypothesis H1 (*The different rotation methods allow viewers to explore the full 360° video content in a stationary position.*) can be accepted. The reason for the different distribution between the GUI method and the other two methods does not seem to have a clear origin, and leads to no other significant conclusions. Although the distributions are significantly different, they do not show a clear improvement of one rotation method over another. We chose to accept H2 (*Viewers will spend an appropriate amount of time for each view section using the rotation.*) has well, although additional studies could yield a different conclusion.

For head movement, it was observed that many participants still resorted to using their head and look around, resulting in the majority (between 74.7% and 58.7%) rotating their head more than 60° at least once during a video. While the average duration of these extreme head rotations can be seen as rather small compared to the overall video duration, it is still reason enough to reject hypothesis H3 (*Using the different rotation methods, viewers will avoid head movements above 60° in the yaw direction, avoiding neck strain.*). However, the three rotation methods performed similarly in terms of head movement, with no statistically significant differences in maximum head rotation as well as duration above 60°, resulting in hypothesis H4 (*There is no significant difference in head movement between the three rotation methods.*) being confirmed.

In terms of viewing behaviour, the GUI method stood out with the slowest average rotation speed, the lowest average of individual rotations, as well as the highest average rotation amount, compared to both the Thumbstick and the Dragging method. This hints at a lower accuracy and less precise rotation, compared to the other two methods. The differences between the Thumbstick method and the Dragging method are less pronounced. Still, there are clear differences between Thumbstick and Dragging in terms of rotation direction and rotation amount, leading to the rejection of hypothesis H5 (*Individual performance and viewing behaviour will not change significantly between rotation methods.*).

Figure 3: Mean percentage of time spent in each video section for the for each rotation method. (Averaged over all tasks and all participants)

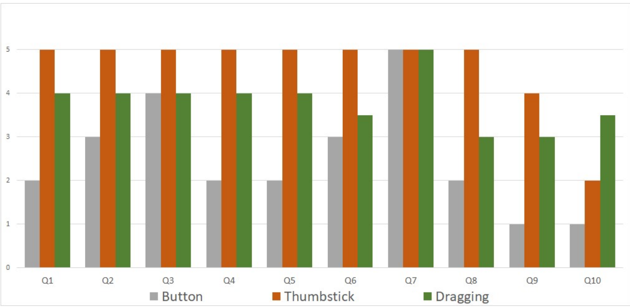

Figure 4: Medians of the questionnaire answers of all participants for each rotation method.

7.2. Subjective Feedback

Out of 25 participants, 22 (88%) indicated that the Thumbstick method was their favorite out of the 3 presented rotation methods. Only 3 participants (12%) prefer the Dragging method. No participant preferred the GUI method.

This result is reflected in the analysis of the questionnaire results, where Thumbstick scored significantly higher than GUI and Dragging for a number of key aspects like *ease-of-use* and *fun*. It was also rated significantly less exhausting than the Dragging method.

Additionally, most participants provided some additional feedback to explain their choice for favorite rotation method. For Thumbstick, many participants noted the relative *ease-of-use* and speed of this method. It was easier to make small adjustments as well as to rotate the video quickly. Some of these participants also noted that dragging felt ”sluggish” and slow, leading to a less responsive control scheme.

On the contrary, other participants also noted the high speed of the Thumbstick method as a problem, and instead preferring the Dragging method. This method was overall more fun and enjoyable to use in their view, since the added hand and arm motion made the movement more engaging.

This leads to the conclusion that the Thumbstick method can be very efficient for most users, but depending on the implemented speed setting, can lead to less accuracy for less experienced users. A more engaging control method like Dragging can be seen as a fun and enjoyable alternative despite the increase in fatigue.

The low rating of the GUI method is most likely due to the fact that the GUI is following the head movement when not pressed. While this was theorized to decrease the need to search and locate the GUI elements during head movement, it seemed to produce the opposite effect. Many participants noted that they had to relocate the GUI buttons and aim at them for the majority of the rotations, leading to frustration and mental exhaustion. GUI elements seem to need a stationary position in the virtual 3D space. Having a GUI element in a stationary position allows users to recognize and remember them in a 3D space much more effectively than if they move around, even if this movement is within the field of view.

Additionally, the speed of the rotation is much less consistent when using GUI buttons. Since gradual input or direct mapping to the rotation is not possible, the rotation speed is either uniform, or increases for longer button presses. However, this often leads to a rotation which was either too slow or too fast. Participants noted that there was no good middle-ground, making rotations much less precise.

7.3. The Problem of Cybersickness

The majority (53.85%) of the participants experienced moderate to severe cybersickness during the experiment. No link between independent variables like gender, age, and VR experience could be drawn. However, this study lacks the precision and focus on cybersickness to draw any definitive conclusions of the effects of these factors. Feedback from participants gives interesting insight to potential causes for this occurrence.

Multiple participants compared the feeling of nausea to being ”sea sick”, and noted that the rotation via head movement reduced their perceived symptoms. This could hint at the fact that rotation without perceived movement causes a disconnect between visual and vestibular perception, leading to cybersickness. Other participants noted that the effect increased for the GUI rotation method. Although this could not be explicitly confirmed by the data, the participants noted that the moving GUI elements and the need to locate and aim at them seemed to be confusing and mentally exhausting, exacerbating the symptoms.

A different contributing factor is the fact that the video content is often shot and edited in a dynamic way, with the probable intent to make the experience more engaging. However, such videos are most likely meant to be consumed on a 2D screen, since excessive movement and jarring camera edits seem to be somewhat responsible to the amount of cybersickness perceived by the participants.

In general, we believe that the high amount of rotation required for the search tasks, in combination with dynamic videos is most likely the reason for the high levels of cybersickness observed. However, to confirm this hypothesis, additional more in-depth studies need to be performed.

8. Conclusion and Future Work

The research goal of this work was to create and evaluate alternative rotation methods for viewing a 360° video with a HMD device while remaining in a stationary position. To this end, three different rotation methods have been implemented and analysed in a user study. Specifically, aspects like head movement, view direction and rotation amount have been measured and compared in order to answer the question if a 360° video can be navigated using one of the provided rotation methods in a stationary position and without moving uncomfortable head rotations.

Additionally, subjective feedback was gathered using multiple questionnaires as well as a post-experiment interview, in an effort to better understand the feasibility and potential design issues with these rotation methods.

The results show that while all three rotation methods achieve their goal of allowing viewers to experience the full content of a 360° video in a stationary position, there exist significant differences in viewer behaviour and viewer preference between these rotations. Different rotation methods will elicit different approaches from a viewer, resulting in different rotation speeds and rotation amounts, as well as different view time distributions. We conclude that it is important to take these differences into account when implementing such control methods in the context of 360° videos. We also conclude that based on the results, it is not likely to eliminate extreme head rotations completely, even with the inclusion of additional rotation controls. Additionally, subjective feedback from study participants has shown that certain design choices have to be considered carefully when implementing rotation methods. Especially rotation speed, precision and ease-of-use are important aspects that need to be fine-tuned and implemented with great care given the context of the control method. For UI elements like graphical buttons, the feasibility is greatly dependant on how easily they can be located and targeted. From this research, we conclude that rotation controls using a physical thumbstick, as well as gestures like dragging motions, are a better fit for 360° video controls, compared to exclusively using GUI elements.

The results of our research also identified the general problem of cybersickness when rotating 360° videos. Additional in-depth studies need to be performed in order to fully understand the different degrees of cybersickness, as well as its underlying causes and influencing factors in the context of 360° videos.

This study made a first effort in assessing the feasibility of additional rotation methods in the context of 360° videos. One idea for future work is to analyse the direct effects of these methods on the level of experienced cybersickness, with the goal of reducing this phenomenon. Potential solutions could include the pausing of the video while rotating. Other future work could focus on formulating overall guidelines for designing rotation methods for 360° videos and define optimal implementation details, like rotation speed and level of precision, as well as formulating these guidelines for the broader context of VR applications in general.

References

[1] Francis C Li, Anoop Gupta, Elizabeth Sanocki, Li-wei He, and Yong Rui. Browsing digital video. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 169–176, 2000.

[2] Suporn Pongnumkul, Jue Wang, Gonzalo Ramos, and Michael Cohen. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 139–142, 2010.

[3] Klaus Schoeffmann and Lukas Burgstaller. Scrubbing wheel: An interaction concept to improve video content navigation on devices with touchscreens. In *2015 IEEE International Symposium on Multimedia (ISM)*, pages 351–356. IEEE, 2015.

[4] Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. Pocketdragon: a direct manipulation video navigation interface for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–3, 2009.

[5] Wolfgang Hürst, Georg Götz, and Philipp Jarvers. Advanced user interfaces for dynamic video browsing. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 742–743, 2004.

[6] Teresa Chambel, Maiur N. Chhaganlal, and Luís A. R. Neng. Towards immersive interactive video through 360° hypervideo. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ACE '11, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308274. doi: 10.1145/2071423.2071518. URL `https://doi.org/10.1145/2071423.2071518`.

[7] Helen Katz. *The Media Handbook: A Complete Guide to Advertising Media Selection, Planning, Research, and Buying (4th ed.).* Routledge, 2010.

[8] Shakeeb Shirazi. Timeline visualization of omnidirectional videos. Master's thesis, 2018.

[9] Matthias Wille, Britta Grauel, and Lars Adolph. Strain caused by head mounted displays. In *Proceedings of the Human Factors and Ergonomics Society Europe*, 2014.

[10] Eunjee Kim and Gwanseob Shin. Head rotation and muscle activity when conducting document editing tasks with a head-mounted display. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 62, pages 952–955. SAGE Publications Sage CA: Los Angeles, CA, 2018.

[11] Yang Hong, Andrew MacQuarrie, and Anthony Steed. The effect of chair type on users' viewing experience for 360-degree video. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11, 2018.

[12] Fanyi Duanmu, Yixiang Mao, Shuai Liu, Sumanth Srinivasan, and Yao Wang. A subjective study of viewer navigation behaviors when watching 360-degree videos on computers. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[13] Mathias Almquist and Viktor Almquist. Analysis of 360 video viewing behaviours, 2018.

[14] Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 360-degree video head movement dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 199–204, 2017.

[15] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang. A dataset for exploring user behaviors in vr spherical video streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 193–198, 2017.

[16] Marc Van den Broeck, Fahim Kawsar, and Johannes Schöning. It's all around you: Exploring 360 video viewing experiences on mobile devices. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 762–768, 2017.

[17] Robert JK Jacob, Audrey Girouard, Leanne M Hirshfield, Michael S Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: unifying the new generation of interaction styles. In *CHI'07 extended abstracts on Human factors in computing systems*, pages 2465–2470, 2007.

[18] Chris Hand. A survey of 3d interaction techniques. In *Computer graphics forum*, volume 16, pages 269–281. Wiley Online Library, 1997.

[19] Marc Baloup, Thomas Pietrzak, and Géry Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[20] Gerwin De Haan, Michal Koutek, and Frits H Post. Intenselect: Using dynamic object rating for assisting 3d object selection. In *Ipt/egve*, pages 201–209. Citeseer, 2005.

[21] Huawei Tu, Susu Huang, Jiabin Yuan, Xiangshi Ren, and Feng Tian. Crossing-based selection with virtual reality head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[22] Susu Huang, QI Daqing, YUAN Jiabin, and TU Huawei. Review of studies on target acquisition in virtual reality based on the crossing paradigm. *Virtual Reality & Intelligent Hardware*, 1(3):251–264, 2019.

[23] Hrvoje Benko and Andrew D Wilson. Multi-point interactions with immersive omnidirectional visualizations in a dome. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 19–28, 2010.

[24] Jeffrey T Hansberger, Chao Peng, Shannon L Mathis, Vaidyanath Areyur Shanthakumar, Sarah C Meacham, Lizhou Cao, and Victoria R Blakely. Dispelling the gorilla arm syndrome: the viability of prolonged gesture interactions. In *International conference on virtual, augmented and mixed reality*, pages 505–520. Springer, 2017.

[25] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. Understanding the heisenberg effect of spatial interaction: A selection induced error for spatially tracked input devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2020.

[26] Henry Jenkins. *Convergence Culture: Where Old and New Media Collide*. New York: New York University Press, 2006.

[27] Goranka Zoric, Louise Barkhuus, Arvid Engström, and Elin Önnevall. Panoramic video: design challenges and implications for content interaction. In *Proceedings of the 11th european conference on Interactive TV and video*, pages 153–162, 2013.

[28] Lizzy Bleumers, Wendy Van den Broeck, Bram Lievens, and Jo Pierson. Seeing the bigger picture: a user perspective on 360 tv. In *Proceedings of the 10th European conference on Interactive TV and video*, pages 115–124, 2012.

[29] Mirjam Vosmeer and Ben Schouten. Interactive cinema: engagement and interaction. In *International conference on interactive digital storytelling*, pages 140–147. Springer, 2014.

[30] Klemen Lilija, Henning Pohl, and Kasper Hornbæk. Who put that there? temporal navigation of spatial recordings by direct manipulation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2020.

[31] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 237–246, 2008.

[32] Lemonia Argyriou, Daphne Economou, and Vassiliki Bouki. Design methodology for 360 immersive video applications: the case study of a cultural heritage vir-

tual tour. *Personal and Ubiquitous Computing*, 24(6): 843–859, 2020.
[33] Benjamin Petry and Jochen Huber. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proceedings of the 6th Augmented Human International Conference*, pages 217–218, 2015.
[34] Shyam Prathish Sargunam, Kasra Rahimi Moghadam, Mohamed Suhail, and Eric D Ragan. Guided head rotation and amplified head rotation: Evaluating seminatural travel and viewing techniques in virtual reality. In *2017 IEEE Virtual Reality (VR)*, pages 19–28. IEEE, 2017.
[35] Toni Pakkanen, Jaakko Hakulinen, Tero Jokela, Ismo Rakkolainen, Jari Kangas, Petri Piippo, Roope Raisamo, and Marja Salmimaa. Interaction with webvr 360 video player: Comparing three interaction paradigms. In *2017 IEEE Virtual Reality (VR)*, pages 279–280. IEEE, 2017.
[36] Sylvia Rothe, Pascal Pothmann, Heiko Drewe, and Heinrich Hussmann. Interaction techniques for cinematic virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1733–1737. IEEE, 2019.
[37] Gustavo Alberto Rovelo Ruiz, Davy Vanacken, Kris Luyten, Francisco Abad, and Emilio Camahort. Multiviewer gesture-based interaction for omni-directional video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4077–4086, 2014.
[38] Jiannan Li, Jiahe Lyu, Mauricio Sousa, Ravin Balakrishnan, Anthony Tang, and Tovi Grossman. Route tapestries: Navigating 360 virtual tour videos using slit-scan visualizations. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 223–238, 2021.
[39] Unity Technologies. Unity. `unity.com`, 2022.
[40] Unity Technologies. Xr interaction toolkit. docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.10, 2018.
[41] Inc. Meta Platforms. Oculus quest, 2019. URL `oculus.com/quest/features/`.

Appendix A

Literature Review

In this section, relevant previous work and their relation to this thesis project are outlined in detail.

Section A.1 outlines research relating to navigating and controlling 2D videos in order to understand the current design paradigms and how they differ and relate to our context of VR. Section A.2 dives into more general studies relating to virtual reality in order to expand on the context of VR and identify general principles which need to be taken into account for 360° videos. In Section A.3, user guidance mechanisms in VR are highlighted. These mechanics are used to help and guide a viewer's attention in different VR applications, including 360° videos. This process can substantially change a viewer's navigation in such a video, to the point that these techniques are worth investigating for other navigation methods. Section A.4 outlines current virtual reality interaction methods and paradigms, which need to be taken into account when creating new control methods for VR applications. Finally, Section A.5 dives into existing navigation and interaction methods for 360° videos. Combining aspects from the previously mentioned research fields, these methods can focus on video interaction, spatial navigation, time navigation, or a combination of these fields.

A.1 Traditional Video Controls

Video navigation and manipulation are important areas of human-computer interaction which have been thoroughly explored since the inception of the video player. Due to constant improvement of encoding and compression methods, streaming services and the internet as a whole, the questions relating to how humans can best interact and control video playback are far from settled.

Early works in video UI have adapted the by then common VCR control schemes to control digital videos [1]. These include play/pause and fast-forward/backward. This limited interface has been extended upon with many features that are expected in modern video players, such as playback speed, skip forward/back buttons, and most notably a timeline for time scrubbing, including preview shots acting as thumbnails for the viewer to orient themselves when navigating the timeline.

More recent work [2] expand on these functions, for example when using interactable bookmarks (similar to shot frames in [1]), which have thumbnails for visual guidance as well as preloading the streamed video to reduce buffering time in case a user navigates to the bookmark. Accompanying user studies in [1] and [2] have shown that these added features reduce buffering time for the videos, as well as reducing the need for manual timeline scrubbing. Instead, viewers were able to identify the preloaded bookmarks using the thumbnails and navigate the content of the video more efficiently.

Specifically for the task of time scrubbing or time navigation, a number of designs and interaction methods have been proposed over the years. One problem of traditional timelines is that they have a limited, finite level of detail for precise time navigation. Especially for longer videos and small UI elements, each interactable pixel of a timeline corresponds to a number of video frames, making fine-grained frame-by-frame browsing impossible.

