Enhanced Interaction in Immersive Virtual Environments

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Loren Frank Puchalla Fiore

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Victoria Interrante

May, 2016

Acknowledgements

First, I would like to thank the 3M Corporation for their graduate fellowship that allowed me to attend graduate school. I would also like to thank the Linda and Ted Johnson Digital Design Consortium and the National Science Foundation for lab funding.

I would like to thank my colleagues in the DDC lab, Lane Phillips, Peng Liu, and Koorosh Vaziri who have served as coauthors and friends during my time at the University of Minnesota. They have on many occasions helped with running user studies, debugging software, and providing useful feedback during the PhD process. I also thank Ying He, who assisted with the user studies described in Chapter 3.

To my adviser, Victoria Interrante, I am deeply grateful. Vicki has been very helpful throughout my years as a graduate student in matters of writing, research, and teaching.

Finally, I would like to thank my family. My parents, Leonard and Joy, for their support during my collegiate studies. And most importantly my wife, Heather, for her encouragement all through my years as a graduate student and before. You endured many late nights and uneventful weekends, helped proofread my papers, and gave me the motivation to work through the difficult projects. I could not have done this without you.

Abstract

Virtual reality (VR) has many uses across diverse areas such as scientific visualization, flight training, and architecture design. The spatial awareness and feeling of presence created by immersive virtual environments (IVE) assists with learning and reasoning tasks within VR and is one reason for its adoption. However, there are still problems of perception within IVEs that can limit the effectiveness of their use. One example is the compression of egocentric distances when using a head-mounted display to view the IVE. The goal for this dissertation is to investigate whether IVEs can be made more effective through the development of enhanced locomotion and interaction methods that provide more accurate visual and vestibular feedback to the user. We investigate the use of color and depth (RGB+D) cameras to generate real-time video-based self-avatars that perfectly match the user's own body without the need for markers or per-user calibration. We perform a user study to determine if the video avatars reduce the errors in egocentric distance perception experienced in virtual environments. Next, we discuss the geometric and perceptual errors present in multi-viewer single-view virtual reality displays, such as CAVEs. Since the calibration and setup of these displays is crucial to limit these errors, we have created open source software for calibration of these displays. Finally, we look at walking in IVE and discuss the benefits of natural movements over *indirect* interfaces such as keyboards and introduce the novel technique of redirected driving. We show that redirected driving has the potential to offer the benefits of walking, such as better spatial understanding and mapping recall, while still allowing movement of large distances.

Contents

List of Figures

1

Introduction

> *He inferred that persons desiring to train this faculty must select localities and form mental images of the facts they wish to remember and store those images in the localities, with the result that the arrangement of the localities will preserve the order of the facts, and the images of the facts will designate the facts themselves, and we shall employ the localities and images respectively as a wax writing tablet and the letters written on it.*
>
> – Cicero, *de Oratore 2.86.354*

The method of loci, described in the quote above and occasionally referred to as the memory palace technique, is a mnemonic device that uses visualization to organize and remember information. It was used in antiquity by Cicero, Quintilian, and others to remember speeches, as paper and ink were still too valuable for speaking notes. It is still used by many in the present day to for memory contests, such as the record 2^{16} digits of π [1], and everyday items such as shopping lists.

The method at its core is straightforward. Simply visualize a place that you have been and know well, such as your house, and within each room of your house place one piece of the information to be remembered. Then at a later time the visualization of the well known place will bring back the ordering and contents of the memory. The method is effective because it is well-suited to how the human brain has evolved to have a keen sense of spatial understanding. Functional magnetic resonance imaging (fMRI) has shown that the brains of individuals in the process of remembering using the method

have activation in the areas of the brain thought to control spatial awareness such as the medial parietal cortex and the right posterior hippocampus [2].

The method of loci shows the strength of the human mind at visualizing spatial information and how one can leverage that strength to help with other unrelated tasks. Leveraging the mind's ability of spatial understanding is also one of the motivations behind the field of virtual reality. Virtual reality (VR) is defined as a computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real manner. VR can take the form of large-screen immersive displays (LSID), such as those found in motion simulators for planes and automobiles, and head-mounted displays (HMD) that allow the wearer to view only the virtual environment. Having the ability to view, navigate, explore, and interact with a virtual environment has been shown to increase task performance on a variety of tasks versus attempting the same task with a traditional monitor and keyboard computer setup [3–7].

1.1 Immersive Virtual Reality

When a virtual environment simulates the user being in the environment and in such a way that they themselves feel physically present, then it is said to be an immersive virtual environment (IVE). The immersion can be as straightforward as head-tracking, so that the user's head motion causes the VR view to update accordingly, to something as involved as physics-based interaction with the environment. These IVEs have a diverse range of uses across areas such as scientific visualization, flight training, design review, and architectural design, to name a few [8, 9]. One reason for this wide-spread adoption is that virtual reality provides the ability to immerse the user in the virtual environment in a way that is fundamentally different from traditional keyboard and monitor setups. Immersive virtual environments can take advantage of the human mind's keen sense of spatial awareness, and systems using head-tracking can provide a metaphor for movement and exploration that is in many ways the same as real life. For instance, users can change their point of view by moving themselves and by turning their head instead of pressing buttons or moving a mouse. Additionally, many immersive virtual reality systems employ stereoscopic displays that allow each eye to see slightly different images allowing for full three-dimensional viewing of the virtual environment.

However, there are limitations to the current implementations of these systems. For one, tracking of any kind is only possible when the user is within the tracked space. Larger tracked spaces require larger lab spaces to move within and additional tracking equipment. Also, stereoscopic displays require calibration and when miscalibrated can create a false sense of space that can fool users into drawing incorrect inferences about the virtual environment. For example, when designing a building in an IVE an architect must be able to trust that the space they and the clients are viewing looks the same as it does in the real world. If there is a miscalibration that causes distances to appear shorter when viewed in the IVE the clients may insist on a requiring a space larger than needed, resulting in wasted resources and additional cost. In the following sections we will discuss the promises and limitations of the specific areas of immersive virtual reality addressed in this dissertation.

