A Virtual Testbed for Fish-Tank Virtual Reality: Improving Calibration with a Virtual-in-Virtual Display

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Dylan Brodie Fafard

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Abstract

With the development of novel calibration techniques for multimedia projectors and curved projection surfaces, volumetric 3D displays are becoming easier and more affordable to build. The basic requirements include a display shape that defines the volume (e.g. a sphere, cylinder, or cuboid) and a tracking system to provide each user's location for the perspective corrected rendering. When coupled with modern graphics cards, these displays are capable of high resolution, low latency, high frame rate, and even stereoscopic rendering; however, like many previous studies have shown, every component must be precisely calibrated for a compelling 3D effect. While human perceptual requirements have been extensively studied for head-tracked displays, most studies featured seated users in front of a flat display. It remains unclear if results from these flat display studies are applicable to newer, walk-around displays with enclosed or curved shapes. To investigate these issues, we developed a virtual testbed for volumetric head-tracked displays that can measure calibration accuracy of the entire system in real-time. We used this testbed to investigate visual distortions of prototype curved displays, improve existing calibration techniques, study the importance of stereo to performance and perception, and validate perceptual calibration with novice users. Our experiments show that stereo is important for task performance, but requires more accurate calibration, and that novice users can make effective use of perceptual calibration tools. We also propose a novel, real-time calibration method that can be used to fine-tune an existing calibration using perceptual feedback. The findings from this work can be used to build better head-tracked volumetric displays with an unprecedented amount of 3D realism and intuitive calibration tools for novice users.

Acknowledgements

I would like to thank my supervisor, Dr. Ian Stavness, for the opportunity to work together for my undergraduate research project, and for his guidance and support throughout my graduate studies. I am grateful to have had the opportunity to join the Biomedical Interactive Graphics (BIG) lab. This work wouldn't have been possible without the technological resources and fellow students from the lab. I would also like to thank Dr. Carl Gutwin and Dr. Regan Mandryk for providing effective strategies for organizing and communicating research within HCI.

I would like to express my sincerest gratitude to my wife, Angela, for her patience and feedback throughout my graduate studies.

This thesis is dedicated to my grandparents, Lucien and Claire

Co-Authorship Statement

Parts of this thesis have been included in technical exhibits and featured in peer-reviewed journals.

The extensions from Chapter 3 build on the Interactive Visual Calibration method published in Wagemakers, Fafard, and Stavness [104]. This method was exhibited at CHI 2017 by Fafard et al. [38] and the extensions were presented at the CoClobe exhibit at SIGGRAPH 2018 by Zhou et al. [117]. I formulated the viewpoint calibration algorithm with Mr. Wagemakers and validated the implementation myself through an experiment using synthetic data. I also designed and implemented a semi-automatic image processing algorithm to quantitatively measure calibration error on a perspective-corrected 3D display.

A version of Chapter 4 has been published in Fafard et al. [39] and was awarded the Polyphony Digital award at VRST 2018. I designed and implemented a virtual testbed for FTVR displays. I integrated the testbed, its virtual rendering pipeline, and the improved viewpoint calibration from Chapter 3 with a project that Mr. Chamberlain and Mr. Wagemakers implemented for mono/stereo rendering with physical FTVR displays. I also designed an occlusion-based 3D navigation algorithm used by the fish in the "find the fish" game.

A version of Chapter 5 has been published in Fafard et al. [37]. I put together the main objectives of the study under the supervision of Dr. Stavness. The design and subsequent statistical analysis of the study was shared in part by myself, Mr. Dechant, Ms. Zhou, and our respective supervisors. I designed and implemented the virtual replica of the physical display, the three experiments, and a stereo acuity test in VR.

Contents

Permission to Use i

Abstract ii

Acknowledgements iii

Co-Authorship Statement v

Contents vi

List of Tables viii

List of Figures ix

List of Equations ix

List of Abbreviations x

1 Introduction 1
- 1.1 General Problem . . . 1
- 1.2 Specific Problem 1: Viewpoint Calibration . . . 2
- 1.3 Specific Problem 2: Design Decisions . . . 3
- 1.4 Solution 1: Improved Visual Calibration . . . 3
- 1.5 Solution 2: VR Testbed for FTVR Displays . . . 4
- 1.6 Solution 3: Evaluation of FTVR Viewing Conditions . . . 5
- 1.7 Research Questions . . . 5
- 1.8 Research Objectives . . . 5
- 1.9 Organization of Thesis . . . 6

2 Related Work 7
- 2.1 FTVR . . . 7
- 2.2 Multi-screen FTVR . . . 9
- 2.3 Handheld FTVR . . . 10
- 2.4 Room-based VR and AR . . . 10
- 2.5 Calibration in FTVR Systems . . . 10
- 2.6 Static and Swept Volume Displays . . . 12
- 2.7 Head-mounted Displays . . . 12
- 2.8 View Independent Rendering . . . 12
- 2.9 Rendering Systems . . . 13
- 2.10 Novelty of Simulation System . . . 13

3 Extensions to Interactive Visual Calibration 15
- 3.1 Adding Orientation . . . 16
- 3.2 Adding Stereoscopic Support . . . 17
- 3.3 Improving Optimization . . . 18
  - 3.3.1 Comparison . . . 19
- 3.4 Viewpoint Model Transformations . . . 21
  - 3.4.1 Position Only . . . 22
  - 3.4.2 Position and Orientation . . . 23
- 3.5 Realtime Refinement of Calibrations . . . 24
  - 3.5.1 Viewpoints . . . 24

3.5.2 Handheld Objects . . . 25
3.5.3 Updating the Calibration . . . 26

4 Virtualization of FTVR **28**
4.1 Rendering Overview . . . 29
4.2 Planar Display Screens . . . 30
4.3 Projected Surfaces . . . 30
4.4 Near-surface Clipping . . . 37
4.5 Virtual FTVR Model . . . 39

5 Virtual FTVR User Study of Viewing Condition **41**
5.1 Experiment Design . . . 42
5.1.1 Viewing Conditions . . . 42
5.1.2 Participants and Procedure . . . 43
5.1.3 Data Analysis . . . 44
5.2 Experiment 1: Viewpoint Pattern Alignment . . . 44
5.2.1 Analysis & Results . . . 45
5.3 Experiment 2: Subjective Preference . . . 47
5.3.1 Analysis & Results . . . 48
5.4 Experiment 3: Point Cloud Performance . . . 48
5.4.1 Analysis & Results . . . 50
5.5 Discussion . . . 52
5.5.1 Virtual Reality Sickness Questionnaire . . . 52
5.5.2 Effectiveness of Viewpoint Calibration with Spherical Displays . . . 52
5.5.3 Subjective Perception of *Viewing Conditions* . . . 53
5.5.4 Task Related Performance . . . 54
5.5.5 Head Movement . . . 54
5.5.6 Novelty of Simulated FTVR . . . 55
5.5.7 FTVR Design Recommendations . . . 55

6 Conclusions **57**
6.1 Limitations and Future Work . . . 57
6.2 Contributions . . . 59
6.3 Concluding Remarks . . . 59

Glossary **60**

References **61**

Appendix A Shaders **69**

Appendix B Virtual Study Documents **71**
B.1 Consent Form . . . 72
B.2 Dominant Eye Test Card . . . 73
B.3 Questionnaires . . . 74
B.4 Questionnaires . . . 75
B.5 Questionnaires . . . 76
B.6 Questionnaires . . . 77
B.7 Virtual Reality Sickness Questionnaire . . . 78

List of Tables

2.1 Design and Capabilities of FTVR Displays 8

3.1 Model Parameters 16

5.1 Viewing Condition Preference 48

List of Figures

1.1 Cubic and Spherical FTVR 2

3.1 Coordinate Frames 16
3.2 3D Calibration Shape 18
3.3 Optimizer Comparisons 21
3.4 Position Only Viewpoint Correction 23
3.5 Realtime Viewpoint Calibration 25
3.6 Realtime Object Calibration 26

4.1 Physical and Virtual Displays 28
4.2 Rendering Overview 29
4.3 Projector Mapping 32
4.4 Rendering Pipeline Overview 33
4.5 LookAt Frustum 35
4.6 Off-axis Frustum 35
4.7 Frustum Comparison 36
4.8 Bunny-in-bunny Display 37
4.9 Near-surface Clipping 38
4.10 Visual Distortion 39

5.1 Viewing Conditions 43
5.2 Viewpoint Pattern Alignment 44
5.3 Measured Viewpoints 45
5.4 Pattern Alignment Task Results 46
5.5 Spatial Impression Task 47
5.6 Point Cloud Tasks 50
5.7 Point Cloud Task Results 51
5.8 Deviation from Perceptual Guidance 53

List of Equations

3.1 Display to Calibration Ray Rotation 17
3.2 Isolation of Head to View Rotation 17
3.3 Approximation of Head to View Rotation 17
3.4 Vector of Optimization Parameters 19
3.15 Position Only Viewpoint Offset 23

4.1 Off-Axis Projection Matrix 30
4.7 Virtual Projector Projection Matrix 31
4.11 Projector Luminosity 31

List of Abbreviations

API	Application Program Interface
VR	Virtual Reality
AR	Augmented Reality
MR	Mixed Reality
FTVR	Fish Tank Virtual Reality
3DoF	3 Degrees of Freedom
6DoF	6 Degrees of Freedom
IMU	Inertial Measurement Unit
PD	Pupillary Distance
HCI	Human Computer Interaction

1 Introduction

Perspective-corrected displays – interactive displays that use images corrected to a user's perspective – create a 3D experience by coupling head movements to the perspective of a rendered scene. This technique has been used effectively with head-mounted displays in augmented reality and virtual reality to depict a fixed virtual environment to the user. In the early 1990s, this technique was used with desktop CRT monitors as an alternative to head-mounted displays and was called "Fish Tank Virtual Reality" (FTVR) by Ware et. al [106]. It offered higher pixel resolution and more natural viewing over head-mounted displays at the time [24, 68, 42, 91, 27] and it was suggested that stereopsis (stereo) may add only marginally to the perception of 3D objects. Given the additional equipment and technical requirements of including stereo in 3D displays, many subsequent FTVR displays omitted stereo and relied primarily on motion parallax for the 3D experience. FTVR displays provide motion parallax by rendering many stationary objects inside the display and updating the perspective for a moving observer.

Room and headset-based VR displays situate and surround the user within the virtual environment which offers an immersive virtual experience at the cost of deemphasizing the physical environment. Head-mounted AR displays are able to overlay virtual content onto the physical environment thereby reducing the virtual-physical gap, however, the virtual content remains intangible. FTVR displays are able to embed and situate a portion of the virtual environment in a physical space while offering tangible interactions through touch or reorientation of the display. They can be used effectively as an addition or enhancement to workspaces by supporting easy transitions and interactions between real-world information, traditional 2D displays, and virtual information.

1.1 General Problem

Recent (post-2010) technology has made the inclusion of stereo an affordable option for FTVR displays and has enabled new convex form factors (e.g. spherical, cubic, and cylindrical shapes) that allow viewing from all sides (see Figure 1.1). With such a high field of view, they are usually designed to encapsulate and create a contained virtual environment to give the illusion that a virtual object is contained within the volumetric display.

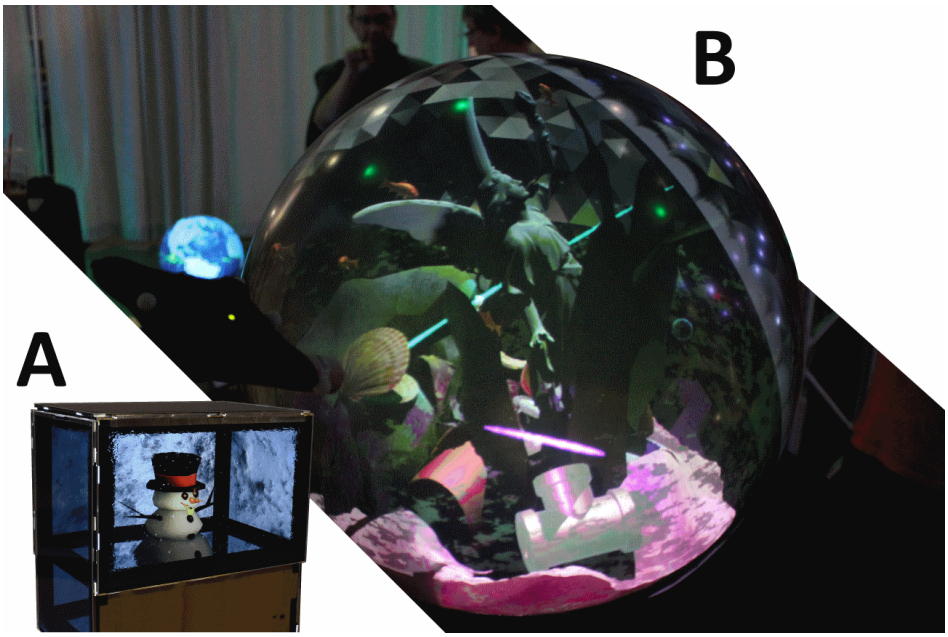


Figure 1.1: New FTVR displays have made use of multiple screens and curved projection surfaces to produce cubic (A, Cubee [104]) and spherical (B, CoGlobe [117]) form factors.

These newer displays are different from the older (pre-1997), CRT-based FTVR displays which made significant use of rendering objects with front-depth to give the illusion that a virtual object was floating in space in front of the display. Given the technology and design gap between these types of FTVR displays, it is unclear if experimental and user research generalizes to both. FTVR is a promising technology, however, it has not been as widely adopted as other VR technologies. This is likely due to the technical challenges that come with designing and calibrating FTVR displays. They are often built from a set of smaller subsystems such as projectors, display screens, and tracking systems. The challenges lie in bringing these distinct subsystems into efficient cooperation regarding latency, accuracy, and ease of use.

1.2 Specific Problem 1: Viewpoint Calibration

For displays to achieve a compelling and physically situated virtual environment using perspective-corrected images, the position of each pixel must be known relative to a viewpoint with a high degree of accuracy and be updated with minimal latency with respect to head movements. Many recent VR headsets have accomplished this by 1) tracking the headset using high-update inertial measurement units (IMUs) and slower-updating external tracking devices to correct for drift errors from the IMUs; and 2) assuming the viewpoints are a fixed distance in front of the screens or measuring the viewpoints directly using eye-tracking. For FTVR displays, the user and display are physically separated and an external tracking system must be used to register the display and user in a consistent coordinate space. A range of tracking systems including infrared, ultrasonic, electromagnetic, optical, and computer vision have been used with FTVR displays; however, they

all have relatively high latency compared to IMUs and introduce a noticeable amount of visual distortion. In addition to this, the displacement between what is being tracked (usually a sensor of some type) and the user's viewpoint has been more difficult to define than with VR headsets. The conventional approach to measuring this displacement has been to either manually calibrate it for each user and assume it remains fixed or to place the sensor close to the viewpoint, usually somewhere on the head, and use the sensor as the viewpoint. Introducing stereo requires an additional displacement, one for each eye, and depends on interpupillary distance, which can be different for each user. Also, it is known that human eyes do not have a rigid center of projection relative to the head [31], but move with respect to the viewer's gaze. A recent, interactive approach to viewpoint and tracking calibration proposed by Wagemakers et al. [104] calibrated a viewpoint to within $\approx 2^\circ$ (5 cm) using 2D patterns on a non-stereo desktop monitor and a DSLR camera on a box-shaped volumetric display; however, this method remains untested with stereo and novice users.

1.3 Specific Problem 2: Design Decisions

Video recordings of recent spherical FTVR displays have been very effective in portraying the display as a realistic "crystal ball" or a palantír from J.R.R. Tolkien's fictional universe – a ball that contains another environment of which you can observe freely [100, 102, 101]. However, unlike the original single-screen FTVR, these systems often omit stereo rendering (a perspective-corrected image for each eye) and instead rely heavily on motion parallax to provide a 3D effect to the user as they walk around the display. The omission of stereo rendering goes unnoticed in monocular video recordings which may misrepresent the quality of the 3D effect when perceived in-person using our naturally binocular vision. Studies on single-screen FTVR have shown the inclusion of stereo depth cues to be beneficial; however, it is currently unclear how detrimental the omission of these cues is for spherical (multi-screen) FTVR which can utilize depth cues from motion parallax much more effectively due to their convex shape and large viewing angles. To continue the design and development of this new display technology efficiently, it would be helpful to be able to reason about the importance of features based on their use cases, performance impact, and cost. Guided by the original FTVR studies [106, 107, 1, 108] and limitations/future work of more recent volumetric FTVR displays [98, 119, 104, 38, 39, 120, 118, 117], we posed a similar set of questions regarding design, calibration, and user interaction, but in the context of volumetric FTVR displays.

1.4 Solution 1: Improved Visual Calibration

The calibration method from Wagemakers, Fafard, and Stavness [104] was shown to be very effective at calibrating an entire FTVR display system and viewpoints from scratch when performed by experts. They also showed that novice users could interpret and minimize visual distortion of a 2D pattern caused by perspective mismatch on a desktop monitor. To improve this calibration method even further, we propose a set of extensions that expand its use cases and accuracy: a stereo calibration procedure using a two

viewpoint model; a realtime, perceptual adjustment procedure of viewpoint or handheld object calibrations; the incorporation of gaze information for a more accurate viewpoint model; and using a new optimizer for faster and more accurate calibrations. We also run novice users through a mock calibration procedure to investigate whether they will be able to minimize visual distortion of the 2D calibration pattern on a spherical FTVR display.

1.5 Solution 2: VR Testbed for FTVR Displays

To investigate the impact of visual errors in volumetric FTVR displays, we required a display with minimal tracking latency and viewpoint registration error as well as ground truth measurements for objective assessment of accuracy. However, even our best physical FTVR display showed noticeable tracking latency with fast head movements and noticeable viewpoint registration error when close to the display. These visual artifacts were not accounted for with the physical display and ground-truth viewpoint measurements were not possible. An initial investigation into the visual distortions showed that they may be caused by perspective-mismatch from non-stereo rendering with binocular viewing (i.e. seeing the same image from both eyes) or a perceived rotation of the virtual content due to viewpoint registration error with stereo rendering [69, 110]. This investigation was supported by a pilot study with a virtual "crystal ball", viewed in VR, that featured the toggling of stereo and the control of tracking latency or calibration error. This initial VR demo proved to be a useful tool for investigating the perception of volumetric FTVR displays, so we continued its development and design a completely virtual volumetric FTVR system that can model the same parameters as a counterpart in the real world. This offers several advantages over using a physical display when running an experiment or user study: 1) the display and tracking system would be co-located in the virtual environment, so ground truth display and viewpoint calibrations could be computed; 2) the IMU-enabled tracking system in VR headsets could be used to provide low latency and accuracy for volumetric FTVR; and 3) the pixel warping techniques that are used to situate the virtual environment for VR headsets would also be applied to the virtual display.

Head-mounted VR has been previously used as a testbed for evaluating new interaction techniques [19], the effectiveness of surgical [87, 46] or industrial [45] training, and the performance of simulated Augmented Reality (AR) experiences [62]. VR has also been used for effectively scaling-up research studies by performing out-of-lab experiments with participants' consumer level VR systems [73]. The benefits of using a virtual system and VR headset for a user study outweigh the disadvantages that come with using a relatively poorly calibrated physical display and tracking system. In addition, the results obtained from studies using a virtual system should still be applicable to the physical display that it was modeled after, since both displays are parameterized in the same fashion and use the same rendering code.

1.6 Solution 3: Evaluation of FTVR Viewing Conditions

With a reliable simulation system available, we design three experiments to compare subjective and objective measurements under different presentation methods (e.g. stereo or non-stereo) of the virtual display. In this study, we investigate the perceptual and performance effects of stereo on a multi-screen FTVR display and challenge the assumption that when a user can freely move about the display, depth cues from motion parallax are so compelling that they overwhelm the need for stereo. To provide ground truth comparisons and to control for as many confounds as possible, we use a VR testbed system that can faithfully simulate FTVR displays inside an immersive virtual environment. This allows us to measure viewpoint(s) precisely, render pixels on the display with virtually no error, minimize overall system latency, and provide the same worn equipment between all conditions. We use a spherical display shape because it has been the most widely adopted shape for volumetric FTVR displays.

1.7 Research Questions

The main research questions regarding the design of volumetric FTVR displays, addressed in this thesis, are as follows:

- How important is the inclusion of stereo for performing 3D tasks with a volumetric FTVR display?
- How accurate does viewpoint registration need to be for visual calibration errors to be imperceptible to users?
- Where exactly is the viewpoint when rendering without stereo?
- Do users notice when stereo is present or absent in a volumetric FTVR display?

1.8 Research Objectives

The research questions described above focus on depth perception of motion parallax and stereo within volumetric FTVR. We incorporate a mix of simulated/synthetic experiments for tool validation and use research methodologies from Human Computer Interaction (HCI) to design user studies that measure the impact or importance of these depth cues. Whenever possible, we use perceptual feedback from users to guide the design and implementation of our solutions. The research objectives of this thesis can be summarized as follows:

1. Extend interactive viewpoint calibration (with and without stereo) to incorporate tracking orientation for a viewer's gaze, provide viewpoint displacement correction for more accurate viewpoint registration, and design a real-time perceptual calibration to fine tune viewpoint registration.

2. Create a simulated, volumetric FTVR display system that models displays in the real world and use it to investigate the source of visual distortions that have been observed in the lab.

3. Perform a user study using simulated FTVR with perceptual and performance tasks to verify the extensions to interactive visual calibration and also challenge previous assumptions about the relative importance of motion parallax and stereoscopic depth cues.

1.9 Organization of Thesis

Chapter 2 covers the background, development, implementation, and calibration of FTVR displays; compares FTVR displays to other types of VR displays; and highlights the novelty of our VR testbed. Chapter 3 describes many improvements to current calibration methods within multi-screen FTVR and proposes a novel, real-time perceptual calibration technique. Chapter 4 describes the implementation and features of our VR tested. Chapter 5 presents three experiments that investigate monocular and binocular depth cues while users complete tasks designed for 3D viewing and interaction. Finally, Chapter 6 concludes with limitations and possible directions for future work, and reiterates important findings and contributions of this work.

2 Related Work

Fish tank virtual reality was developed as an alternative to early head-mounted displays and was shown to be an affordable and effective technology for exploring 3D content and tasks [106, 107, 108]. Its technological underpinnings are similar to that of other VR and AR displays; for example, they may use the same perspective-corrected rendering pipeline or realtime head or eye tracking. FTVR displays situate and constrain a portion of a virtual environment inside a physical one; thus, they can be considered a type of volumetric display. However, FTVR is different in how the perspective is delivered and how the subsystems are calibrated together. The following sections describe other display technologies that share similarities with FTVR displays and goes on to explain the novelty of our simulated FTVR testbed.