Timeline implementation designs tackling this problem work by dynamically extending the basic timeline to allow for finer granularity [3][4][5]. The simplest way is to extend or add additional timelines for smaller periods of a video, allowing for finer time scrubbing in a given interval. This approach has been adopted by [3], allowing for frame-by-frame time navigation.

In the work by Hürst, Götz and Jarvers [4], three different approaches have been outlined in their 2004 paper. The authors proposed three UI designs for timeline interaction. A first UI, called *ZoomSlider*, allows for finer granularity when browsing as the pointer or mouse cursor moves vertically. The higher the pointer, the larger the interactable timeline becomes, allowing for more precise navigation. A second UI, called *NLslider*, works in a non-linear fashion, with a timeline where the points in time are not uniformally distributed. The distribution can be adapted, either by user input or pointer speed, allowing for both coarse and fine video navigation. The third UI, called *Elastic Skimming*, works by allowing for different timeline scrubbing speeds, dependant on the distance between pointer and actual position of the thumb on the timeline. The authors note that all of their designs have their benefits as well as drawbacks, dependant on the use case, and that there is no "one-size-fits-all" UI design solution.

The paper by Pongnumkul et al. [5] noted that users typically engage in both fast and fine skipping during timeline scrubbing. Their proposed design works by adapting the *Elastic Slider* function from [4], increasing playback speed of the timeline scrubbing dependant on the position of the mouse cursor and the timeline thumb. Additionally, the system employs a dynamic timeline speed, where the video will start jumping to predefined points of interest for scrubbing speeds above x2, allowing a user to make out the content before jumping to the next position. This way, the system is able to achieve speed of 8x original speed while still being able to convey relevant information and support the user in his time navigation.

The problem of a small timeline also extends to the thumbnails displayed when time scrubbing large videos. Small scenes are not displayed when jumping multiple frames per pixel, making it impossible to find those scenes when using these thumbnails to navigate through the video. The paper by Matejka, Grossman and Fitzmaurice [6] and their approach, called *Swifter*, tackle this problem by extending the displayed thumbnail. In the proposed interface, a large number of frames are displayed, using the whole video screen as space to display up to 12x12 interacteable thumbnails. A user study found that Swifter significantly improved search task completion time as well as general user satisfaction and preference, compared to the traditional timeline and thumbnail design.

Another approach to video navigation is called *direct manipulation* [7][8][9]. In this approach, the video can be manipulated directly by dragging the content in order to navigate through it. This *space-centric* browsing (as compared to traditional, *time-centric* browsing) is a way to interact directly with the video content by matching the dragging motion to the motion of the content of the video.

In order to keep the cursor movement in line with the video motion, [7] introduces *relative flow dragging*. With this method, a user can manipulate an object along the predefined trajectory as well as account for object deformation and moving backgrounds. The motion data is extracted by the player directly, and a visual hint of the path of motion is portrayed when the dragging cursor moves too far from the object trajectory. The background is also stabilized, compensating for camera pans.

Another video navigation method, employing the same general concept of direct manipulation, is by Karrer et al. from 2008 and called *DRAGON* [8], which stands for *draggable object navigation*. The objects in question behave as if on rails, attached to the cursor via rubber band. In their own study, they demonstrated this method to be between 19% and 42% faster for search tasks than the traditional timeline design. The authors argue that DRAGON is specifically useful for in-scene navigation, compared to broad scene location. They later extended their work in 2009 with *PocketDRAGON* [9], a direct manipulation interface for mobile devices. Since traditional timelines and UI elements in general tend to take up valuable screen space on mobile devices, the implementation of the direct manipulation has added benefits on mobile devices. The computationally expensive optical flow extraction is offloaded to a dedicated server in order to work on small, less powerful devices introducing only a small and negligible amount of delay.

Other approaches to improve and adapt video manipulation involve more elaborate systems, such as a paper by Nguyen, Niu and Liu [10] from 2012. Their interface for 2D video summarization and navigation, called *Video Summagator*, creates a 3D timeline of a video using object detection. A 3D cube is created, showing the movement of objects in the video along their trajectory, creating a visual summary of the video. This summary allows for a general overview of the video content which conveys visual information which would otherwise be easy to miss. The cube can be rotated and interacted with, allowing for a user to navigate to a specific point in the video by selecting a frame of an object along it's trajectory. These keyframes can be sampled with a controllable sampling rate, or based on the amount of motion happening in a scene. In the case of a moving camera, the cube becomes a panoramic view, stitching together shots to create a concise 3D image of the moving background.

Another timeline design which questions the traditional, horizontal bar design is called *Scrubbing Wheel*, outlined in a 2015 paper by Schoeffmann and Burgstaller [11]. Current video navigation follows the same design principles as desktop video navigation.

Due to the small screen space of mobile devices, this method of video navigation can become tiresome and inefficient, especially for small timelines in portrait mode. The Scrubbing Wheel is based on the clickwheel of the first iPod generations, which had an analog interaction wheel, used to scroll through music lists and later on through video timelines. This recent implementation of the design creates a similar surface for interaction on a touch screen display, allowing for forward/backward scrubbing by wiping the circle (counter-)clockwise. In a user study, the authors found that this implementation increases the performance for video search tasks as well as reducing the mental and physical workload for these tasks. Participants also rated this method higher in terms of overall fun and support compared to the traditional timeline design.

Many of the controls and functions for traditional 2D videos can be adapted and implemented into the context of VR for the use of 360° videos. However, the unique aspects of 360° videos, such as the need for spatial navigation and the lack of a mouse and keyboard, need to be taken into consideration when adapting these functions. In this work, the focus lies on spatial navigation and rotation in 360° videos. For this purpose, other video controls like a timeline and thumbnails are only implemented in a basic fashion. This allows for basic video interaction like time scrubbing which is needed for performance analysis such as search tasks. It was however deemed relevant to research and understand both basic and elaborate video controls. Existing control methods need to be taken into consideration to avoid creating confusing, overlapping or conflicting control schemes.

A.2 Studies on 360° video viewer behaviour and proposed guidelines

The following section will outline findings and derived design conclusions from studies relating to viewer behaviour and experience when consuming 360° videos. The focus is mostly on movement along the spatial dimensions of yaw, pitch and roll, but also on effects like presence, immersion and interactivity. It is argued that these last aspects are the main benefits and unique characteristics of these types of videos. Viewer behaviour with this relatively unfamiliar medium can give interesting insight into how humans tend to interact with 360° videos, different display methods and interaction controls in general. Based on these findings, a number of guidelines have been proposed in an effort to help developers and film makers make their applications and videos as user friendly and enjoyable as possible.

User behaviour is an essential aspect when designing VR applications. Studies by Wu et al. [12], and by Corbillon, Simon and De Simone [13] in 2017 aimed to create extensive data sets which focus on extracting viewer behaviour and movement data while watching 360° videos. While their main contribution was the creation of these data sets, they also gave analysis examples with relevant findings. For example, in the study by [12], the proposed software extracted movement data for movement along the three dimensions of yaw, pitch and roll for viewers watching 360° videos with a HMD device. Their analysis of the results show that viewers tend to move their head mostly along the yaw dimension, with up to 330° on the horizontal plane being explored regularly, and more than half of the watch time spent outside of the front view. While [13] noted that behaviour tends to vary between viewers as well as the different types of videos, these aspects are observed throughout numerous studies.

For example one study from 2018, Mathis and Victor Almquist [14] analyzed viewer behaviour when watching 360° videos to possibly try and reduce streaming data and 360° video file size. The viewer movement patterns when watching a 360° video using a HMD were analyzed, along the dimensions of yaw, pitch and roll, as well as velocity in degrees per second. Different categories of videos were tested:

- Exploration: No particular point or object of interest.
- Static focus: The point of interest is always at the center and does not move too much.
- Moving focus: The point of interest moves around the entire 360° degrees.
- Rides: The camera is moving forward at a high speed, for example a car ride or a roller coaster.
- Miscellaneous: 360° videos with a mix of the previous types.

The study showed that the yaw direction ("left and right") was the most dominant rotation direction, with angle changes around +/- 60° and up to 90°. The other dimensions roll and pitch were mostly centered around 0°. The most changes in orientation were done for exploration and moving focus video types. In exploration, the viewpoint tends to change and vary a lot, since there is no defined focus point and participants tend to look around and explore at their own pace. If a point of interest is present, like in the static focus and rides videos, participants tend to adapt their viewpoint so the point of interest stays relatively close to the center of their field of view (+/- 30°).

The authors also noted the presence of an exploration phase, in which the viewpoint varies a lot more in all dimensions with a relatively high velocity in the first 20 to 50 seconds of the video. Even in this phase, the most extreme angle change is happening along the yaw dimension.

Another study by Duanmu et al. from 2018 [15] confirms the existence of this initial phase of exploration. In their paper, they created and analysed a data set of behavioural data of users watching a 360° video on a 2D desktop. The sample videos used cover a variety of topics, such as sports, movies, stage-performance, gaming, etc. These videos were played using a custom streaming application, and the extracted data was the movement in both yaw and pitch dimension, along with the corresponding timestamps. After the initial exploration, defined by a scattered focus and exploration of the environment, the viewer focus is primarily driven by the video content type. For example, game and sports videos tend to have a single focus point. Other videos, like city tour, have points of interested distributed all around the horizontal (yaw) axis, leading to more dynamic viewer focus. The authors noted that in general, this horizontal movement is more common than vertical (pitch) movement. Additionally, most movement happens slow ($\pi/2$ per 2 seconds or slower). The authors also compared their findings against HMD data sets from [13], noting that desktop viewers generally perform slower, less frequent movement than HMD viewers.

A number of studies focusing on 360° video viewing were done with close cooperation with participants in order to analyze their reactions and preferences when it comes to watching and enjoying 360° videos [16][17]. In these human-centered design studies, participants acted as development partners. Overall, participants noted the desire for a high level of control over the video, especially for fast-paced videos like sports videos, for example by quickly jumping between predefined camera positions instead of orientation by dragging a mouse across a 2D screen [16].

Other feedback included the sentiment that too much control and interaction can distract from the viewing experience [17]. Certain videos encourage a more passive, *lean-back behaviour* as defined by Katz [18], to which viewers are already used to from their previous TV watching habits. The use of too many different interaction methods can interfere with this viewing experience and act as a distraction.

In general, the findings of these studies point to responsive, accurate and fast spatial navigation methods. Jumping between predefined viewpoints and zooming in allows viewers to effortlessly follow the main point of interest of a 360° video without getting lost.

Other studies focused on analyzing differences in viewing devices on task performance and user preference [19][20]. The tasks in [19] are *search tasks* which involve counting different people over a number of videos. Devices like HMD, tablets, desktops and smartphones were compared. Generally, participants noted that HMD had the highest performance for search tasks, followed by desktop devices. Tablets generally performed badly, and participants quickly adopted the touch-based rotation instead of the orientation based method, which was described as tiring and cumbersome. Tablets were also performing worse than HMD and desktop for complex search tasks.

Participants also had higher engagement and explored the 360° video environment more when using HMD devices compared to smartphones or tablets [20]. However, this came at the cost of HMD devices being cumbersome and not ergonomic after prolonged use. Additionally, HMD devices have the added possibility of inducing cybersickness [21]. This leads to a negative bias towards HMD, with participants feeling less competent when using them, even though studies show that they are used as accurately as desktop devices [19]. Other notable study feedback was the negatively perceived low resolution of HMD devices as well as the need for a mobile, rotating setup like a swivel chair to allow for optimal navigation.

Additionally, [20] also noted the presence of the initial exploration phase, similar to [14] and [15].

The effect of chair type has also been explored in the context of 360° video viewing experience. In a study from 2018, Hong, MacQuarrie and Steed [22] did a study on viewer behaviour using different chair setups. They tested three chair conditions: *fixed*, which did not allow for any rotation, *half-swivel*, which allowed for rotation of +90° and -90° for a total of 180°, and *full-swivel*, a chair which could rotate the full 360°. A range of video types were used for the study, including music videos, documentaries, dance, narrative, among others. The authors measured the level of exploration using eye tracking as well as the mean rotation away from the initial center position. In addition, effects on spatial awareness, incidental memory and engagement were analyzed, as well as cybersickness. The results showed that viewers on the fixed and half-swivel chairs focused on the front part of the video, while viewers on the full-swivel chair tended toexpand their focus of attention along the horizontal (yaw) axis significantly more. Additionally, viewers on the fixed and half-swivel chairs reported higher fear of missing relevant content compared to the full-swivel group. The full-swivel group also performed better on spatial awareness tasks than the other two groups. The ability to rotate leads to more exploration as well as better spatial understanding of the video content. There was no significant difference in engagement or cybersickness. In a final interview, study participants on the fixed chair condition also reported physical discomfort when needing to turn their bodies and heads, which kept them from exploring. Fixed and half-swivel group participants also noted difficulties tracking moving objects throughout the video.

These ergonomic problems of HMD devices have been explored by other studies focusing on physical and visual strain and discomfort [23][24]. In 2013, Wille, Grauel and Adolph [23] performed a study analysing physical and mental strain during task completion using a monocular HMD device, compared to a 2D tablet PC. The HMD device was fixed to a head carrier, had a front camera and a headset and weighed 380 grams. The task was to assemble a virtual model car using building bricks, while observing vertical bars on a second monitor and tracking their position. The authors found significantly higher strain for the HMD device as well as increased visual fatigue. Taking off the device during breaks alleviated these effects.

Similarly in 2018, Duanmu et al. [24] analyzed the muscle activity when performing typical document editing tasks with a HMD device compared to a desktop screen. The HMD device was a HTC Vive, weighing 660 grams. The task was to edit virtual documents for 60 minutes at a time. Neck and shoulder activation were analyzed as well as head tilt. The HMD condition had significantly higher neck activation and mean head tilt in both yaw and pitch dimensions compared to the desktop condition. There was no significant difference in shoulder activation.

Participants were more productive on the desktop device and also noted higher levels of subjective discomfort which increased over time. The authors speculate that more advanced HMD devices with higher resolution and lower weight might reduce these problems in the future.

Another important part of VR experiences is the level of immersion. In a study from 2018, Harth et al. [25] studied the effects of VR immersion on users, and found that the level of immersion is largely dependant on the viewer themselves. Study participants had to explore different virtual environments. In the qualitative analysis, the authors found that different participants had varied, individual experiences, differing significantly from other participants in the level of immersion and perception. Even though there was a lot of common ground as well, these differences highlight the level of individuality each user has when interacting with VR, and the need for a certain degree of customization in application design.

A number of studies have also focused on viewing experiences using specifically VR environments and head-mounted displays. One study from 2014 by Vosmeer and Schouten [26] analyzed the impact of this new technology on the narrativity, engagement and interactivity of both video games and movies. For video games, the engagement is generally more *lean-forward*, while movies usually offer a more passive, relaxed, *lean-back* experience [18]. VR offers a new perspective into these established experiences. For general storytelling in a 360° video, the viewer is generally not able to control the movement through space. They can experience in the position of a passive observer (fly-on-the-wall), or can be included in a more immersive way, with events and characters in the video addressing the camera position and by proxy the viewer. Another possible avenue especially suited for VR are *scenescapes*. Adding VR to convey difficult scenes, like dreams, also fits well into the concept of transmedia storytelling as outlined by Jenkins [27]. The authors concluded that the interactivity in a VR movie or 360° video is neither a completely lean-back, nor lean-forward experience.

In 2016, Argyriou et al. [28] analyzed the challenges and benefits that come with creating gamified 360° videos. The goal of these videos is to increase immersion and engagement compared to traditional 360° videos by adding other humanoid characters, missions and interactable elements into the video. The authors noted that other elements, such as UI or virtual 3D elements, should be added in a non-intrusive and non-distracting way. These elements can make a scene more interactable and immersive. Additional UI elements like a 2D radar minimap can help with the feeling of presence and orientation. The authors outlined an exploration scenario at a virtual cultural heritage site as an interesting and engaging use case.

However, combining 2D and 3D for virtual videos can create problems in other ways. Previous studies, for example by Navqi et al. [29], found that viewing a 360° video with 3D technology could lead to increased cybersickness, nausea and general visual strain. However, these findings could not by a study from 2016, where Bessa et al. [30] analyzed if a full 3D 360° video offers a better viewing experience in VR than regular panoramic videos. Their conclusion was that 3D viewing technology does not significantly increase presence, immersion or cybersickness. The research on this seems to be inconclusive, as another recent study from 2019 [31] did not find any significant differences on cybersickness between 2D monoscopic and 3D stereoscopic 360° videos.

The findings in this section lead to two main conclusions which will be addressed by this work.

Firstly, when watching 360° videos, viewers tend to navigate and explore mostly along the yaw axis. Almost the entire range is explored regularly, showing the importance for easy and comfortable rotation along this axis. Navigation along the pitch axis, while also present, tends to be less prevalent, and it can be concluded that navigation along this axis is probably not needed as much and deemed less interesting by viewers. Rotation along the roll axis seems to be not relevant, since it is a movement that is either very hard and limited to execute (in the case of HMD devices) or not enabled at all (in the case of desktop devices).

These types of viewer behaviours are unique to 360° and panoramic videos, and traditional control methods outlined in A.1 do not address the need for this type of navigation. Because of this, our work will focus on analyzing spatial navigation, most importantly rotation along the yaw axis, through the use of different control methods, other than the standard method of head movement.