1.1.1 Video-Based Self-Avatars

Immersive virtual environments experienced through head-mounted displays (HMDs) confer several benefits. If head-tracking is used, the display can update in real-time in response to the motion of the wearer who can move around the virtual environment and examine it from different angles in a natural manner. Another benefit of the HMD system is that, most of the time, the real-world is completely obscured from view, forcing the user to focus on the virtual environment and enhancing the feeling of being physically present. However, this is also a major disadvantage because the obscuring of the real-world also obscures the user's view of his or her own body. This can be quite disorienting when navigating a virtual environment and has a detrimental effect on the immersion. A solution to this problem is the give the user a VR avatar.

An avatar is a graphical representation of the user in the virtual environment. Avatars are often created from generic models, sometimes with minor customization options, and rendered using the tracking data of the user's body. Depending on the application and the tracking system used, the avatar's arms and legs may move in the virtual environment to match the motions of the user in the real world, or the limbs may be static and only the head position of the avatar follows the user movement. In [10], Slater *et al.* show that immersive virtual environments employing a generic avatar increase the feeling of presence and immersion in the environment compared to the same

environment without any avatar. However, a generic avatar is not as realistic as the real-world. The physical appearance, clothing, and even the gender of the avatar may not match that of the user. Also, the tracking systems available often require per-user calibration resulting in long setup times before the virtual reality system can be used.

We propose, in Chapter 3, to use a video see-thru HMD, that is one with attached video cameras, in order to create a video-based self-avatar for the user. This will allow the user to see an avatar matching his or her own likeness within the virtual environment. Our hypothesis is that this method of avatar creation will show an increase in presence greater than that of a generic avatar. The use of a camera-based system instead of one involving tracking markers, such as optical motion capture, makes the system easier to use because no per-user calibration is needed prior to use.

1.1.2 Immersive Projector Calibration

A popular display modality in the virtual reality community is the Large Screen Immersive Display (LSID). These displays are usually on the scale of two meters or more in both width and height and are sometimes referred to as *display walls* or *virtual windows*. A LSID may have a single screen surface that subtends a large portion of the viewer's field of view or multiple surrounding screen surfaces such as in a CAVE system. An advantage of these large immersive displays is that they are large enough for multiple users to view the scene and, at the same time, interact with each other in a natural way. This is often useful in design review or data visualization tasks to facilitate discussion between the designers and researchers viewing the virtual environment.

A critical step in the creation of a LSID is calibration of the display surface. This is especially true with stereoscopic LSIDs using head-tracking, where the center of project of the rendered view must match the vantage point of the user. Normally, in stereo vision, feature points between images seen by both eyes can be matched and triangulated by our eyes to obtain depth information about a scene. When the rendered views do not match, or are rendered incorrectly, the geometric solution may not exist and the human brain will approximate a solution [11]. This approximate solution, if believed, can lead users to experience the virtual environment differently than if it were a real physical space. For example, when viewing a CAD design of a building a user may believe a virtual wall to be 2.5 meters distant, but in the design the wall is only 2

meters distant. In the use case of architectural design this can, and will, cause incorrect designs because the user's senses are tricked into perceiving the environment incorrectly, and this incorrect understanding will affect any decisions made to alter the design.

In Chapter 4, we describe the development of an open-source software package for the calibration of arbitrary shaped rear-projection large-screen immersive displays. This software was developed at the University of Minnesota in order to calibrate the LSID at Digital Design Consortium lab. This particular screen is saddle-shaped (i.e. curving in opposite directions along two axes) and as such was not capable of calibration using the traditional parametric approaches. The software makes it easy, using only a webcam, several printed fiducial markers, and time to calibrate any shape of screen by a piece-wise planar approximation. Techniques for rendering using the calibrated surface are also discussed in this chapter.

1.1.3 Locomotion in Large Environments

Creating the experience of natural self-motion in virtual environments is a fundamental problem facing the field of virtual reality [12]. While tracking technologies allow users of virtual reality to perform motions in the virtual world that match their real-world motions, many such setups provide this natural interaction over the range of little more than a few square meters. This can be because of technical limitations of the tracking method, the cost of covering a larger area, or simply the limiting size of the available physical space. To overcome these limitations, users of virtual environments often have to revert to indirect forms of traveling involving keyboard, joystick, or wand input. These systems allow the users to travel in a large virtual environment but at the expense of losing the ability for natural motion. This is detrimental because several groups have shown that users of indirect locomotion systems have substantially greater difficulty in remembering their motions and the layout of the virtual environment than users who are able to move about naturally [3, 13, 14].

Natural walking redirection techniques have shown great potential for enabling users to travel in large-scale virtual environments while their physical movements are limited to a much smaller laboratory space [15]. In the real world, walking is primarily used to cover short distances, but a variety of methods of transportation are employed when

traversing longer distances. In Chapter 5 we introduce a novel approach to the concept of redirection by using physical traveling devices. We show that using a modified electric wheelchair it is possible to redirect users using visual rotations as well as direct re-steering of the wheelchair itself. To evaluate this novel locomotion technique we designed and conducted two psychovisual experiments comparing the technique to existing redirection and locomotion strategies in terms of effectiveness at environment spatial understanding, redirection sensing thresholds, and rate of cybersickness. We have designated this technique *redirected driving* and believe it fills an important niche of natural-seeming locomotion covering large distances as it allows users to draw on their everyday experiences of driving or riding in moving vehicles.