2.1 FTVR

Early FTVR displays and experiments used a single flat display to compare combinations of head-coupled rendering, stereoscopic rendering (stereo), and monocular/binocular viewing [1]. They found that both head-coupled rendering and stereo increased a user's performance in a 3D path tracing task, with head-coupled rendering having a larger effect, and that users preferred head-coupled rendering (both monocular and binocular) without stereo. It was speculated that the apparent distaste for stereo may have been a result of the slight ghosting (inter-ocular crosstalk) caused by the slow phosphor decay of the stereoscopic display. Further studies examined the effect that head-coupled motion and structure motion (i.e. constant rotation about an axis) had on a user's understanding of a 3D graph [107]. The findings suggested that either approach offered benefits over static rendering, that the type of motion did not matter that much, and that head-coupled rendering without stereo was significantly worse than all stereo-containing motion conditions. Another experiment studied the difference in task performance in FTVR versus the real world and found that different participants had different responses to the virtual task [99]. Participants in this study noted that "a virtual cube appeared to move along with their head movement," which we will refer to as a floating effect — the perception that objects inside the display seem to float in space rather than remaining stationary like real objects would. Possible causes of this floating effect include non-stereo rendering, viewpoint registration errors, and system latency. A re-evaluation of stereo and kinetic depth effect was undertaken with a much higher resolution display (3840x2400 per eye), but without head-tracking. The kinetic depth effect was implemented by rotating the object of interest back and forth at a constant rate. They found that the object's motion was a stronger depth cue for experienced users and that both stereo and motion increased a

Table 2.1: Design and capabilities of FTVR displays.

Year	Display	Stereo	Multi Screen	Planar Screen	Viewpoint Calibration	Tracking
2019	Virtual FTVR [37]	Yes	Yes	No	Fixed	Infrared + IMU
2018	CoGlobe [117]	Yes	Yes	No	Perceptual	Infrared
2017	HandheldBall [12]	Yes	No	No	Fixed	Infrared
2017	OrbeVR [7]	Yes*	Yes	No	Fixed	Infrared
2017	3DPS [120, 118]	No	Yes	No	Kinematic	Computer Vision
2017	Calibration [104, 38]	No	Yes	Yes	Perceptual	Multiple
2014	Dynamic Stereo [28]	Yes	No	Yes	Head	Infrared
2014	Spheree [98, 40, 25]	No	Yes	No	Head	Infrared
2012	Telehuman [60]	Yes	No	No	Kinematic	Computer Vision
2012	Holodesk [52]	Yes*	No	Yes	Fixed	Computer Vision
2012	MirageTable [10]	Yes	No	No	Fixed	Computer Vision
2011	Snowglobe [17]	Yes	No	No	Fixed	Infrared
2010	pCubee [88, 92]	No	Yes	Yes	Realtime tune	Electromagnetic
2009	Polyhedral [49]	Yes	Yes	Yes	Head	Computer Vision
2008	FaceTrack [90]	No	No	Yes	Midpoint	Computer Vision
2008	Pseudo-3D [51]	No	No	Yes	Fixed	Computer Vision
2007	E-conic [74, 26]	No	Yes	Yes	Head	Ultrasonic
2006	Cubee [89]	No	Yes	Yes	Head	Electromagnetic
2005	Virtual Showcase [13]	Yes	Yes	Yes	Head	Electromagnetic
2001	BNAVE [56, 55]	Yes	Yes	Yes	Head	Electromagnetic
1997	Media^3 [58]	No	Yes	Yes	Head	Electromagnetic
1997	Cubby [35]	No	Yes	Yes	Head	Infrared
1996	ECP 3000 [22]	Yes*	No	Yes	Head	Electromechanical
1995	Tele-window [16, 15]	Yes	No	Yes	Head****	Electromagnetic
1995	Image Warping [71]	Yes*	No	Yes	Fixed	Electromagnetic
1995	Vison [82]	No	No	Yes	Midpoint ***	Computer Vision
1993	FTVR [1, 106, 107]	Yes*	No	Yes	Fixed	Electromechanical
1993	CAVE [29]	Yes	Yes	Yes	Fixed	Electromagnetic
1993	Kinect3D [65]	No	No	Yes	Fixed	Electromagnetic
1993	Virtual Portal [32]	Yes*	Yes	Yes	Fixed	Ultrasonic
1992	High Resolution VR [31]	Yes	No	Yes**	Fixed	Ultrasonic
1992	Interactive Viewpoint [70]	No	No	Yes	Fixed	Electromagnetic
1983	Interactive Stereo [86]	Yes*	No	Yes	Fixed	Electromagnetic
1982	Viewpoint Dependent [41]	Yes	No	Yes	Fixed	Electromagnetic
1982	One-eyed Guys [34]	No	Yes	Yes	Head	Optical
1973	Stereomatrix [109]	Yes	No	Yes	Fixed	Infrared

* Stereo rendering with low refresh rate (¡30 Hz per eye)
** Curvature of CRT monitor was taken into account
*** Distance from display was fixed
**** Strictly horizontally parallel motions

user's ability to read short paths in high density graphs [108].

To generalize Fitts' law to 3D, selection tasks using a stylus and mouse have been studied with FTVR displays. When targets were rendered in front of the display, their distance from the screen had a significant effect on throughput [93, 94, 95]. The farther targets were from the screen, the harder they were to select and the worse they fit a Fitts' model. The effects of stereo and head-tracking were also studied; head-tracking resulted in worse performance when targets were far from the screen, stereo had a larger effect than head-tracking, and stereo negatively affected mouse performance [6].

HoloDesk [52] used a half silvered mirror and perspective-corrected images to enable hand-based interaction with virtual objects. The authors hypothesized that monocular depth cues (e.g. motion parallax) would result in similar task performance as stereo rendering and shutter glasses, but they found that stereo was significantly faster than their monocular condition. However, this finding was only consistent with objects being rendered in front of their display; they found no significant difference between their monocular and stereo conditions when objects were placed behind the display. They also noted that motion parallax was underutilized during the experiment, but they gave no explicit instruction about motion parallax to the participants.

In these studies, the user was seated in front of a single, flat display with a limited range of viewing angles. The introduction of walkaround multi-screen FTVR displays resulted in significantly more viewing angles for non-seated users. It is unclear if results from these studies also apply to walkaround displays and how additional viewing angles affect performance in 3D selection tasks or more generalized 3D tasks.

2.2 Multi-screen FTVR

A recent resurgence in FTVR features volumetric displays that use multiple-screens and a walkaround design to enhance motion parallax cues for compelling 3D [89, 60, 18, 40]. To physically contain the virtual content inside these volumetric displays, a convex display shape, like a box, cylinder, or sphere, is used. Stereo was often excluded from these displays due to technical limitations and the emphasis on motion parallax in single-screen FTVR displays (see Table 2.1). While numerous new designs for multi-screen FTVR systems have been recently proposed, few studies have evaluated how well these different designs provide perceptually correct 3D information and effective 3D interaction. Hagemann et al. reported on a study of eye contact that had users looking at a 3D avatar on a spherical screen while trying to determine if the avatar was making eye contact. Users correctly identified eye contact much more accurately on the FTVR display compared to a static spherical screen [47]. Qian et al. compared depth perception between spherical and flat FTVR displays and reported significantly better performance on the spherical display in both depth-ranking and size-matching tasks [116]. Similar to the single-screen FTVR studies, users remained seated in both of these experiments. This has left motion parallax, and its relative importance to stereo, an understudied depth cue when users are free to move around a volumetric, FTVR display.

2.3 Handheld FTVR

Handheld versions of volumetric FTVR displays have also been explored. A handheld FTVR display without stereo was shown to be faster and more accurate in a 3D tree-tracing task than a conventional 2D monitor [88]. An externally projected spherical display, well-liked by the participants, permitted stereo and head-tracking, but was shown to have lower performance than a planar perspective-corrected display in an object examination task [12]. The conflicting results from these two studies suggest that aspects of the physical display design, system latency, and/or calibration accuracy may be confounding variables.

2.4 Room-based VR and AR

CAVE Automatic Virtual Environment (CAVE) was used to compare presentation technologies, such as stereo and field of view, and found that head-tracked stereo significantly reduced task time over head-tracked non-stereo, but did not affect error [79, 63]. Another surround VR system, RoomAlive, is able to render view-dependent content onto dynamic projected surfaces, creating the possibility for new display shapes and configurations [57, 113, 111, 112]. The authors noted that head-tracking calibration and accurate screen-to-screen mappings are crucial for seamless and accurate presentations of multi-screen projected displays.

2.5 Calibration in FTVR Systems

Many multi-screen FTVR displays use multiple projectors to illuminate a seamless display surface. However, a careful screen calibration is required, so that overlapping projector geometry can render without visual artifacts, such as ghosting or disparity in brightness. Multi-projector calibration procedures for planar [114] and curved [84, 119, 118] display surfaces have been reported that use a camera to automatically compute accurate transformations and blending between projection regions. Box displays have used LCD screens for a compact design, but have the downside of relatively thick seams between screens [88, 49, 104]. Seams themselves can enhance the 3D effect by providing occlusion cues, but thick seams can be obtrusive and potentially disruptive to viewing. Multiple display panels also require screen calibration, but the accuracy requirements are lower, as the screens do not overlap like projected screens. Camera-based calibrations for box displays using checkerboard patterns [104] and AR Toolkit [2] markers [49] have been shown to be quick, accurate, and accessible.

A study of eye angular error of pixels on a spherical multi-screen FTVR display showed that errors in head-tracking had a larger effect than screen calibration and that they distort the perspective-corrected image in different ways [119]. Head-tracking error can cause virtual content to look like it is floating in space or appear visually distorted, whereas screen calibration error can cause double images or ghosting on the surface of the display. Head-tracking error is an important source of error to minimize, not only because of its larger

impact on overall error, but also because convex-shaped FTVR displays have a large field of view which increases the use of head-tracking as viewers move around more.

Cosmo et al. proposed a quantitative method of assessing the accuracy of any view-dependent display by using a camera and a fiducial marker placed at the image plane [28]. However, this approach is inapplicable to volumetric displays because the image plane is not always accessible or coincident with the surface. Wagemakers et al. proposed a method that uses image processing on visual distortions of a regular pattern to quantitatively measure calibration accuracy on a cubic, volumetric display [104]; however, this is only applicable to displays with planar surfaces.

Benko et al. used projection-based AR to superimpose perspective-corrected images on top of occluding geometry [10]. By using the depth map and colour camera from a Kinect [115] sensor (Microsoft, Redmond, WA), they were able to create a tangible virtual world that allowed seamless interaction between virtual and real objects. A single Kinect was used for display surface registration, head tracking, and user interaction, which reduced the complexity of realtime calibration. More elaborate calibration methods must be used when a single sensor system cannot be used for all calibration phases of a volumetric display.

Madritsch et al. proposed a detailed model of a user's eyepoints [66] for stereo-enabled FTVR based on Deering's recommendations for high quality FTVR [31]. It consisted of a geometric model of the relationship between a user's eyes and two beacons on a pair of glasses. It was assumed that the user would be looking straight at the display (i.e. no turning of the head left/right), so that correcting the eye position based on eye gaze would be unnecessary. The two beacons on the glasses were only tracked with positional data relative to a main sensor on the glasses, so an additional assumption was made: the user's head would be in an upright position (i.e. no nodding of the head up/down). This assumption allowed the rotation of the beacons to be computed and the eyepoints to be found using a fixed offset per beacon. Since it is practically impossible for a user to adhere to these assumptions, a system that could provide position and rotation (6DoF) data for the beacons would result in much more accurate eyepoints.

Perceptual calibration methods that generate a geometric viewpoint model have been shown to be more effective than manual measurements or tuning [78, 104]. Ponto et al. proposed a technique that had users align a physical object with known location with a virtual object in a CAVE display. They found that perceptually calibrated viewpoints were wider and deeper than standard approaches assumed, which could significantly improve depth acuity, distance estimation, and the perception of shape [78]. Wagemakers et al. proposed a novel perceptual calibration technique that could generate a viewpoint model at the same time as calibrating the display-to-tracking system transformation without the use of physical calibration tools or alignment objects. They used static 2D patterns that would appear aligned from a known position and had the user align the pattern in key locations around the display [104]. The approach was verified quantitatively on a box-shaped FTVR display and through a user study on a traditional 2D display. They used simulated FTVR displays in the shape of a box with seams, a cube without seams, and a sphere. It was found that participants were able to align a pattern of circles and lines on a spherical display to less than 1.5° by orbiting

a virtual camera at a fixed distance and the approach was able to calibrate the box-shaped display much more accurately than conventional methods.

Despite these improved calibration approaches, an analysis of visual cues may still be confounded by practical issues, such as ghosting, viewpoint errors, and tracking latency, with even the best available volumetric FTVR display.

2.6 Static and Swept Volume Displays

Approaches to building volumetric displays without headtracking exist, however, they employ different mechanisms than FTVR – like a volume medium or fast moving display components – to generate the display volume. They usually fix the viewing angles at regular intervals around the horizontal and vertical axes, which can remove the dependency on headtracking. Compared to FTVR, they generally have higher technical requirements, limited viewing angles, lower refresh rate, and/or lower resolution. For an in-depth review of volumetric displays that do not use headtracking, see Blundell and Schwarz's classification [14].

2.7 Head-mounted Displays

A study on eye-head coordination in head-mounted VR showed that users exhibited more head movement (in the form of head rotation) in VR than in physical reality during simple visual attention tasks [77]. Our user study assumed that this effect would be minimal since the simulated FTVR display would be in the user's field of view, thereby limiting unnecessary head rotation.

3D task performance has also been evaluated in AR with see-through, head-mounted displays and mobile phones. A recent study reported a comprehensive comparison of different viewing and interaction conditions for exploring and manipulating 3D point cloud visualizations as compared to a traditional 2D desktop [5]. It was found that the performance benefits of AR depended on the presentation method and the level of interaction and perception in the task; using a tablet for AR resulted in performance drops in almost all tasks, whereas using head-mounted AR resulted in better performance for high interaction and perception tasks. We adopt the same point cloud visualizations to assess task performance in a spherical FTVR display for our user study.

2.8 View Independent Rendering

In view-independent rendering, there is no perspective-correction, so virtual content cannot be rendered inside the volume of the display and must be rendered entirely on its surface. At the cost of reduced virtual space, this approach allows any practical number of viewers, which makes it very effective in co-located, collaborative workspaces. Benko et al. developed a calibration approach for infrared (IR) touch-sensing along with multi-touch input techniques that proved to be very useful and intuitive on a view-independent

spherical display [11]. To offer more flexibility in our rendering system, we included a view-independent rendering mode that was could also be used as a fallback mode when head-tracking was unavailable.

2.9 Rendering Systems

Single-pass rendering approaches for multiple perspectives and projectors on planar display surfaces have been implemented using homography prewarping [80, 4] and barycentric coordinate interpolation with back-projected ray casting [53]. Additionally, a fully parallel single-pass rendering approach for multi-planar FTVR displays was presented by Harish and Narayanan [50]. A parallel scene-sorting algorithm assigned triangles to facets, or split them up if they bordered facets, and per facet homographies were computed for a perspective-corrected view. Multiple facet geometries were rendered onto a high-resolution rectangular quilt in a single rasterization pass, followed by a per-fragment depth correction shader. This is a highly scalable approach to rendering high quality FTVR displays that was shown to be more efficient and higher-quality than conventional off-axis projections on multiple types of multi-planar polygonal displays. This framework was also shown to work with curved display surfaces by approximating them as polygons with many triangles.

Single-pass rendering algorithms avoid many of the artifacts that may come from the sampling stage in a two-pass rendering approach; however, they are difficult to generalize to non-planar surfaces due to the non-linear transformation between projector pixels and display surface pixels. For our rendering system, we implemented a simple two-pass approach, so that it could easily be incorporated into a fully featured rendering pipeline.

2.10 Novelty of Simulation System

We have observed that latency is more noticeable in stereo rendering and also results in pronounced floating of the virtual content. The Oculus Rift system provided extremely accurate and low latency viewpoint tracking using built-in inertial measurement units (IMUs) and kinematically constrained prediction models [64, 33]. Minimizing head-tracking error was crucial because it is a significant contributor to eye angular error of pixels on a spherical FTVR display [119]. In addition to low latency tracking, VR headsets apply pixel warping techniques to interpolate frames using motion tracking data that provides even more accurate pixel positions. The Oculus API provided the position and orientation of the viewpoints for every image our simulated FTVR display generated. We were able to use these reported viewpoints as ground truth for Experiment 1.

Running a user study with our simulated FTVR allowed more control of the experiment between conditions than a traditional user study with a physical FTVR display. Participants always wore the same equipment and the visual fidelity of the VR headset was always the same. The system also provided a perfectly calibrated FTVR experience. Since the tracking system and display system were colocated in the virtual environment, ground truth viewpoint and tracking-space calibrations could be computed, and pixel-perfect projection of

perspective-corrected images on the display surface was possible. This ensured that any noticeable visual distortions would be caused solely by the participants' perception of the display content.

We implemented projector image prewarping using two-pass, per-pixel rendering. Because this approach only requires access to programmable vertex/fragment shaders and the ability to render to a texture, it offers more flexibility when integrating it with an already existing graphics pipeline. It features dynamic projection matrices that maintain pixel density to overcome sampling artifacts, requires no ray-tracing, and easily outputs to either real or virtual projectors and planar screens. This allowed us to integrate real and virtual FTVR rendering with Unity software (Unity Technologies, San Francisco, CA) and leverage its fully featured graphics pipeline for more advanced rendering and post-processing.

In addition to implementing virtual projectors, display surfaces, and planar screens, we modeled the parameters of an FTVR display so that synthetic error could be introduced to the system. We were able to control the viewpoint model, projector lens model, and tracking system calibration/latency. This feature was used to investigate and replicate unknown sources of error observed in real FTVR displays.

3 Extensions to Interactive Visual Calibration

The interactive visual calibration approach proposed by Wagemakers, Fafard, and Stavness [104] relies solely on visual cues from the display to guide a viewer into known locations while still allowing natural and unimpeded movement around the display. It renders patterns on the display such that they will appear without distortion if viewed from a known calibration position ($\mathbf{V}$ in Figure 3.1). The patterns are designed so that when they appear distorted, the user can easily determine the location at which it would look correct. This process aligns a user's viewpoint to known locations relative to the display, but only the head position is recorded. This is repeated through a set of predetermined calibration positions that define a path that minimizes backtracking and maximizes workspace coverage. All the pairs of head positions and calibration locations are used to create the viewpoint-to-head and tracker-to-display transformations. This method only requires positional information (orientation is not needed), so it works well with many types of trackers.

Only relying on positional data (3DoF) increases the accessibility of the calibration because it can be used with a broad range of tracking systems; however, without orientation data (6DoF), the viewpoint model may become inaccurate post-calibration. During the calibration phase, a user's gaze is assumed and instructed to be coincident along the calibration rays; however, after the calibration phase, it is difficult to enforce the same gaze assumptions without strict adherence to the assumptions. This 3DoF viewpoint model assumes that the user looks directly at the origin of the display without any side-to-side or up/down head tilt; violating these gaze assumptions would put the user's eyepoint into an unexpected location, potentially causing noticeable visual distortion on the display. We propose the following set of extensions designed to increase the accuracy of a calibration in a variety of new use cases within volumetric FTVR. We derived 3DoF and 6DoF post-calibration transformations for accurate eyepoint placement; relaxed the gaze assumptions after calibration with compatible 6DoF tracking systems; introduced a stereo viewpoint model that can skip the measurement of user PD; improved the accuracy and execution time of the original optimization program; and designed a realtime, perceptual calibration tool that can adjust and improve user-dependent parameters of existing calibrations. The result is an easy to use, fully featured, perceptual calibration toolbox within volumetric FTVR displays that does not require any error-prone, manual measurements.

$\mathbf{R}_{TD}$	rotation portion of $\mathbf{X}_{TD}$
$\mathbf{d}_{TD}$	translation portion of $\mathbf{X}_{TD}$
$\mathbf{R}_{VH}$	rotation from $\mathbf{V}$ to $\mathbf{H}$

Table 3.1: $\mathbf{X}_{TD}$ was split into its constituent parts for clarity and $\mathbf{R}_{VH}$ was added to define gaze and relax some of the viewpoint model assumptions.

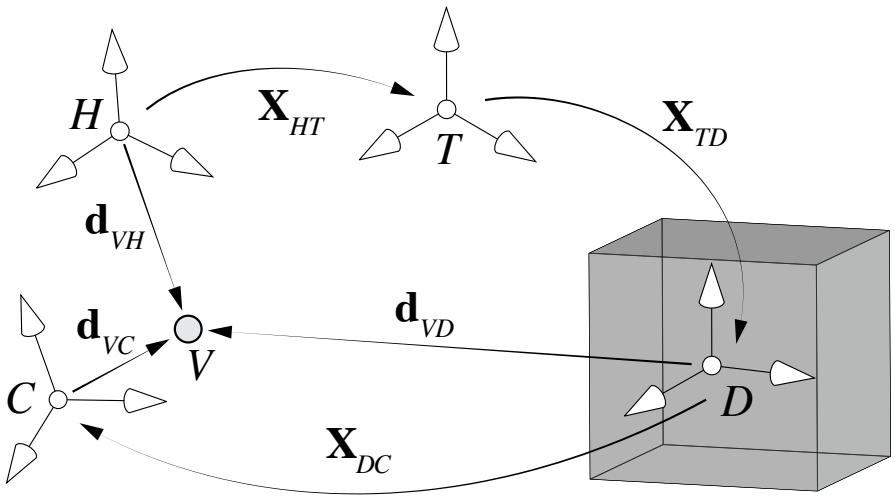


Figure 3.1: Coordinate frames in our calibration problem, including the display (**D**), tracking system (**T**), head (**H**), and calibration point (**C**) frames, and viewpoint position (**V**). **X** and **d** denote rigid transformations and translations, respectively.

Source: Wagemakers et al [104]

In addition to the variables defined in Figure 3.1, Table 3.1 summarizes the additional variables that are used to formalize the extensions.

3.1 Adding Orientation

When the tracking system provides only positional (3DoF) data, the orientation (rotation) of the eyepoint(s) must be reconstructed from a set of assumptions. We assumed, similar to previous research with perceptual calibrations [78], that the viewer looks directly at the origin (center) of the display with their head upright (no tilt). The reconstruction recovers the coordinate frame using a view matrix – a transformation most commonly used to represent the position and orientation of a pinhole camera in computer graphics (also known as a camera matrix). The rotation is composed of three basis vectors named the forward, right, and up axes. The forward-axis is the displacement of the viewpoint position normalized to unit length (i.e. the direction from the eyepoint to the display), the right-axis is the cross product of the forward-axis with the normal of the ground plane (i.e. what "up" is in the physical room), and the up-axis is the cross product

of the right-axis and forward-axis. It is common to level the display with the ground so that "up" from the display's point of view is the same as the room's "up" direction. This makes the display calibration much easier since there will be no need to measure, and correct for, a transformation between the physical room and the display (i.e. they share the same ground plane).

With the addition of rotational (6DoF) tracking, the head frame of reference (R_{HiT}) can be measured directly; however, a fixed rotation (R_{HV}) between the viewpoint and the head remains unknown. We approximate this unknown rotation by recording the orientation of the head at each calibration ray and use the following relationship:

$$\mathbf{R}_{DCi} = \mathbf{R}_{HV}\mathbf{R}_{THi}\mathbf{R}_{DT} \tag{3.1}$$

Isolating $\mathbf{R}_{HV}$ yields,

$$\mathbf{R}_{HV} = \mathbf{R}_{DCi}\mathbf{R}_{TD}\mathbf{R}_{HiT} \tag{3.2}$$

and with n points,

$$\mathbf{R}_{HV} = f\begin{pmatrix} \mathbf{R}_{DC1}\mathbf{R}_{TD}\mathbf{R}_{H1T} \\ \mathbf{R}_{DC2}\mathbf{R}_{TD}\mathbf{R}_{H2T} \\ \vdots \\ \mathbf{R}_{DCn}\mathbf{R}_{TD}\mathbf{R}_{HnT} \end{pmatrix} \tag{3.3}$$

where $f(\mathbf{x})$ is a function that produces an average rotation given a vector of similar rotations. We used unit quaternion representation for rotations in 3-space and equation 17 from Markley [67] to average the measurements. Because of the assumption that the user's viewpoint is aligned to the calibration rays, the coordinate frame change between the viewpoint and calibration ray, $\mathbf{R}_{VC}$, would be the identity (i.e. no rotation) and has thus been omitted from Equations (3.1) to (3.3).