Secondly, the findings of the outlined studies show the difficulties when using head-mounted display devices for prolonged periods of time for a variety of tasks as well as video viewing. The combined weight and center of gravity of the device leads to physical discomfort, especially when the viewer is in a stationary position and needs to use constant head movement in order to navigate and rotate themselves. This discomfort inhibits exploration and reduces the amount of exploration during 360° videos, leading to missed content and reduced enjoyment.

This aspect is unique to HMD devices, leading to the conclusion that these types of devices have a unique need for facilitating and allowing spatial navigation other than head movement. Because of this, our work will focus on these types of devices.

A.3 Guidance for Virtual Reality Content

Guidance is an important topic for virtual reality video content. Different methods exist to guide viewer attention to allow for cinematic experiences without taking away control. Keeping viewer attention focused on specific points of interest during a movie or video is crucial for enjoyable and convincing viewing experiences. The freedom of exploration brings with it the risk of viewers missing key narrative and visual elements, so the goal is to guide a viewer in an enjoyable and non-intrusive way.

Multiple studies have investigated methods of guiding user attention using different visual and audio cues [32][33][34]. Audio cues come in the form of both diagetic and non-diagetic sound, either giving a viewer a spatial, 3D audio hint as to where to look, or giving them outright verbal directions (”look left / look right”) [33]. Visual cues can come in the form of directional arrows, border squares around a point of interest, or other symbols. These two forms of guiding viewer attention can also be combined in an effort to make a stronger incentive for a viewer to change their viewpoint. The main goal of these cues is to have minimal impact of the video itself and the viewing experience while still effectively capturing attention and guiding the viewer effectively to their respective point of interest.

Another method of VR guidance takes shape in the form of autopilot. Here, the viewer's viewpoint is automatically changed to center on a point of interest in the video. There are certain benefits to this method. For example, participants of a study comparing autopilot to visual guides described the effect as positive, saying that it was improving their enjoyment for sports videos, reducing the need for them to look around on their own [34]. Other participants noted that this automatic viewpoint change caused nausea, dizziness and disorientation, a sentiment shared by participants from a study from [33]. The addition of a black screen during the viewpoint change was helpful, making the whole process feel more like a transition from one scene to the next, but was still perceived as negative, compared to audio and visual guides. Another finding from these studies was that this kind of autopilot was mostly useful for fast paced videos, like sports videos, but not enjoyable for slow, explorative videos, like for example city tour videos.

One other way of automatically changing the viewpoint was done by having a motorized swivel chair carry out the task of rotating [35]. The act of physically rotating a viewer reduced the induced cybersickness, to the point that it was no longer increased compared to the baseline. Additionally, study participants noted that their presence and enjoyment of the video were increased with the addition of the automatic viewpoint change from the motorized swivel chair, compared to having to look and search for the points of interest themselves.

Automated guidance is one way certain 360° videos and applications address the problem of rotation. They allow a person to remain stationary without missing essential content and without the need to manually rotate themselves.

However, they require either additional software or manual video annotation in order to extract predefined points of interest. These POI are chosen beforehand, meaning they can be more or less relevant for different viewers. Additionally, they do not address the need for autonomous exploration, and the guidance cues (visual or auditory) can distract from the actual content of the video.

While being relevant to the understanding of current 360° video player applications and possible solutions to the rotation problem, we do not employ guidance for VR in this work.

A.4 Interactions in Virtual Reality

Numerous methods for interaction, navigation and orientation have been tested in and proposed for virtual reality. Methods like controller input and gestures are used alongside other tools, widgets and virtual UI elements to interact with. Studies on these methods also reveal guidelines for designers and programmers to consider when implementing VR applications and interaction methods.

Interaction in VR can occur via head rotation, gestures, controller input and UI elements, often creating a type of *reality-based interaction* [36] which aims to recreate natural, intuitive motion, increase learning speed as well as reducing information overload.

In 1997, Chris Hand [37] conducted a survey of 3D interaction techniques in virtual reality. Since spatial navigation in virtual reality has always been a feature which is difficult achieve, *action-at-a-distance interaction* (AAAD) is necessary for object interaction in VR. The most prevalent method is *ray* (or laser) *casting*, which sends a line out from the user's controller to intersect the object to interact with. These types of interaction methods are essentially evolved mouse techniques (point-and-click) which have been the norm for desktop interaction for a long time.

The efficiency of these interaction methods is partially dependant on user preference, and many systems allow for some degree of customization in order to accommodate for this. Still, small objects, such as small and cluttered UI elements, are generally difficult to select and even harder to dwell on.

Different implementations of the traditional raycast method try to tackle these problems [38][39][40][41][42]. Selection can be facilitated using additional object score functions, like in the method *IntenSelect* [38]. Here, a cone selection is used for general selection, in combination with a scoring function with is used to disambiguate close objects. To avoid fast switching between close objects, the scores are accumulated over time, and the object with the highest score is selected, using a bending ray snapped to the target as visual feedback.

This model is further extended in 2006 [40], proposing an enhanced cone selection algorithm with a selection field and influence field. The narrow selection field allows to deselect objects easily, while influence fields allow for disambiguation and specific selection. These influence fields (originating from the objects themselves) are dependant on volume, with smaller objects having larger influence fields compared to large objects. This allows for easier selection of smaller or partially obscured objects.

Another pointing technique based on raycasting is called *raycursor* [42]. A small sphere on the raycast line, acting like a cursor, can improve target selection is added, acting like a selection cursor. This cursor is manually adjustable along the ray by a user using the controller thumbstick. The study compared this implementation of the raycursor against a traditional raycasting technique, and found that while raycursor had comparable selection times, the error rates were significantly smaller across different conditions.

In another study by Pfeuffer et al. [41], the combination of pinching gestures with eye gaze was analyzed in the context for VR selection. The eye tracking is used to locate and select the target, and pinching gestures are used for the actual object manipulation. This selection and interaction can be added into existing UI methods and has the advantages of not requiring controllers or mid-air gestures, removing excessive jitter and strain present when using other, less comfortable interaction methods.

The effect of jitter and accuracy issues for ray casting is explained in a paper from 2020 by Wolf et al. [43]. The effect of pushing a controller button when using that same controller to keep a ray on a selected target can cause this ray to slightly move, resulting in errors. This effect is also called the *Heisenberg effect* on spatial interaction. The force of a button press has the potential to displace a selection ray. The resulting errors are called *heisenberg errors*. To avoid this kind of error, targets should be designed with a radius larger than the potential *heisenberg magnitude* (size of heisenberg error, difference between original aim position and final selection point at button press). In a user study, participants had to perform multiple pointing tasks with both stationary and ballistic targets. The authors noticed a systematic upwards shift in during button presses. This effect was observed to relatively large compared to normal hand jitter, and especially higher for smaller targets. The authors outline some compensation strategies that can be employed to solve this effect, for example shortening click duration or using a correction function. Different options come with different benefits as well as drawbacks.

Another selection method is the so called *crossing paradigm*. This method was first introduced for the context of virtual reality interaction in 2019 by Tu et al. [44]. The crossing selection works by selecting an object by crossing its boundaries by a selection ray or cursor. This paradigm is often included alongside other selection methods, and the study outlined in this paper focuses on selection in VR. The authors found that crossing selection generally works faster, or at least not slower, than traditional raycast pointing, especially for targets further away. The depth level still has a negative impact on accuracy, with higher depth correlating with a higher error rate. The authors also found that crossing can be modelled by *Fitt's law*, a model for human movement adapted to the field of human-computer interaction by MacKenzie in 1992 [45]. The authors suggest that point selection in VR is difficult and unlikely for small bars and discs, and should be selectable using the crossing method. Small objects should be made crossing friendly by modelling them like 2D surfaces or 1D bars.

The crossing paradigm was analyzed further for selection in VR by Huang et al. [46], analyzing other aspects like possible feedback methods and handling occlusion.

On the topic of interaction fidelity, McMahan et al. described the existence of an *uncanny valley effect* for VR interactions [47]. They analyzed user studies involving natural, semi-natural and non-natural interaction techniques in terms of interaction fidelity (degree of exactness with which real world actions are reproduced in an interactive system). They found that natural interaction techniques generally perform the best, followed by non-natural interaction techniques. Semi-natural interaction techniques performed the worst overall. This is most likely due to the fact that these interaction techniques are neither natural enough to be intuitive, nor abstract enough to be simple and easy to learn.

Interaction in VR is a relevant aspect that needs to be considered when implementing any sort of VR application, including a 360° video player. Our work will take many different interactions methods from the general VR context into consideration while focusing on the most popular ones. The effects of jitter and selection errors will be taken into consideration when necessary. The goal of this work is not to develop new interaction methods per se, but to use tested and established methods and apply them to new control schemes for 360° video players. To this end, we focus on the most popular interaction methods as to find controls that are intuitive enough to use and easy to implement with existing VR technology and interaction design.

A.5 Design and interaction challenges for 360° Videos

The unique way virtual reality combines different media aspects [27] means that applications, such as 360° videos, have multiple design aspects that need to be considered. This can be done by distinguishing between the experience design layer, involving the different media resources and narrative structures employed, and the interaction design layer, as outlined by [48]. In this section, the focus will be on the latter aspects.

For the interaction layer, numerous methods for interaction, control and navigation exist in the context of 360° videos. Most of these are a combination of established video and VR interaction controls outlined in subsection A.4. More recent implementations try to combine the benefits of VR for the context of a video player, in order to create more specified interaction methods, taking full advantage of the medium.

The most common method for orientation, rotation and spatial navigation is done via head movement in the case of HMD devices. This has some benefits, as the head movement is generally seen as a very intuitive and natural way of changing the viewpoint in a virtual reality setting. Certain aspects of this interaction can be adapted and made more powerful, for example in [49]. Here, the head movement is amplified to allow for larger rotation along the yaw-axis while requiring less head movement. As HMD devices can be heavy and cumbersome, reducing the need for extreme movement is seen as a benefit and allows users to fully take in the surrounding video content without having to physically turn around all the way.

Other methods use head rotation for selection and interaction in a different way. Head movement can be interpreted like input for selection, for example in [50]. A pointer following the head movement can be used to interact with traditional video UI, requiring an additional button press to activate a function. The reliance on a controller can be an issue when using a HMD, as argued by [51], since the controller cannot be seen and has to be interacted with without visual feedback. In [51], selection and interaction can both be performed using eye-based head movement. The eyes stay fixed on a target, while head movement (for example nodding up or down) indicates interaction. It is argued that these types of interaction techniques should however not impede the overall viewing experience.

While head-based interaction seems to be intuitive and natural for orientation and rotation, the effectiveness of these methods on timeline interaction and video controls seems to be somewhat lacking. Other input methods, like traditional raycasting on UI elements and controller input, such as in [50], seem to work faster and more accurately. For this reason, time navigation and spatial navigation are decoupled most of the times, such as it is proposed by [52], and hand movement in the form of controller input or gestures [53] are used for the video control.

One common method is to use hand gestures for video control [50][52][53]. The goal is to find gestures that are simple yet distinct, do not interfere with the viewing experience and are easy and comfortable to do over an extended period of time, lest they become physically straining [54]. A study from 2014 [53] let participants interact with 360° videos in a CAVE setup (dome structure with the video projected onto it from the inside). Participants were asked to come up with control gestures themselves. The results showed that viewers generally prefer dynamic, linear, and coarse gestures. The authors also noted that participants had strong mental models from previous desktop experiences, which had a large influenced their gesture interaction, a finding which had previously been explored by [55] in the context of 2D surface computing. Here, 72% of the user-created gestures were closely modeled after mouse interactions. One other example of such a mental model is the left/right motion for time control, which is seen in many UI designs and video control schemes [52][56][57].

In general, mental models have a large impact on how natural and easy to learn a gesture for human-computer interaction is [58], and are important to consider when designing interaction methods in VR.

Other interaction methods adapt previous experimental methods of direct interaction into the VR context, such as [59]. Even in a scenario without a fixed position (such as a 3D animation video), spatial relations can be difficult to track and follow during the entirety of the video. Adding a direct manipulation interaction method allows users to navigate a scene based on any object that has a trajectory. Similar as in [8], a viewer can grab and drag an object along it's trajectory, getting information about the state and time of the object in question. These interaction methods are especially effective for search tasks, since they allow to visualize and jump to any point of a desired object's path in a video. This amount of control is generally seen by viewers as very positive in 360° video interaction.

Previous work already analyzed different types of timeline designs for possible improvement on task performance, temporal relations and user preference [57]. Multiple designs are possible to display a temporal timeline on a 2D surface. The traditional, horizontal bar can be flipped to be vertical instead, with time either starting from the top or the bottom. Both design ideas can have corresponding mental models (time flowing down, like sand in an hourglass, or upward, like time progression on an evolutionary tree). Other designs include circular or spiral designs. Studies outside of the VR context showed that participants generally prefer linear timelines over circular or spiral designs, and are also faster at reading and navigating these designs [57]. These design choices are interesting to consider when implementing new control schemes and possible UI elements for 360° video players.

In the context of VR, the current standard for timeline design is almost the exact same as for desktop applications, meaning a 2D bar going from left to right, with a thumbnail displayed during time scrubbing to show the video content at the selected time to facilitate search tasks and orientation. The design on the thumbnail for a 360°

video is an interesting challenge all on its own. Different methods to transpose and display a 2D thumbnail with 360° video content exist, for example using an equilateral projection, used in the widely available Youtube video player. Another option is to display rounded or curved thumbnails, such as in [56]. These thumbnails lead to less distortion of the original content, to which study participants generally responded in a positive way. Another paper from 2021 [60] introduces a different timeline and thumbnail representation in the form of route tapestries. This design is especially relevant for virtual tour videos, where most of the relevant content and landmarks are located on the left and right sides, relative to the camera movement. By stitching together a flat view of both sides of the camera, this method creates two continuous orthographic-perspective projections which act like thumbnails, with a line displaying the current position of the camera. In a user study, this method had the lowest task completion times as well as the lowest error rates when compared to traditional equirectangular thumbnails and tapestries with sampled, normal field-of-view thumbnails.

The discussed work in this section focuses specifically on design aspects relevant for 360° video players. In our work, we will take inspiration from this previous work and take the outlined difficulties and drawbacks into consideration. More concretely, we will create video controls that do not rely on head movement, but instead use other tested methods like raycasting on UI elements, gestures and controller input to interact with the video. Raycasting has been shown to be efficient and easy to grasp while being prone to jitter. Gestures have the possibility to be faster and more enjoyable, provided they are simple and quick to execute. Controllers are among the most accurate, as long as they are within the reach of the viewer and can be accessed without looking at them directly.

The goal is to use existing interaction methods that are tried and tested, intuitive enough and effective. They need to make sense to the viewer on a mental level, and need to be easily integrated into the overall VR system and feel intuitive to a viewer.

Appendix B

Methodology Details

This section outlines additional and specific details for the methodology and implementation of the Masters project. Section B.1 outlines additional details related to the exact implementation of the program used in the study. Section B.2 goes into detail about the three individual rotation methods. Section B.3 outlines the data gathering process.

B.1 General Implementation

For this study, a custom 360° video player was developed. For the development, a personal laptop (ASUS ROG Strix G553V 15" Core i7-6700HQ 2,6 GHz, GTX960M) with Windows 10 Home edition was used. The program was developed using the Unity IDE [61], version 2019.4.35f1. This IDE is specialised for game development, and recent developments in development for VR have produced specialised libraries for VR support which are easy to include and adapt into Unity. In the case of this work, the XR Interaction Toolkit [62] was used (version 0.10.0). This toolkit allows for easy inclusion of a VR rig in Unity, adapting the camera rotation to the head movement of the VR HMD, and to map controller input to specific functions. In addition to the camera object, the VR rig also includes controller objects which are mapped to the physical controllers in order to copy their movement and orientation.

The program was developed specifically for the Oculus Quest [63]. This is a standalone VR device, meaning that it does not need to be connected to a PC or other source in order to run applications. Custom programs, like this project built in Unity, can be installed onto the device and executed at different times and places.

The Oculus Quest consists of the HMD and 2 controllers. The HMD has 6DOF (degrees of freedom) tracking, and a dual OLED display (1600x1440, 72Hz) with adjustable lens distance. The controllers also have 6DOF tracking.

For each rotation method, a specialised Unity scene was created. A scene is a single 3D environment which usually has its own 3D objects and scripts. In this case, each scene has a video sphere and a video player which projects the video texture onto the sphere. Inside of the sphere, the VR rig with the camera is fixed, allowing only for rotation to change the orientation of the camera. When the video is played, the 2D equilateral video texture is transposed onto the sphere, creating the 360° video effect.

Each scene has individual scripts which contain the rotation functions for the 3 methods of GUI, Thumbstick, and Dragging. The scripts were written in C# in Microsoft

Visual Studio 2019. Each scene contains the basic VR rig setup (camera, controllers, video sphere), as well as a basic video player interface. Additionally, each scene contains a questionnaire interface. After completing the tasks for each rotation method, the questionnaire is displayed similarly to the basic user interface in front of the camera. After the questionnaire is completed, the data is compiled and the next scene is loaded.

The basic interaction method in this program is the raycast method. In this method, a virtual ray extends from the controller object inside the virtual world and is used to aim at interaction targets. To interact with a target, a button on the physical controller is pressed. In this case, the trigger button on the Oculus controller is the standard raycast button. This interaction method is used to navigate from one task to the next, as well as selecting search tasks answers and questionnaire answers. This interaction method is also used in the GUI rotation method (B.2.1), since these graphical buttons are integrated into the overall UI.