1.2 Our Approach

1.2.1 Thesis Statement

The goal of this dissertation is to increase the effectiveness of immersive virtual environments by investigating the following thesis statement:

> *Immersive virtual environments can be made more effective through the development of enhanced locomotion and interaction methods that provide more accurate visual and vestibular feedback to the user.*

Our research accomplishes this investigation through three focused research tasks:

- Develop and evaluate video-based methods for virtual self-avatars that operate without the need of body tracking or per-user calibration. (Chapter 3)
- Create open-source projector display calibration software that allows for arbitrary screen shape and flexible projector configuration using an off-the-shelf web camera for calibration input. (Chapter 4)
- Propose and evaluate the novel locomotion technique of *redirected driving* that allows for natural metaphors for the exploration of large virtual environments via direct and indirect redirection of user movements. (Chapter 5)

In the pursuit of these research avenues, we have developed new software and algorithms for virtual reality that are applicable to a large variety of immersive virtual environments. Throughout this dissertation we will focus on the driving use case of architectural design, however, we expect the contributions to be useful in other domains. For the self-avatar and redirected driving research, we have performed multiple user-studies of the effectiveness and will present and discuss the results.

1.3 Dissertation Overview

The dissertation begins in Chapter 2 with an overview of the research creating virtual reality, discovering its limitations, and several of the key experiments and efforts by others to overcome these limitations. Then, in Chapter 3, we investigate the use of color (RGB) and color plus depth (RGB+D) cameras for the purpose of generating real-time video-based self-avatars to give users an avatar that perfectly matches their own body yet requires no markers or per-user calibration. Next, in Chapter 4, we discuss the types of geometric and perceptual errors present in multi-viewer single-view virtual reality display setups, such as CAVEs or other large stereoscopic displays. We show how these errors can result from corresponding errors in calibration and describe the development of an open source software package for arbitrary shaped, multiple projector, screen calibration. Finally, in Chapter 5, we look at walking in immersive virtual environments and discuss the benefits of natural movements over more *indirect* interfaces such as keyboards and joysticks. Furthermore, we introduce the novel technique of *redirected driving*. We demonstrate the potential of redirected driving to confer the benefits of walking while maintaining its usefulness for covering great distances similar to the methods involving indirect interfaces, such as flying using a wand or joystick interface.

2

Background & Related Work

Immersive virtual environments (IVEs) are made possible by a specialized set of computer hardware and software. In this dissertation we focus on two types of immersive virtual environments, those using head-mounted displays (HMDs) and those using large-screen immersive displays (LSIDs).

An HMD contains two small computer screens positioned one in front of each eye. By rendering slightly different images to each screen a three-dimensional stereoscopic view is presented to the wearer. An example HMD, and the one used in this dissertation, is the SX60 manufactured by NVIS [16]. Recently, consumer grade HMDs have begun to appear on the market using smartphone technology in order to drastically reduce prices such as the Oculus Rift [17], Samsung Gear VR [18], and Google Cardboard [19].

The CAVE Automatic Virtual Environment (CAVE), first proposed and built by Cruz-Neira *et al.* in 1992 [20, 21] is one type of large-screen immersive display, and the second display modality discussed in this dissertation. A CAVE is a cubic room with full-extent stereoscopic screens on several of its surfaces. The original CAVE had four of its six surfaces as screen (three walls and the floor) but several modern CAVEs, such as the ones built by Mechdyne Corporation at Iowa State University and KAUST [22], have all six surfaces as displays. Another type of LSID is the large projector wall. Typically several meters in both width and height this can be flat or curved. An extreme example of a curved LSID wall is the CAVE2 at the University of Illinois at Chicago [23] that is a cylindrical room with a tiled 37-megapixel stereoscopic display wrapping 320-degrees around the user.

Regardless of the display modality being used any IVE application will also require some form of user tracking. Tracking the user's head position and orientation is critical for rendering their correct first-person viewpoint of the virtual world. The current state-of-the-art tracking system is the optical tracking system, such as that manufactured by Vicon [24], that uses reflective markers attached to the HMD and watched by multiple cameras placed around the perimeter of the tracked physical space. The video from the cameras is sent to a central computer server that localizes the marker position in each image and uses that to triangulate its real-world position. By attaching multiple markers to the HMD a full six degree of freedom (6-DOF) tracking can be achieved. Markers may also be placed on the arms, legs, and body of the user for full-body motion capture. An interesting variation on the optical tracking system is the Hiball [25], that switches the roles of the markers and the camera. In the Hiball system the cameras are placed on the HMD and the markers, each a flashing infrared light, are place on the ceiling of the room. This enables greater stability in the localization at the expense of a more complex head-worn display and sensor unit.

Another commonality between display modalities is the software. Depending on the application an immersive virtual environment may require physically-based rendering, physics simulation, artificial intelligence for virtual agents and a host of other complicated software. To ease the burden of this integration reusable software is often used to create IVEs. A notable example of a commercial software is Vizard by WorldViz [26]. Another advantage for virtual reality is the recent shift in the video games industry towards independent, or indie, developers. This has created many affordable video game engines, several of which are open source. In this dissertation we make use of the G3DEngine [27], Unity3D [28], and, most recently, the Unreal 4 Engine [29] to create immersive virtual environments.

Existing research related to this dissertation falls into three broad categories: Avatars & Embodiment, Large-Screen Immersive Display (LSID) Design & Calibration, and Natural Locomotion in IVEs. We will now briefly discuss each of these categories including the established techniques and technologies in each as well as any remaining unsolved problems. More detailed and in-depth review of prior work is left for the start of each related chapter.

2.1 Avatars & Embodiment

An avatar is a graphical representation of the user. In immersive virtual environments this typically appears as a three-dimensional model of person. The avatar can be designed to match the user's physical appearance in the real-world or be purposefully different depending on the application. In multi-user environments it is possible to show avatars for the other users as well as virtual agents for AI characters. In this dissertation we focus specifically on first-person avatars of the user.