3.2 Adding Stereoscopic Support

Wagemakers, Fafard, and Stavness evaluated several 2D calibration patterns on spherical and cubic display shapes [104]. We attempted to add stereoscopic support to this calibration approach by designing patterns in 3D with additional depth cues (see one example in Figure 3.2). We tested many iterations of 3D calibration shapes using headset VR; however, several issues arose that kept them from being as intuitive and effective as the 2D patterns. Headtracked stereoscopic rendering naturally requires two viewpoints, but, due to the variance of human pupillary distance (PD), no general viewpoint model could be assumed at the start of calibration that would work well for all users. An initial measurement step for PD could have been

introduced, but this would have been counter-productive to the design of a quick, easy, and perceptually-based calibration. In addition to this, visual distortions caused by incorrect perspective are more difficult to interpret in stereo. When a 2D pattern is perceived in mono, straight lines bend and shapes squish and stretch. Using a 3D pattern in stereo, an incorrect perspective can be perceived as a rotation of the virtual content [69, 110] in addition to the distortions present in mono. Early testing suggested that the perceived rotations made the calibration pattern ineffective at guiding a user to the intended perspective and that viewing stereo from incorrect perspectives caused significant eye discomfort. To circumvent these issues with non-headtracked stereo during the calibration phase, we presented the original 2D pattern in mono and performed two calibrations at once by interleaving the left and right eye measurements. This produces optimized calibrations for each eye and skips the PD measurement step.

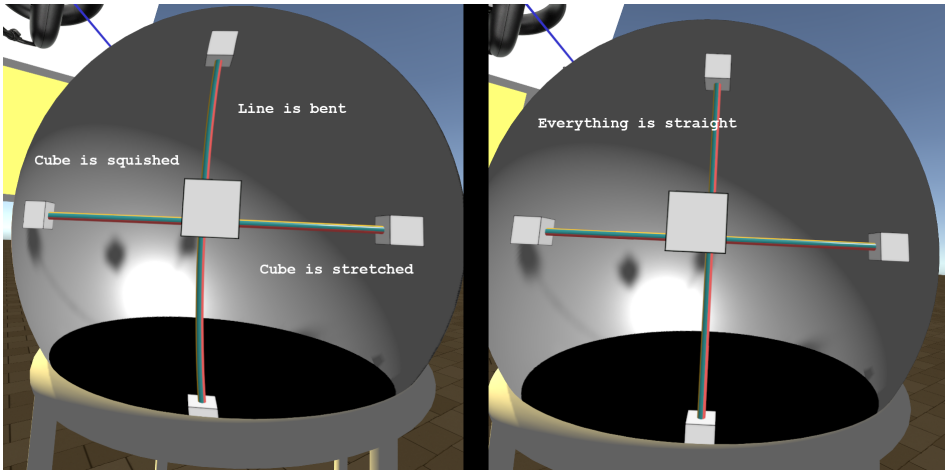


Figure 3.2: A 3D shape consisting of textured cylinders with cubes attached to the end is shown from the correct perspective (right) and an incorrect perspective (left). The distortion from incorrect perspective in this 3D pattern is manifested as bent lines and stretched/squished cubes.

3.3 Improving Optimization

The viewpoint calibration optimization proposed by Wagemakers et al. was written in MATLAB and compiled to a standalone executable for Windows [104]. This offered some portability and interoperability, however, the executable depended on a bulky (800+ MB) MATLAB runtime and required data conversion between processes. Since all of our development with FTVR displays used C#.NET 3.5+ in Unity software, we reimplemented the optimizer in C# using the ALGLIB optimization library to increase portability and interoperability with our codebase.

The MATLAB optimizer used the sequential quadratic programming (sqp) non-linear algorithm described in Chapter 18 of Nocedal and Wright [75], which does not require a Jacobian matrix to be specified. However, the ALGLIB optimizer used the Levenberg-Marquardt algorithm [72], which does require a Jacobian matrix.

In total, the new optimizer required a vector of parameters $\mathbf{x}$, lower and upper bounds for each parameter, an objective function $O(\mathbf{x})$, a Jacobian matrix $\mathbf{J}_O$, and an initial guess for each parameter. The parameters remained the same as in the original formulation: $\mathbf{X}_{TD}$ and $\mathbf{d}_{VH}$. The rotation of the transformation was parameterized as a unit quaternion ($\mathbf{q}$) and the depth (forward) component of the offset was not included due to the depth invariance of the calibration method. This meant there were nine parameters in total. The bounds of $\mathbf{x}$ were defined in a human readable configuration file for easy access.

$$\mathbf{x} =< \mathbf{q}_{wTD}, \mathbf{q}_{xTD}, \mathbf{q}_{yTD}, \mathbf{q}_{zTD}, \mathbf{d}_{xTD}, \mathbf{d}_{yTD}, \mathbf{d}_{zTD}, \mathbf{d}_{xVH}, \mathbf{d}_{yVH} > \tag{3.4}$$

The objective function was implemented using an n-dimensional formulation of the distance between point and a line in vector notation [48]. In this formulation, the points ($\mathbf{p}$) are the measured head points transformed into display space with offset correction (i.e. the estimate of the viewpoint) and the lines are the corresponding calibration rays. Because the calibration rays originate from the display, only their direction ($\mathbf{n}$) is needed. This objective function satisfies the depth-invariance requirement because the calibration rays are naturally coincident with the forward/looking direction, thus the point-line distance is unaffected by the distance (depth) the point is measured at.

$$\mathbf{n}_i = \mathbf{d}_{CiD} / \left\| \mathbf{d}_{CiD} \right\| \tag{3.5}$$

$$\mathbf{p}_i = \mathbf{X}_{TD}\mathbf{d}_{HiT} + \mathbf{R}_{CiD}\mathbf{d}_{VH} \tag{3.6}$$

$$O(\mathbf{x}) = \sum \left\| -\mathbf{p}_i - (-\mathbf{p}_i \cdot \mathbf{n}_i) * \mathbf{n}_i \right\| \tag{3.7}$$

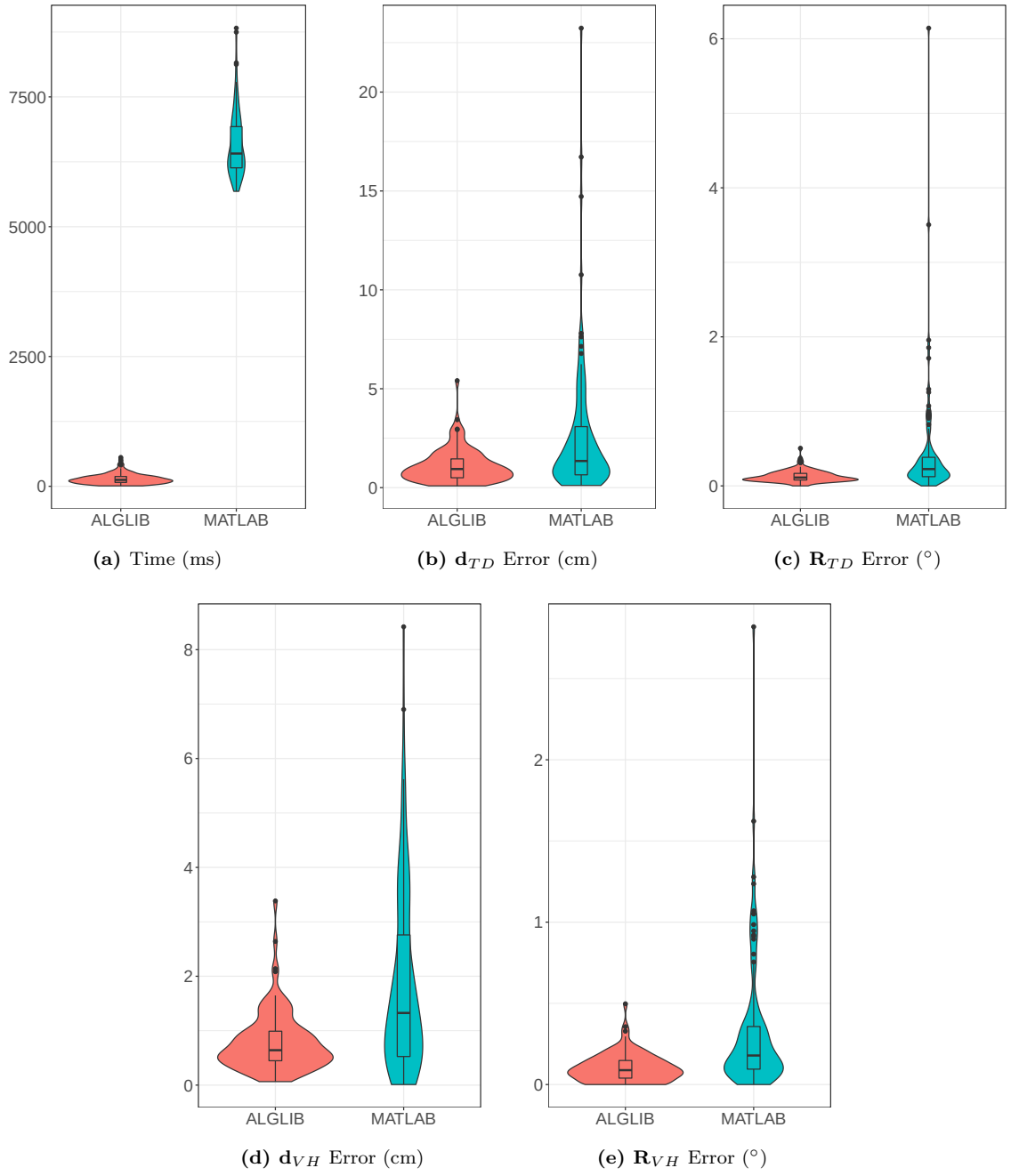
We were not able to find a closed-form representation of $\mathbf{J}_O$, so the first row was generated using the Symbolic Toolbox and `jacobian` function from MATLAB and saved as C# code. A dynamic loop was used to generate the remaining rows based on the number of calibration point correspondences. The toolbox expanded the vectors into component-based operations and just the first row was over 221,000 characters. For the interested reader, the full Jacobian matrix be found in the `Optimization.cs` source file.

To achieve an accurate optimization that reduces the chance of finding local minima, the parameters must be initialized close to the ground truth. The initial guess for $\mathbf{q}_{TD}$ was computed using a least-squares fitting of two 3D point sets [3]. The first set was the calibration points in $\mathbf{D}$, and the second set was the measured points in $\mathbf{T}$, minus their geometric mean. The initial guess for $\mathbf{d}_{TD}$ was computed using a least-squares fitting of the measured points to the calibration rays. The derivation and explanation of this method can be found in Section 5.3 and Appendix B of Wagemakers' Master's thesis [103]. The initial guess of $\mathbf{d}_{VH}$ was computed using the mean displacement between measured points and calibration rays.

3.3.1 Comparison

A synthetic test was designed to compare execution time and error between the MATLAB and ALGLIB optimizers. Time was measured in milliseconds (ms) and included the time it took for each optimizer to

terminate with a result. Both optimizers were set to the same maximum number of iterations (1500). MATLAB had a disadvantage with respect to time because it had to be called as an external process that uses file I/O to transfer data, whereas, ALGLIB could be called from our C# environment. Errors were measured as the distance (cm) between ground truth and estimated displacements or as the angle (°) between ground truth and estimated rotations. Optimizers were tested with 100 randomized, simulated calibrations that added uniform noise – ±0.5 cm up/right and ±5 cm forward in the viewer's frame of reference – to each calibration correspondence. The plots in Figure 3.3 indicate that the ALGLIB optimizer is consistently faster, more accurate, and produces fewer outliers.



(a) Time (ms) **(b)** $\mathbf{d}_{TD}$ Error (cm) **(c)** $\mathbf{R}_{TD}$ Error (°)

(d) $\mathbf{d}_{VH}$ Error (cm) **(e)** $\mathbf{R}_{VH}$ Error (°)

Figure 3.3: Violin plots with inset box plots of optimizer performance vs. optimizer type.

3.4 Viewpoint Model Transformations

After a perceptual calibration is completed, the estimated parameters must be used to transform the stream of headtracking data, so that the viewpoint corresponds with the viewer's eyepoint rather than the tracked point. The following subsections describe how to apply this transformation depending on the capabilities of

the headtracking system and calibration.

3.4.1 Position Only

When an interactive visual calibration is performed using a 3DoF tracking system, such as the Microsoft Kinect, orientation is not provided, so the user's gaze must be assumed. During calibration, patterns guide the user to look down a known calibration ray which gives a good approximation of their gaze. However, after calibration, this approximation is no longer possible because the user may direct their gaze anywhere they wish. This makes determining the eyepoint difficult because the coordinate frame in which the offset is defined – the user's gaze – is unknown. Even if we assume that a user would always be looking at the center of the display with their head upright, we would still be unable to compute the gaze because the eyepoint location is unknown. This creates a cyclic dependency between eyepoint and gaze.

We formulated a geometric solution to this problem by using the constituent components of the offset ($\mathbf{d}_{HV}$: $< \mathbf{o}_x, \mathbf{o}_y, \mathbf{o}_z >$) by observing that each component had a distinct effect on the relationship between the eyepoint and head point in spherical coordinates.

Observation 1: $\mathbf{o}_x$ affects θ and r. The azimuthal angle (γ) between the head point and eyepoint can be isolated by projecting the components of the head point onto the XZ-plane of $\mathbf{D}$. With the XZ-distance (a) and $\mathbf{o}_x$ known, γ can be solved.

$$a = \|(\mathbf{h}_x, \mathbf{h}_z)\| \tag{3.8}$$

$$\gamma = \sin^{-1}(\mathbf{o}_x / a) \tag{3.9}$$

$$\theta = \tan^{-1}(\mathbf{h}_z / \mathbf{h}_x) - \gamma \tag{3.10}$$

Observation 2: $\mathbf{o}_y$ affects ϕ and r. The polar angle (ψ) between the head point and eyepoint can be isolated by projecting the components of the head point onto the YZ-plane of $\mathbf{V}$. However, since the eyepoint is unknown at this time, the location of this plane was computed by rotating the YZ-plane of $\mathbf{H}$ around the y-axis of $\mathbf{D}$ by γ. With the YZ-distance (b) and $\mathbf{o}_y$ known, ψ can be solved.

$$b = \|(\mathbf{h}_x \cos\gamma, \mathbf{h}_y, \mathbf{h}_z \cos\gamma\| \tag{3.11}$$

$$\psi = \sin^{-1}(\mathbf{o}_y / b) \tag{3.12}$$

$$\phi = \cos^{-1}(\mathbf{h}_y / b) - \psi \tag{3.13}$$

Observation 3: $\mathbf{o}_z$ affects only r. The effect that each component has on r is accounted for here by rotating b by ψ towards $\mathbf{V}$ and rescaling using $\mathbf{o}_z$.

$$r = b\cos\psi + \mathbf{o}_z \tag{3.14}$$

Finally, the Cartesian coordinates of $\mathbf{d}_{VD}$ can be computed by converting from spherical coordinates.

$$\mathbf{d}_{VD} = \begin{bmatrix} r\cos\theta\sin\phi \\ r\cos\phi \\ r\sin\theta\sin\phi \end{bmatrix} \tag{3.15}$$

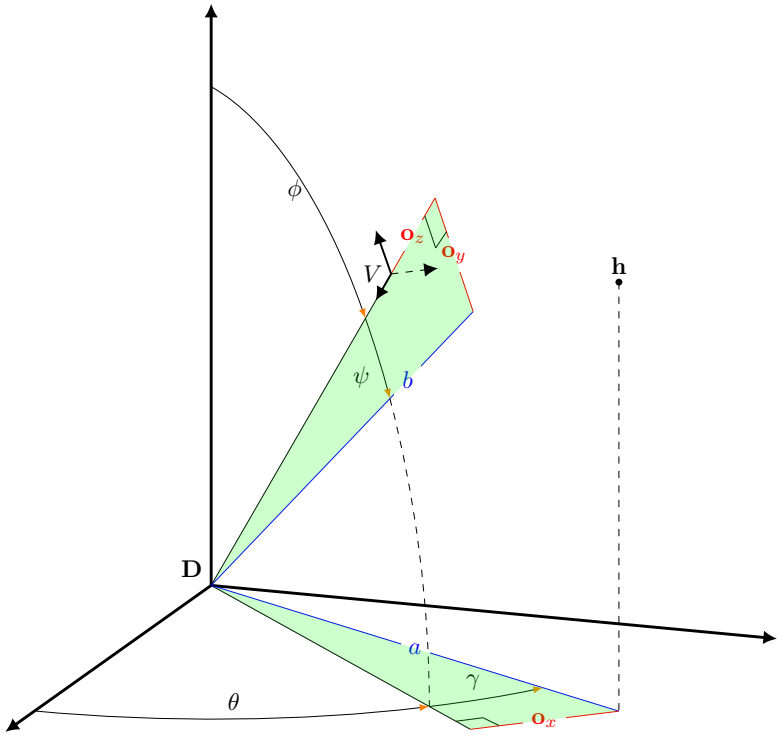


Figure 3.4: A geometric representation in ($\mathbf{D}$) of the unknown viewpoint ($\mathbf{V}$) coordinate frame, the measured head point ($\mathbf{h}$), the components ($< \mathbf{o}_x, \mathbf{o}_y, \mathbf{o}_z >$) of the offset ($\mathbf{d}_{HV}$), and the spherical coordinates being computed (θ, ϕ, r). Here, γ and ψ represent the azimuthal and polar effects of the offset, respectively. Highlighted in green are the right-angled triangles that exist when the viewing assumptions are met.

In these equations, θ is the angle measured from the z-axis (forward) of $\mathbf{D}$, ϕ is the polar angle from the y-axis (up) of $\mathbf{D}$, r is the distance from the origin $\mathbf{D}$, and $< \mathbf{h}_x, \mathbf{h}_y, \mathbf{h}_z >$ are the components of $\mathbf{d}_{HD}$. Refer to Figure 3.4 for an illustration of the problem and relevant variables.

3.4.2 Position and Orientation

When an interactive visual calibration is performed using a 6DoF tracking system, such as an OptiTrack or Polhemus Fastrak system, the calibration provides all the necessary parameters needed to represent the eyepoint and gaze relative to the display. Eyepoint position and rotation can be computed using the following transformations.

$$\mathbf{R}_{VD} = \mathbf{R}_{TD}\mathbf{R}_{HT}\mathbf{R}_{VH} \tag{3.16}$$

$$\mathbf{d}_{VD} = \mathbf{X}_{TD}\mathbf{d}_{HT} + \mathbf{R}_{VD}\mathbf{d}_{VH} \tag{3.17}$$

3.5 Realtime Refinement of Calibrations

Interactive visual calibration provides faster and more accurate calibrations than manual; however, even with the extensions described above, this method is not suitable for several common FTVR use cases. If a new user attempts to use a previous individual's calibration, the display-to-tracking transformation will be correct, but the head-to-viewpoint transformation(s) may be inaccurate due to differences in PD between the users or how the tracker is affixed. Instead of recalibrating the entire system from scratch with this new user, it would be convenient to use a calibration method that only updates the parameters that changed: the user-dependent parameters. Another use case can occur when a calibration is very accurate, but still contains noticeable distortion. It would again be convenient to perform small adjustments to the calibration where visual imperfections are noticed, instead of recalibrating from scratch. Given these common situations, we propose a realtime, perceptual calibration technique that uses an already existing calibration to apply small corrections anywhere the user notices imperfections.

In addition to head-tracking for viewpoint rendering, FTVR displays often track handheld objects to control virtual pointer into the 3D scene or act as manipulation tools for 3D content [93, 94, 95]. Similar to the previous discussion of viewpoint models, tracked objects require a calibration to correct for any differences in position or rotation between the tracking sensor and object origin. Interactive viewpoint calibration is not compatible with handheld objects; the perceptual patterns require a viewpoint, which handheld objects do not have. The same realtime correction approach for viewpoints can be performed to visually align a physical object/pointer to the virtual object/pointer rendered in the scene.

3.5.1 Viewpoints

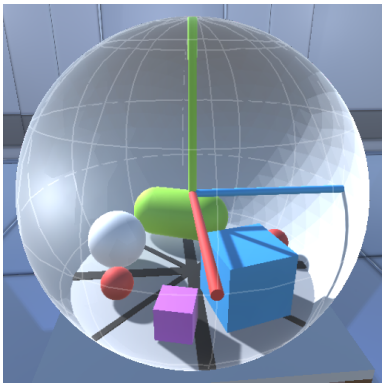
For realtime viewpoint corrections, the user would be provided with the following instructions.

1. Walk around the display to look for visual imperfections. The manifestations of the imperfections depend on the display shape. For example, on a spherical display, distortions are most evident near the edges of the display, while on a box display, distortions are most evident across the seams of screens.

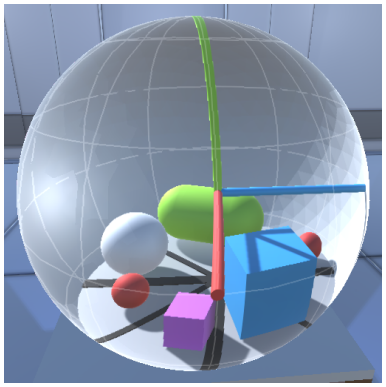
2. When a visual imperfection is found, stand comfortably and freeze the virtual content in place. (See Figure 3.5 (b).)

3. Make small movements with your head to alter your perspective until the virtual content has the least distortion. (See Figure 3.5 (c).)

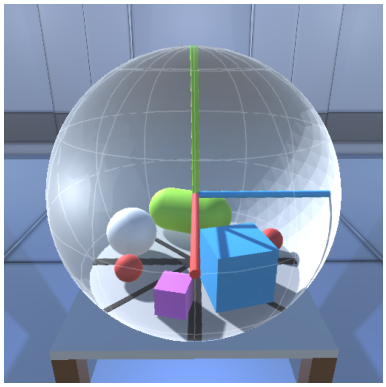
4. Repeat steps 1-3 as many times as needed.



(a) Exact calibration is perceived as consistent lines, angles, or shapes from all perspectives.



(b) Miscalibration is perceived as the distortion of lines, angles, or shapes.



(c) The virtual content is frozen and the user moves their perspective to minimize distortion.

Figure 3.5: The shapes, lines, and angles of the virtual content are affected by a user's calibration. The calibration can be exact (a) or miscalibrated (b). The calibration from (b) can be corrected to (a) by using the change in a user's position and rotation between (b) and (c).