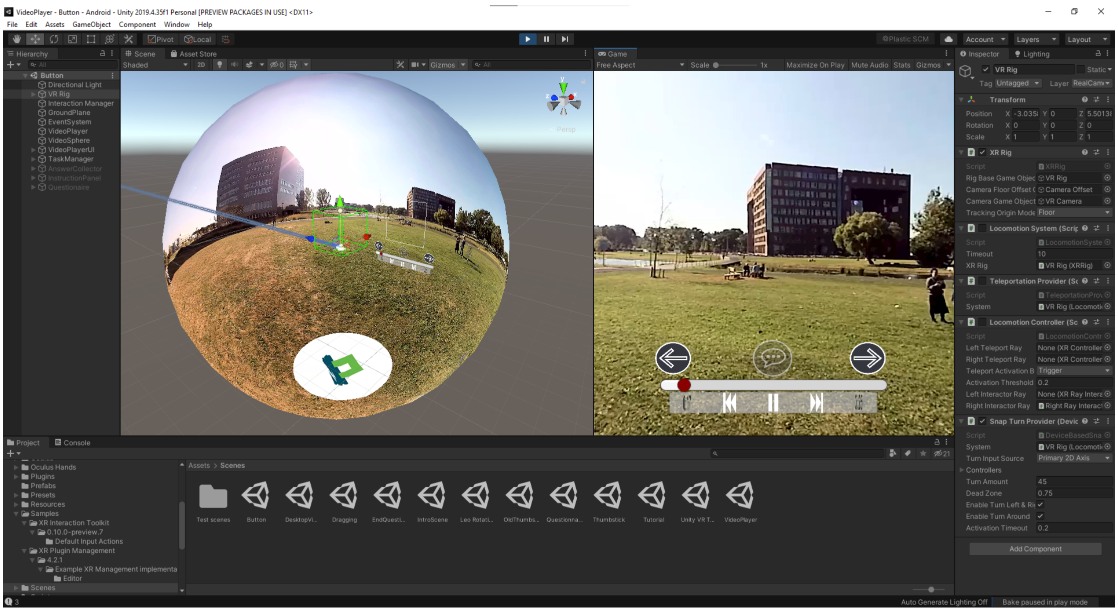

Figure B.1: Screenshot of the basic setup in Unity. The left panel shows the video sphere and the camera object inside. The left side shows the in-game view with the basic video UI and the graphical rotation Buttons.

The basic UI consists of a timeline, a play/pause button, and skip forward/backward buttons. This UI is added to the scenes in order to give the context of a basic 360° video player, as well as having a basis to integrate the GUI method. In addition, the video player UI has a button which displays the task instructions again, should the participant need to verify this information during the video. The functions of the basic UI elements are disabled during the study. The reason for disabling these controls is to make the study more uniform for every participant. In a search task, the ability to pause or rewind the video could elicit drastically different search strategies from participants. For example, one such strategy could involve pausing the video, rotating the video slowly, before playing the video again. Another strategy could be playing the video without rotating, before rewinding all the way back and rotating the video 180°. These strategies would create vastly different interaction with the rotation methods. For this reason, it was deemed to be more consistent to disable the video controls. This decision also has the effect that the study duration is more controlled and consistent.

The UI elements are set up to follow the head movement of the viewer along the yaw direction. This means that at all times, the UI is at a fixed distance in the center view of the viewer. This design choice was made to facilitate the interface in the GUI method, and to keep the user interface consistent between rotation methods for a fair comparison.

B.2 Rotation Controls Methods

The following sections outlines the implementation details of the GUI method B.2.1, the Thumbstick method B.2.2 as well as the Dragging method B.2.3.

B.2.1 GUI Control Method

For this method, two graphical buttons are displayed on the left and right side of the basic video player UI. Interacting with these buttons via ray cast rotates the video sphere around the yaw axis in either the left or right direction, depending on which button is pressed. The longer the button is pressed, the faster the video sphere rotates.

The speed is increased by 70% for every 0.5 seconds the button is held. The increase in speed is necessary to allow for greater rotations in a smaller time period. This gives a viewer more control over the rotation for small adjustments as well as being able to quickly perform large rotations. Similarly to Thumbstick and Dragging, a viewer should have the option to determine the speed of the rotation. This is not as straight forward with a graphical interface, since a button can only be pressed or not pressed. Gradual pressure which could correspond to an increase or decrease in speed is not possible. Increasing the speed over time is a design choice which overcomes this issue.

Like the basic video UI, the graphical buttons also follow the head motion to stay in the field of view at all times. This design choice was made to facilitate interaction with the GUI method while moving the head to look around the video. The buttons are always in the field of view, and a viewer does not have to move their head back to the initial starting position to find the buttons if they want to perform a rotation. During the button press, the GUI remains stationary. This way, the viewer can still move their head freely and independently while keeping the rotation buttons pressed. After releasing the button, the GUI smoothly returns back into the center of the field of view.

The results and feedback (E) show that this design choice did not have the intended effect of facilitating the video rotation. Instead, study participants noted that having the GUI move around with their head movement made it harder to track them in 3D space, since small head adjustments required to identify and target the buttons made them move away again.

B.2.2 Thumbstick Control Method

For this method, the physical thumbstick on the right Oculus controller is used to control the video sphere rotation. Tilting the stick towards either the left or the right side will rotate the sphere in the corresponding direction. The thumbstick input returns a 2D vector. The x and y values of this vector are between -1 and 1, depending on the direction and the amount of tilt of the thumbstick. Putting the thumbstick all the way to one side will return either $(-1, 0)$ or $(-1, 0)$, the center position returns the vector $(0, 0)$.

The x value of the thumbstick is plugged into the rotation function which rotates the sphere. This allows the thumbstick to control the direction of the rotation (negative value for left rotation, positive value for right rotation), as well as the speed of the rotation. The higher the tilt value of the thumbstick, the faster the sphere rotates, up to a fixed speed in the case of a full tilt. This method allows for fast rotations as well as very small, precise rotations, depending on the amount of tilt the viewer puts on the thumbstick.

B.2.3 Dragging Control Method

For this method, a viewer presses and holds the "Grab" trigger button (usually accessed by the middle finger on the side of the controller) and drags the controller to the left or right. The video sphere rotates in the corresponding direction. This creates a gesture which enables the viewer to rotate the sphere at the same speed and for the same amount as the movement of the controller / hand.

The Dragging method works by tracking the initial and current orientations of the video sphere and the right hand controller for the yaw direction in 3D space. Once the Grab trigger is pushed, the function calculates the difference between this initial controller orientation and the current controller orientation. The video sphere orientation is then updated by subtracting this difference from its initial orientation. This results in rotating the sphere in real time with the hand motion, giving the feeling of a natural, direct interaction. The speed and rotation amount are dependant on the viewer's hand motion, allowing for an intuitive understanding and feeling for the motion which allows both fast and precise rotation.

Since the rotation of the sphere is based on the local orientation of the controller, a viewer does not need to put their arm out in front in order to perform the dragging gesture. Even if the controller is at a downwards angle, the correct rotation is still applied. This reduces the physical stress extending the arm over a longer period of time and avoids the so called "gorilla arm" effect which usually accompanies gesture controls.

The rotation of the video sphere starts after a threshold of a 5° difference has been passed. This is done to avoid micro rotations which can be induced from hand shaking. This design decision was made to increase the accuracy of the rotation. However, the threshold was large enough to be noticed by some participants, and received some negative feedback (outlined in section E).

B.3 Data Gathering

In order to evaluate the performance and viewer behaviour during the search tasks, quantitative data is gathered during the experiment. This data gathering is done in the background of the program while the search task is performed. Each search task is counted as one *run*, and data is gathered for each run individually.

B.3.1 Types of Data

The main types of quantitative data are *rotation speed*, *individual rotations*, *head movement*, and *view time distribution*.

For *rotation speed*, the maximum and average rotation speed is calculated. Speed is defined as degrees over time, meaning the distance (on a circle, in degrees) rotated over the duration of the rotation (in seconds). The maximum rotation speed is defined as the highest speed for a single rotation for one run. The average rotation speed is the sum of all individual rotation speeds divided by the number of individual rotations.

For *individual rotations*, both the number of individual rotations, as well as the maximum and average rotation amount is calculated. The number of individual rotations is calculated differently for each rotation method.

For the GUI method, a rotation starts when one of the two graphical buttons is pressed. The rotation stops after the button is released. After every button release, one additional rotation is added to the count.

For the Thumbstick method, a rotation starts when the thumbstick is being tilted from the rest position ($x = 0$) to either side ($x! = 0$). The rotation stops when the thumbstick is moved back to its original position with value $x = 0$. If the thumbstick value goes from a non-zero position back to 0, one rotation is added to the count.

For the Dragging method, a rotation starts when the "Grab" button is triggered. The rotation stops when the Grab button is released, and one rotation is added to the count. Additionally, if a direction change occurs (going from rotating left to rotating right or vice versa) while keeping the Grab button pressed, the rotation stops and starts again. This is done because it is technically possible to not let go of the Grab button and perform multiple rotations to the left and right over the course of a long period of time. Counting this as one big rotation would be inaccurate, since such a movement could result in a small rotation amount (if the initial and final orientation are similar), while still having explored much of the content. It is more accurate to count the smaller, individual rotations with their individual amount. The direction change is calculated by measuring the difference between initial and current orientation of the video sphere between each frame. This returns either a positive or negative value, and is compared against the previously stored direction and updated.

The maximum rotation amount is the most degrees rotated for a single rotation. The average rotation amount is calculated by adding up the individual rotation amounts and dividing them by the number of individual rotations.

For *head movement*, the rotation of the head is measured by calculating the difference between the initial head orientation and the current head orientation. The initial head orientation is measured at the start of each task, once the instructions have been read and the start button is pressed. Both the maximum head rotation (between 0° and 180° in either direction) is measured, as well as the amount of time the head rotation is larger than 60° to either side (in seconds). 60° has been chosen as the threshold for this data point, because it was deemed a suitable midsection between 45° (not a straining head rotation) and 90° (a straining head rotation which is uncomfortable to hold for an extended period of time). It is also threshold for the binocular field of view of humans, which is considered to be 120° (from left end point to right end point).

For *view time distribution*, the video sphere is divided into 8 sections at the horizontal (yaw) axis. The position of the center line of the head orientation is measured by calculating the difference between the current head orientation and the current video sphere orientation. This results in the view angle between the two objects, corresponding to the section of the video sphere towards which the camera (center view point) is pointed at. The amount of time (in seconds) for each of the 8 sections is individually summed up, resulting in 8 data points which together form the view time distribution.

Appendix C

Study Details

The following chapter contains additional details regarding the detailed setup of the study (C.1), a listing and description of the individual videos and search tasks (C.2), the individual questionnaire items (C.3) and additional details of the study participants (C.4).

C.1 Setup

The general setup of the study is comprised of the following steps, in order:

1. Introduction to the study and signing of consent form
2. Tutorial of rotation method 1
3. Search task 1; 2; 3
4. Questionnaire relating to rotation method 1
5. Tutorial of rotation method 2
6. Search task 4; 5; 6
7. Questionnaire relating to rotation method 2
8. Tutorial of rotation method 3
9. Search task 7; 8; 9
10. Questionnaire relating to rotation method 3
11. Post-experiment interview

The order of the rotation method and search tasks was balanced using a latin square design. Additionally, the pairing of the search task and the rotation method was also balanced. Between each of the outlined steps, the participants could take a break if needed. Taking a break during a search task was not possible, since there was no way of pausing a video. On average, the experiment duration was 30.92 ± 7.37 minutes. The minimum duration was 24 minutes, and the maximum duration was 52 minutes.

C.2 Videos and Tasks

The videos used in the study were downloaded from different websites in an equirectangular projection format. They were chosen based on their similar length, file size and suitability for search tasks. The search tasks were defined based on the content of the videos.

Video Index	Video Name	Length	Task	Correct Task Answer
Tutorial	Crystal Shower Falls	00:25	None	None
1	Campus Tour Wageningen	03:25	"Count the number of beer bottles."	between 3 and 5
2	Cat Cafe Moscow	02:19	"Count the number of cats with complete or partially orange fur."	between 3 and 5
3	BrightFarms Virtual Reality Greenhouse Tour	02:18	"Count the number of people wearing gloves."	between 5 and 10
4	San José City Tour	02:14	"Count the number of flags in the video."	more than 5
5	Our City of 360	01:15	"How many cactus plants can you see in the video?"	less than 4
6	Savanna City Project	02:01	"Count the number of umbrellas you see in the video."	more than 5
7	Grayline Miami Key West	02:32	"Count the number of people wearing yellow."	between 5 and 10
8	Puerto Vallarta City Tour	01:53	"Count the number of people wearing sunglasses."	less than 10
9	Manolin Restaurant	01:50	"Count the glasses with red wine."	less than 10

Table C.1: Listing of the individual videos and corresponding search tasks.

C.3 Questionnaire

After each rotation method has been tested on three search tasks, the participants are presented with a questionnaire to gather subjective feedback. These questionnaires are the same for each rotation method and are in the form of a 5 point Likert scale with 10 questions:

1. It was easy to use the rotation controls.
2. I felt in control of the video rotation.
3. The rotation controls did what I expected them to do.
4. I felt compelled to explore the video content.
5. The rotation controls allowed me to easily explore the video content.
6. The rotation controls were responsive.
7. I knew how to operate the rotation control.
8. The rotation controls felt natural and intuitive.
9. Using the rotation controls was fun and exciting to use.
10. Using the rotation controls was demanding and tiring to use.

The data is compiled and sent to a separate Google Form, similarly to the qualitative data outlined in B.3.

C.4 Participant Details

Index	gender	age	completion time (min)	cybersickness	VR experience	Favorite
1	male	27	26	no	high	Thumbstick
2	female	25	25	no	low	Thumbstick
3	male	27	28	yes	low	Thumbstick
4	female	28		yes	very low	
5	male	29	30	yes	high	Thumbstick
6	female	27	28	no	low	Thumbstick
7	female	26	35	yes	very low	Dragging
8	female	28	48	yes	very low	Thumbstick
9	female	28	39	yes	very low	Thumbstick
10	male	28	26	no	low	Thumbstick
11	female	27		yes	very low	Thumbstick
12	male	23	35	no	very low	Thumbstick
13	female	28	34	yes	very low	Thumbstick
14	male	29	25	no	moderate	Dragging
15	male	26	30	yes	low	Thumbstick
16	male	27	27	no	very low	Thumbstick
17	male	29	24	no	high	Thumbstick
18	male	26	24	no	very high	Thumbstick
19	female	23	38	yes	very low	Dragging
20	female	33	26	no	very low	Thumbstick
21	male	27	31	yes	moderate	Thumbstick
22	male	32	27	yes	low	Thumbstick
23	male	29	33	yes	very low	Thumbstick
24	male	26	24	no	low	Thumbstick
25	female	28	52	yes	low	Thumbstick
26	male	25	27	no	very low	Thumbstick

Table C.2: List of participants. Participant 4 dropped out due to excessive cybersickness. Completion time for participant 11 is missing because of a timer malfunction.

Appendix D

Additional Results

D.1 Quantitative Data Results

For the quantitative data, each participant was tested 3 times per rotation method. One participant had to stop midway through the experiment due to excessive cybersickness, resulting in a total of 25 participants and $N = 75$ individual measurements.

D.1.1 Rotation Speed

The speed of the rotation is measured in degrees per second deg/s. Both the maximum speed as well as the average speed for all rotations is taken and measured for significant differences between the rotation methods. Data is noted as mean $\pm$ standard deviation, unless specified otherwise.

After removing the most extreme outliers, $N = 66$ data points remained. For maximum speed, the Thumbstick method achieved the highest result with 331.82 ± 47.06 deg/s, followed by the Dragging method with 324.09 ± 187.34 deg/s. The GUI method measured 87.76 ± 35.35 deg/s.

The data was analyzed for normality using a Shapiro-Wilk test and deemed to be not normally distributed. ($p < .001$ for each rotation method). A Friedman test was performed to check for statistically significant differences between the rotation methods, $\chi^2 = 84.273, p < .001$. Further analysis with Bonferroni correction reveals that there is significant difference in maximum speed between GUI and Thumbstick ($p < .001$), as well as between GUI and Dragging ($p < .001$), but not between Thumbstick and Dragging ($p = .172$).

For average speed, the Thumbstick method achieved the highest result with 98.88 ± 58.34 deg/s, followed by the Dragging method with 85.34 ± 37.28 deg/s,. Lowest was the GUI method with 36.38 ± 11.91 deg/s.

The data was analyzed for normality using a Shapiro-Wilk test. The average speed distribution for GUI was deemed to be distributed normally ($p = .076$), Thumbstick and Dragging were however not distributed normally ($p = .004$ and $p = 0.034$ respectively).

Regardless of non-normality, a one-way repeated measures ANOVA test was performed to test for statistically significant differences in average speed, since this test is fairly robust against non-normality.