Giving the user a virtual avatar creates a greater sense of realism in the IVE as well as an awareness of the effects of his or her actions. The avatar also provides several spatial cues to help the user understand the extent of the virtual space and his or her location within. It is strongly believed that virtual avatars increase a user's sense of presence in the virtual environment [10]. Lok [30] *et al.* has shown that the use of virtual avatars can lead to significant improvement in performing tasks involving reaching distances.

In many virtual reality systems the only tracked object is the head-mounted display. A common occurrence in this instance is to not render any avatar for the user, creating an abrupt visual disconnect from the laws of physics as the user experiences the immersive virtual environment from a disembodied view. An alternative is the give the user a rigid, non-articulated, avatar that is always aligned with the position of the tracked head-mounted display. There will still be some disconnect as the motion of the user's arms and legs will not be reflected in the virtual environment.

Adding additional tracked objects to the virtual reality system is usually the next step for more robust avatars. Wands and hand markers can be used to track the location of the user's hands and feet in order to articulate the arms using inverse kinematics. More elaborate systems involving full-body motion capture can allow accurate tracking of the entire skeleton and avoid the jumps in motion often present in inverse kinematics as the solver fluctuates between multiple correct solutions.

An open area of research in this field is reducing the number of tracking markers that must be used in order to create an avatar. In the extreme case the goal is marker-less avatar creation. Marker-less systems promise to reduce the costs of providing articulated avatars in virtual reality and also simplify the process of using the virtual reality system as there will be no markers or special clothing to wear in order to use the system.

2.2 LSID Design & Calibration

A popular display modality in virtual reality is the Large Screen Immersive Display (LSID). These displays are usually on the scale of 2m or more in height, and 2m or more in width and are sometimes referred to as *display walls* or *virtual windows*. A LSID may have a single screen surface that subtends a large portion of the viewer's field of view, or multiple surrounding screen surfaces such as in a CAVE system.

A critical step in the construction of any LSID is the determination of the mapping from display pixels to world space. This mapping is needed in order to display the correct stereo image given a specific vantage point. The process of determining this mapping is known as screen calibration.

Calibration is often done manually, using specially designed screen hardware and mounting. However, this is time consuming and does not scale well. Several researchers have therefore investigated automatic calibration of LSIDs using feedback from one for more video cameras, [31–39]. However, even with all of this knowledge relating to automated calibration of LSIDs, the the authors were unable to find any readily available software package for performing automated calibration of arbitrary LSID configurations. Therefore, in Chapter 4, we present the results of our work to create such a software package.

2.3 Natural Locomotion in IVEs

One of the primary means of interaction within immersive virtual environments is exploration. Users need to be able to move about within the environment and examine it from different perspectives. The most straight-forward way of exploring a virtual environment when viewed through an HMD is to use head-tracking and walk about in a natural way with motions in the real-world mapping one-to-one with motions in the IVE. However, because of the physical limitations of the available physical space and the tracking equipment involved it is very often the case that the virtual environment displayed is much larger than the tracked space. A notable exception is the HIVE, Huge Immersive Virtual Environment, where the tracking system along with all the required VR hardware is wearable and can be used outdoors in ample space [40]. To overcome the size limits of the tracked space many researchers have studied additional locomotion

techniques.

The most common form of locomotion in immersive virtual environments is natural walking, where real-world movements map one-to-one to virtual movements, limited to the tracked workspace with button-controlled flying or jumping to new locations within the virtual environment [41]. When the tracked space is too small to allow actual walking, walking-in-place is an alternative. In [42], Slater *et al.* develop a walking-in-place algorithm that analyzes tracked head motion to count steps. In [43], Usoh *et al.* discuss the results of a user study showing that walking interfaces have much higher levels of presence and immersion that those of walking-in-place or flying.

In [44], Interrante *et al.* propose the "seven-league boots" metaphor for exploring virtual environments. In this mode the user holds a tracked wand device with a single button. Without using the wand, motion within the virtual environment is matched exactly to the head-tracked motion. When the button is pressed and the user walks forward the program determines the direction of travel using the head tracking information and amplifies the motion along this direction. This allows for rapid movement between different regions of the virtual environment when the button is pressed, and for natural movement when the user has reached the destination. A disadvantage of this system is its unnatural modal nature requiring the user to press a button to switch locomotion modes.

An interesting fusion between walking-in-place and true walking is given by methods that move the floor under the user. These systems allow the user to take physical steps while remaining in the same place. In [45], Souman *et al.* describe the CyberWalk omni-directional treadmill that is used for immersive virtual environments. Iwata *et al.*, in [46], develop the CirculaFloor alternative to a omni-directional treadmills. The CirculaFloor consists of several flat, rectangular robotic pads that rearrange themselves under the user's feet to produce the same effect as an omni-directional treadmill but without needing a special building or room to house the treadmill. A low-cost alternative to these setups is the VirtuSphere system [47]. The VirtuSphere uses a 10-foot hollow sphere that the user walks within. The sphere rests atop a set of roller wheels, so that the user's steps only cause the sphere to rotate in-place instead of moving about the physical room.

A promising new form of locomotion in immersive virtual environments is *redirected*

walking. Originally described by Razzaque *et al.* in 2001 [15], redirected walking uses small carefully orchestrated manipulations of the visual display prompting the user to unconsciously make corrective adjustments to their walking. Done correctly this can make it possible to redirect the user to walking in a circle even though they believe they are walking in a straight path.

This is only the briefest overview of the many locomotion methods available within immersive virtual environments. The fact that there are so many methods points to the largest unsolved problem that there is no best solution. Natural walking is preferable to walking-in-place or disorienting flying motions, but requires large spaces to work effectively. Redirection provides a way to decrease the size needed, but the exact threshold required is still quite large, by some estimates as much as 40 $\times$ 40 meters [48]. Also, redirection is still a walking motion and covers distance gradually. For large immersive virtual environments some combination of a faster travel methods, such as flying or seven league boots, with redirection will be required to effectively explore the space. The exact combination, and its implications on the required workspace size, are an unsolved problem.