3.5.2 Handheld Objects

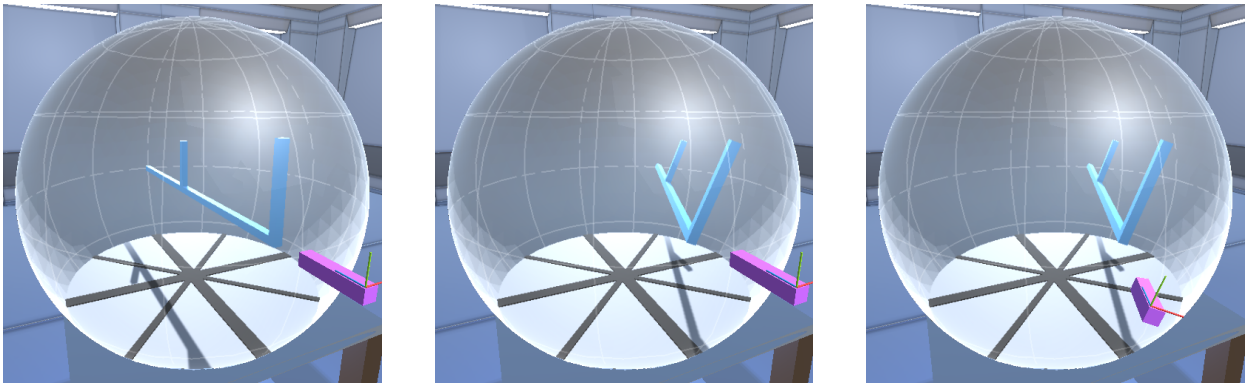
For realtime handheld object corrections, the user would be provided with the following instructions.

1. Manipulate the handheld object and observe the relationship between it and its virtual counterpart.

2. When there is a perceptual mismatch between them, stand comfortably and freeze the virtual content in place. (See Figure 3.6 (b).) When using a tracked pointing device, the user may perceive a mismatch of the targeting direction between the handheld and virtual pointers; it may seem like the handheld pointing device is pointed accurately while the virtual pointer is not (or vice versa).

3. While remaining at the same perspective, move the handheld object into perceptual alignment with its virtual counterpart and unfreeze the virtual content. (See Figure 3.6 (c).)

4. Repeat steps 1-3 as many times as needed.



(a) Exact calibration is perceived as perfect alignment between the real and virtual objects.

(b) Miscalibration is perceived as misalignment of the real and virtual objects.

(c) The virtual content is frozen and the handheld object is moved into alignment.

Figure 3.6: A real object (purple) and virtual object (blue) are connected through a calibration. The calibration from (b) can be corrected to (a) by using the change in the handheld object's position and rotation between (b) and (c).

3.5.3 Updating the Calibration

Performing this calibration adjustment with a viewpoint or handheld object produces a difference in rotation and position between the start and end of a sample. An arbitrary number of samples (n) may be taken and the calibration can be updated using the mean sample values. We assume, and present some results in Chapter 5, that perceptual misalignment from errors in user-dependent parameters will guide the user towards the correct perspective in a consistent (in their frame of reference) direction. This produces samples that are numerically close to each other. For example, if a viewpoint calibration caused the perspective to render 5 cm to the right of the eyepoint, then this calibration procedure should guide the user 5 cm to the right, regardless of where a sample is taken.

We define the ith sample as the rotation ($\mathbf{R}_{GFi}$) and translation ($\mathbf{d}_{GFi}$) of the realtime coordinate frame (G) in the frozen coordinate frame (F).

$$\mathbf{R}_{GFi} = \mathbf{R}_{DF}\mathbf{R}_{GD} \tag{3.18}$$

$$\mathbf{d}_{GFi} = \mathbf{X}_{DF}\mathbf{d}_{GD} \tag{3.19}$$

We use the functions f and g to compute weighted means of the rotations and translations, respectively.

The user-dependent parameters are then updated by adding the mean sample values to the current values.

$$\hat{\mathbf{R}}_{VH} = f\begin{pmatrix}\mathbf{R}_{GF1}\\ \mathbf{R}_{GF2}\\ \vdots\\ \mathbf{R}_{GFn}\end{pmatrix}\mathbf{R}_{VH} \tag{3.20}$$

$$\hat{\mathbf{d}}_{VH} = g\begin{pmatrix}\mathbf{d}_{GF1}\\ \mathbf{d}_{GF2}\\ \vdots\\ \mathbf{d}_{GFn}\end{pmatrix} + \mathbf{d}_{VH} \tag{3.21}$$

We used uniform weights; however, as future work, it could be useful to weight each sample based on a user rating of perceptual misalignment.

4 Virtualization of FTVR

This chapter describes the multi-screen FTVR rendering system from Fafard et al. [39] and focuses on the steps taken to implement a virtual FTVR display within a virtual reality environment. The objectives of the virtual output were to provide a convenient output for desktop or headset VR when a physical FTVR display was unavailable, to offer a preview inside the development environment for easier content design and testing, and to provide a perfectly calibrated FTVR experience for perceptual and performance studies. To implement accurate virtual simulations of physical FTVR displays, each component of the physical display – projectors, screens, surfaces, and coordinate frames – was modeled and colocated alongside the virtual content. In addition to this, we reused as much of the physical rendering pipeline as possible to reduce code management and to increase the accuracy of the simulation.

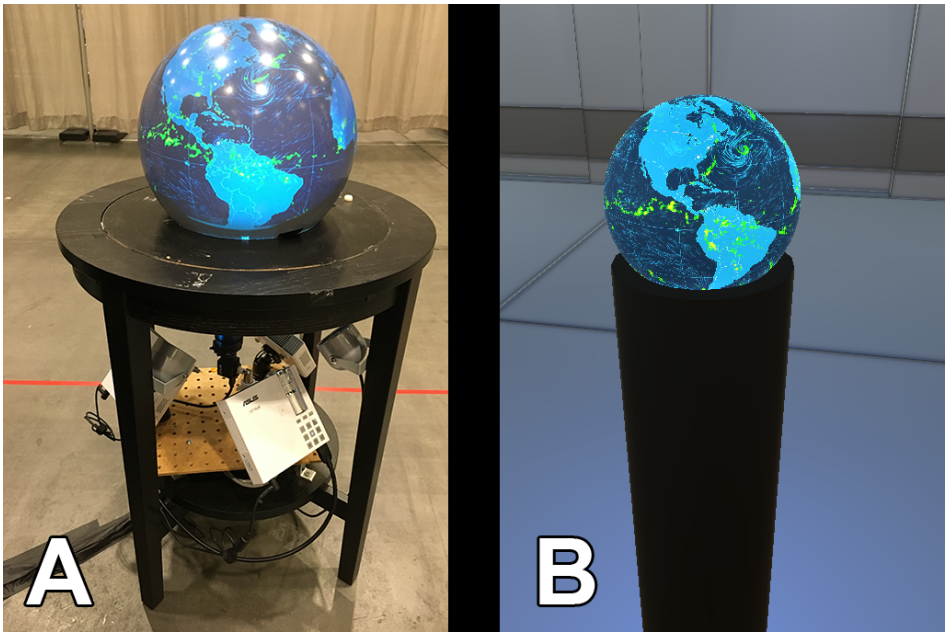


Figure 4.1: Projectors mounted underneath a physical FTVR display (A, left) render content onto a spherical surface. The same process can be replicated in a virtual environment (B, right) with virtual projectors (not visible) and a spherical mesh.

4.1 Rendering Overview

A flexible rendering system was designed to support stereoscopic rendering, multiple viewing modes depending on the number of viewers, and physical or virtual outputs (see both outputs in Figure 4.1). The rendering system was capable of responding to changes in stereo and viewers in realtime and changes in display output through a configuration file loaded at startup. The rendering overview of an image is shown in Figure 4.2. This rendering system features a two-pass approach for rendering perspective-corrected images: 1) render the image(s) from a viewer's perspective(s), and 2) render the pixels on the output display.

During the first pass, perspective images for each viewpoint are rendered into a `RenderTexture` using a `CoRoutine` — a repeating function — that executes immediately after the physics, input, and game logic updates every frame. The system supports one fully stereoscopic viewer using two eye passes or two monoscopic viewers using a single eye pass each. If there are no tracked viewers, then a cubemap camera rig is used to render the virtual content into a cubemap `RenderTexture`.

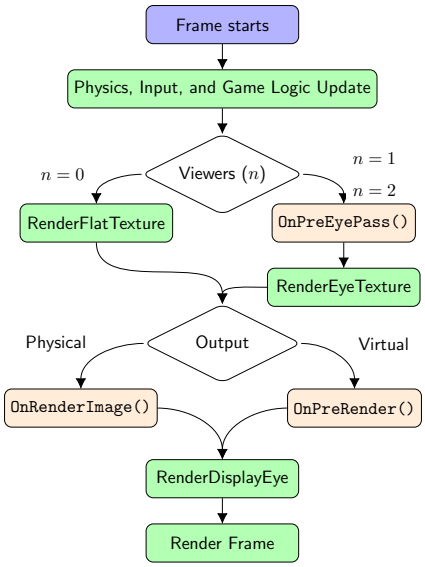


Figure 4.2: The `OnPreEyePass()` event occurs just before an eye texture is rendered and is used for near-surface clipping. The `OnRenderImage()` event occurs when the physical display renders. The `OnPreRender()` event occurs just before the display (desktop monitor or VR headset) renders. In stereoscopic rendering, a frame is rendered for each eye following the diagram.

The second pass is slightly different between physical and virtual outputs. For a physical output, projectors or screens are joined together as a single display – a mosaic – and an `OnRenderImage()` event occurs during the rendering `CoRoutine`; this event triggers the rendering system to sample the appropriate `RenderTexture` and build the mosaic image using a 2D texture pass. For a virtual output, an `OnPreRender()` event occurs when the display output (either a desktop monitor or VR headset) begin to render a frame; this event triggers the rendering system to sample the appropriate `RenderTexture` and project the pixels onto the virtual display

surface. The ordering of these events ensured that the virtual display was updated prior to the display for proper frame syncing (see Figure 4.2).

4.2 Planar Display Screens

FTVR rendering pipelines that output to flat screens, such as used in cubic FTVR displays [88], often use an off-axis projection to render a perspective-correct image. This approach also has the ability to clip the virtual content at the screen boundary by placing the near clip plane coincident with the screen.

The following off-axis projection matrix was used as defined in Unity documentation [97]. Let l, r, t, b be the left, right, top, and bottom offsets and n, f be the near and far plane distances that define the projection frustum.

$$P = \begin{bmatrix} 2n/(r-l) & 0 & (r+l)/(r-l) & 0 \\ 0 & 2n/(t-b) & (t+b)/(t-b) & 0 \\ 0 & 0 & -(f+n)/(f-n) & -2fn/(f-n) \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{4.1}$$

This projection was used for both planar display surfaces and mobile display screens since they both acted as "windows" into the virtual world. A function was implemented that could compute the necessary frustum parameters given the screen size (s_w, s_h is width and height respectively), position ($\mathbf{s}$), and orientation in the virtual environment. Since the camera must be orthogonal to the screen, the camera's ($\mathbf{c}$) axes (c_f, c_r, c_u is forward, right, and up respectively) were used. The far plane distance (f) was assumed to be known. The equations follow.

$$n = \|\mathbf{s} - \mathbf{c}\| \cdot c_f \tag{4.2}$$

$$r = \|\mathbf{s} - \mathbf{c}\| \cdot c_r + s_w/2 \tag{4.3}$$

$$l = \|\mathbf{s} - \mathbf{c}\| \cdot c_r - s_w/2 \tag{4.4}$$

$$t = \|\mathbf{s} - \mathbf{c}\| \cdot c_u + s_h/2 \tag{4.5}$$

$$b = \|\mathbf{s} - \mathbf{c}\| \cdot c_u - s_h/2 \tag{4.6}$$

It is also possible to use the more general, two-pass approach described in the next section with flat screens.

4.3 Projected Surfaces

Following the camera calibration and 3d reconstruction model from OpenCV [23, 76], a projector with intrinsic and distortion parameters was implemented using vertex and fragment shader programs. The projector was

modeled as the inverse of the OpenCV camera model and was injected into the rendering pipelines by using the Unity `Projector` component. This component has programmable vertex and fragment shaders that run during a transparent pass late in the pipeline. The extrinsic properties of the projector were modeled by the position and orientation of the component in the scene, which make up the view matrix (V). The intrinsic parameters – focal lengths (f_x, f_y), principle point (c_x, c_y), and skew (α) – were modeled by the projection matrix (P) of the component using the following equation.

$$P = \begin{bmatrix} 2f_x/w & 2f_y \tan\alpha & 2c_x/w - 1 & 0 \\ 0 & 2f_y/h & 2c_y/h - 1 & 0 \\ 0 & 0 & -(f+n)/(f-n) & 2fn/(n-f) \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{4.7}$$

Where w and h are the width and height of the projector image in pixels, respectively, and f and n are the far and near plane distances, respectively.

The distortion equations of the camera model are used to generate distorted pixel positions ($\hat{u}$) from projector pixel positions (u). The distorted pixel positions ($\hat{u}$) are computed as follows.

$$d_r = u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{4.8}$$

$$d_t = \begin{bmatrix} 2p_1 u_x u_y + p_2(r^2 + 2u_x^2) \\ 2p_2 u_x u_y + p_1(r^2 + 2u_y^2) \end{bmatrix} \tag{4.9}$$

$$\hat{u} = d_r + d_t \tag{4.10}$$

Where radial distortion coefficients (k_1, k_2, k_3) form the radial component (d_r) and tangential distortion coefficients (p_1, p_2) form the tangential component (d_t) and r is the magnitude of u.

Luminosity of a projector was simulated by computing the apparent brightness for each projected pixel using a radiometric model. This model assumes that the projector is a point light source that emits light equally in all directions. The apparent brightness (B) was computed as follows.

$$B = \frac{L}{4\pi d^2} \tag{4.11}$$

Where L is the luminosity of the projector and d is the distance to the fragment.

Occlusion of projection surfaces was implemented using a two-pass rendering approach for the projectors that is similar to two-pass shadow mapping, but the difference being that both passes occur from the projector's point of view. On the first pass, the depth of the closest fragment is written to a texture. On the second pass, the texture is sampled using standard shadow map sampling. A fragment is rejected if the depth is larger (i.e. the fragment was occluded) than the value in the texture.

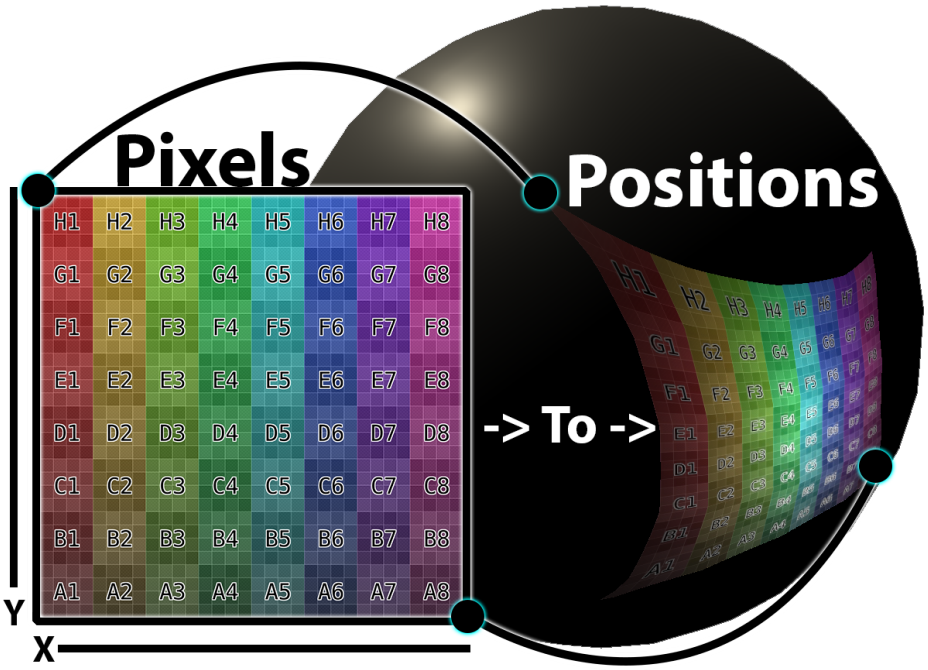


Figure 4.3: A mapping between 2D projector pixels to 3D surface positions on a non-planar surface (Sphere) using extrinsic, intrinsic, distortion, and luminosity equations is accomplished using programmable vertex and fragment shader programs.

To determine the final colour and position of projected pixels, the intrinsic parameters get applied in the vertex shader using the view (V) and projection (P) matrices and the distortion coefficients, luminosity, and occlusion rejection in the fragment shader.

The image source of the virtual projector can be specified as either a 2D texture for static image projection (e.g. Figure 4.3) or a `RenderTexture` – a texture coupled to a render source – for dynamic image projection. Stereoscopic output is also supported by syncing the projection render pass with the appropriate left/right stereoscopic output pass using a callback triggered by a rendering event. Virtualizing projectors in this manner meant that we could reuse the code that generates prewarped, perspective-corrected images for real projectors on a physical spherical display.

A step-by-step overview of projector prewarping is illustrated in Figure 4.4. In the first step, an image of the virtual content – as seen from the viewer's perspective – is stored in a `RenderTexture` by rendering from a camera placed at the tracked eyepoint. There would be a separate `RenderTexture` and camera for each viewpoint in the case of multi-viewer or left/right stereo rendering. In the second step, the prewarped image source is created by sampling the `RenderTexture` in a shader program using a projector-to-surface mapping that maps 2D projector image pixels to 3D display surface positions. These per-projector mappings are generated from a calibration phase during the configuration and setup of the display. For physical displays, the mappings are created using the automatic multi-projector calibration approach from Zhou et al. [118]. For virtual display, a much simpler approach was possible because the position and orientation of the projectors

and display are known a priori due to the fact that they are colocated in the virtual environment. Using this information, plus the projector intrinsics, we implemented a virtual calibration that was capable of outputting in the same format as the physical calibration (see Appendix A for details).

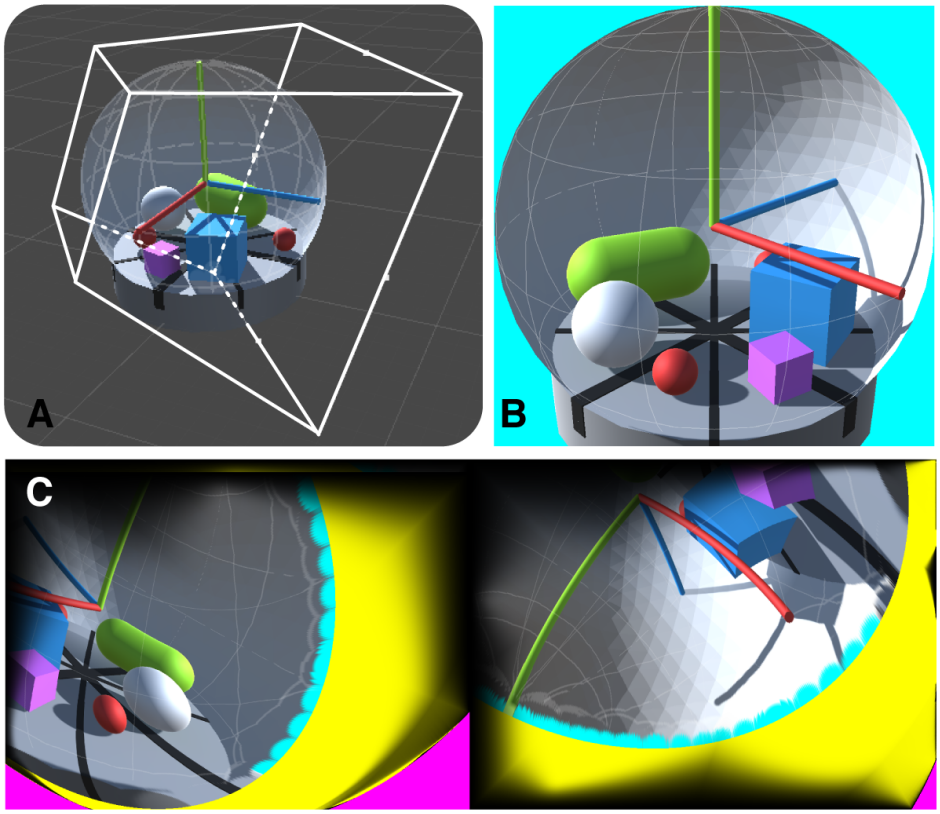


Figure 4.4: Overview of our rendering pipeline. We generate a view frustum for the user's viewpoint of the scene (A) and render to an off-screen texture (B). The mapping from projector-space to sphere-space is used to non-uniformly sample the rendered texture to generate the prewarped image for each projector (C, two projectors shown). For illustration, the prewarped image (C) is coloured to show different projector regions, including: magenta regions that are not visible on the spherical surface (because they do not pass through the bottom hole of the sphere), yellow regions that are visible on the spherical surface, but not from the user's current viewpoint, and black regions that are alpha blended for a seamless transition in the overlap between projectors.

Source: Fafard et al. [39]

To maximize sampling quality from the `RenderTexture` in the first stage, a custom camera frustum was used that tightly fit to a specified bounding region – the volume of the display. This ensured that pixel density remained high regardless of viewing angle or distance. An illustration of the frustum can be seen in Figure 4.4 (top left (A)). Depending on the particular setup of the system one of two frustums were used.

When rendering in non-stereo or using 3DoF tracking data, a symmetric frustum is constructed using the

assumption that the camera is facing the display with no roll. The field of view (f), far plane distance (a), and near plane distance (b) can be computed using the following equations.

$$f = 2\sin^{-1} r/d \tag{4.12}$$

$$a = d + r \tag{4.13}$$

$$b = d - r \tag{4.14}$$

Where r is radius of the display and d is the distance from the display. Using these parameters, the final projection matrix is generated using the `Matrix4x4.Perspective` function from the Unity API. We called this approach the **LookAt** frustum because it requires that the viewpoint be looking at the origin of the display (see Figure 4.5).

When rendering in stereo, special care must be taken when deciding on what kind of frustum to use. The differences in stereo image pairs affect the perceived depth of objects and depending on the approach may hinder the 3D effect. For an in-depth explanation of stereo image pairs and how to generate them correctly, we recommend these excellent resources from Bourke [20, 21]. If we were to use the above frustum for stereoscopic rendering, then we would be using the toe-in approach for stereo image pairs which introduces vertical parallax which will cause increased eye discomfort. An off-axis frustum that fit the spherical display was implemented so that no vertical parallax would be introduced. This offered the most comfortable viewing experience for stereoscopic rendering while still maintaining pixel density of the `RenderTexture`. Figure 4.6 shows the camera position (d_{VD}) and orientation (l) relative to the display all projected onto the horizontal plane; circle-circle intersection was used to compute tangential points on the display to compute tightly fitting left/right (l/r) frustum offsets. The same approach was used to find the top/bottom (t/b) plane offsets. The near plane offset (n) is the magnitude of the displacement vector between the camera and the display along the forward direction of the camera and far plane offset (f) is the near plane plus the diameter ($2r$) of the display. These offsets were used with the same off-axis projection matrix defined in Equation (4.1).