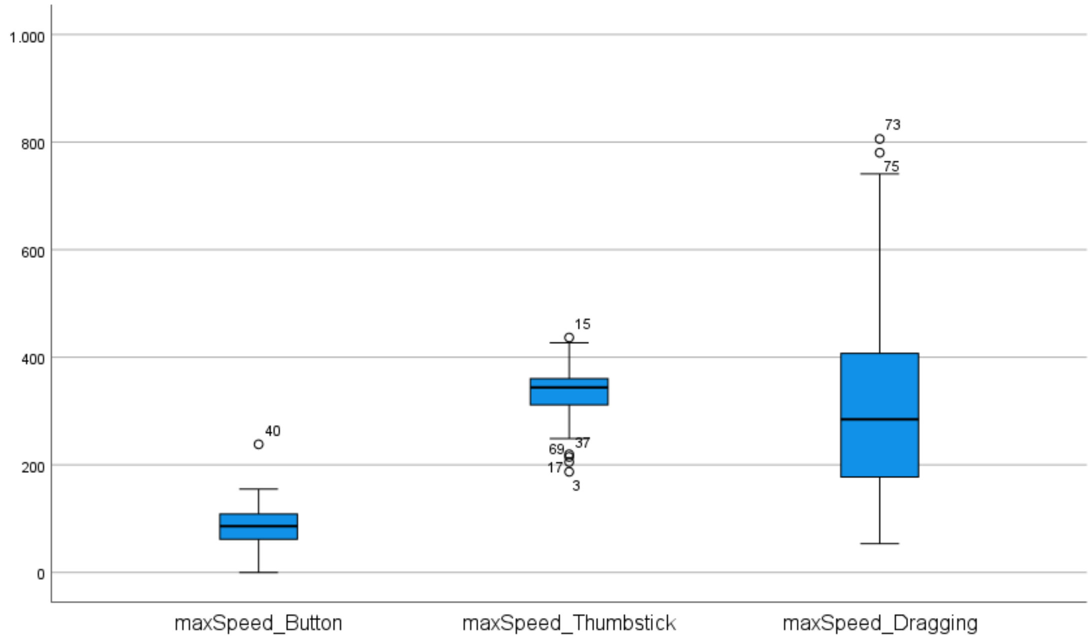

Figure D.1: Maximum speed in deg/s for each rotation method.

Mauchly's test of sphericity indicated that the assumption of sphericity had been violated, $\chi^2 = 23.579$, $p < .001$. Greenhouse-Geisser correction was applied ($\epsilon = .784$). There was significant difference in average speed depending on the rotation method, $F(1.567, 115.985) = 56.49$, $p < .001$, partial $\eta^2 = .433$.

Post hoc analysis with Bonferroni correction revealed average speed was significantly slower for the GUI method, compared to the Thumbstick method (difference of 62.50, 95% CI, 46.47 to 78.53 deg/s, $p < .001$), as well as compared to the Dragging method (difference of 48.56, 95% CI, 38.29 to 59.64 deg/s, $p < .001$).There was no significant difference between Thumbstick and Dragging (13.54, 95% CI, -4.29 to 31.37 deg/s, $p = .200$).

Additionally, a Friedman test with post hoc Bonferroni correction were performed and achieved similar results ($\chi^2 = 74.187$, $p < .001$), with average rotation speed on GUI being significantly lower than both Thumbstick ($p < .001$) as well as Dragging ($p < .001$), and no significant difference between Thumbstick and Dragging ($p = .391$).

D.1.2 Individual Rotations

Data is noted as mean ± standard deviation, unless specified otherwise.

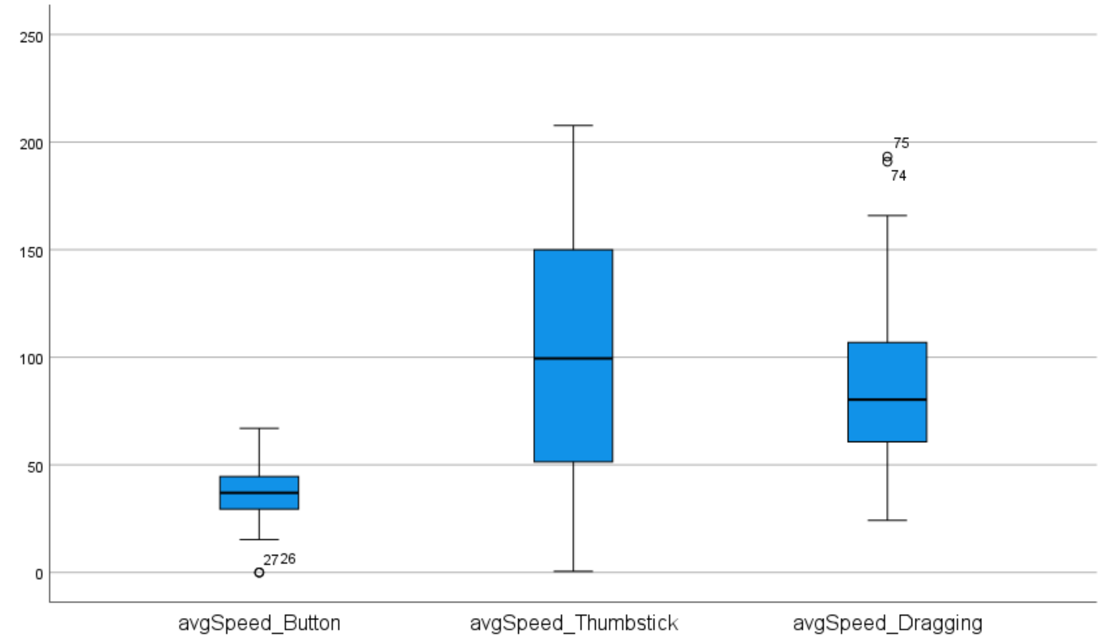

Figure D.2: Average speed in deg/s for each rotation method.

For number of individual rotations, an initial Shapiro-Wilk test revealed the data to not be normally distributed, but moderately positively skewed. After removing 3 outliers and transforming the data using the square root function, a second Shapiro-Wilk test noted the data as normally distributed ($p_{GUI} = .561$, $p_T = .473$, $p_D = .171$). The following results have been obtained by analyzing the transformed data of $N = 72$ individual observations.

A one-way repeated measures ANOVA test was performed on the transformed data. Mauchly's test for sphericity indicated that the assumption of sphericity was not violated, $\chi^2 = 3.986$, $p = .136$. Number of individual rotations was significantly different for the three different rotation methods, $F(2, 142) = 104.371$, $p < .001$, partial $\eta^2 = .595$. GUI has the lowest amount of rotations, with a mean of 39.43 ± 21.46, followed by Dragging, 104.75 ± 55.38, and finally Thumbstick with the most amount of rotations, 137.99 ± 69.26.

Post hoc analysis with Bonferroni correction revealed that the 3 rotation methods are pairwise significantly different in number of rotations ($p_{GUI;T} < .001$, $p_{GUI;D} < .001$, $p_{T;D} = .002$).

For maximum amount rotated per single rotation, measured in degrees deg, the non-linear nature of the data, combined with the high number of outliers requires a Friedman test instead of a one-way repeated measures ANOVA test. Removing extreme outliers resulted in $N = 65$.

The Friedman test showed significant differences between the three rotation methods,

$\chi^2 = 84.554$, $p < .001$. Pairwise comparison with Bonferroni correction reveals significant difference between GUI (199.87 ± 91.67 *deg*) and Thumbstick (310.55 ± 175.50 *deg*); $p = .009$, between GUI and Dragging (76.95 ± 28.66 *deg*); $p < .001$, and Thumbstick and Dragging; $p < .001$.

For average rotation (in degrees *deg*), Shapiro-Wilk revealed normal distribution after removing the most extreme outliers, resulting in $N = 63$ ($p > .05$ for each rotation method).

Mauchly's test revealed that the assumption of sphericity had been violated, $\chi^2 = 11.470$, $p = .003$. Greenhouse-Geisser correction was applied ($\epsilon = .854$).

There was significant difference in average amount rotated between the rotation methods, $F(1.707, 105.855) = 31.901$, $p < .001$, partial $\eta^2 = .340$. Pairwise comparison with Bonferroni correction revealed significant difference between GUI (50.52 ± 23.44 *deg*) and Thumbstick (34.55 ± 16.85 *deg*); difference of 15.97, 95% CI, 8.19 to 23.76 *deg*, $p < .001$; GUI and Dragging (29.07 ± 6.14 *deg*) ; difference of 21.45, 95% CI, 14.19 to 28.70 *deg*, $p < .001$; as well as Thumbstick and Dragging; difference of 5.48, 95% CI, 0.17 to 18.78 *deg*, $p = .041$.

For the direction of the rotations, the number of left and right rotations have also been measured. Shapiro-Wilk determined the data to be non-normally distributed. After removing outliers and transforming the data using the square root function, a one-way repeated measures ANOVA test has been performed for both left rotations as well as right rotations, $N = 73$.

For left rotations (L), Mauchly's test revealed that the assumption of sphericity was not violated, $\chi^2 = 3.528$, $p = .171$. There was significant difference in the number of left rotations between the rotation methods, $F(2, 144) = 72.128$, $p < .001$, partial $\eta^2 = .524$. Pairwise comparison with Bonferroni correction revealed significant difference between GUI (12.78 ± 11.04) and Thumbstick (36.08 ± 22.70); $p < .001$, between GUI and Dragging (56.60 ± 36.46); $p < .001$, as well as between Thumbstick and Dragging; $p < .001$.

For right rotations (R), Mauchly's test revealed that the assumption of sphericity was not violated, $\chi^2 = .218$, $p = .897$. There was significant difference in the number of left rotations between the rotation methods, $F(2, 144) = 18.783$, $p < .001$, partial $\eta^2 = .207$. Pairwise comparison with Bonferroni correction revealed significant difference between GUI (27.33 ± 20.27) and Thumbstick (41.37 ± 25.64); $p < .001$, between GUI and Dragging (45.77 ± 27.80); $p < .001$, as well as between Thumbstick and Dragging; $p < .001$.

Additionally, the difference between L and R rotations have been calculated (in percentage % change, positive percentage for $L > R$, negative percentage for $L < R$). Shapiro-Wilk revealed that the difference for the GUI method is non-normally distributed; $p = .019$, but the distributions for Thumbstick and Dragging are normally distributed; $p = .820$ and $p = .151$ respectively. A one-way repeated measures ANOVA test has been performed, since only one distribution is non-normal, and ANOVA is considered robust with non-normal distributions.

Mauchly's test revealed that the assumption of sphericity was not violated, $\chi^2 = 4.345$, $p = .114$. The difference between left and right rotations was significantly different for the three different rotation methods, $F(2, 142) = 20.548$, $p < .001$, partial $\eta^2 = .222$.

GUI has the has the highest number of L compared to R, with a mean of $-58.19\% \pm 93.43$, followed by Thumbstick, $-14.15\% \pm 72.20$, and finally Dragging with more R compared to L, $17.51\% \pm 63.90$.

Pairwise comparison with Bonferroni correction revealed significant difference between GUI and Thumbstick (difference of 44.04%, 95% CI, 12.26 to 75.82, $p = .003$), GUI and Dragging (difference of 75.70%, 95% CI, 46.20 to 105.20, $p < .001$), as well as Thumbstick and Dragging (difference of 31.66%, 95% CI, 6.05 to 57.27 deg, $p = .010$.

D.1.3 Head Movement

Data is noted as mean $\pm$ standard deviation, unless specified otherwise.

For the maximum head rotation, measured in degrees deg, a Shapiro-Wilk test revealed that the data is not distributed normally. A Friedman test has been performed to test for significant difference of maximum head movement between the rotation methods, with $N = 75$.

GUI had a mean of 89.98 ± 40.98 deg, Thumbstick had a mean of 73.23 ± 94.94 deg, and Dragging had a mean of 79.69 ± 33.99 deg. The test showed no significant difference for the three rotation methods in terms of maximum head movement, $p = .257$.

For the frequency of head tilted more than 60° in either direction in the yaw dimension, the GUI control method had $56/75 = 74.7\%$ cases larger than 0 seconds, with a mean of 18.19 ± 22.60 seconds. The Thumbstick method had $44/75 = 58.7\%$ cases $> 0s$, $8.59 \pm 9.83s$. The Dragging method had $51/75 = 68.0\% > 0s$, $14.43 \pm 17.86s$. A chi-square test has been performed and found no statistical significance in number of times that the head was tilted above 60° between the three rotation methods $p = .158$.

In a first test for the time spent above the 60°, only measures greater than 0 were analyzed, resulting in $N = 23$. The GUI method measured a mean of $13.58 \pm 21.05s$, the Thumbstick method measured $5.04 \pm 8.62s$, and the Dragging method measured $9.82 \pm 16.17s$. A Shapiro-Wilk test revealed the data to distribution to be non-normal, heavily positively skewed, so a log10 function was used to transform the data and to achieve normality.

Mauchly's test revealed that the assumption of sphericity was not violated, and a one-way repeated measures ANOVA test was performed; $F(2, 44) = 1.256$, $p = .295$. These results show that the differences in time spent above 60° are not statistically significant ($p = .295$).

Additionally, a Friedman test was run for all of the values, including 0 vales, for an $N = 75$, $\chi^2 = 5.610$, $p = .060$; and for the time above 60° excluding the 0 values (non-transformed), for an $N = 23$, $\chi^2 = .800$, $p = .670$; confirming the results obtained by the previous tests.

D.1.4 View Time Distribution

The time spent looking at each of the 8 sections of the video $q_0; ...; q_7$ is measured as a percentage of the whole task duration. First, the means ± std.dev. of each rotation method per quadrant is noted. Then, one-way repeated measures ANOVA tests are performed to check for significant differences in time spent per video sphere section between the three methods. The data was deemed to be distributed close enough to normality using Shapiro-Wilk to justify using ANOVA, with $N = 75$.

- q_0: GUI $11.77 \pm 0.57\%$; Thumbstick $9.59 \pm 0.47\%$; Dragging $9.29 \pm 0.43\%$
- q_1: GUI $9.11 \pm 0.58\%$; Thumbstick $9.31 \pm 0.46\%$; Dragging $10.86 \pm 0.49\%$
- q_2: GUI $9.84 \pm 5.80\%$; Thumbstick $12.30 \pm 6.20\%$; Dragging $12.48 \pm 4.31\%$
- q_3: GUI $9.73 \pm 0.67\%$; Thumbstick $14.76 \pm 0.62\%$; Dragging $14.90 \pm 0.70\%$
- q_4: GUI $10.52 \pm 0.47\%$; Thumbstick $20.57 \pm 1.11\%$; Dragging $18.74 \pm 1.14\%$
- q_5: GUI $17.64 \pm 8.72\%$; Thumbstick $11.99 \pm 5.79\%$; Dragging $11.45 \pm 4.31\%$
- q_6: GUI $18.60 \pm 1.11\%$; Thumbstick $11.83 \pm 0.47\%$; Dragging $13.72 \pm 0.70\%$
- q_7: GUI $12.90 \pm 0.50\%$; Thumbstick $8.59 \pm 0.38\%$; Dragging $8.54 \pm 0.36\%$

q_0: Mauchly's test revealed sphericity was not violated ($\chi^2 = 1.266$, $p = .531$). ANOVA revealed significant differences for rotation method for time spent in q_0; $F(2, 148) = 6.829$, $\eta^2 = .084$, $p = .001$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 2.175%, 95% CI, .321% to 4.029%, $p = .016$), as well as between GUI and Dragging (difference 2.474%, 95% CI, .632% to 4.315%, $p = .005$), but not between Thumbstick and Dragging ($p = 1.000$).

q_1: Mauchly's test revealed sphericity was not violated ($\chi^2 = .414$, $p = .813$). ANOVA revealed significant differences for rotation method for time spent in q_1; $F(2, 148) = 3.196$, $\eta^2 = .041$, $p = .044$. However, pairwise comparison with Bonferroni correction revealed no significant differences between GUI and Thumbstick ($p = 1.000$), GUI and Dragging ($p = .065$), and Thumbstick and Dragging ($p = .119$). A Friedman test has been performed and determined no significant differences between the three rotation methods. ($\chi^2 = 3.057$, $p = .217$).

q_2: Mauchly's test revealed sphericity was not violated ($\chi^2 = 3.438$, $p = .179$). ANOVA revealed significant differences for rotation method for time spent in q_2; $F(2, 148) = 5.261$, $\eta^2 = .066$, $p = .002$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 2.453%, 95% CI, .009% to 4.898%, $p = .049$), as well as between GUI and Dragging (difference 2.634%, 95% CI, .579% to 4.689%, $p = .007$), but not between Thumbstick and Dragging ($p = 1.000$).

q_3: Mauchly's test revealed sphericity was not violated ($\chi^2 = 1.708$, $p = .426$). ANOVA revealed significant differences for rotation method for time spent in q_2; $F(2, 148) = 27.547$, $\eta^2 = .271$, $p < .001$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 6.033%, 95% CI, 3.954% to 8.112%, $p < .001$), as well as between GUI and Dragging (difference 5.175%, 95% CI, 3.112% to 7.237%, $p < .001$), but not between Thumbstick and Dragging ($p = 1.000$).

q_4: Mauchly's test revealed sphericity was violated ($\chi^2 = 28.963$, $p < .001$); Greenhouse-Geisser correction was applied, $\epsilon = .753$. ANOVA revealed significant differences for rotation method for time spent in q_4; $F(1.507, 111.488) = 27.006$, $\eta^2 = .267$, $p < .001$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 10.053%, 95% CI, 7.068% to 13.038%, $p < .001$), as well as between GUI and Dragging (difference 8.226%, 95% CI, 5.178% to 11.274%, $p < .001$), but not between Thumbstick and Dragging ($p = .962$).

q_5: Mauchly's test revealed sphericity was violated ($\chi^2 = 35.184$, $p < .001$); Greenhouse-Geisser correction was applied, $\epsilon = .723$. ANOVA revealed significant differences for rotation method for time spent in q_5; $F(1.447, 107.058) = 19.661$, $\eta^2 = .210$, $p < .001$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 5.477%, 95% CI, 2.431% to 8.522%, $p < .001$), as well as between GUI and Dragging (difference 6.013%, 95% CI, 3.116% to 8.910%, $p < .001$), but not between Thumbstick and Dragging ($p = 1.000$).

q_6: Mauchly's test revealed sphericity was violated ($\chi^2 = 10.258$, $p < .006$); Greenhouse-Geisser correction was applied, $\epsilon = .884$. ANOVA revealed significant differences for rotation method for time spent in q_6; $F(1.768, 130.847) = 19.396$, $\eta^2 = .208$, $p < .001$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 6.771%, 95% CI, 3.881% to 9.661%, $p < .001$), as well as between GUI and Dragging (difference 4.880%, 95% CI, 1.811% to 7.949%, $p < .001$), but not between Thumbstick and Dragging ($p = .119$).

q_7: Mauchly's test revealed sphericity was not violated ($\chi^2 = 2.657$, $p = .265$). ANOVA revealed significant differences for rotation method for time spent in q_7; $F(2, 148) = 34.308$, $\eta^2 = .317$, $p < .001$. Pairwise comparison with Bonferroni correction revealed significant differences between GUI and Thumbstick (difference 4.308%, 95% CI, 2.844% to 5.772%, $p < .001$), as well as between GUI and Dragging (difference 4.363%, 95% CI, 2.759% to 5.966%, $p < .001$), but not between Thumbstick and Dragging ($p = 1.000$).