2.4 Summary

Each new generation of virtual reality hardware has brought with it more accurate data, either in the form of tracking accuracy or visual realism, and lowered the latency of the system. Latency is very critical as too great of a latency creates a disconnect between user motions and the update of the virtual environment. This can lead to a "swimming" sensation in the best case, and severe cyber-sickness in the worst case. Latency and the effects of motion/visual disconnect will be discussed further in subsequent chapters.

3

Enhanced Immersion

Virtual environments have become increasingly realistic in recent years, especially those experienced through head mounted displays (HMDs). However, a major disadvantage of HMDs is that any view of the real world is completely occluded, including the user's view of his or her own body. This can be quite disorienting when navigating a virtual environment. One solution is to track the user's body and create a virtual avatar within the virtual environment. This allows the user to see a generic body in place of his or her own and generally reduces disorientation and increases presence as shown by Slater and Usoh in [10]. This works well for some situations, but it is still not the user's own body, and thus is not as realistic as it could be. Furthermore, in some situations this may hurt the sense of presence more than if no body was rendered, for instance if the body is re-targeted incorrectly resulting in limbs that are out of place or proportioned incorrectly. When interacting in immersive virtual environments several groups have shown that even a partial avatar embodiment can enable enhanced task performance within the environment [30, 49, 50]. It has also been shown, by Ries *et al.* in [51], that the avatar fidelity and quality can have a beneficial effect on task performance. We would like to determine if a self-avatar, one that is a near-perfect replica of the user, increases task performance even further.

The purpose of this research is to use a video see-thru HMD (VSTHMD), which has cameras mounted to the outer-front of the visor, to capture video of the real-world as it would be seen by the user wearing the display. Then, using computer vision techniques, we will segment the user's body (hands, feet, arms, and legs) from the

captured video and composite them into the rendering of the virtual environment. This will allow the user to see his or her own body within the virtual environment and may, in theory, increase the sense of presence the user feels within such an environment, an effect describe by Slater *et al.* in [10]. Also, if we have separate video for each eye, and can locate the user's hands within the video from each camera, we would have the potential to use stereo vision to determine the hands' approximate location in three dimensions. By using the segmentation of the user's hands to determine stereo we could essentially side-step the general form of the stereo correspondence problem. This localization would afford interaction within the virtual environment without the need for additional complexity of a separate hand tracking system.

There are a few problems, however, keeping us from jumping right in and studying interaction with such a VSTHMD. Currently, commercially available video-based see-thru HMDs are prohibitively expensive, costing tens of thousands of dollars. Also, there is often no calibration of the video cameras from the factory, especially for viewing position, causing the composited video to look like a 2d cutout and breaking any sense of presence. The calibration criteria needed for accurate perception is discussed in the following sections and in great detail by State *et al.* in [52]. Simple HMDs, however, are relatively cheap because of their application as a portable television, see for example the Sony HMZ-T1 which costs $800, and can be modified into VSTHMD while taking into account the camera calibration issues. Because of this cost differential, we first investigated building our own VSTHMD using the HMD our lab already owned. Once we have a VSTHMD, we must develop an algorithm that is capable of robustly identifying pixels belonging to the user's body in the video streams coming from the cameras in the head-mounted display. This is a generalized form of the background/foreground segmentation problem from computer vision. Our case is interesting because both the camera, attached to the user's head, and the foreground consisting of the user's body are moving arbitrarily in 3D, which makes the problem inherently intractable unless additional information can be gathered. Thankfully, there are several assumptions we can make that simplify the problem. These design considerations, the assumptions we can make, and the hardware development of our VSTHMD will be discussed in the Section 3.2.1.

First, we would like to take some time to discuss the related prior work, before

moving on to an overview of our assumptions, hardware, and design considerations. After that, in Section 3.2 we will describe early results we have obtained using multiple histogram modeling of foreground and background color distributions. We will also discuss the difficulties of using only 2D cameras to do background segmentation and show results using 3D structured light sensors in Section 3.3, and time-of-flight RGB+D sensors in Section 3.4, that allow robust detection of the user's body in three dimensions and eliminate many of difficulties in segmentation. The work using webcams was originally published as [53] in the workshop on Off-The-Shelf VR at IEEE VR 2011, it was later expanded upon using the Kinect sensors, Section 3.3, and published as [54] in the International Journal of VR.

3.1 Background

3.1.1 See-Thru HMD

In order to create the proposed video-based self-avatars we will be using a see-thru head mounted display. There are two varieties of see-thru HMD available, optical and video. In an optical see-thru display the backing of the LCD panel is removed and light from outside the HMD is allowed to pass through into the eye of the user. This allows computer graphics to be displayed overlaid onto the view of the real-world. The downside to optical devices is that it is impossible for the LCD pixels to completely block the light from the real-world and so the virtual environment always appears translucent. This is not desired for our proposed self-avatar application.

The second type of see-thru HMD are the video-based variety. In this setup a standard HMD is outfitted with cameras that can view the real-world. The computer application can then decide how to composite the video from the real-world with the generated view of the virtual environment. This has the benefit of allowing the complete occlusion of the real-world if desired, since the light reaching the eye is entirely under computer control.

However, the placement of the cameras can lead to downsides not present with the optical see-thru HMDs. For instance it is very important that the cameras capture the view of the real-world from as near as possible the same vantage point as the user's own eyes, otherwise the stereoscopic cues will mismatch between the virtual and real

environments. In [52], State *et al.* describe the development of a video-based see-thru HMD that uses a system of attached mirrors in order to align the camera optical centers and focal length with that of the wearer's eyes. The system was designed using 3D CAD software and 3D printed for use in augmented reality medical applications. One major practical problem with the proposed design was that the mirrors needed to be open to the world so that they could accept incoming light, but that also left them accessible to people's hands. This resulted in people continually touching the mirrors when putting the HMD on and moving them out of alignment or smearing the mirror surface with fingerprints.