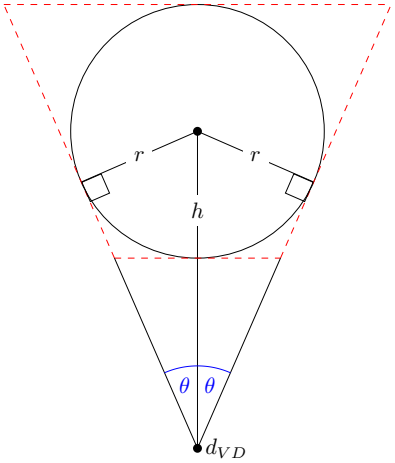


Figure 4.5: An illustration of the toe-in, on-axis frustum is outlined in red. The field of view of the projection matrix (2θ) is computed using one of the right-angled triangles made from the radius (r) of the display and the distance (h) from the display.

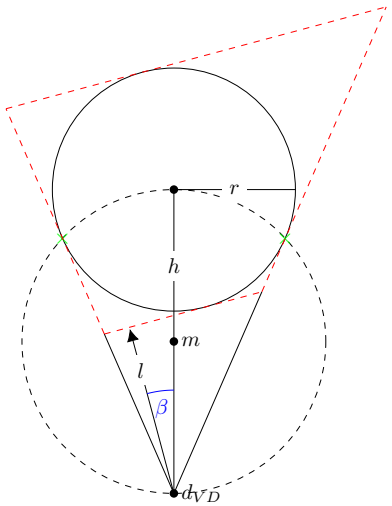


Figure 4.6: An illustration of the off-axis frustum is outlined in red. When a user looks away from the display at an angle (β), the near and far planes of the frustum remain perpendicular to the viewing direction (l). The midpoint circle (m) is used to compute the tangential intersection points (green marks) which are used to define the offsets for an off-axis projection matrix.

One problem that we encountered when using these display-constrained frustums is that the image on the display would become pixelated when the user was very close to the display. This was due to the fact that the frustum was always fitting the display even though the user could only see a small portion of the display. This problem was alleviated by clamping the horizontal and vertical field of views to reasonable maximums. Based on human retinal field of views 150 and 110 for horizontal and vertical respectively [85], we defined the clamping limits to be slightly larger than these values. They were chosen to be larger to account for untracked eye movements and to prevent the user from seeing portions of the display that get clipped (i.e. shouldn't be visible to the user) that may still be visible due to calibration error. We called this the **Off-axis** frustum since it uses an off-axis projection matrix.

We also included a default, or fallback, frustum for degenerate cases. If there was ever an error when computing one of the above frustums, a fixed 75° field of view frustum was used. We called this frustum the **Fixed** frustum. A visual comparison of frustums is demonstrated in Figure 4.7.

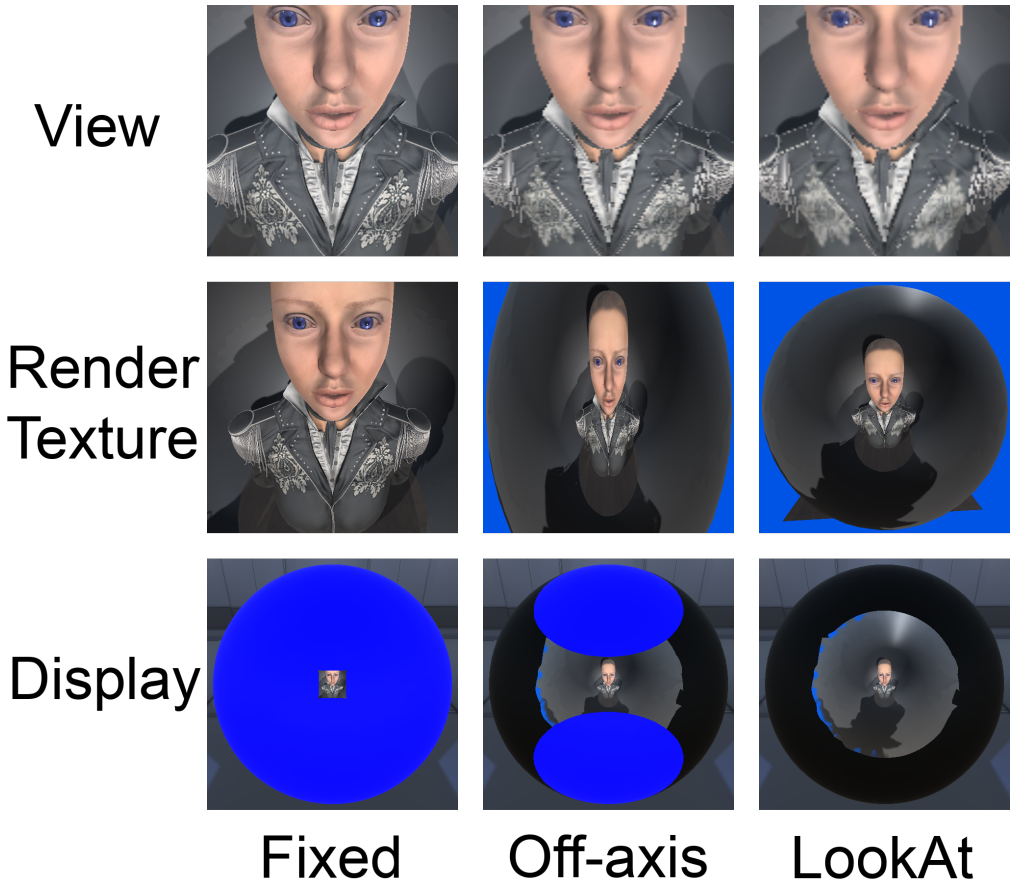
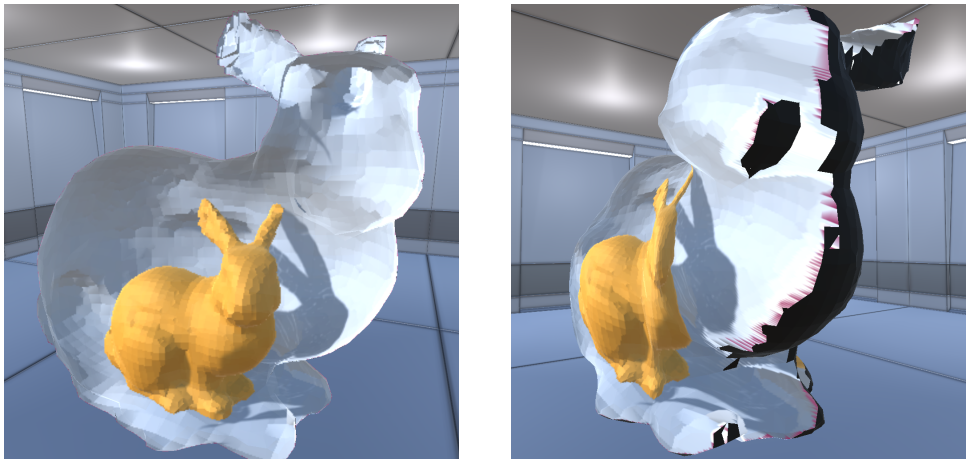


Figure 4.7: A comparison of three frustums when rendering a perspective-corrected image to a texture. This is an exaggerated case of when the viewpoint is extremely close to the display to illustrate the difference between display coverage and resolution of the render texture. Blue and black areas represent clipped regions on the display surface and ideally would not be visible to the user. The **View** row shows what the image would look like to the viewer. The **RenderTexture** row shows the stored image. The **Display** row shows the reprojected image on the display surface from a different perspective. **Fixed** offers the highest resolution by using a fixed 75° field of view, **LookAt** offers the most coverage at the cost of the lowest resolution by fitting the frustum around the entire display, and **Off-axis** offers a balance of resolution and coverage by clamping the horizontal and vertical field of views to 150° and 110° respectively.

The two-pass rendering approach that we used is simple to implement and offers the flexibility to render any practical shape of volumetric FTVR display. Common FTVR display shapes (e.g. boxes, spheres, and cylinders) are easy to implement using quads and screens for flat displays or high polygon objects and projectors for curved displays. If sophisticated enough calibration methods are used, then any realistic shape could be used as a display, like the bunny-in-bunny display in Figure 4.8.



(a) When looking from the correct perspective, a bunny can be seen inside a bunny-shaped display.

(b) When looking from an incorrect perspective, you can see that the perspective-corrected image is pasted onto the surface of a bunny-shaped display.

Figure 4.8: An example of a more complex, volumetric FTVR display in the shape of a bunny is rendered using the same shaders as any other projected FTVR shape. This virtual bunny display is composed of 2,710 faces and renders in realtime (120+ fps).

4.4 Near-surface Clipping

The projection frustum defines six distinct planes (top, bottom, left, right, near, and far) that are used to clip the geometry in the scene so that only the objects that are within the frustum are processed and rendered. Standard depth testing will process fragments (potential pixels) and choose only the closest one to the camera to render. The surface of an FTVR display may not match the near-plane in shape or position. For example, when using curved display surfaces and having virtual content that is not contained within the display, then the content may be rendered with front-depth — due to the space between the surface and near-plane — and be clipped at an unnatural angle; this example is illustrated in Figure 4.9. Front-depth is not necessarily a bad thing though, often, it is intended to provide an effect that the object is in front of (or coming out of) the display. However, special care must be taken when rendering with front-depth to ensure that edges or seams of the display do not interfere with the placement of the virtual content otherwise the effect will be broken by *frame cancellation* — "near-edge cut-off for objects with front depth" [81] (see Figure 4.9 B for an example). With a headtracked non-planar display and unlimited viewing angles, it is impossible to place virtual content with intended front-depth without causing *frame-cancellation* from some viewing angles. Harish and Narayanan's multi-facet rendering approach [49] could perform per-facet culling of geometry, however, using this approach with a perfectly curved display surface could introduce noticeable

clipping discontinuities depending on the level of polygonization of the display surface.

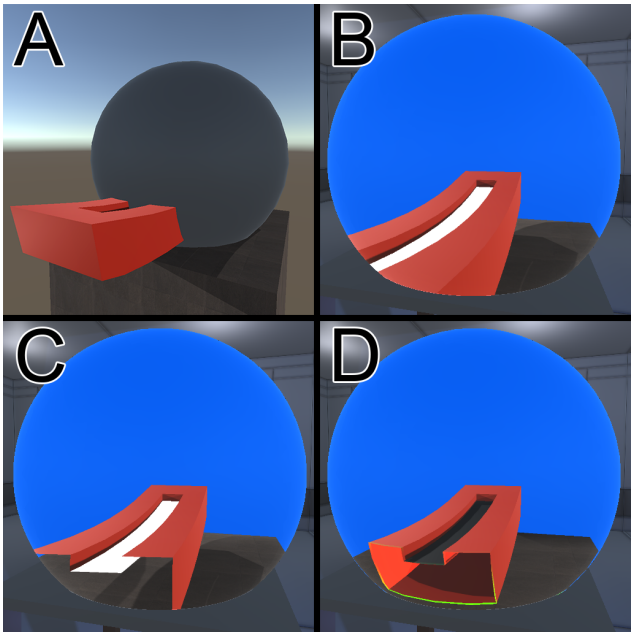


Figure 4.9: An illustration of a virtual object that exceeds the display volume and extends outside of the display surface (A). Renderings from the user's perspective are shown with no clipping (B), standard near-plane clipping with a display-constrained frustum (C), and near-surface clipping (D). The green highlights in (D) show the intersection of the object with the display surface. The different lighting in (D) is a result of the shader replacement procedure. The left edge of the display in (B) disrupts the 3D effect and causes *frame cancellation*.

To expand the range of possible content for non-planar FTVR displays, we implemented a near-surface clipping approach that would clip the virtual content right at the display surface, turn on backface rendering, and highlight intersections to provide a more realistic crystal ball effect (see Figure 4.9 D). For per-fragment surface-clipping, we needed the eye camera(s) depth of the display surface to perform an additional near-depth rejection test. We added a render pass using the eye camera(s) on the `OnPreEyePass` event that rendered the depth of the display surface — represented as a predefined mesh — to a depth only `RenderTexture`. To add the rejection test to all rendered geometry, we used a technique referred to as shader replacement in Unity documentation [96]. This technique replaces the shader programs of objects in the scene with a user specified one.

While shader replacement technically worked for our purposes, it came with several disadvantages with respect to content design. It was not possible to write a single shader that could replace all types of shaders. For example, objects with custom shaders (common on the Unity Asset Store) could not be replaced accurately since they deviate from the standard built-in shader. It was also not possible to clip particle effects at the fragment level because they are typically billboarded objects that are rendered on a special pass. Thus,

content with near-surface clipping had to be designed without particles and custom shaders to properly render as intended.

4.5 Virtual FTVR Model

One advantage of a virtual FTVR setup is that one can precisely and deliberately control the visual calibration. This control could be used to provide a perfectly calibrated FTVR display experience or to inject specific visual errors into the display to evaluate how noticeable or problematic they are for user perception or performance. Our VR testbed modeled each of the parameters from interactive visual calibration (see Figure 3.1 from Chapter 3). Each parameter could take on one of three values depending on the particular use case as follows:

- *GroundTruth* – The true, exact value of the parameter. Used to present a perfect calibration or as a ground truth value for comparisons to estimated, computed, or recorded measurements.
- *ErrorParameter* – A user-defined specific error to add to the exact value of the parameter. Used to inject synthetic or specific error into the system. Usually used to replicate an observed visual distortion from a physical FTVR display.
- *Approximation* – An approximated value for the parameter. Used when testing or validating calibration tools. The outputs of the calibration tools would get stored here and compared to *GroundTruth*

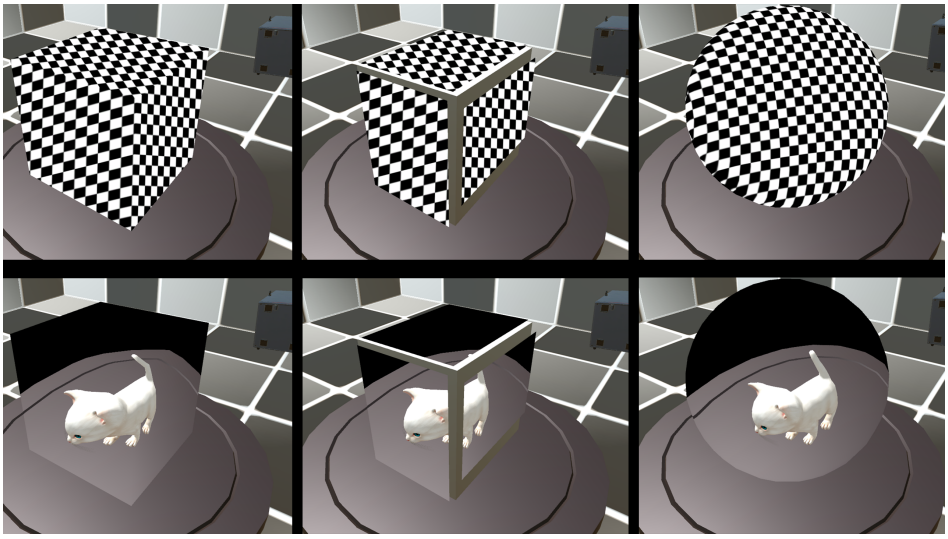


Figure 4.10: An illustration of *ErrorParameter* being used to simulate visual distortion of a 2D pattern (top row) and a 3D object (bottom row) on a cubic display (left), a box display with seams (middle) and a spherical display (right). The error added to the parameters is exaggerated for illustration purposes.

In addition to these three values, each parameter had an error measurement associated with it that represented the difference between *Approximation* and *GroundTruth*. The way in which this error was calculated depended on the type of the parameter; for positions, it was the norm of the displacement between the vectors, and for rotations, it was the minimum geodesic distance between them as described by equation ϕ_3 from Huynh [54].

We used these model parameters to investigate *shear distortion* — "perspective distortion with viewpoint changing" [81] — and classify the emergent visual distortions specific to display shape. We found with planar screens, visual distortion was most noticeable around the seams of the display (see left two columns in Figure 4.10) and with curved surfaces, the same error resulted in perceptually less visual distortion and was most noticeable along the edges of the display (see right column in Figure 4.10). We used these observations to design better perceptual calibration tools, such as the realtime refinement of calibrations as discussed in Chapter 3.

5 Virtual FTVR User Study of Viewing Condition

Video portrayals of spherical fish tank virtual reality (FTVR) displays convincingly depict a magical "crystal ball" experience that can show virtual 3D objects and scenes inside. However, unlike the original single-screen FTVR, these systems often omit stereo and instead rely primarily on motion parallax to provide the 3D effect as the viewer moves around the spherical display. Since motion parallax is a monocular depth cue, the FTVR display need only render a single perspective for the user as they move around. Rendering with stereo (a perspective for each eye) requires special optics to keep the left eye from seeing the perspective image intended for the right eye and vice versa. Implementing stereo also doubles the rendering cost, which may impact the overall frame rate. The lack of stereo does not affect video footage of these displays, since the view of a video camera is naturally monoscopic, but it is not clear how much the omission of stereo degrades the in-person experience. The evaluation of 3D perception and interaction performance among different viewing conditions is critical to guide future designs of spherical and similarly volumetric FTVR displays.

FTVR displays can be constructed from inexpensive commodity components, while maintaining high visual fidelity (bright, high resolution, etc.), making the technology practical and poised for widespread adoption. Unlike room or headset VR, FTVR creates a 3D illusion that is situated within the real world. This allows a user to easily transition between real-world information, traditional 2D displays, and virtual information on the FTVR display. Mobile phone and headset augmented reality (AR) can also create the illusion of virtual information on a tabletop, but because virtual imagery is overlaid on one's view, the experience is not physically tangible, whereas FTVR displays can be touched and held [88, 12]. The original single-screen FTVR user studies showed that although the combination of head-tracking and stereo resulted in the best user performance, head-tracking alone outperformed stereo alone for a range of 3D tasks, such as path tracing and shape assessment [106, 1]. The result that parallax was a more effective 3D cue than stereo was surprising, given that "3D" was (and still is) often considered synonymous with stereo viewing. This early result in FTVR research has led many follow-on displays to de-emphasize the need for binocular depth cues and omit stereo from their designs. Dropping stereo makes FTVR displays cheaper and easier to build: stereo glasses or auto-stereoscopic lens overlays are not needed and inexpensive 60$\dot{\text{H}}$z screens or projectors can be used. However, the design trade-off between ease of construction and perceptual fidelity has not been evaluated.

In this study, we challenge the assumption of recent spherical FTVR systems: that stereo is less important now that users can walk around the display and benefit from increased motion parallax. We simulate a

spherical FTVR display within a VR environment, which allows us to carefully measure 3D performance differences among viewing conditions, while controlling for all other factors, including calibration errors, head-gear, and latency. Our simulated FTVR platform allows us to evaluate any display shape and form factor. We focus on the spherical form factor in this study because it has been most widely adopted for volumetric FTVR displays.

This study makes the following four research contributions: 1) We evaluated 3D performance on distance estimation and 3D selection/manipulation with a spherical FTVR display for stereo vs. non-stereo viewing conditions. Our results show that stereo provided significantly faster and more accurate performance across a range of 3D tasks. 2) We evaluated noticeability and user preference of different viewing conditions for a spherical FTVR display and found no strong user preference for the stereo-viewing condition. 3) We evaluated a visual pattern-alignment scheme for viewpoint calibration with a spherical FTVR display under different viewing conditions and found that stereo-viewing resulted in the most accurate viewpoint alignment and that binocular-viewing was aligned, on average, to the mid-point between the two eyes rather than to the dominant eye. Together, our results show that, contrary to the prevailing design of spherical FTVR displays, stereo should not be neglected if 3D perception and task performance are a priority.

5.1 Experiment Design

We used our simulated FTVR display to evaluate the effect of viewing conditions on a range of 3D tasks, including: visual pattern alignment [104], forced-choice viewing preference [1], and a series of point cloud visualization tasks previously used for AR evaluation [5]. To reduce the likelihood of accidental user input (e.g. a double-click), buttons were disabled immediately following a click and during the transition between conditions.

5.1.1 Viewing Conditions

The primary independent variable used in all experiments is *Viewing Condition* with three levels (as illustrated in Figure 5.1). In *Stereo*, the FTVR display and headset render distinct images for each eye. In *NonStereo*, the FTVR display renders one image the midpoint of the eyes and the headset renders to both eyes. In *Monocular*, the FTVR display and headset render one image to one eye.

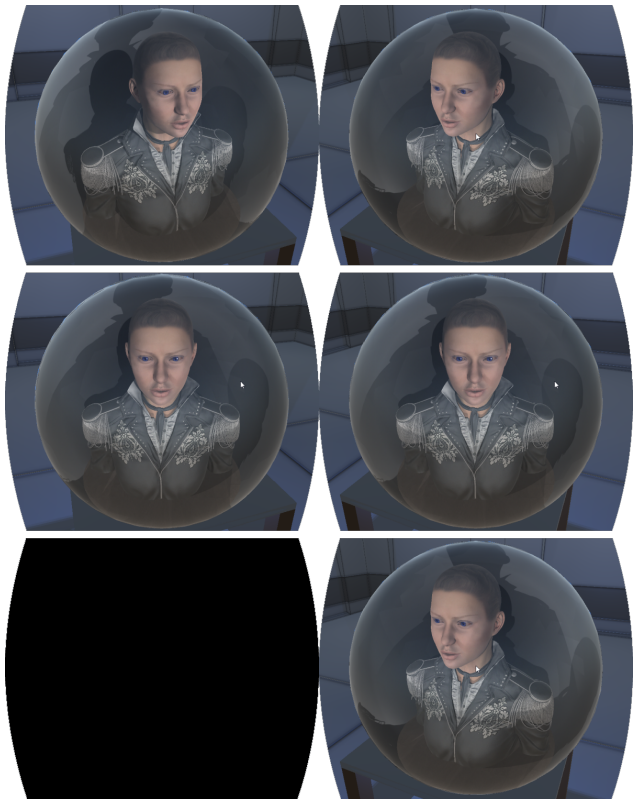


Figure 5.1: An example, with exaggerated stereo disparity, of what the left and right eyes would see in the *Stereo* (top), *NonStereo* (middle), and *Monocular* (bottom) viewing conditions. Note that non-stereo rendering creates a perspective mismatch between the background 3D world and the display.

Source: Fafard et al. [37]

5.1.2 Participants and Procedure

We used a within-participant design so that all participants performed all three experiments. Experiments were analyzed separately, so the order of experiments was the same for all participants, whereas the viewing conditions and choice order were counter-balanced within each experiment.

Twenty four participants were recruited from a local university. Before starting the experiments, they responded to a questionnaire, performed an eye dominance test, chose which hand and corresponding controller they would use, and underwent a stereo acuity test [44] in VR. A Virtual Reality Sickness Questionnaire [59] (VRSQ) and general task questionnaire were interleaved between experiments to give participants a break from VR and to monitor any ill effects. Three participants were excluded based on their results from the stereo acuity test and VRSQ responses, leaving a total of 21. Of these participants, 13 had used VR before, 19 were right eye dominant, 21 used the right-handed controller, and 20 used VR less than once per week.

5.1.3 Data Analysis

We investigated outliers — measurements with high variability within or across participants — to determine if any should be excluded from our analysis, but we found no measurements that could be reliably explained by system malfunctions or measurement errors. Significance values are reported in brackets for $p < .05(*)$, $p < .01(**)$, and $p < .001(***)$ respectively. Numbers in brackets indicate mean ($\bar{x}$), median ($\widetilde{x}$), and standard deviation (σ) of their respective measurement. P-values were adjusted using the expected proportion of false discoveries amongst the rejected hypotheses [8, 9].