D.2 Qualitative Data Results

D.2.1 Questionnaire Results

The following results have been obtained by comparing the 10 different questionnaire answers for the rotation methods of GUI, Thumbstick and Dragging using a Friedman

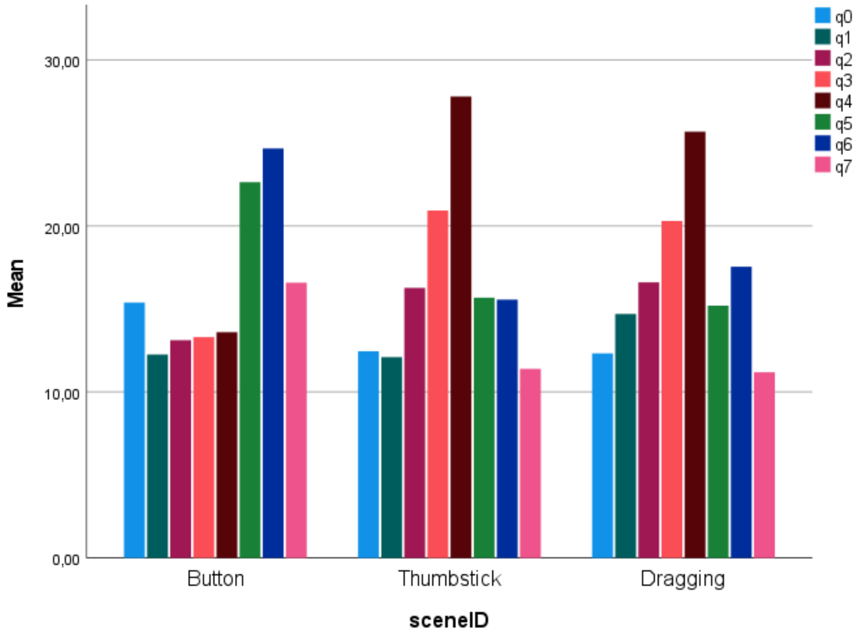

Figure D.3: View direction distribution (percentage of task completion time)

test to determine statistical significance as well as post-hoc analysis with a Bonferroni for pairwise comparisons.

Q1: "It was easy to use the rotation controls."

The Friedman test determined statistically significant differences in reported "ease of use" between the three rotation methods, $\chi^2 = 21.925$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 2$) and Thumbstick (median $mdn = 5$) ($p < .001$), between Thumbstick and Dragging ($mdn = 4$) ($p = .049$), but not between GUI and Dragging ($p = .231$).

Q2: "I felt in control of the video rotation."

The Friedman test determined statistically significant differences in reported "level of control" between the three rotation methods, $\chi^2 = 19.000$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 3$) and Thumbstick ($mdn = 5$) ($p < .001$), but not Thumbstick and Dragging ($mdn = 4$) ($p = .059$), and also not between GUI and Dragging ($p = .413$).

Q3: "The rotation controls did what I expected them to do."

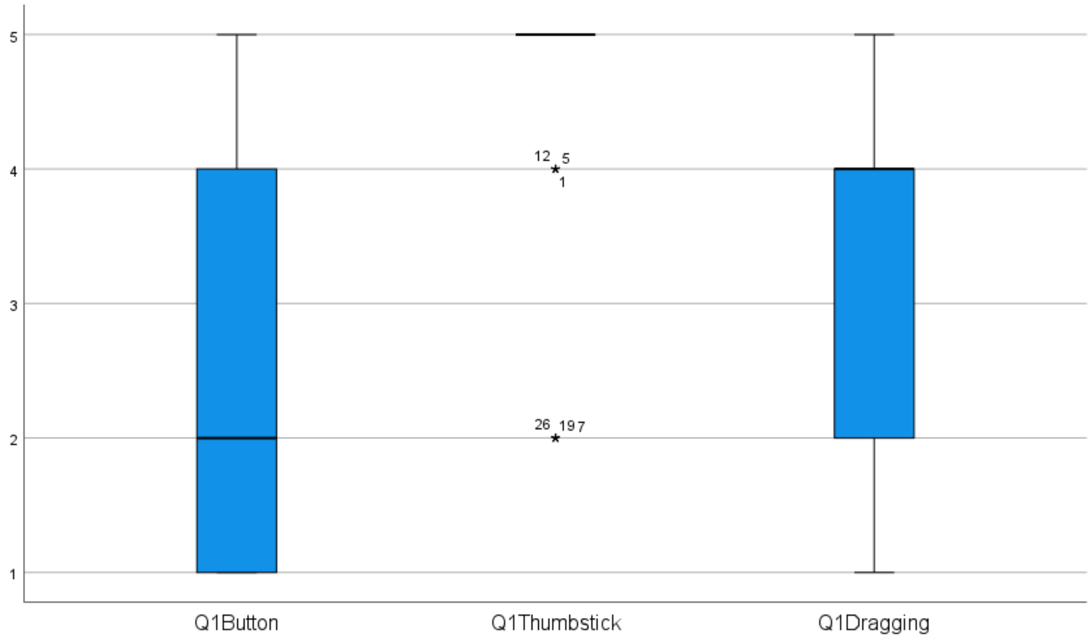

Figure D.4: Distribution for Question 1

The Friedman test determined statistically significant differences in reported ”fulfillment of expectation” between the three rotation methods, $\chi^2 = 18.092$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 4$) and Thumbstick ($mdn = 5$) ($p = .009$), as well as Thumbstick and Dragging ($mdn = 4$) ($p = .009$), but not between GUI and Dragging ($p = 1.000$).

Q4: ”I felt compelled to explore the video content.”

The Friedman test determined statistically significant differences in reported ”encouragement of exploration” between the three rotation methods, $\chi^2 = 18.092$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 2$) and Thumbstick ($mdn = 5$) ($p = .001$), but not Thumbstick and Dragging ($mdn = 4$) ($p = .102$), and not between GUI and Dragging ($p = 0.413$).

Q5: ”The rotation controls allowed me to easily explore the video content.”

The Friedman test determined statistically significant differences in reported ”enabling of exploration” between the three rotation methods, $\chi^2 = 25.049$, $p < .001$, pairwise

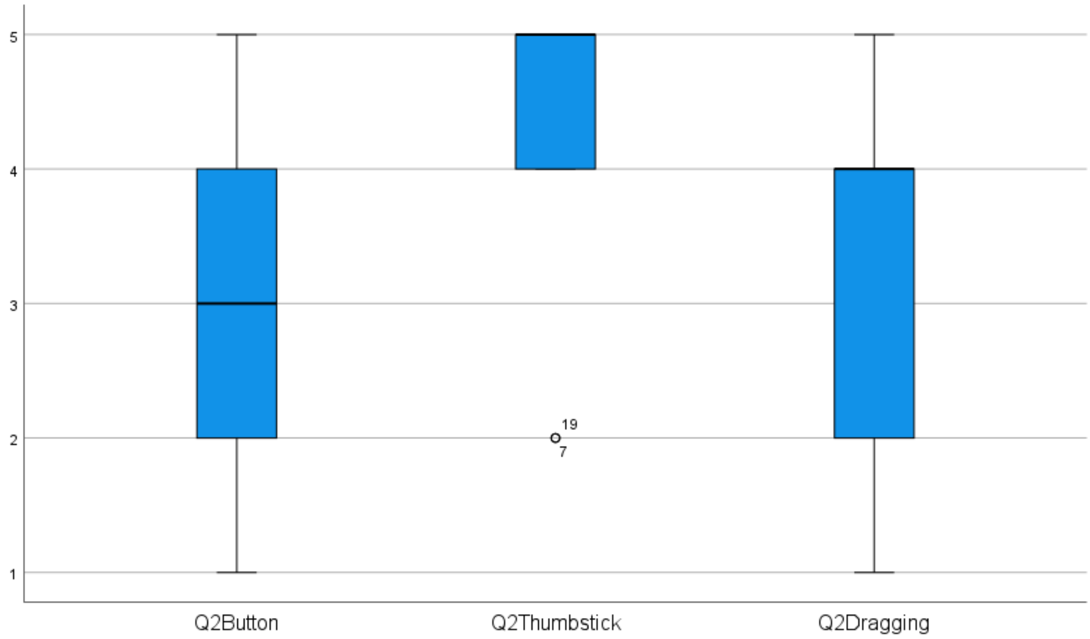

Figure D.5: Distribution for Question 2

comparison revealing significance between GUI ($mdn = 2$) and Thumbstick ($mdn = 5$) ($p < .001$), as well as Thumbstick and Dragging ($mdn = 4$) ($p = .040$), but not between GUI and Dragging ($p = 0.121$).

Q6: ”The rotation controls were responsive.”

The Friedman test determined statistically significant differences in reported ”responsiveness” between the three rotation methods, $\chi^2 = 21.726$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 3$) and Thumbstick ($mdn = 5$) ($p = .001$), as well as Thumbstick and Dragging ($mdn = 3.5$) ($p = .007$), but not between GUI and Dragging ($p = 1.000$).

Q7: ”I knew how to operate the rotation control.”

The Friedman test determined no statistically significant differences in reported ”knowledge on how to operate” between the three rotation methods. $\chi^2 = 3.152$, $p = .207$, with GUI $mdn = 5$; Thumbstick $mdn = 5$; Dragging $mdn = 5$.

Q8: ”The rotation controls felt natural and intuitive.”

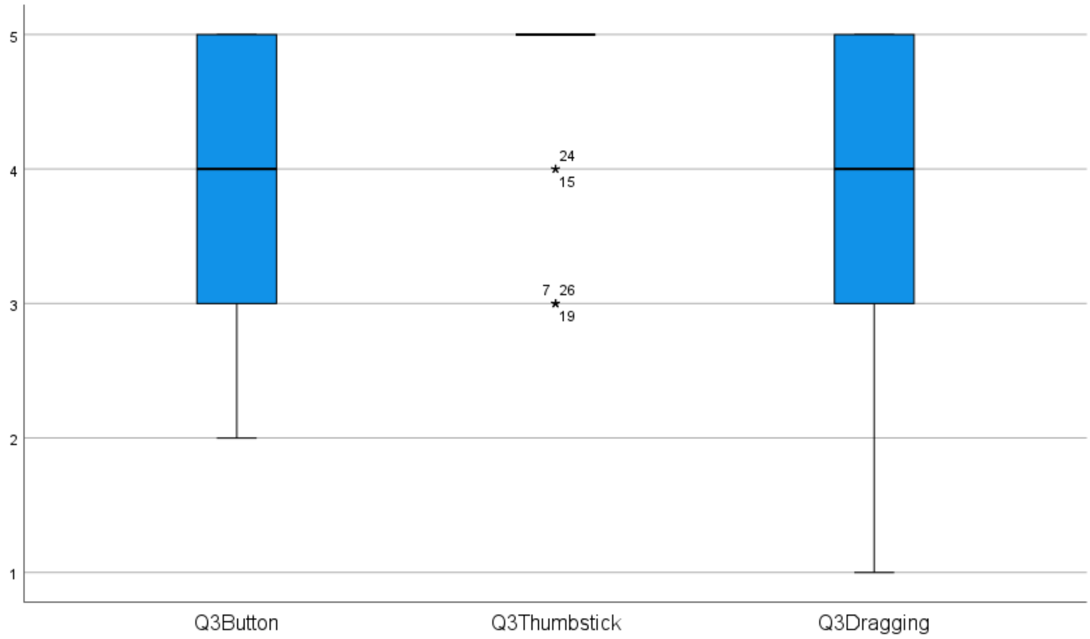

Figure D.6: Distribution for Question 3

The Friedman test determined statistically significant differences in reported ”intuitiveness” between the three rotation methods, $\chi^2 = 23.816$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 2$) and Thumbstick ($mdn = 5$) ($p < .001$), as well as Thumbstick and Dragging ($mdn = 3$) ($p = .022$), but not between GUI and Dragging ($p = 0.198$).

Q9: ”Using the rotation controls was fun and exciting to use.”

The Friedman test determined statistically significant differences in reported ”fun” between the three rotation methods, $\chi^2 = 19.227$, $p < .001$, pairwise comparison revealing significance between GUI ($mdn = 1$) and Thumbstick ($mdn = 4$)($p < .001$), as well as Thumbstick and Dragging ($mdn = 3$) ($p = .042$), but not between GUI and Dragging ($p = 0.511$).

Q10: ”Using the rotation controls was demanding and tiring to use.”

The Friedman test determined statistically significant differences in reported ”exhaustion” between the three rotation methods, $\chi^2 = 12.025$, $p = .002$, pairwise comparison

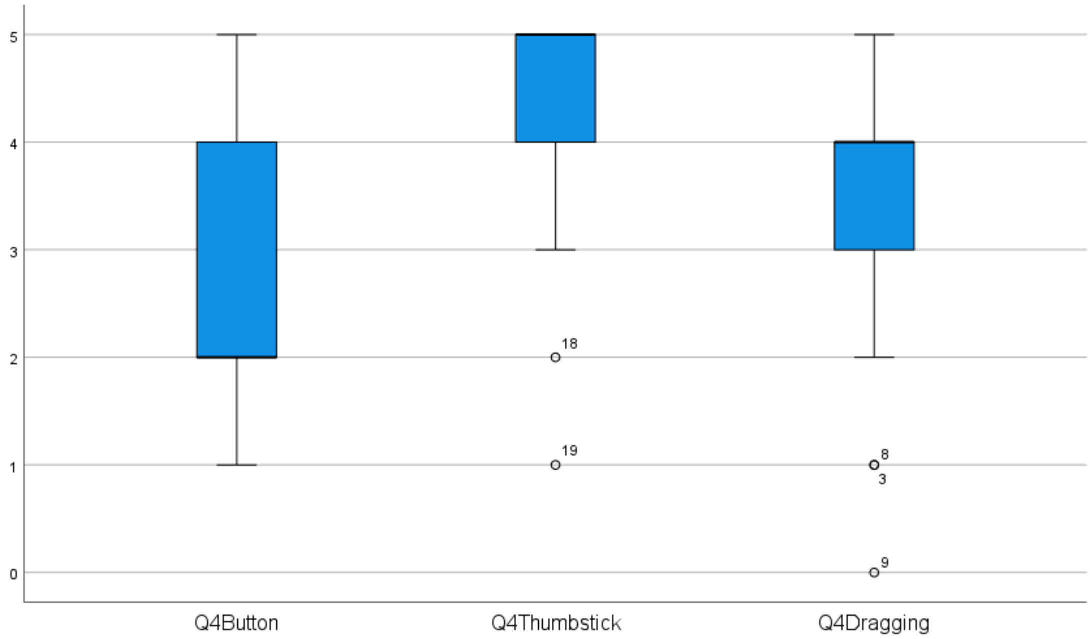

Figure D.7: Distribution for Question 4

revealing no significant difference between GUI ($mdn = 1$) and Thumbstick ($mdn = 2$) ($p = 1.000$), but a significant difference between Thumbstick and Dragging ($mdn = 4$) ($p = .040$), and between GUI and Dragging ($p = 0.014$).

D.2.2 Factors and Effects on Cybersickness

Of the 26 participants, 14 participants (53.85%) experienced moderate to severe cybersickness over the duration of the experiment. To analyze the presence of cybersickness, other independant variables are examined for a possible predictor effect.

For gender, a chi-square test for association was performed. All the expected cell frequencies were greater than 5. It was determined that there was no statistically significant association between gender and the presence of cybersickness, $\chi^2 = 2.735$, $p = .098$.