3.1.2 Video Segmentation

A critical step in any camera-based avatar system is the analysis of the camera data for regions of interest. In the case of avatars the human body needs to be located in the image and separated from the rest of the image data; this process is known as image *segmentation*. This process is often modeled as a task of finding a binary mask image (M) that separates the area of interest, known as the *foreground*, from the rest of the image, called the *background*. During segmentation pixels within M are assigned a value of 1 if they are foreground or 0 if they are background. The segmentation can then be described as,

$$I = M \cdot F + (1 - M) \cdot B,$$

where I is the input image from the camera, F and B are the foreground and background images, respectively, and $\cdot$ is the per-element multiplication operator.

There has been a massive amount of work on image segmentation in the image processing and computer vision communities. The work usually deals with single-image segmentation, segmentation from a stationary video camera, or segmentation from a moving video camera. The objects of interest, which are referred to as the foreground, are usually moving with respect to the background. Examples of this are cars on a freeway, or people moving through an airport. Our situation then is a blend of several of these areas that computer vision usually focuses on. Depending on how the user moves, the video from the VSTHMD can be thought of as coming from a moving or a nearly stationary camera. Also dependent on the user's motion, the arms and feet can

be moving or stationary with respect to the background. In the end we will have to blend techniques and ideas from these different use cases in order to fit our situation. We will now give an overview of a small portion of the segmentation literature, and point out things that we believe will help us achieve our goals.

The most basic form of image segmentation is that of image differencing. If the background is static, and known a priori, foreground objects can be inferred by computing the difference between the current and background images. This algorithm is highly sensitive to changes in illumination and objects that appear similarly colored as the background. In applications where the camera and the background are static and unmoving, the background image can be computed as the average of several frames of video. Several frames are needed in averaging to remove image noise typically caused by camera white-balancing and real-time compression. We have a 3D model of our VR lab room that we will use in Section 3.2.2 to generate a rendered background image.

Human skin segmentation in images and videos has been studied for some time. In our application one of the main parts of the user that we wish to segment is the hands and arms, which if wearing a short sleeved shirt, will be skin colored. We will then investigate the use of skin segmentation as part of our approach. The results from our investigations of skin segmentation will be discussed in the Sections 3.2.3 and 3.2.4 of this paper.

In the survey paper [55], it is noted that skin detection is largely unaffected by choice of colorspace. This is important because it eliminates a possible step in our image processing pipeline. Also, they show through several tests that Bayesian based classifiers using histograms (such as the one described in [56]) are the most effective at segmenting human skin tones, and this is where began our implementation research in Section 3.2.

3.1.3 Distance Perception in VR

Virtual reality has many applications in architectural design review, scientific visualization, and task training. A key component that determines the effectiveness of these applications is the that the user perceives the virtual environment correctly and believes the view that is being presented to them of the virtual environment. In 1993, Henry and Furness first investigated the differences in distances perception across multiple

viewing conditions commonly encountered in virtual reality. Their results found that when comparing egocentric distance estimations, that is the subjective distance measured from a human observer to an object, people significantly underestimated distances in the virtual environment [57]. In the 20 years that followed many other researchers confirmed this finding and made some progress in determining its cause. In [58], Renner *et al.* survey the work to date and determine that, on average, individuals in virtual environments only report distances at about 75% of the modeled distance. This is an interesting result because Waller found that when allocentric distances, i.e. distances between objects as measured by a human observer, are studied it is found that the measurements are similar to those done in the real world [59].

One of the goals of this dissertation is to examine the effects of a video-based self-avatar on egocentric distance perception. It is our hypothesis that a video-based avatar that one that matches the user's movements and appearance much more closely than a 3d rendered and tracked avatar will improve the accuracy of egocentric distance estimations. There is some prior work that supports this idea. The study by Ries *et al.* showed that using a 3d rendered and tracked avatar improved distance estimates, while an avatar made of moving spheres did not improve distance estimates [51]. Ries *et al.* hypothesize that this improvement in accuracy was the result of an increased sense of presence created by the tracked avatar. However, this is in contrast to studies by Leyrer *et al.* [60], Lin *et al.* [61], and McManus *et al.* [62] neither of whom found any significant improvement on distance estimation when a tracked avatar was used.

Measuring egocentric distances is a critical component to understanding the differences between the real and the virtual world. When using an HMD to view the virtual environment the standard way to measure distances is blind walking. In this case the participant looks at a virtual object some distance from themselves, they then press a button or signal to the experimenter to begin the test. This disables the HMD screens and leaves them blind. They then walk to where they believe the object is, stop, and the distance is recorded. After this, the participant is led away from the place they have stopped along a circuitous route back to the starting point. Then, the HMD displays can be turned back on and a second trial measurement attempted.

When it is impossible to walk the entire distance from the starting point to the object, because the actual space is too small, blind imagined walking can be used.

In this case the steps are the same only instead of walking the participant imagines walking, and sometimes walks in place, and signals when to stop. Blind walking can also be replicated easily in the real world by having the participant walk with his or her eyes closed. Another alternative to blind imagined walking is triangulated walking. In triangulated walking the participant is asked to face the object and judge its distance. When ready they turn ninety degrees, the hmd is blanked, and they begin walking. When the experimenter stops them after a predefined, but unknown to the participant, amount of walking they must turn with the HMD still turned off, and face the object. The implicit angle estimation can be used to compute an estimate of the initial distance from the object.