5.2 Experiment 1: Viewpoint Pattern Alignment

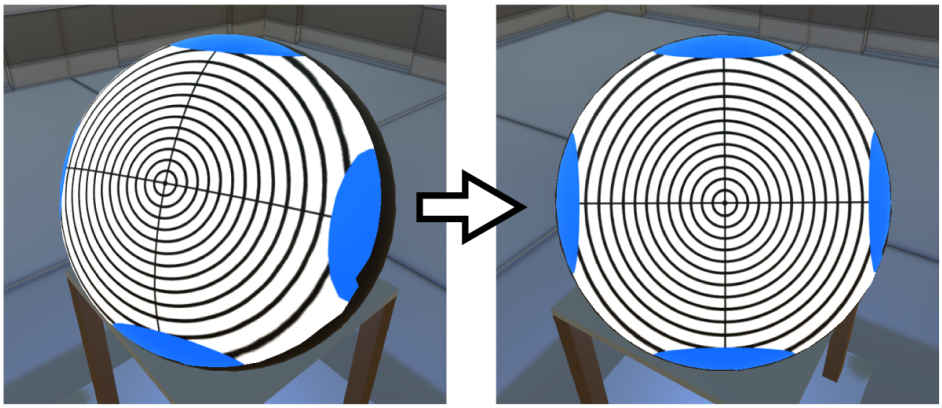


Figure 5.2: Pattern alignment task: the pattern starts distorted (left) and then the participant moves their head left, right, up, or down to align their viewpoint so that the pattern appears to have straight lines and circular rings (right).

Source: Fafard et al. [37]

FTVR requires rendering to the user's viewpoint in real-time to provide the correct perspective as they move around the display. Accurately calibrating the viewpoint to the display is important, otherwise the 3D scene appears distorted. Recently, a visual calibration method was proposed where a user aligns a 2D pattern on the display by moving their head to minimize the visual distortion in the pattern [104]. However, this method was evaluated for monoscopic viewing on a cubic display. It is not clear how well it works for binocular viewing conditions, where the viewpoint should be defined in binocular non-stereo, and how accurate viewpoint calibration needs to be in order to render convincing 3D scenes. We recreated the pattern-alignment task within our simulated FTVR environment, which allows us to measure viewpoint error across viewing conditions relative to the ground-truth eye locations provided by the VR headset.

Participants performed three trials with two dependent measures, *Time* and *Error*, for each *ViewingCondition*. The order of conditions in this experiment was randomized for each participant.

Our main hypothesis regarding viewpoint alignment were that:

- `H1-1` *Monocular* level will be faster to align because, with only one eye receiving images, there is less information to process;
- `H1-2` *NonStereo* will have more variability because the pattern will never look perfect from both eyes;
- `H1-3` *NonStereo* will have a mean measured viewpoint that is near the geometric mean of the eyes. This hypothesis is motivated by assumptions made in previous work. The midpoint of the eyes (rather than a single eye) has been assumed to be the most appropriate viewpoint for binocular *NonStereo* viewing with FTVR [82, 90, 1, 88].

`H1-1` and `H1-2` followed from Wagemakers et al. [104] and observations made in our lab. `H1-3` is based on a common viewpoint model adopted by previous research [117, 104, 89, 88, 98, 90, 38].

5.2.1 Analysis & Results

The resulting times and errors for pattern alignment are shown in Figure 5.4.

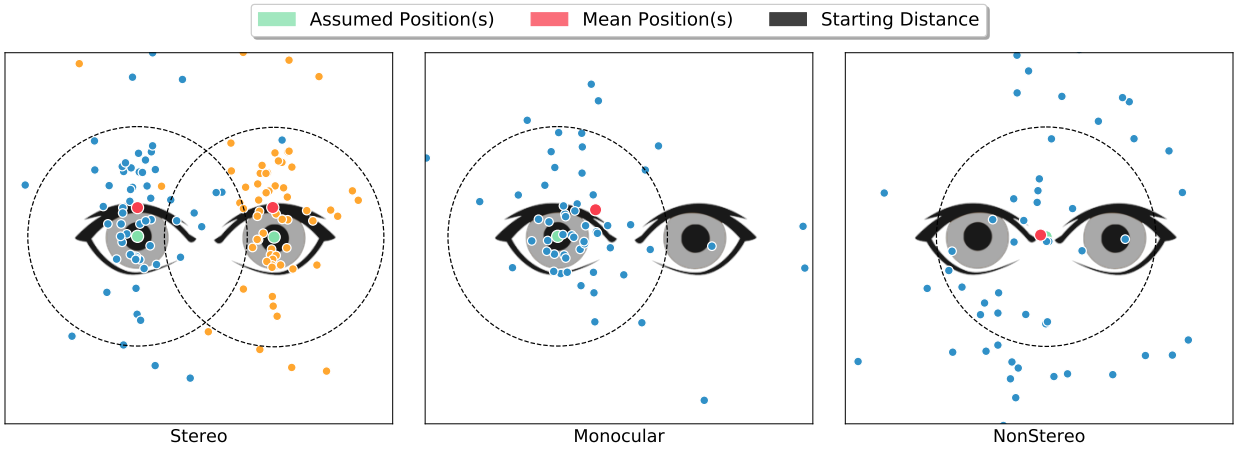


Figure 5.3: The geometric mean (red) of measurements (blue/orange) is shown relative to the ground truth (green). Calibrations were perturbed by 5 cm (black circles) at the start of each trial. Plots are scaled to 6.3 cm pupillary distance.

Source: Fafard et al. [37]

An `RM-ANOVA` was performed followed by `Tukey's` pairwise significance test. There was a significant $(***)$ difference in means across levels for *Error* ($F(2, 40) = 13.8,\ p < .001$), but not *Time* ($F(2, 40) = 0.71,\ p = .497$). The mean *Error* for *Stereo* ($\bar{x} = 3.3cm,\ \sigma = 2.2cm$) and *Monocular* ($\bar{x} = 4.1cm,\ \sigma = 3.4cm$) was lower $(***)$ than *NonStereo* ($\bar{x} = 6.6cm,\ \sigma = 3.1cm$). Therefore, we reject `H1-1` (*Monocular* is faster) and accept `H1-2` (*NonStereo* is more variable). A `One-sample Wilcoxon signed-rank` test was performed

on the lateral (from the left eye towards the right eye) displacement of measurements (normalized to a 65 mm pupillary distance) to the midpoint of the eyes under the *NonStereo* condition. The displacement was significantly $(***)$ closer to the midpoint than the left eye ($V(\mu < -16.25\ mm) = 1687,\ p < .001$) and right eye ($V(\mu > 16.25\ mm) = 369,\ p < .001$) with a 99% confidence interval of (−7.14 mm to 6.42 mm). The geometric mean for *NonStereo* is closer to the center of the eyes than either eye (see Figure 5.3), therefore we accept `H1-3` (*NonStereo* is aligned to mid-point of the eyes).

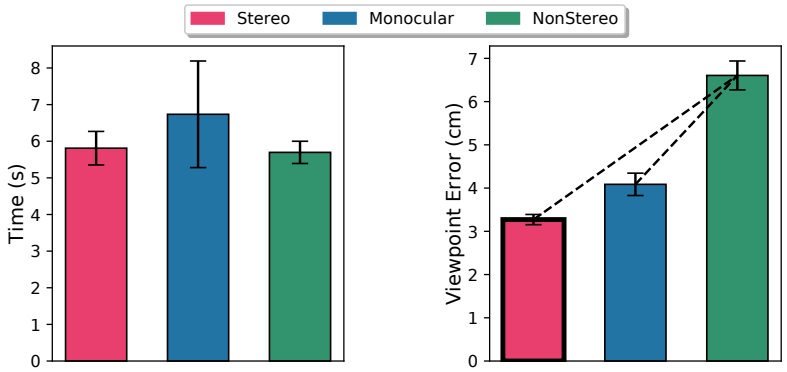


Figure 5.4: Mean *Time* and *Error* vs. *Viewing Condition* for the Pattern Alignment task. Error bars represent the standard error of the mean, highlighted bars indicate significant best results, and dashed lines indicate a significant difference.

In a follow-up questionnaire, 86% of participants agreed that the distortion in the pattern helped them align the image and most participants stated that their strategy to align the pattern was to move their upper body left/right or up/down until the lines were straight.

5.3 Experiment 2: Subjective Preference

Figure 5.5: Subjective preference task: the participant was forced to move left and right to induce a minimum amount of head motion before selecting their preference between a pair of viewing conditions.

Source: Fafard et al. [37]

Some use cases for FTVR displays do not involve an explicit 3D task, but rather attempt to convey a compelling 3D impression or experience. For example, an FTVR display could be used to exhibit high fidelity 3D scans of an artifact, which would bypass the need to travel to or exhibit the original artifact. Because of this use case, we were interested in assessing the general spatial impression and subjective preference of different viewing conditions on a spherical display using methodologies from flat screen FTVR studies [1, 107, 108]. In addition to the levels already described, we added the following variants where the viewpoint was rendered to the dominant (D) or non-dominant (ND) eyes: *NonStereoDominant*, *NonStereoNonDominant*, and *MonocularNonDominant*. In this experiment, participants were instructed to pay close attention to how 3D the scene appeared and to notice any perceived movement of the scene coupled to head movements. They were instructed to choose the condition that appeared most 3D and perceived with the least head-coupled movement. We also enforced head movement for each pair by requiring participants to cross two virtual bars in order to proceed (see Figure 5.5). Once a participant had crossed the bars on the second condition, buttons appeared that allowed them to record their preference. Early pilots of this experiment showed that repeated toggling between viewing conditions was disorienting; to minimize this disorientation, participants

	Monocular	*NonStereo*	*Stereo*	**All**
Monocular	-	50%	32.1%	41.0%
NonStereo	50%	-	52.4%	51.2%
Stereo	67.9%	47.6%	-	57.7%

Table 5.1: View condition preferences: the row label was preferred $X\%$ of the time over the column label, e.g., *Stereo* was preferred 67.9% of the time over *Monocular*.

Source: Fafard et al. [37]

were given one pair (AB) of conditions to inspect at a time and switched viewing conditions only once per pair. There were 15 pairs of conditions (AB) and 30 when the inter-pair order were reversed (AB to BA). To counterbalance inter-pair ordering, we split participants into AB and BA groups. The order of pairs was randomized and inter-pair order followed the AB or BA sequence with respect to the participant's group.

Our main hypothesis regarding user preferences were that:

- `H2-1` *Stereo* will be preferred over all other levels. This condition provides the most depth cues, thus the best visual fidelity, so it should be the most preferred;
- `H2-2` Binocular levels (*Stereo*, *NonStereo*) will be preferred over *Monocular* because closing or blocking one eye will be uncomfortable. There are not many examples of monocular viewing in FTVR research, and we found it uncomfortable during preliminary testing.

5.3.1 Analysis & Results

A recording error resulted in the loss of seven of the participant's preferences; however, we maintained group counterbalancing with the remaining participants. There were inconsistent judgments within and across participant responses for the variants related to eye dominance; therefore, these variants were summed with their corresponding parent level, as reported in Table 5.1. Data did not meet the minimum number of agreeing judgments necessary to establish significance using a two-tailed `Paired preference` test [83]. Therefore, we reject `H2-1` and `H2-2`.

In a follow-up questionnaire, 90% of participants agreed that, in at least one condition, the statue looked 3D and 86% agreed that the statue seemed to remain physically fixed (i.e. no evidence of floating). Between conditions, 71% of participants agreed or strongly agreed that the change in 3D appearance was noticeable and 62% agreed or strongly agreed that there was noticeable floating.

5.4 Experiment 3: Point Cloud Performance

To assess 3D perception and interaction performance, we followed a recently proposed set of tasks for exploration of 3D visualizations in AR [5]. The tasks featured structured point clouds because they are commonly

used to represent 3D data (e.g. data produced by 3D scanners). The authors described parameters to generate the point clouds so that the overall density, number of clusters, and size of the points were appropriate for each of the tasks. We adopted these same parameters, but adjusted the size of the points because we used a much larger display. We chose three tasks relevant in a FTVR setting: perceptual distance estimation, target selection with occlusion, and 3D object manipulation. The tasks differed mainly in the amount of interaction with the virtual content, from a minimum in the *Distance* task to a maximum in the *Manipulation* task. In the *Distance* task, participants were asked to judge which pair of points (red or yellow) had the smallest distance between them; in the *Selection* task, they were asked to select four target points that were highlighted red; and in the *Manipulation* task, they were asked to align a semi-transparent cutting plane to intersect three coplanar clusters of red points (which turn blue when intersected). Participants were instructed to prioritize speed over accuracy for all tasks. They performed 3 training trials followed by 10 recorded trials for each viewing condition and task. The order of tasks remained fixed so that participants could practice and build up their expertise with the interaction tool. Participants were told that they could move around the front 180° of the display as much as they would like, especially if they found that their view was occluded by irrelevant points. We included a reflective surface on the virtual display to ensure that there was a clear separation between the VR and virtual FTVR environments.

Participants performed 10 trials with two dependent measures, *Time* and *Error*, for each *ViewingCondition*. Within each task, the point clouds were generated from the same parameters using a pseudo-random generator to distribute the points. Primarily due to the occlusion of important points, some point clouds were more difficult than other. To control for the variation in measurements due to relative difficulty, *PointCloud* was used as a blocking variable. The order of tasks was the same for each participant, but the viewing condition order in each task was counterbalanced by splitting participants into three groups (ABC, BCA, CAB).

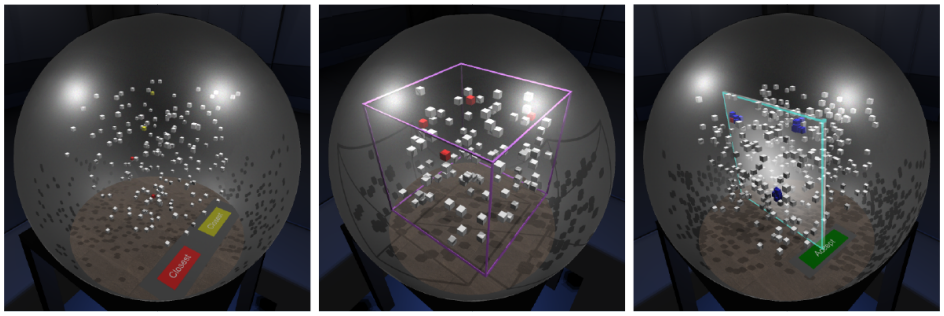


Figure 5.6: *Distance* task (left): judge which pair of points (red or yellow) had the smallest distance between them. *Selection* task (middle): select all four red cubes. *Manipulation* task (right): move and rotate the semi-transparent cutting plane so that it intersects the three coplanar clusters of red points (which turn blue when intersected).

Source: Fafard et al. [37]

Our main hypothesis regarding user performance were that:

- `H3-1a` *Stereo* will have the lowest completion time because of its additional depth cues;
- `H3-1b` *Stereo* will have the lowest error because of its additional depth cues;
- `H3-2` *Monocular* will have lower mean error than *NonStereo* because a single accurate view into the scene is better than two inaccurate views;
- `H3-3a` More head movement will be observed in non-stereo levels (*Monocular*, *NonStereo*) because motion parallax is needed for depth cues;
- `H3-3b` More head movement will result in better task performance in non-stereo levels (*Monocular*, *NonStereo*) because of the additional motion parallax.

These hypotheses were based on the combination of previous research in flat FTVR [106, 107, 108], more recent research with volumetric FTVR [98, 104, 117, 38], and observations made using the simulation system and various display shapes.

5.4.1 Analysis & Results

Data did not meet the normality and homoscedastic assumptions for using `ANOVA`. A `Friedman ranked sum` test was performed followed by an `Eisinga, Heskes, Pelzer & Te Grotenhuis all-pairs` test [36] for pairwise significance testing. The distributions of values for each group had a similar shape and spread for both *Time* and *Error* for all tasks. Mean completion times and task errors are shown in Figure 5.7.

Distance: There was a significant $(***)$ difference in median values across groups for *Time* ($\chi^2(2) = 16.8,\ p < .001$) and a borderline significant difference $(*)$ in *Error* ($\chi^2(2) = 6,\ p = .05$). The median *Time*

for *Stereo* ($\bar{x} = 3.7s,\ \sigma = 0.5s$) was lower ($***$) than *NonStereo* ($\bar{x} = 5.2s,\ \sigma = 1.0s$) and lower ($*$) than *Monocular* ($\bar{x} = 4.7s,\ \sigma = 0.6s$).

Selection: There was a significant ($***$) difference in median values across groups for *Time* ($\chi^2(2) = 18.2,\ p < .001$) and ($*$) *Error* ($\chi^2(2) = 8.6,\ p < .05$). The median *Time* for *Stereo* ($\bar{x} = 6.0s,\ \sigma = 0.8s$) was lower ($***$) than *NonStereo* ($\bar{x} = 11.5s,\ \sigma = 1.4s$) and lower ($*$) than *Monocular* ($\bar{x} = 9.7s,\ \sigma = 1.7s$). The median *Error* for *Stereo* ($\bar{x} = 1.8,\ \sigma = 0.8$) was lower ($*$) than *Monocular* ($\bar{x} = 3.8,\ \sigma = 1.3$).

Manipulation: There was a significant ($***$) difference in median values across groups for *Time* ($\chi^2(2) = 14.6,\ p < .001$), but not *Error* ($\chi^2(2) = 5.6,\ p = .061$). The median *Time* for *Stereo* ($\bar{x} = 11.7s,\ \sigma = 4.0s$) was lower ($***$) than *NonStereo* ($\bar{x} = 15.3s,\ \sigma = 4.7s$) and lower ($*$) than *Monocular* ($\bar{x} = 14.4s,\ \sigma = 4.0s$).

Given the results from all tasks, we accept `H3-1a` (*Stereo* is fastest), partly accept `H3-1b` (*Stereo* is most accurate), and reject `H3-2` (*Monocular* is more accurate than *NonStereo*).

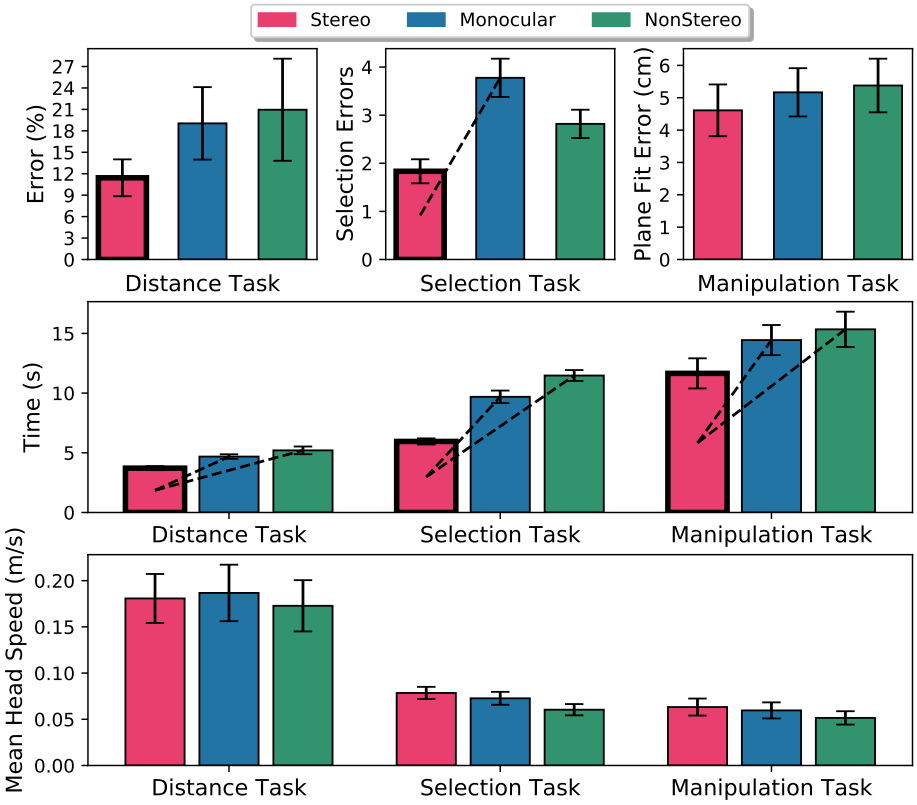


Figure 5.7: Mean *Time*, *Error* and *Head Speed* vs. *Viewing Condition* grouped by *Task*. Error bars represent the standard error of the mean, highlighted bars indicate significant best results, and dashed lines indicate a significant difference.

Source: Fafard et al. [37]

Head Movement Magnitude: A two-way `ANOVA` was performed to find any main effects or interactions between *Viewing Condition* and *Task* using *Head Speed*. There was a significant ($***$) main effect for *Task*

($F(2, 180) = 44.3,\ p < .001$), but not *Viewing Condition* ($F(2, 180) = 0.48,\ p = .62$). There were no significant interactions between *Task* and *Viewing Condition*. Therefore, we reject `H3-3a` (*NonStereo* has more head movement).

Head Movement and Performance: Mean speed (m/s) of the VR headset was used to quantify head movement per *Viewing Condition*. An analysis of covariance (`ANCOVA`) using type II sum of errors was performed to examine interactions between *Head Movement* and *Time/Error* within the *Viewing Conditions*. A significant ($**$) difference was noted in the intercepts among levels for *Time* ($F(2, 59) = 7.7,\ p < .01$) in the *Selection* task and the maximal R-squared value across all tasks and measures was $r^2 = 0.3482$. Therefore, we reject `H3-3b` (Head movement improves accuracy).

5.5 Discussion

Overall, our evaluation of viewing conditions for spherical FTVR suggests that users did not have a strong subjective preference for stereo viewing, but it did improve either their speed or accuracy for all tasks performed.

5.5.1 Virtual Reality Sickness Questionnaire

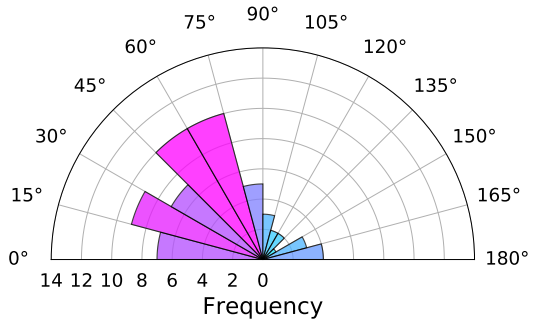
The majority of participants (14+) noted no effects on the VRSQ through all experiments, except for the last: roughly half of the participants noted slight or moderate effects (*Eyestrain*, *Fatigue*, or *General Discomfort*). Three participants noted severe effects (*Blurred Vision* or *Difficulty Focusing*) in the first and second experiments, however, these effects were reported as less than severe by the last experiment.

5.5.2 Effectiveness of Viewpoint Calibration with Spherical Displays

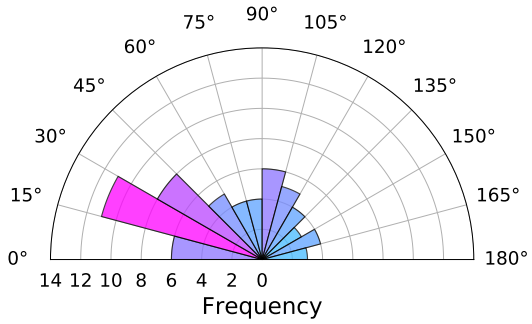
Previous work [104] suggested that the visual distortions of the pattern, when viewed from the wrong perspective, were more noticeable for a cubic display than for a curved display surface. With a cubic display, pattern distortions appear as sharp kinks or discontinuities across the screens of different faces of the display; with a curved display surface, the distortions manifest as curved distortion of straight lines. They performed a desktop study using a mouse and standard 3D projection rendering without stereo or motion parallax. We improved upon this by faithfully recreating the visual alignment cues on a spherical display, while also having ground-truth information about the viewpoint(s). This way, we quantitatively evaluated how accurately participants were able to perform the visual alignment task. We used their most effective visual pattern (bullseye with horizontal and vertical lines) and found that alignment error on a spherical display was consistent with their results: *Stereo* ($\bar{x} = 1.3°$), *Monocular* ($\bar{x} = 1.7°$), and *NonStereo* ($\bar{x} = 2.8°$) versus Desktop 2D ($\bar{x} = 1.1°$).