For level of VR experience, a Cochran-Armitage test of trend was used to determine wether a trend exists between level of VR experience and cybersickness. The experience level was encoded on a 5-point Likert scale, with 1-very little experience ($N = 12$), 2 - little experience ($N = 8$), 3 - moderate experience ($N = 2$), 4 - high level of experience ($N = 3$), and 5 - very high level of experience ($N = 1$), and the percentage proportion of participants experiencing cybersickness were 42.2%, 30.8%, 7.8%, 11.5%, and 3.8% respectively. Cochran-Armitage showed no significant linear trend, $p = .130$.

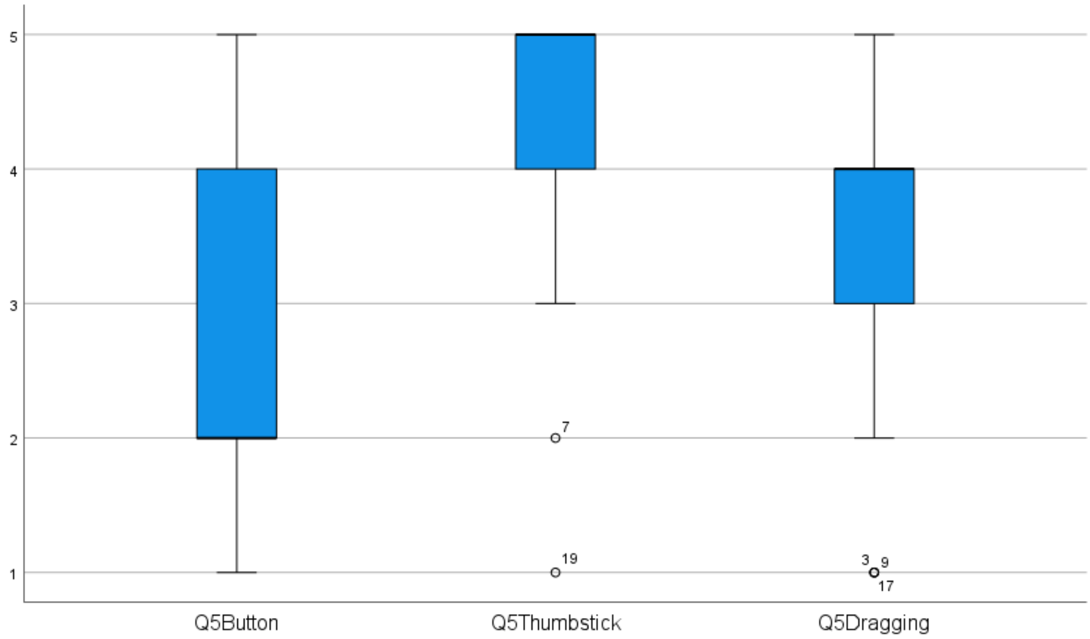

Figure D.8: Distribution for Question 5

For age, a binomial logistic regression was performed to ascertain the effect on the likelihood of experiencing cybersickness. The resulting model was deemed not statistically significant, $\chi^2 = .326$, $p = .568$. The model only explained 1.7% of variance in cybersickness (Nagelkerke R^2).

		B	S.E.	Wald	df	Sig.	Exp(B)	95% C.I.for EXP(B)	
								Lower	Upper
Step 1[a]	age	,104	,185	,318	1	,573	1,110	,772	1,596
	Constant	-2,700	5,074	,283	1	,595	,067		

a. Variable(s) entered on step 1: age.

Table D.1: Logistic Regression Prediction Likelihood of cybersickness based on age.

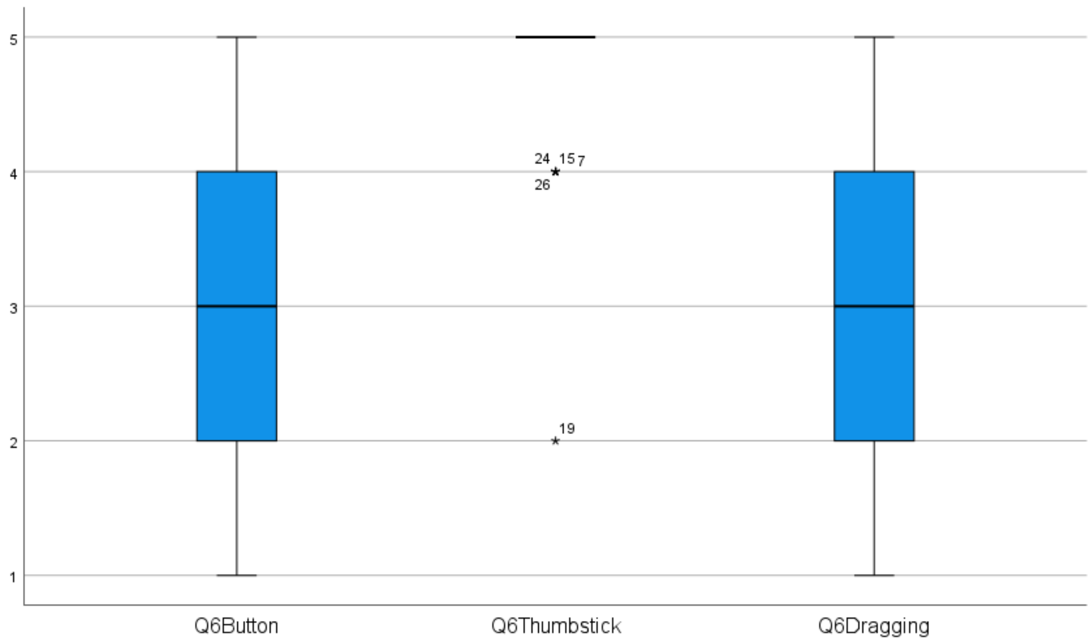

Figure D.9: Distribution for Question 6

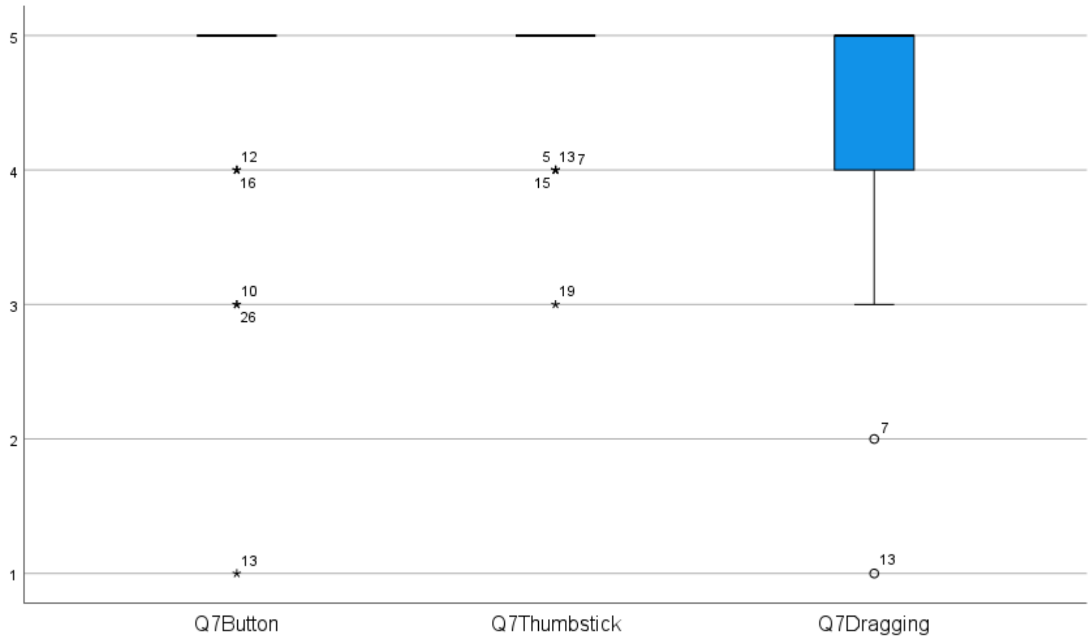

Figure D.10: Distribution for Question 7

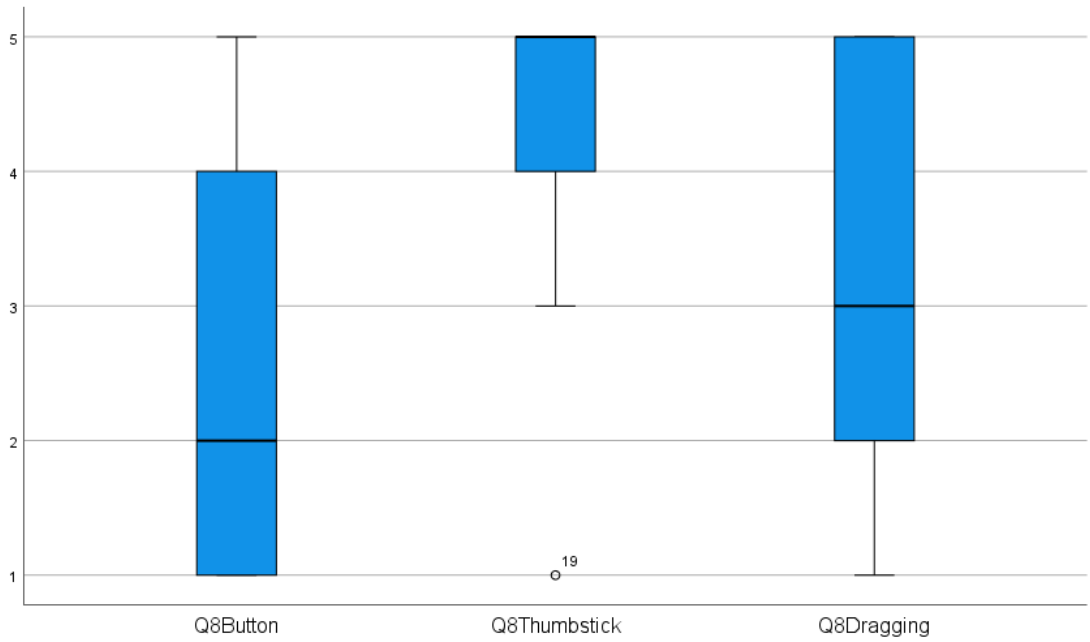

Figure D.11: Distribution for Question 8

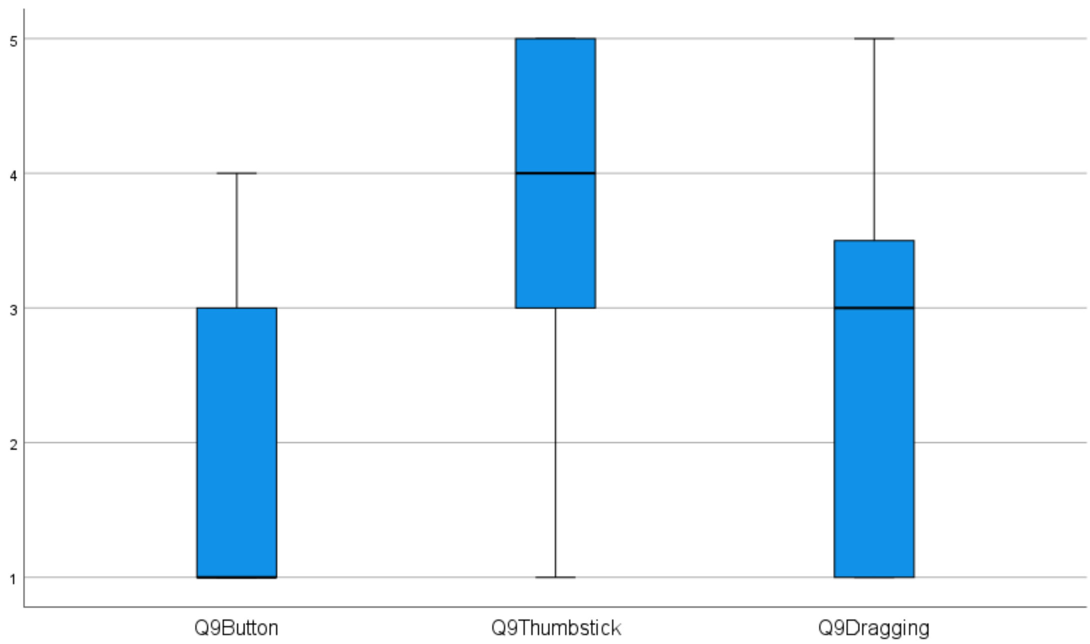

Figure D.12: Distribution for Question 9

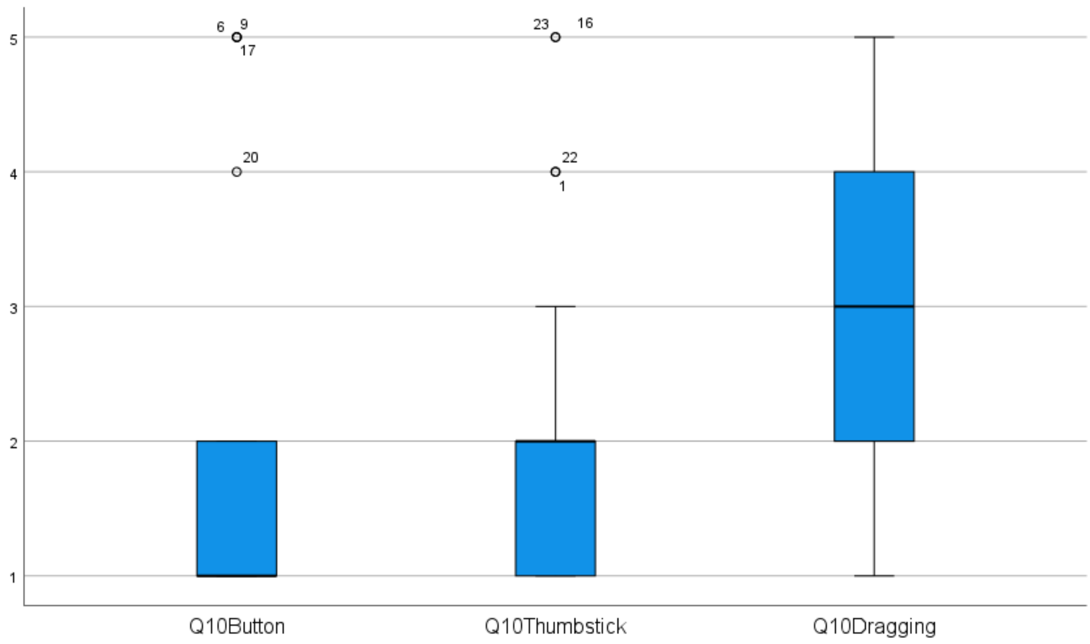

Figure D.13: Distribution for Question 10

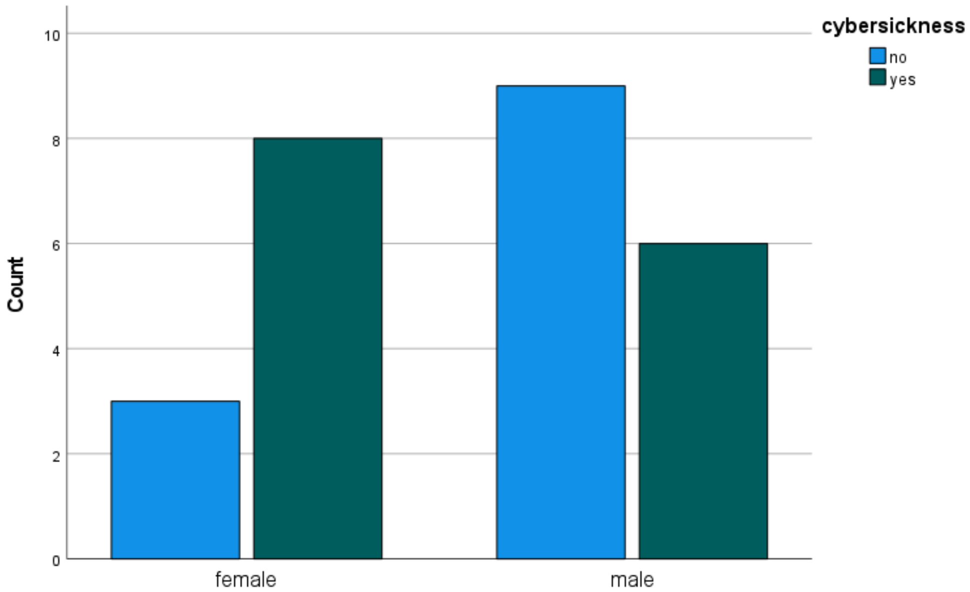

Figure D.14: Count of reported moderate to severe cybersickness by gender.