3.1.4 Video-Based Avatars

With this abundance of research in the computer vision field on image segmentation and 3D reconstruction there are many groups working within the VR community to create real-time, camera-based avatars. A key difference between many of the works presented below and ours is that our system focuses on cameras mounted to the HMD and not mounted externally around the available tracked physical space. This is because our goal is to allow users to be able to see themselves from a point of view that is as close as possible to that of their own eyes. When using room-mounted cameras it is difficult to obtain sufficiently detailed, high-quality images from the user's point of view since often the user's body and head will be occluding the view of the hands.

Petit *et al.* in [63] introduce the GrImage system that uses a room completely covered in green-screen material to simplify the background subtraction process. Multiple RGB cameras placed around the room then perform chroma-key segmentation and reconstruct a mesh of the observed scene (users and objects) along with texture information. The 3D shape information is used to give the users a self-avatar and compute collisions with virtual objects. Both the self-avatar and the physics of collision act together to give the user an increased immersion in the virtual environment. The advantages of this approach are that it gives a 3D volume of the user from only 2D camera images, and the cameras are room-mounted and can be accurately calibrated offline. The disadvantage is that the room must be purpose built entirely with green-screen material in order for reliable segmentation.

In [64], Bruder *et al.* use a head-mounted camera and background subtraction via color-segmentation in order to give the user a self-avatar of their arms and legs. Offline calibration is used to determine the user's skin color as it appears to the camera as well as that of the floor, and two segmentation models are trained depending on whether the user is looking at their hands or their feet. The system performs brightness compensation on the input video that is then overlaid on top of the virtual scene. Because the avatar is 2D video it cannot be reprojected to match the center of view of the virtual camera.

At the 2014 Consumer Electronics Show, Alexx Henry Studios demonstrated their xxArray System that uses more than 68 cameras mounted around a person to capture an avatar [65]. The avatar is created by segmenting the megapixel images from the camera and computing the mesh from the silhouettes. The resulting mesh can contain over 100k polygons. The mesh is fitted to a skeleton and can be immediately animated in 3D games or animation software. The work of Wiess, Hirshberg, and Black [66] uses RGB+D cameras also mounted around a person to generate a body scan. The scanned body model is morph-able, in addition to being pose-able, so that, for example, different meshes can be generated based on body type or fitness level for a user from a single set of images.

The previous applications all make use of one or more RGB camera and triangulation is used when 3D data is needed. With the recent appearance of affordable depth sensing cameras (e.g. PrimeSense & Kinect) the availability of 3D data, without the need for PC processing, has greatly increased. Suma, Krum and Bolas in [67] demonstrate a system using a PrimeSense camera mounted on an HMD that allows a user to see and interact with other users and objects sharing the same available physical space. The user cannot see their own hands or feet because of the range limits of the camera.

The work of Maimone and Fuchs [68] shows how multiple Kinect sensors can be used to obtain overlapping depth data. The core idea is to attach a vibrating pager motor to each Kinect. Because of the differences in oscillations each Kinect will vibrate out-of-sync with the others. This will cause the projected IR pattern from the other Kinect cameras to appear blurry, since the camera shutter is not synchronized to the other projectors, and will reduce the interference caused by overlapping patterns.

In [69], Maimone *et al.* describe a telepresence system using multiple Microsoft

Kinect cameras to perform 3D scene capture. The paper describes a technique to calibrate each Kinect for errors in color and depth so that the resulting merged 3D data is consistent and blends smoothly between the regions covered by different cameras. The entire system is implemented as a GPU processing pipeline that can perform data merging, hole-filling, color correction and surface generation at rates of up to 200 million triangles per second. The viewer of the telepresence system uses an autostereoscopic display with an additional Kinect for head-tracking thus requiring no shutter glasses or markers for viewing.

One issue with the above work is that there is still noise present in the image since each image in the video is merged independently of the others. In a follow-up work by Dou & Fuchs [70], a temporal enhancement method is demonstrated that can reconstruct the 3D dynamic scene and can compensate for noise, occlusion and dynamic moving objects while still generating a complete 3D reconstruction of the room. The limit of the system described is that it currently only runs on the CPU, not GPU, and is therefore not real-time.

In [71], Beck *et al.* present a telepresence system using multiple Kinect depth cameras and a novel 6-user wall-sized autostereoscopic display. The system can use pre-calibrated and registered Kinect cameras along with real-time tracked Kinect cameras that enable the two groups to change the reconstructed area at run-time. The image data is compressed after processing, which reduces network congestion, and temporally filtering is used to smooth the depth images. The low resolution of the depth image resulted in some visual artifacts, but a user study revealed that the result was precise enough to understand gestures and interact with the other group.

3.1.5 Relative advantages & disadvantages of video-based avatars

Our self-avatar solution is inspired by the approach of Bruder *et al.* [64] and Suma *et al.* [67], in that we wish to use color and depth data from head-worn sensors to build the self-avatar. Our end goal is a system that can work in any room, regardless of the color, shape, or layout of that room, and that can work for feet and hands without any active mode switching on the part of the user.

Our proposed system uses camera(s) mounted onto the HMD that can view the real world and the user's body from a vantage point very near the user's eyes. The proposed

video self-avatar does not need any tracking suit or devices attached the user's hands or feet and is entirely contained in the head-worn apparatus.

Because the cameras are so near the user's eyes they have the advantage of creating an image that exactly matches the user's appearance at the exact time they are in the virtual environment. This means the virtual avatar will match in skin color, clothing and movement. This is advantageous to create a sense of presence in the virtual environment, and useful when the goal of the virtual application is the environment or the interaction with the environment. In some psychological or military training applications the goal may require that the avatar not match the user. For instance, when treating Post-Traumatic Stress Disorder (PTSD) it may be desirable for the avatar to wear a military uniform.