We also informally investigated an important assumption that viewpoint calibration makes: the perceptual distortion created when viewing a 2D pattern from an incorrect perspective will consistently guide the user

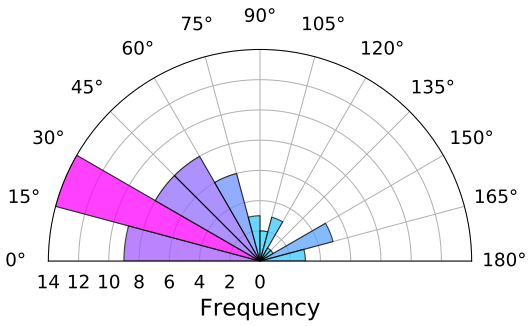
towards the correct perspective. Using the data from Experiment #1 (pattern alignment), we measured the angle between the true direction of correction and the direction the participant actually went. We split the measurements by level (*NonStereo*, *Monocular*, and *Stereo*) and found that, for all levels, the pattern guided the participants towards the correct perspective far more often than away from it (see Figure 5.8). We also found that *Stereo* and *Monocular* offered much more consistent guidance than *NonStereo*.



(a) *NonStereo*



(b) *Monocular*



(c) *Stereo*

Figure 5.8: These polar histograms illustrate the angular deviation of measurements by *Viewing-Condition*. Angular deviation is measured as the angle between the guiding direction of the perceptual pattern and the measured direction of correction.

5.5.3 Subjective Perception of *Viewing Conditions*

Data from Experiment 2 show that two-thirds of users preferred *Stereo* over *Monocular*, but the remaining preferences were not in agreement. These findings are different from a similar subjective preference task [106], however, this is not surprising given the differences in how the 3D content was presented. There was improvement in the quality of stereoscopic rendering and 3D real-time graphics, the accuracy and latency of head-tracking, and the calibration of the FTVR display. In addition to these technological differences, our

participants walked around a spherical display instead of being seated in front of a "flat" CRT monitor. We also changed the protocol to minimize eye strain and general discomfort of the user by limiting the number of times *Viewing Condition* was switched; this meant that we relied more on a user's first impression than a rigorous comparison. Overall, we found that it was difficult for inexperienced users to perceive the subtle differences in *Viewing Conditions* on a spherical FTVR display with a static scene. Although no strong preference was found for stereoscopic rendering, task performance measures were significantly improved with stereo.

5.5.4 Task Related Performance

We chose three of the tasks from [5] that balanced the benefit of stereo and motion/interaction: *Distance* could be accomplished from binocular depth cues or by finding a perpendicular vantage point to the paired-points, *Selection* required avoiding occlusion and potential ambiguity in the direction of the selection ray, and *Manipulation* required depth cues for the visual feedback needed to align the plane to points in 3D. On these tasks, our results showed a significant reduction in task time across all tasks and significant reduction in error across some tasks when using *Stereo*. This is consistent with previous studies that have shown faster performance (in surround VR studies [79, 63]) and more accurate performance (in single-screen FTVR studies [107, 60]) when stereo is included.

We observed that the benefit that *Stereo* offered over *NonStereo* varied a lot between point clouds: mean error across point cloud #4 in the *Distance* task decreased from 76% to 19% when switching from *NonStereo* to *Stereo*, but for point cloud #7, it went from 5% to 0%. The only task where *Stereo* did not improve accuracy was the *Manipulation* task, which had similar errors across *Viewing Conditions*. This result matches that of the original AR study using this task [5]. This may be due to the high difficulty of the task, particularly for novice VR users.

5.5.5 Head Movement

In the non-stereo conditions, the distance task in Experiment 3 should be nearly impossible without using the depth cues from motion parallax. We expected participants to adopt a strategy featuring more head movements under non-stereo *Viewing Conditions*, but, similar to previous research [52], motion parallax remained an underutilized depth cue. We informed the participants that it may be helpful to move their head to different viewpoints around the display, but we did not provide them explicit instructions on the potential benefits of motion parallax in non-stereo conditions. For non-stereo, spherical FTVR displays, it may be useful to give explicit training on head movement strategies for new users if task performance is important at all.

We also noticed that most participants were more comfortable standing upright and avoided crouching if at all possible. The variance of head position was roughly eight times larger along the ground plane than vertical ($\approx$ 8 cm vs. $\approx$ 0.9 cm, respectively). It is possible that this behaviour was responsible for the

unequal variance in pattern alignment (most notable in *Stereo*) visible in Figure 5.3: participants were more accurate side-to-side than up-and-down, which could mean that they would rather stand comfortably than get a more accurate calibration. This finding could be used to improve the pattern-alignment calibration technique by generating the calibration positions at a comfortable height for each user. Additionally, none of the participants walked into, or put their face inside of, the display at any time during the study which really shows how compelling VR can be.

5.5.6 Novelty of Simulated FTVR

We have observed that latency is more noticeable with stereoscopic FTVR rendering than monoscopic rendering and also results in pronounced floating of the virtual content. The Oculus Rift system provided extremely accurate and low latency viewpoint tracking using built in inertial measurement units (IMUs) and kinematically constrained prediction models [64]. Minimizing head-tracking error was crucial because it is a significant contributor to eye angular error of pixels on a spherical FTVR display [119]. In addition to low latency tracking, VR headsets apply pixel warping techniques to interpolate frames using the motion tracking data to provide even more accurate pixel positions. The Oculus API in Unity provided us the position and orientation of the user's eyes for every image our simulated FTVR display generated. We were able to use these reported viewpoints as ground truth for Experiment 1.

By using a VR headset, we were able to minimize confounding variables between *Viewing Conditions*. Participants always wore the same equipment and the visual fidelity of the VR headset was always the same. We also provided a perfectly calibrated FTVR experience. Since the tracking system and display system were colocated in the virtual environment, ground truth viewpoint and tracking-space calibrations could be computed, and pixel-perfect projection of perspective-corrected images on the display surface was possible. This ensured that any noticeable visual distortions (e.g. floating content) would be caused solely by a user's perception under the respective *Viewing Condition*.

5.5.7 FTVR Design Recommendations

Our study provides empirical evidence for a number of design recommendations for FTVR displays. The visual alignment error we measured in Experiment 1 gives a sense of the precision that is required for accurate viewpoint calibration. Presumably, the users were only able to align the pattern as precisely as they could notice the visual distortion in the pattern, so calibrating the viewpoints any more accurately than that could go unnoticed. It may be sufficient to provide viewpoint calibration techniques for spherical FTVR displays that are accurate to within ≈ 3 cm, depending on its stereo capabilities. We also found that, while the visual alignments in the *NonStereo* condition were quite variable, they were centered around the mid-point between the eyes. Therefore, for binocular non-stereo rendering on an FTVR display, it is best to render between the user's eyes rather than to their dominant eye.

In the *Selection* task, participants had significantly increased accuracy with *Stereo*, and during the training

in *NonStereo* or *Monocular* conditions, several participants reported that they had a very difficult time selecting the targets. Without stereo, there may be an ambiguity between the start and end points of the selection ray inside the display and, therefore, a simple visual indicator of the endpoints may be helpful in such use cases.

We did not find evidence that *Stereo* was preferred over *NonStereo* nor that *Stereo* was even particularly noticeable for our participants. Therefore, our recommendation is that *NonStereo* spherical FTVR displays would be reasonable for use cases that simply provide a 3D effect or impression to a participant, such as when used as an attention-drawing showcase display or a casual entertainment device. However, the results for 3D performance with a spherical FTVR display are conclusive: if stereo is omitted to make the system easier to build or glasses-free, then the in-person experience will be degraded and users will have trouble perceiving and interacting with 3D scenes within the display.

6 Conclusions

This thesis evaluates 3D calibration, perception, and performance of spherical FTVR displays within a virtual reality testbed. By recreating interactive visual calibration in our virtual system, we found that participants could notice viewpoint error when it was more than ≈ 3 cm without stereo, and a little less with stereo. Visual alignment was also more consistent with stereo, and the viewpoint without stereo was measured to be near the midpoint of the eyes. Empirical measurements of performance tasks showed that user performance was degraded when stereo was disabled. However, in a subjective perceptual task, users were unable to determine whether stereo was present. These findings can be used to guide the design and development of volumetric FTVR displays. For example, if task performance is important, then include stereo. Also, if stereo is not included, then render the perspective to the midpoint of the eyes. These guidelines could be used to improve viewpoint registration, not only for non-stereo capable displays, but also displays that split stereo rendering into two distinct monocular views (see CoGlobe [117]) for co-located, multi-user experiences.

6.1 Limitations and Future Work

The viewpoint model that we used for our extensions and new calibration method makes the assumption that the eyepoint(s) and tracked point are rigidly connected to each other. For viewpoints, this implies that the eyes never move and look straight ahead. It is known that users easily deviate from this assumption when they look around freely [31]. This introduces a small amount of error to the position and orientation of the eyepoint(s). The results from our pattern alignment experiment show that this would be a noticeable amount of error for some users, however, it seems unreasonable to simply ask them to not move their eyes. If additional accuracy is required, it would be possible to extend these calibration methods with a non-rigid viewpoint model by incorporating gaze tracking.

For handheld tool calibration, this viewpoint model works well because the tools that we used were in fact rigid bodies with no moving parts. However, other limitations to this type of calibration still exist. The familiarity that a user has with the handheld tool would have a direct impact on calibration accuracy. The realtime calibration relies on aligning both the rotation and position of the tool. It follows that the better a user knows the tool, the better they would be able to calibrate it. For instance, for proper alignment, the user must know which way is "up", "forward", or "side" for the tool. This mapping may be unnatural for tools that either have an ambiguous frame (e.g. a cube, sphere, or symmetric object) or have multiple ways

that they can be held (e.g. a cylinder/stick/wand). Thus, this calibration approach is limited by the user's familiarity and the specific design, modality, and use case of the tool being calibrated.

For the simulation system, many optical properties of real projectors were not modeled by the virtual projectors. For example, distinct spectral radiance of colour channels and non-linear brightness response were omitted for simplicity. We chose to focus more on the spatial mapping between projector image pixels to projected pixel locations instead of colour and luminosity. Having the virtual and real projectors match in luminosity and colour was not an important part in ensuring calibration accuracy for our system and, thus, was omitted. If the projector-to-display surface calibration was being investigated or studied using our testbed, this would be a natural place to improve.

Our study was performed in a simulated environment, rather than with a physical display. This was done to control for various perceptual factors; however, as future work, we plan to perform a similar evaluation with real spherical [117] and cubic [104] displays that we have built. Our software and experimental methodologies can be used to evaluate any display shape, either simulated in our VR platform or with a physical AR display.

We chose to study a spherical shape because it is the most common form-factor for volumetric FTVR displays. We expect that our findings would extend to cubic and cylindrical shaped displays because, like a sphere, their shape is convex. These display shapes offer wide viewing angles (sometimes 360° around), which can increase the 3D effect when motion parallax is taken advantage of. We also expect that our finding regarding the importance of stereo for task performance would be even more pronounced for a planar FTVR screen. This is because there are fewer opportunities for motion parallax, due to smaller viewing angles as compared to a spherical display.

One of the known limitations of headset VR is the mismatch in vergence and accommodation cues: the user focuses on a screen very close to their eyes, while their eyes converge toward virtual objects located distances far from the screens [105, 30, 61]. Volumetric FTVR screens largely avoid this problem by maintaining a metaphor of virtual objects contained within the bounds of the volumetric display. While it is possible to render virtual objects to appear to float in front of the globe, or exist far in the distance when looking through the globe, the primary usage is making objects appear within the globe. This metaphor helps the virtual content to appear more naturally situated within the real world, as if it were real objects within a glass globe or display case. Within such a metaphor, a viewer focuses their eyes on the front surface of the globe and can, at most, change the vergence of their eyes to objects virtually located at the back side of the globe. Therefore, the accommodation-vergence conflict is limited. We expect this is part of the reason why we did not have any reported VR sickness from participants. In general, since the vergence-accommodation conflict is further reduced when using a real FTVR display, we expect that our results will transfer well to the physical display and are likely to be stronger in the real-world use of a spherical FTVR.

6.2 Contributions

The work in this thesis has been shown to meet the research objectives outlined in Chapter 1, and has opened up new avenues of research within FTVR as described in *Limitations and Future Work*. The primary contributions correspond to the activities and solutions described in Chapter 3, Chapter 4, and Chapter 5. These contributions include:

1. The improvements and extensions from Chapter 3 were effective at increasing the accuracy and reducing the runtime of viewpoint calibration, and added a novel calibration technique to improve viewpoint and handheld object calibrations in realtime using perceptual feedback.

2. The FTVR testbed from Chapter 4 was an effective tool to investigate known defects of physical FTVR displays, explore calibration techniques, and study the perceptual effects of stereo.

3. The user study from Chapter 5 investigated the research questions outlined in Chapter 1. We showed that the improved calibration technique was effective at guiding novice users towards a more accurate calibration, that users could not reliably discern the presence of stereo, and that stereo increased task efficiency (time or error) for all performance tasks.

These solutions were built in parallel, and feedback from each was incorporated throughout the design and implementation phases. With a better simulation system came a better testbed; with a better testbed came better user feedback; with better user feedback came better calibration methods; and with better calibration came a better 3D experience in the simulation system.

6.3 Concluding Remarks

Fish tank virtual reality has been popularized by recent systems that eschew stereo rendering for a "glasses-free" 3D experience using head-tracked rendering alone. While these monoscopic spherical FTVR displays look perfect when shown in video, the in-person experience with and without stereo rendering has not been previously interrogated. In Chapter 3, we improved upon the best available viewpoint calibration methods, but showed that it may still be possible for some users to notice errors in these rigid viewpoint models. In Chapter 4, we developed a virtual testbed for investigating, testing, and studying a wide variety of FTVR display systems. In Chapter 5, we showed that, while users do not have a strong preference for stereo rendering on spherical displays, their performance in pattern alignment, distance estimation, 3D selection, and 3D manipulation is consistently better when stereo cues are included. Therefore, future designs of spherical and volumetric FTVR displays should include stereo in use cases for which performance is a priority.

Glossary

Fitts' law A predictive model of the amount of time it will take a human to move a cursor to a target area. The time it takes is dependent on the distance to the target, the size of the target, and two input-specific parameters [43].

Motion parallax A depth cue that requires only one of our eyes (monocular). Objects that are closer to an observer appear to move faster than objects farther away. This cue arises from relative motion between objects and an observer.

Stereo A depth cue that requires both of our eyes (binocular). An observer can use two different views of an object from binocular vision to perceive its depth and 3D structure. This cue can be simulated by presenting two different images separately to each eye; this is called stereoscopy (or stereoscopic rendering in the context of computer graphics).

Two-pass shadow mapping A common approach to rendering shadows in computer graphics. Shadow mapping determines which areas of a rendered scene should be considered unlit and then darkens them. The first pass looks at the scene from a light source's point of view and saves the depth (distance) of every surface in a texture called the shadow map. The second pass looks at the scene from a camera's point of view and darkens surfaces that are farther from the light than the depths stored in the shadow map.

References

[1] Kevin W. Arthur, Kellogg S. Booth, and Colin Ware. Evaluating 3d task performance for fish tank virtual worlds. *ACM Transactions on Information Systems*, 11(3):239–265, 1993.

[2] ARToolKit. Artoolkit home page. `http://www.hitl.washington.edu/artoolkit/`.

[3] K. Somani Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 698–700, 1987.

[4] Mark Ashdown, Matthew Flagg, Rahul Sukthankar, and James M. Rehg. A flexible projector-camera system for multi-planar displays. In *Computer Vision and Pattern Recognition*, volume 2. IEEE, 2004.

[5] Benjamin Bach, Ronell Sicat, Johanna Beyer, Maxime Cordeil, and Hanspeter Pfister. The hologram in my hand: How effective is interactive exploration of 3D visualizations in immersive tangible augmented reality? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):457–467, 2018.

[6] Bartosz Bajer, Robert J. Teather, and Wolfgang Sturzlinger. Effects of stereo and head tracking in 3d selection tasks. In *Proceedings of the 1st symposium on Spatial user interaction*, pages 77–77. ACM, 2013.

[7] O. R. Belloc, M. R. Nagamura, D. Fonseca, André Rodrigues, D. A. R. Souza, Celso Setsuo Kurashima, M. M. Almeida, Eduardo Zilles Borba, R. D. Lopes, and Marcelo Knörich Zuffo. OrbeVR: a handheld convex spherical virtual reality display. In *ACM SIGGRAPH 2017 Emerging Technologies*, page 19. ACM, 2017.

[8] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[9] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

[10] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. MirageTable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 199–208. ACM, 2012.

[11] Hrvoje Benko, Andrew D. Wilson, and Ravin Balakrishnan. Sphere: multi-touch interactions on a spherical display. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 77–86. ACM, 2008.

[12] Francois Berard and Thibault Louis. The object inside: Assessing 3d examination with a spherical handheld perspective-corrected display. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4396–4404. ACM, 2017.

[13] Oliver Bimber, Bernd Fröhlich, Dieter Schmalstieg, and L. Miguel Encarnação. The virtual showcase. In *ACM SIGGRAPH 2005 Courses*, page 3. ACM, 2005.

[14] Barry G. Blundell and Adam J. Schwarz. The classification of volumetric display systems: characteristics and predictability of the image space. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):66–75, 2002.

[15] Martin Böcker, Werner Blohm, and Lothar Mühlbach. Anthropometric data on horizontal head movements in videocommunications. In *Conference Companion on Human Factors in Computing Systems*, pages 95–96. ACM, 1996.

[16] Martin Böcker, Detlef Runde, and Lothar Mühlbach. On the reproduction of motion parallax in videocommunications. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 39, pages 198–202. SAGE Publications Sage CA: Los Angeles, CA, 1995.

[17] John Bolton, Kibum Kim, and Roel Vertegaal. Snowglobe: a spherical fish-tank vr display. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1159–1164. ACM, 2011.

[18] John Bolton, Kibum Kim, and Roel Vertegaal. A comparison of competitive and cooperative task performance using spherical and flat displays. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 529–538. ACM, 2012.

[19] John Bolton, Mike Lambert, Denis Lirette, and Ben Unsworth. Paperdude: a virtual reality cycling exergame. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 475–478. ACM, 2014.

[20] Paul Borke. Calculating stereo pairs, 1999. `http://paulbourke.net/stereographics/stereorender/`.

[21] Paul Bourke and Peter Morse. Stereoscopy: Theory and practice. In *Workshop at 13th International Conference on Virtual Systems and Multimedia*, 2007.

[22] Mark F. Bradshaw and Brian J. Rogers. The interaction of binocular disparity and motion parallax in the computation of depth. *Vision research*, 36(21):3457–3468, 1996.

[23] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[24] Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc T. Olano. 3DM: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 135–138. ACM, 1992.

[25] Marcio Cabral, F. Ferreira, Olavo Belloc, Gregor Miller, C. Kurashima, R. Lopes, Ian Stavness, J. Anacleto, Sidney Fels, and M. Zuffo. Portable-Spheree: A portable 3D perspective-corrected interactive spherical scalable display. In *Virtual Reality (VR), 2015 IEEE*, pages 157–158. IEEE, 2015.

[26] Jorge H. dos S. Chernicharo, Kazuki Takashima, and Yoshifumi Kitamura. Seamless interaction using a portable projector in perspective corrected multi display environments. In *Proceedings of the 1^{st} symposium on Spatial user interaction*, pages 25–32. ACM, 2013.

[27] James H. Clark. Designing surfaces in 3-D. *Communications of the ACM*, 19(8):454–460, 1976.

[28] Luca Cosmo, Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello. Design and evaluation of a viewer-dependent stereoscopic display. In *International Conference on Pattern Recognition*, pages 2861–2866. IEEE, 2014.

[29] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20^{th} annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.

[30] Varuna De Silva, Anil Fernando, Stewart Worrall, Hemantha Kodikara Arachchi, and Ahmet Kondoz. Sensitivity analysis of the human visual system for depth cues in stereoscopic 3-D displays. *IEEE Transactions on Multimedia*, 13(3):498–506, 2011.

[31] Michael Deering. High resolution virtual reality. *ACM SIGGRAPH Computer Graphics*, 26(2):195–202, 1992.

[32] Michael F. Deering. Making virtual reality more real: Experience with the virtual portal. In *Graphics Interface*, pages 219–219. CANADIAN INFORMATION PROCESSING SOCIETY, 1993.

[33] Parth Rajesh Desai, Pooja Nikhil Desai, Komal Deepak Ajmera, and Khushbu Mehta. A review paper on oculus rift-a virtual reality headset. *arXiv preprint arXiv:1408.1173*, 2014.

[34] R. Diamond, A. Wynn, K. Thomsen, and J. Turner. Three dimensional perception for one-eyed guys, or the use of dynamic parallax. *Computational Crystallography*, pages 286–293, 1982.

[35] J. P. Djajadiningrat, Gerda J. F. Smets, and C. J. Overbeeke. Cubby: a multiscreen movement parallax display for direct manual manipulation. *Displays*, 17(3-4):191–197, 1997.

[36] Rob Eisinga, Tom Heskes, Ben Pelzer, and Manfred Te Grotenhuis. Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers. *BMC bioinformatics*, 18(1):68, 2017.

[37] Dylan Fafard, Ian Stavness, Martin Dechant, Regan Mandryk, Qian Zhou, and Sidney Fels. FTVR in VR: evaluating 3D performance with a simulated volumetric fish-tank virtual reality display. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, 2019.

[38] Dylan Fafard, Andrew Wagemakers, Ian Stavness, Qian Zhou, Gregor Miller, and Sidney S. Fels. Calibration methods for effective fish tank VR in multi-screen displays. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 373–376. ACM, 2017.

[39] Dylan Fafard, Qian Zhou, Georg Hagemann, Chris Chamberlain, Sidney Fels, and Ian Stavness. Design and implementation of a multi-person fish-tank virtual reality display. In *ACM Symposium on Virtual Reality Software and Technology*, 2018.

[40] Fernando Ferreira, Marcio Cabral, Olavo Belloc, Gregor Miller, Celso Kurashima, Roseli de Deus Lopes, Ian Stavness, Junia Anacleto, Marcelo Zuffo, and Sidney Fels. Spheree: a 3D perspective-corrected interactive spherical scalable display. In *ACM SIGGRAPH 2014 Posters*, page 86. ACM, 2014.

[41] Scott Fisher. Viewpoint dependent imaging: An interactive stereoscopic display. Master's thesis, Massachusetts Institute of Technology, 1982.

[42] Scott S. Fisher, Micheal McGreevy, James Humphries, and Warren Robinett. Virtual environment display system. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 77–87. ACM, 1987.