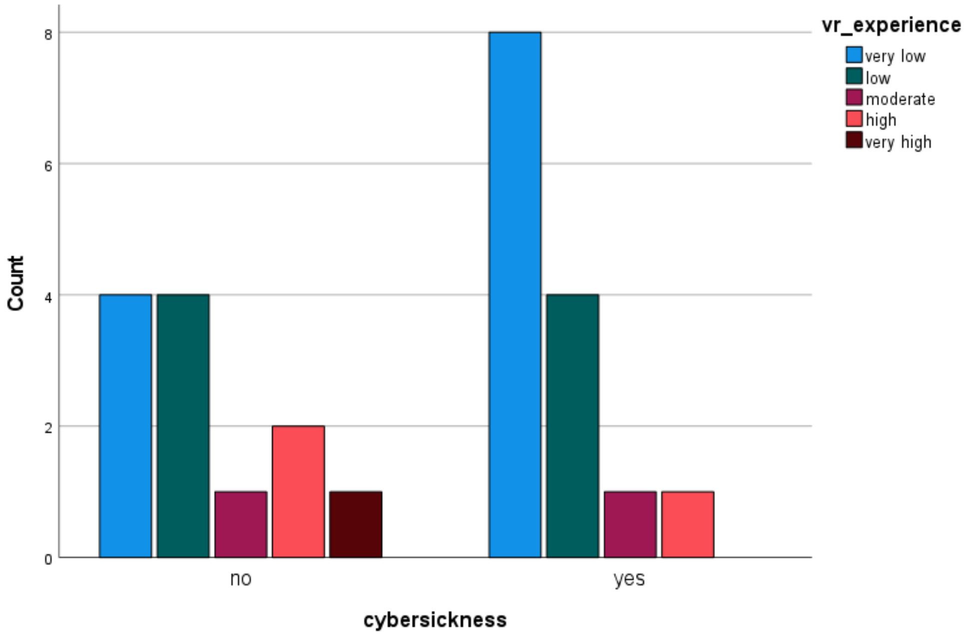

Figure D.15: Reported experience level with VR and experienced cybersickness.

Appendix E

Conclusions and Future Work

The goal of this work was to analyze different methods to navigate a 360° video using a HMD in a stationary position. Based on previous research, 3 different rotation methods have been created:

1. GUI: Two graphical buttons that are displayed on screen and which can be targeted via raycasting
2. Thumbstick: The physical thumbstick on the HMD controller which can tilted to the left or right side
3. Dragging: A physical gesture which includes holding a button on the controller and moving/rotating the controller itself

The research aim was to analyse these rotation methods for potential differences in viewer behaviour and viewer preference. The results presented in this work have identified specific differences between the rotation methods, namely rotation speed, rotation amount, and left/right rotation distribution. The results also show no significant differences in head rotation between the 3 methods. Disparities in view time distribution have been observed, but cannot be confidently interpreted without additional data.

Additionally, subjective feedback from participants has shown significant differences for experienced ease-of-use, immersion, engagement, physical exhaustion as well as overall enjoyment. The Thumbstick method was chosen as the favorite rotation method by the vast majority of participants, whereas the GUI method had distinct implementation aspects which made it significantly less enjoyable in the context of 360° videos.

Participants noted that specific design choices, like movement of the GUI elements with the head rotation, did not have their intended effect but made the rotation more difficult instead. In order to target a GUI button, the participants would move their head to focus on the button. From this movement, the button would move again slightly, requiring the participant to target them again, making sure their head did not rotate this time. It is believed that this design choice made the rotations more difficult and required more mental energy compared to the other rotation methods. One participant noted:

"It was hard to keep track of the rotation buttons. When I wanted to look at them, but they kept moving around."

Another issue of the GUI method was the implementation of the rotation speed. Finding the correct speed settings is a difficult task on its own, since different viewers will have different preferences for speed. Additionally, the speed increase over time has multiple values which must be fine-tuned. One participant noted:

"I did not like the rotation buttons. They were either way too slow or way too fast. It was difficult to make precise adjustments."

Another design decision which got negative feedback was the initial threshold of 5° for the dragging method. The goal of this threshold was to avoid small micro rotations from hand shaking. However, the threshold was large enough for participants to notice. One participant noted:

"I liked the Dragging method, but it was not as smooth as it could be. It felt *laggy*."

Based on these results, research topic for additional studies can be formulated. One of these topics would be the additional studies performed on the validity of the GUI control method with an improved implementation. It is possible that the poor performance of the GUI method was caused mainly by the moving GUI elements, and that a different implementation could improve the performance to the point of being comparable to other rotation methods. Another similar study could focus on the Dragging method with the same goal in mind.

Similarly, performing a user study on a larger population could either validate the findings of this work, find larger differences in viewing behaviour, or conclude that these differences are not as significant as initially reported. Additional studies could also focus on analysing viewer behaviour in a more organic way, for example by observing participants and their level of exploration without the presence of a search task. These studies could also focus on first finding other tasks which would better simulate the context of a 360° video before testing any control methods. Another focus could be implementing rotation methods which work on multiple axes instead of only one dimension. Including the pitch-dimension in the control scheme could help establish one of the presented rotation methods as the objectively better choice.

Before being able to include a new control method into an existing player, additional studies need to determine the validity of these new controls with the existing control scheme for the specific context. In the case of a 360° video player, the rotation methods need to be integrated alongside the basic video controls like play/pause, time skimming, etc. In order for a new control scheme to be viable, potential conflicts with other controls need to be examined and potentially eliminated.

We conclude that in order to chose a rotation method to integrate into a video player, additional, more focused studies should be performed.

Besides these possible studies into the validity of rotation methods, another area of research which presented itself in this work is the area of cybersickness. While this phenomenon is not uncommon for VR applications in general, the addition of visual rotation could have exacerbated these symptoms significantly. One area of future research would include a more detailed study on the amount of cybersickness experienced for each rotation method, and the effect of certain video types have on cybersickness. Such a study could focus on the feasibility of rotating a 360° video in the first place, or focus on the type of 360° videos which are less prone to inducing cybersickness. Such a study could not only find design guidelines for control methods, but for the creation of videos targeted towards HMD VR users as well, for example stationary videos without excessive movement and with minimal cuts between different camera angles. Other studies could focus on the effects of other independent variables, like age, gender or previous VR experience. While these aspects were taken into account in this work, it is very well possible that their distribution was not large enough to find significant influences. Performing a study on cybersickness would require a larger, more diverse participant group. It can be stated that studies in the area of VR controls and applications should take the problem of cybersickness into consideration, and possibly include an analysis on the severity along with their main body of work. Other future studies can be more specific to the context of cybersickness during 360° videos, and focus on finding design solutions which would alleviate this problem. One possible solution could include the pausing of the video content. Another solution could be to extend the field of view during rotation, facilitating object identification and requiring less rotation overall.

Bibliography

[1] Francis C Li, Anoop Gupta, Elizabeth Sanocki, Li-wei He, and Yong Rui. Browsing digital video. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 169–176, 2000.

[2] Axel Carlier, Vincent Charvillat, and Wei Tsang Ooi. A video timeline with bookmarks and prefetch state for faster video browsing. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 967–970, 2015.

[3] Heather Anne Richter, Jason Alan Brotherton, Gregory D Abowd, and Khai Nhut Truong. A multi-scale timeline slider for stream visualization and control. Technical report, Georgia Institute of Technology, 1999.

[4] Wolfgang Hürst, Georg Götz, and Philipp Jarvers. Advanced user interfaces for dynamic video browsing. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 742–743, 2004.

[5] Suporn Pongnumkul, Jue Wang, Gonzalo Ramos, and Michael Cohen. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 139–142, 2010.

[6] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Swifter: improved online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1159–1168, 2013.

[7] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 237–246, 2008.

[8] Thorsten Karrer, Malte Weiss, Eric Lee, and Jan Borchers. Dragon: a direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 247–250, 2008.

[9] Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. Pocketdragon: a direct manipulation video navigation interface for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–3, 2009.

[10] Cuong Nguyen, Yuzhen Niu, and Feng Liu. Video summagator: An interface for video summarization and navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 647–650, 2012.

[11] Klaus Schoeffmann and Lukas Burgstaller. Scrubbing wheel: An interaction concept to improve video content navigation on devices with touchscreens. In *2015 IEEE International Symposium on Multimedia (ISM)*, pages 351–356. IEEE, 2015.

[12] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang. A dataset for exploring user behaviors in vr spherical video streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 193–198, 2017.

[13] Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 360-degree video head movement dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 199–204, 2017.

[14] Mathias Almquist and Viktor Almquist. Analysis of 360 video viewing behaviours, 2018.

[15] Fanyi Duanmu, Yixiang Mao, Shuai Liu, Sumanth Srinivasan, and Yao Wang. A subjective study of viewer navigation behaviors when watching 360-degree videos on computers. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[16] Lizzy Bleumers, Wendy Van den Broeck, Bram Lievens, and Jo Pierson. Seeing the bigger picture: a user perspective on 360 tv. In *Proceedings of the 10th European conference on Interactive TV and video*, pages 115–124, 2012.

[17] Goranka Zoric, Louise Barkhuus, Arvid Engström, and Elin Önnevall. Panoramic video: design challenges and implications for content interaction. In *Proceedings of the 11th european conference on Interactive TV and video*, pages 153–162, 2013.

[18] Helen Katz. *The Media Handbook: A Complete Guide to Advertising Media Selection, Planning, Research, and Buying (4th ed.)*. Routledge, 2010.

[19] Fabrizio Pece, James Tompkin, Hanspeter Pfister, Jan Kautz, and Christian Theobalt. Device effect on panoramic video+ context tasks. In *Proceedings of the 11th European Conference on Visual Media Production*, pages 1–9, 2014.

[20] Marc Van den Broeck, Fahim Kawsar, and Johannes Schöning. It's all around you: Exploring 360 video viewing experiences on mobile devices. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 762–768, 2017.

[21] Lisa Rebenitsch. Managing cybersickness in virtual reality. *XRDS: Crossroads, The ACM Magazine for Students*, 22(1):46–51, 2015.

[22] Yang Hong, Andrew MacQuarrie, and Anthony Steed. The effect of chair type on users' viewing experience for 360-degree video. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11, 2018.

[23] Matthias Wille, Britta Grauel, and Lars Adolph. Strain caused by head mounted displays. *Proceedings of the Human Factors and Ergonomics Society Europe*, pages 267–277, 2013.

[24] Eunjee Kim and Gwanseob Shin. Head rotation and muscle activity when conducting document editing tasks with a head-mounted display. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 62, pages 952–955. SAGE Publications Sage CA: Los Angeles, CA, 2018.

[25] Jonathan Harth, Alexandra Hofmann, Mike Karst, David Kempf, Annelie Ostertag, Isabell Przemus, and Bernhard Schaefermeyer. Different types of users, different types of immersion: A user study of interaction design and immersion in consumer virtual reality. *IEEE Consumer Electronics Magazine*, 7(4):36–43, 2018.

[26] Mirjam Vosmeer and Ben Schouten. Interactive cinema: engagement and interaction. In *International conference on interactive digital storytelling*, pages 140–147. Springer, 2014.

[27] Henry Jenkins. *Convergence Culture: Where Old and New Media Collide.* New York: New York University Press, 2006.

[28] Lemonia Argyriou, Daphne Economou, Vassiliki Bouki, and Ioannis Doumanis. Engaging immersive video consumers: Challenges regarding 360-degree gamified video applications. In *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, pages 145–152. IEEE, 2016.

[29] Syed Ali Arsalan Naqvi, Nasreen Badruddin, Aamir Saeed Malik, Wan Hazabbah, and Baharudin Abdullah. Does 3d produce more symptoms of visually induced motion sickness? In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6405–6408. IEEE, 2013.

[30] Maximino Bessa, Miguel Melo, David Narciso, Luís Barbosa, and José Vasconcelos-Raposo. Does 3d 360 video enhance user's vr experience? an evaluation study. In *Proceedings of the XVII International Conference on Human Computer Interaction*, pages 1–4, 2016.

[31] Miguel Correia Melo, José Vasconcelos Raposo, António Coelho, David Gonçalves Narciso, and Maximino Bessa. Immersive 360° video user experience: impact of different variables in the sense of presence and cybersickness. 2019.

[32] Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Liu, Hsien-Tzu Cheng, Yung-Ju Chang, and Min Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1396–1405. IEEE, 2017.

[33] Tejal Mate. Design and evaluation of guiding methods for 360° videos. Master's thesis, 2018.

[34] Yen-Chen Lin, Yung-Ju Chang, Hou-Ning Hu, Hsien-Tzu Cheng, Chi-Wen Huang, and Min Sun. Tell me where to look: Investigating ways for assisting focus in 360 video. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2535–2545, 2017.

[35] Jan Gugenheimer, Dennis Wolf, Gabriel Haas, Sebastian Krebs, and Enrico Rukzio. Swivrchair: A motorized swivel chair to nudge users' orientation for 360 degree storytelling in virtual reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1996–2000, 2016.

[36] Robert JK Jacob, Audrey Girouard, Leanne M Hirshfield, Michael S Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: unifying the new generation of interaction styles. In *CHI'07 extended abstracts on Human factors in computing systems*, pages 2465–2470, 2007.

[37] Chris Hand. A survey of 3d interaction techniques. In *Computer graphics forum*, volume 16, pages 269–281. Wiley Online Library, 1997.

[38] Gerwin De Haan, Michal Koutek, and Frits H Post. Intenselect: Using dynamic object rating for assisting 3d object selection. In *Ipt/egve*, pages 201–209. Citeseer, 2005.

[39] Sangyoon Lee, Jinseok Seo, Gerard Jounghyun Kim, and Chan-Mo Park. Evaluation of pointing techniques for ray casting selection in virtual environments. In *Third international conference on virtual reality and its application in industry*, volume 4756, pages 38–44. International Society for Optics and Photonics, 2003.

[40] Anthony Steed. Towards a general model for selection in virtual environments. In *3D User Interfaces (3DUI'06)*, pages 103–110. IEEE, 2006.

[41] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. Gaze+ pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pages 99–108, 2017.

[42] Marc Baloup, Thomas Pietrzak, and Géry Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[43] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. Understanding the heisenberg effect of spatial interaction: A selection induced error for spatially tracked input devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2020.

[44] Huawei Tu, Susu Huang, Jiabin Yuan, Xiangshi Ren, and Feng Tian. Crossing-based selection with virtual reality head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[45] I Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1):91–139, 1992.

[46] Susu Huang, QI Daqing, YUAN Jiabin, and TU Huawei. Review of studies on target acquisition in virtual reality based on the crossing paradigm. *Virtual Reality & Intelligent Hardware*, 1(3):251–264, 2019.

[47] Ryan P McMahan, Chengyuan Lai, and Swaroop K Pal. Interaction fidelity: the uncanny valley of virtual reality interactions. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 59–70. Springer, 2016.

[48] Lemonia Argyriou, Daphne Economou, and Vassiliki Bouki. Design methodology for 360 immersive video applications: the case study of a cultural heritage virtual tour. *Personal and Ubiquitous Computing*, 24(6):843–859, 2020.

[49] Shyam Prathish Sargunam, Kasra Rahimi Moghadam, Mohamed Suhail, and Eric D Ragan. Guided head rotation and amplified head rotation: Evaluating semi-natural travel and viewing techniques in virtual reality. In *2017 IEEE Virtual Reality (VR)*, pages 19–28. IEEE, 2017.

[50] Toni Pakkanen, Jaakko Hakulinen, Tero Jokela, Ismo Rakkolainen, Jari Kangas, Petri Piippo, Roope Raisamo, and Marja Salmimaa. Interaction with webvr 360 video player: Comparing three interaction paradigms. In *2017 IEEE Virtual Reality (VR)*, pages 279–280. IEEE, 2017.

[51] Sylvia Rothe, Pascal Pothmann, Heiko Drewe, and Heinrich Hussmann. Interaction techniques for cinematic virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1733–1737. IEEE, 2019.

[52] Benjamin Petry and Jochen Huber. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proceedings of the 6th Augmented Human International Conference*, pages 217–218, 2015.

[53] Gustavo Alberto Rovelo Ruiz, Davy Vanacken, Kris Luyten, Francisco Abad, and Emilio Camahort. Multi-viewer gesture-based interaction for omni-directional video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4077–4086, 2014.

[54] Jeffrey T Hansberger, Chao Peng, Shannon L Mathis, Vaidyanath Areyur Shanthakumar, Sarah C Meacham, Lizhou Cao, and Victoria R Blakely. Dispelling the gorilla arm syndrome: the viability of prolonged gesture interactions. In *International conference on virtual, augmented and mixed reality*, pages 505–520. Springer, 2017.

[55] Jacob O Wobbrock, Meredith Ringel Morris, and Andrew D Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1083–1092, 2009.

[56] Shakeeb Shirazi. Timeline visualization of omnidirectional videos. Master's thesis, 2018.

[57] Sara Di Bartolomeo, Aditeya Pandey, Aristotelis Leventidis, David Saffo, Uzma Haque Syeda, Elin Carstensdottir, Magy Seif El-Nasr, Michelle A Borkin, and Cody Dunne. Evaluating the effect of timeline shape on visualization task performance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[58] Nancy Staggers and Anthony F. Norcio. Mental models: concepts for human-computer interaction research. *International Journal of Man-machine studies*, 38 (4):587–605, 1993.

[59] Klemen Lilija, Henning Pohl, and Kasper Hornbæk. Who put that there? temporal navigation of spatial recordings by direct manipulation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2020.

[60] Jiannan Li, Jiahe Lyu, Mauricio Sousa, Ravin Balakrishnan, Anthony Tang, and Tovi Grossman. Route tapestries: Navigating 360 virtual tour videos using slit-scan visualizations. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 223–238, 2021.

[61] Unity Technologies. Unity. `www.unity.com`, 2022.

[62] Unity Technologies. Xr interaction toolkit. `docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.10`, 2018.

[63] Inc. Meta Platforms. Oculus quest. `oculus.com/quest/features/`, 2019.

List of Figures

List of Tables