[43] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[44] Davide Gadia, Gianfranco Garipoli, Cristian Bonanomi, Luigi Albani, and Alessandro Rizzi. Assessing stereo blindness and stereo acuity on digital displays. *Displays*, 35(4):206–212, 2014.

[45] Nirit Gavish, Teresa Gutiérrez, Sabine Webel, Jorge Rodríguez, Matteo Peveri, Uli Bockholt, and Franco Tecchia. Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23(6):778–798, 2015.

[46] Teodor P Grantcharov, VB Kristiansen, Jørgen Bendix, L Bardram, J Rosenberg, and Peter Funch-Jensen. Randomized clinical trial of virtual reality simulation for laparoscopic skills training. *British Journal of Surgery*, 91(2):146–150, 2004.

[47] Georg Hagemann, Qian Zhou, Ian Stavness, Oky Dicky Ardiansyah Prima, and Sidney S. Fels. Here's looking at you: A spherical FTVR display for realistic eye-contact. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, pages 357–362. ACM, 2018.

[48] Andrew J Hanson. Geometry for n-dimensional graphics. *Graphics Gems IV*, 443:149–170, 1994.

[49] Pawan Harish and P. J. Narayanan. A view-dependent, polyhedral 3D display. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, pages 71–75. ACM, 2009.

[50] Pawan Harish and P. J. Narayanan. Designing perspectively correct multiplanar displays. *IEEE transactions on visualization and computer graphics*, 19(3):407–419, 2013.

[51] Chris Harrison and Scott E. Hudson. Pseudo-3d video conferencing with a generic webcam. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, pages 236–241. IEEE, 2008.

[52] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. HoloDesk: direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2421–2430. ACM, 2012.

[53] Xianyou Hou, Li-Yi Wei, Heung-Yeung Shum, and Baining Guo. Real-time multi-perspective rendering on graphics hardware. *Rendering Techniques*, 6:93–102, 2006.

[54] Du Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[55] Jeffrey Jacobson and Zimmy Hwang. Unreal tournament for immersive interactive theater. *Communications of the ACM*, 45(1):39–42, 2002.

[56] Jeffrey Jacobson, Mark S. Redfern, Joseph M. Furman, Susan L. Whitney, Patrick J. Sparto, Jeffrey B. Wilson, and Larry F. Hodges. Balance NAVE: a virtual reality facility for research and rehabilitation of balance disorders. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 103–109. ACM, 2001.

[57] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 637–644. ACM, 2014.

[58] Naoki Kawakami, Masahiko INAMI, Taro MAEDA, and Susumu TACHI. Proposal for the object-oriented display: The design and implementation of the media3. In *Proceedings of ICAT*, volume 97, pages 57–62, 1997.

[59] Hyun K Kim, Jaehyun Park, Yeongcheol Choi, and Mungyeong Choe. Virtual reality sickness questionnaire (vrsq): Motion sickness measurement index in a virtual reality environment. *Applied ergonomics*, 69:66–73, 2018.

[60] Kibum Kim, John Bolton, Audrey Girouard, Jeremy Cooperstock, and Roel Vertegaal. TeleHuman: effects of 3d perspective on gaze and pose estimation with a life-size cylindrical telepresence pod. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2531–2540. ACM, 2012.

[61] Gregory Kramida. Resolving the vergence-accommodation conflict in head-mounted displays. *IEEE Transactions on Visualization & Computer Graphics*, (1):1–1, 2016.

[62] Max Krichenbauer, Goshiro Yamamoto, Takafumi Taketom, Christian Sandor, and Hirokazu Kato. Augmented reality versus virtual reality for 3d object manipulation. *IEEE transactions on visualization and computer graphics*, 24(2):1038–1048, 2018.

[63] Bireswar Laha, Doug A. Bowman, and John J. Socha. Effects of VR system fidelity on analyzing isosurface visualization of volume datasets. *IEEE Transactions on Visualization and Computer Graphics*, 20:513–522, 2014.

[64] Steven M. LaValle, Anna Yershova, Max Katsev, and Michael Antonov. Head tracking for the Oculus Rift. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 187–194. IEEE, 2014.

[65] Jiandong Liang and Mark Green. Geometric modeling using six degrees of freedom input devices. In *3rd Int'l Conference on CAD and Computer Graphics*, pages 217–222. Citeseer, 1993.

[66] Franz Madritsch, Franz Leberl, and Michael Gervautz. Camera based beacon tracking: accuracy and applications. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 101–108. Association of Computing Machinery, 1996.

[67] F. Landis Markley, Yang Cheng, John Lucas Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[68] Ian E. McDowall, Mark T. Bolas, Steven D. Pieper, Scott S. Fisher, and Jim Humphries. Implementation and integration of a counterbalanced CRT-based stereoscopic display for interactive viewpoint control in virtual-environment applications. In *Stereoscopic Displays and Applications*, volume 1256, pages 136–147. International Society for Optics and Photonics, 1990.

[69] Michael Wallace McGreevy and Stephen R. Ellis. The effect of perspective geometry on judged direction in spatial information instruments. *Human factors*, 28(4):439–456, 1986.

[70] Michael McKenna. Interactive viewpoint control and three-dimensional operations. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 53–56. ACM, 1992.

[71] Leonard McMillan and Gary Bishop. Head-tracked stereoscopic display using image warping. In *Stereoscopic Displays and Virtual Reality Systems II*, volume 2409, pages 21–31. International Society for Optics and Photonics, 1995.

[72] Jorge J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[73] Aske Mottelson and Kasper Hornbæk. Virtual reality studies outside the laboratory. In *Proceedings of the 23rd acm symposium on virtual reality software and technology*, page 9. ACM, 2017.

[74] Miguel A. Nacenta, Satoshi Sakurai, Tokuo Yamaguchi, Yohei Miki, Yuichi Itoh, Yoshifumi Kitamura, Sriram Subramanian, and Carl Gutwin. E-conic: a perspective-aware interface for multi-display environments. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 279–288. ACM, 2007.

[75] Jorge Nocedal and Stephen J. Wright. *Nonlinear equations*. Springer, 2006.

[76] OpenCV. Camera calibration and 3D reconstruction — OpenCV 2.4.13.7 documentation. `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`.

[77] Kevin Pfeil, Eugene M. Taranta II, Arun Kulshreshth, Pamela Wisniewski, and Joseph J. LaViola Jr. A comparison of eye-head coordination between virtual and physical realities. In *Proceedings of the 15th ACM Symposium on Applied Perception*, page 18. ACM, 2018.

[78] Kevin Ponto, Michael Gleicher, Robert G. Radwin, and Hyun Joon Shin. Perceptual calibration for immersive display environments. *IEEE transactions on visualization and computer graphics*, 19(4):691–700, 2013.

[79] Eric D. Ragan, Regis Kopper, Philip Schuchardt, and Doug A. Bowman. Studying the effects of stereo, head tracking, and field of regard on a small-scale spatial judgment task. *IEEE transactions on visualization and computer graphics*, 19(5):886–896, 2013.

[80] Ramesh Raskar. Immersive planar display using roughly aligned projectors. In *Virtual Reality, 2000. Proceedings. IEEE*, pages 109–115. IEEE, 2000.

[81] Stephan Reichelt, Ralf Häussler, Gerald Fütterer, and Norbert Leister. Depth cues in human visual perception and their realization in 3D displays. In *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, volume 7690. International Society for Optics and Photonics, 2010.

[82] Jun Rekimoto. A vision-based head tracker for fish tank virtual reality-VR without head gear. In *Virtual Reality Annual International Symposium, 1995. Proceedings.*, pages 94–100. IEEE, 1995.

[83] EB Roessler, RM Pangborn, JL Sidel, and H Stone. Expanded statistical tables for estimating significance in paired–preference, paired–difference, duo–trio and triangle tests. *Journal of Food Science*, 43:940–943, 1978.

[84] Behzad Sajadi and Aditi Majumder. Automatic registration of multi-projector domes using a single uncalibrated camera. In *Computer Graphics Forum*, volume 30, pages 1161–1170. Wiley Online Library, 2011.

[85] Peter J Savino and Helen V Danesh-Meyer. *Color Atlas and Synopsis of Clinical Ophthalmology–Wills Eye Institute–Neuro-Ophthalmology*. Lippincott Williams & Wilkins, 2012.

[86] Christopher Schmandt. Spatial input/display correspondence in a stereoscopic computer graphic work station. In *ACM SIGGRAPH Computer Graphics*, volume 17, pages 253–261. ACM, 1983.

[87] Neal E Seymour, Anthony G Gallagher, Sanziana A Roman, Michael K O'brien, Vipin K Bansal, Dana K Andersen, and Richard M Satava. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery*, 236(4):458, 2002.

[88] Ian Stavness, Billy Lam, and Sidney Fels. pCubee: a perspective-corrected handheld cubic display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1381–1390. ACM, 2010.

[89] Ian Stavness, Florian Vogt, and Sidney Fels. Cubee: a cubic 3D display for physics-based interaction. In *ACM SIGGRAPH 2006 Sketches*, page 165. ACM, 2006.

[90] Tsuyoshi Suenaga, Yasuyuki Tamai, Yuichi Kurita, Yoshio Matsumoto, and Tsukasa Ogasawara. Poster: Image-based 3D display with motion parallax using face tracking. In *IEEE Symposium on 3D User Interfaces*, pages 161–162. IEEE, 2008.

[91] Ivan E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[92] Yichen Tang, Ian Stavness, and Sidney S. Fels. The new pCubee: Multi-touch perspective-corrected cubic display. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 419–422. ACM, 2014.

[93] Robert J. Teather and Wolfgang Stuerzlinger. Target pointing in 3D user interfaces. *Poster at Graphics Interface*, 2010.

[94] Robert J. Teather and Wolfgang Stuerzlinger. Pointing at 3D targets in a stereo head-tracked virtual environment. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 87–94. IEEE, 2011.

[95] Robert J. Teather and Wolfgang Stuerzlinger. Pointing at 3d target projections with one-eyed and stereo cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 159–168. ACM, 2013.

[96] Unity Technologies. Unity - manual: Rendering with replaced shaders. `https://docs.unity3d.com/Manual/SL-ShaderReplacement.html`.

[97] Unity Technologies. Unity - scripting API: Camera.projectionmatrix. `https://docs.unity3d.com/ScriptReference/Camera-projectionMatrix.html`.

[98] F. Teubl, Celso Setsuo Kurashima, Marcio Calixto Cabral, R. D. Lopes, Júnia Coutinho Anacleto, Marcelo Knörich Zuffo, and Sidney Fels. Spheree: An interactive perspective-corrected spherical 3D display. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pages 1–4. IEEE, 2014.

[99] Michael Treadgold, Kevin Novins, Geoff Wyvill, and Brian Niven. What do you think you're doing? Measuring perception in Fish Tank Virtual Reality. In *Computer Graphics International 2001. Proceedings*, pages 325–328. IEEE, 2001.

[100] HCT UBC. Calibration methods for effective fish tank VR in multi-screen displays - YouTube. `https://www.youtube.com/watch?v=Kbs-3XGa2YQ`.

[101] HCT UBC. Siggraph2018 - YouTube. `https://www.youtube.com/watch?v=mN5opwUxkuI`.

[102] HCT UBC. Spherical fish tank VR research demo for IEEE VR 2017 - YouTube. `https://www.youtube.com/watch?v=w_5OZGEt6C8`.

[103] Andrew J Wagemakers. Calibration methods for head-tracked 3d displays, 2018.

[104] Andrew John Wagemakers, Dylan Brodie Fafard, and Ian Stavness. Interactive visual calibration of volumetric head-tracked 3d displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3943–3953. ACM, 2017.

[105] John P. Wann, Simon Rushton, and Mark Mon-Williams. Natural problems for stereoscopic depth perception in virtual environments. *Vision research*, 35(19):2731–2736, 1995.

[106] Colin Ware, Kevin Arthur, and Kellogg S. Booth. Fish tank virtual reality. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 37–42. ACM, 1993.

[107] Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics*, 15(2):121–140, 1996.

[108] Colin Ware and Peter Mitchell. Reevaluating stereo and motion cues for visualizing graphs in three dimensions. In *Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, pages 51–58. ACM, 2005.

[109] Stephen Whiteside. *The stereomatrix 3-D display system*. PhD thesis, University of Illinois, Urbana-Champaign, 1973.

[110] Christopher D. Wickens. Three-dimensional stereoscopic display implementation: Guidelines derived from human visual capabilities. In *Stereoscopic displays and applications*, volume 1256, pages 2–12. International Society for Optics and Photonics, 1990.

[111] Andrew D. Wilson and Hrvoje Benko. Projected augmented reality with the RoomAlive toolkit. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, pages 517–520. ACM, 2016.

[112] Andrew D. Wilson and Hrvoje Benko. Holograms without headsets: Projected augmented reality with the RoomAlive toolkit. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 425–428. ACM, 2017.

[113] Andy Wilson. Roomalive toolkit. *GitHub* *https://github.com/Kinect/RoomAliveToolkit*, 2015.

[114] Björn Wöldecke, Dionysios Marinos, and Christian Geiger. Flexible registration of multiple displays. In *Proceedings of The International Symposium on Pervasive Displays*, page 86. ACM, 2014.

[115] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[116] Qian Zhou, Georg Hagemann, Dylan Fafard, Ian Stavness, and Sidney Fels. An evaluation of depth and size perception on spherical fish tank virtual reality display. In *IEEE VR 2019 Journal Papers*, 2019.

[117] Qian Zhou, Georg Hagemann, Sidney Fels, Dylan Fafard, Andrew Wagemakers, Chris Chamberlain, and Ian Stavness. Coglobe: a co-located multi-person FTVR experience. In *ACM SIGGRAPH 2018 Emerging Technologies*, page 5. ACM, 2018.

[118] Qian Zhou, Gregor Miller, Kai Wu, Daniela Correa, and Sidney Fels. Automatic calibration of a multiple-projector spherical fish tank vr display. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1072–1081. IEEE, 2017.

[119] Qian Zhou, Gregor Miller, Kai Wu, Ian Stavness, and Sidney Fels. Analysis and practical minimization of registration error in a spherical fish tank virtual reality system. In *Asian Conference on Computer Vision*, pages 519–534. Springer, 2016.

[120] Qian Zhou, Kai Wu, Gregor Miller, Ian Stavness, and Sidney Fels. 3DPS: An auto-calibrated three-dimensional perspective-corrected spherical display. In *Virtual Reality (VR), 2017 IEEE*, pages 455–456. IEEE, 2017.

Appendix A

Shaders

```
Shader "Biglab/GeometryTexture/Position" {
  SubShader {
    Tags {
      "Queue"="Geometry"
      "RenderType"="Opaque"
      "IgnoreProjector"="True"
    }

    LOD 100

    Pass {
      CGPROGRAM
      #pragma vertex vert
      #pragma fragment frag
      #include "UnityCG.cginc"

      struct appdata {
        float4 vertex : POSITION;
      };

      struct v2f {
        float4 position : SV_POSITION;
        float4 vPosition : TEXCOORD0;
      };

      uniform float4x4 _WorldToVolume;

      v2f vert(appdata v) {
        float4 w_position = mul(unity_ObjectToWorld, v.vertex);

        v2f o;
        o.position = UnityObjectToClipPos(v.vertex);
        o.vPosition = mul(_WorldToVolume, w_position);
        return o;
      }

      float4 frag(v2f i) : SV_Target {
        return i.vPosition / i.vPosition.w;
      }

      ENDCG
    }
  }
}
```

```
Shader "Biglab/GeometryTexture/Normal" {
  SubShader {
    Tags {
      "Queue"="Geometry"
      "RenderType"="Opaque"
      "IgnoreProjector"="True"
    }

    LOD 100

    Pass {
      CGPROGRAM
      #pragma vertex vert
      #pragma fragment frag
      #include "UnityCG.cginc"

      struct appdata {
        float4 vertex : POSITION;
        float3 normal : NORMAL;
      };

      struct v2f {
        float4 position : SV_POSITION;
        float3 vNormal  : TEXCOORD0;
      };

      uniform float4x4 _WorldToVolumeNormal;

      v2f vert(appdata v) {
        float3 w_normal = UnityObjectToWorldNormal(v.normal);

        v2f o;
        o.position = UnityObjectToClipPos(v.vertex);
        o.vNormal = mul(_WorldToVolumeNormal, float4(w_normal, 0));
        return o;
      }

      float4 frag(v2f i) : SV_Target {
        float3 vNormal = normalize(i.vNormal);
        return float4(vNormal, 1.0f);
      }

      ENDCG
    }
  }
}
```

Appendix B

Virtual Study Documents

B.1 Consent Form

INFORMED CONSENT FORM

UNIVERSITY OF SASKATCHEWAN
College of Arts and Science
DEPARTMENT OF COMPUTER SCIENCE
ARTSANDSCIENCE.USASK.CA

Research Project: **Investigating presentation modes for Fish Tank Virtual Reality (FTVR) in Virtual Reality (VR)**

Investigators: **Dr. Ian Stavness,** Department of Computer Science (ian.stavness@usask.ca)
Dylan Brodie Fafard, Department of Computer Science (dylan.fafard@usask.ca)
Christopher Chamberlain, Department of Computer Science (chris.chamberlain@usask.ca)
Martin Dechant, Department of Computer Science (Martin.Dechant@usask.ca)

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with FTVR 3D displays and aims to measure the location your viewpoint(s) and any noticeable differences in presentation modes (whether or not headtracking is used and either one or both eyes are used). The goal of the research is to examine FTVR displays in a consistent, high fidelity environment to guide the research and development of real FTVR displays.

The session will require **60 minutes**, during which you will be asked to perform the following tasks under each FTVR presentation mode: **guided viewpoint calibration, 3D spatial impression comparison, and point cloud selection/estimation/cutting in the Biomedical & Interactive Graphics Lab at the University of Saskatchewan. You will be asked questions before/during/after these tasks.**

Feel free to answer only the questions that you are comfortable with answering. At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for **your time and any travel costs you may have incurred, you will receive a $10 honorarium at the end of the session**. The data collected from this study will be used in articles for publication in journals and conference proceedings. As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled. This summary will outline the research and discuss our findings and recommendations. This summary will be available on the BIGLAB lab's website: **http://www.biglab.ca**

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred, and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

Dr. Ian Stavness, Professor, Dept. of Computer Science, (306) 966-7995, ian.stavness@usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about your rights as a participant, please contact:

Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature: _____ **Investigator's signature:** _____

Date:_____ Date:_____

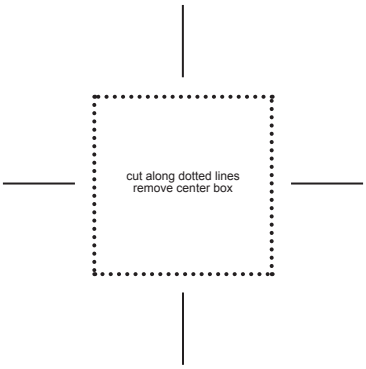
A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

B.2 Dominant Eye Test Card

USA*Eyes*.org

Dominant Eye Test Card

1. Print USA*Eyes*.org Dominant Eye Test Card on your computer printer.
2. Cut across dotted line to make smaller rectangle. Cut out center square.
3. Hold USA*Eyes*.org Dominant Eye Test Card with both hands at arm's length and centered in front of you.
4. With both eyes open, focus on any still object viewed through the center hole.
5. While continuing to keep focus on the object, keeping the object centered in the hole, and with both eyes open, slowly bring the USA*Eyes*.org Dominant Eye Test Card toward you until you touch your face.
6. The eye over which you have the USA*Eyes*.org Dominant Eye Test Card centered is your dominant eye.
7. Repeat test to verify.



The dominant eye is the eye that looks directly at an object. The non-dominant eye looks at the same object at a slight angle. This small difference provides depth perception.

Being right or left handed will not necessarily determine if you are right or left eye dominant. Eye dominance is an important consideration for monovision correction to reduce the need for reading glasses or bifocals.

cut along dotted line

B.3 Questionnaires

Investigating presentation modes for Fish Tank Virtual Reality (FTVR) in Virtual Reality (VR)

ID of the Questionnaire	

Virtual Reality Familiarity

Have you used a Virtual Reality headset before?

Please circle one answer

- O Yes
- O No

Which Virtual Reality headset(s) have you used before?

Please circle any that apply

- O Google Cardboard
- O Samsung Gear VR
- O Sony PlayStation VR
- O HTC Vive
- O Oculus Rift
- O Google Daydream View
- O Other (Please Specify): _____

How often do you use Virtual Reality headsets?

Please circle one answer

- O Less than once per week
- O Between 1-3 times per week
- O More than 3 times per week

1

B.4 Questionnaires

Investigating presentation modes for Fish Tank Virtual Reality (FTVR) in Virtual Reality (VR)

ID of the Questionnaire	

Viewport Calibration

Consider how the pattern changed as you moved your head for the following questions. Did the pattern move in a way that helped you align it to the target image?

Please circle one answer

What strategy(s) did you use to align the pattern?

Please write down your thoughts

B.5 Questionnaires

Investigating presentation modes for Fish Tank Virtual Reality (FTVR) in Virtual Reality (VR)

ID of the Questionnaire	

3D Spatial Impression

In any of the conditions, did the scene inside the display appear three dimensional (3D)?
Please circle one answer

O Yes

O No

In any of the conditions, did the scene inside the display seem to stay still as you moved around?
Please circle one answer

O Yes

O No

If you answered No to both of the questions above, please skip the following section and continue with Page 4.
For the following questions please consider the comparison of conditions on the scene inside the display.

The change in 3D appearance was noticeable.
Please circle one answer

O Strongly Disagree O Disagree O Undecided O Agree O Strongly Agree

There was a noticeable change in it staying still.
Please circle one answer

O Strongly Disagree O Disagree O Undecided O Agree O Strongly Agree

B.6 Questionnaires

Investigating presentation modes for Fish Tank Virtual Reality (FTVR) in Virtual Reality (VR)

ID of the Questionnaire	

General Impression and Comments

Please leave any comments, suggestions, or feedback you have about the study.
Please write down your thoughts

Thank you very much for participating! 😊

4

B.7 Virtual Reality Sickness Questionnaire

Investigating presentation modes for Fish Tank Virtual Reality (FTVR) in Virtual Reality (VR)

ID of Questionnaire	
Task ID	

Virtual Reality Sickness Questionnaire

Hyun K. Kim, Jaehyun Park, Yeongcheol Choi, Mungyeong Choe[1]

Circle how much each symptom below is affecting you right now.

Please circle one answer

General discomfort	O	None	O	Slight	O	Moderate	O	Severe
Fatigue	O	None	O	Slight	O	Moderate	O	Severe
Eyestrain	O	None	O	Slight	O	Moderate	O	Severe
Difficulty focusing	O	None	O	Slight	O	Moderate	O	Severe
Headache	O	None	O	Slight	O	Moderate	O	Severe
Fullness of head	O	None	O	Slight	O	Moderate	O	Severe
Blurred vision	O	None	O	Slight	O	Moderate	O	Severe
Dizzy (eyes closed)	O	None	O	Slight	O	Moderate	O	Severe
Vertigo	O	None	O	Slight	O	Moderate	O	Severe

*Vertigo is experienced as a loss of orientation with respect to vertical upright.

[1] *Kim, H. K., Park, J., Choi, Y., & Choe, M. (2018). Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. Applied Ergonomics, 69, 66-73.*