Another potential problem with the proposed video based system is that the avatar geometry only exists when it is being viewed by the cameras. While this poses no problem for a first-person system, as the camera frustum overlaps the user's view frustum, it would be an issue in a multi-user setting where it is required to see the avatars of the other participants. It is, however, a performance benefit since we are not wasting rendering resources on avatar geometry that is not visible to the user.

3.2 Video-based self-avatars using head-worn off-the-shelf webcams

3.2.1 Assumptions & Hardware

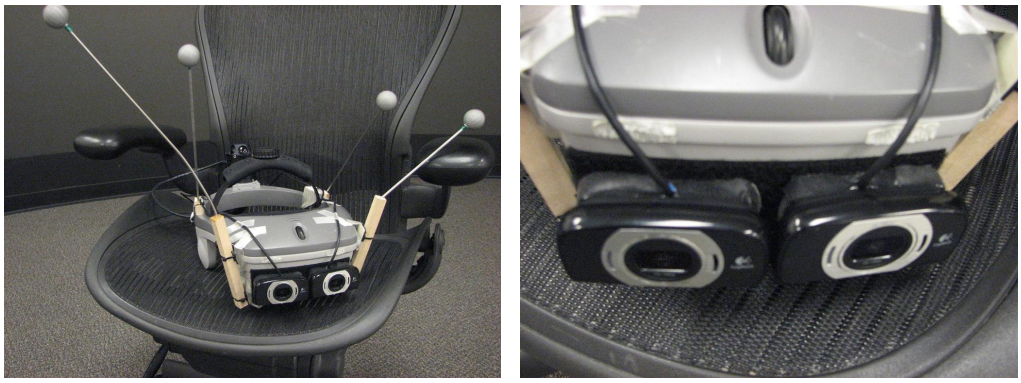
As discussed, when the camera and the user (i.e. the foreground) are both moving arbitrarily there is very little in general that we can be sure about. Thankfully, however, there are some assumptions that we can make. The first assumption is that this VR system will always be used in a room that we know about ahead of time. This allows us to use an offline training phase where we can learn various image and geometric statistics about the room. The second assumption that we make is the user's head, and by extension the HMD and cameras attached to it, are tracked with 6 degrees of freedom within this room. Since this technology is to be used along with an immersive virtual environment that requires this tracking, it is a reasonable assumption. The third

assumption that we can acquire some information about the user in advance of them using the system. Currently we do this by taking a few seconds of video of the user looking at their hands and feet while wearing the VSTHMD. This video is then used in a short calibration step prior to the system use. In the examples shown below the virtual environment we use is a replica of our real-world lab rendered in a non-photo-realistic black-and-white style.

In our lab we already have an SX60 HMD manufactured by NVIS. This HMD can display 1280×1024 resolution stereoscopic images to the user using two Liquid Crystal on Silicon (LCOS) displays, and has a Field of View (FOV) of 60 degrees diagonally. We therefore wanted to find cameras that could match these display properties. What we decided on was the Logitech C615 USB webcam that can capture 1920×1080 video at 24fps with a FOV of 74 degrees diagonally. Two of these webcams are mounted onto the SX60 HMD using velcro that has been attached to the webcams with hot melt glue. The glue is applied to the back of the webcams, and then sanded down in order to give a smooth surface with which to apply the velcro and also act as an alignment guide to aim the webcams. The final result is shown in Figure 3.1(a) with detail of the attachment shown in Figure 3.1(b). The images are captured from the cameras using the OpenCV library over USB 2.0. Once in our program, they are cropped, resized, and rendered onto an OpenGL quad to display on the HMD screens. The exact amount of cropping is determined via overlaying the video with a rendering of our lab room model and then adjusting by hand until the images match.

The webcams themselves have a fairly short depth-of-field, and as a result if the user looks rapidly around the room the image will become blurred while the camera autofocus attempts to adjust to the scene. Using a camera with a different lens, and a camera with a faster shutter and auto-focus response would alleviate these problems, but the additional weight and size could make mounting the cameras aligned with the eyes and wearing them more difficult. A system of mirrors, as demonstrated by State *et al.* in [52], would be one way to incorporate larger cameras without negatively affecting the weight distribution and alignment of the cameras.

The velcro was used to attach the webcams in the hope that they could be moved to adjust for different inter-pupillary distances between users. However, using velcro has made it difficult to align the cameras with the lab room model for longer than a



(a) COTS VSTHMD (b) COTS VSTHMD attachment detail.

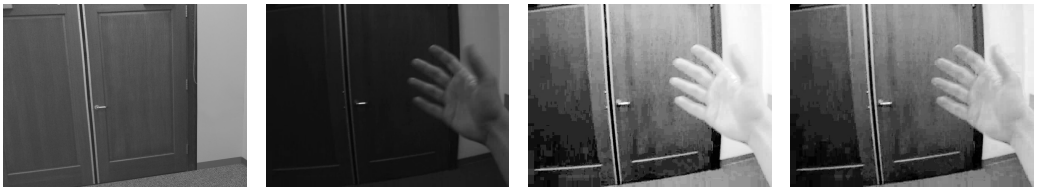
Figure 3.1: The prototype Commercial-Off-The-Shelf Video See-Thru HMD built using two USB webcams attached to our existing NVIS SX60 HMD.

few hours of use, since they have a tendency to move slightly during motion of the user's head. The development of a different attachment mechanism is therefore a highly desired future project.

3.2.2 Image Differencing

The first method we tried was one of simple frame differencing between a rendered scene of the 3D model of our room and the live video. The motivation for this is that any difference between the rendered room model and the video will show up as foreground. As a result, anything not included precisely in the room model, such as chairs and desks, will show as foreground. To work around this we could in the future use a Microsoft Kinect, or similar device, to obtain a room model that includes the furniture instead of the hand-built model we are currently using. The difference was computed as the absolute value of the difference between the rendered image and the video image, in the grayscale color space. A threshold was then applied and any pixels with an error larger than this threshold were marked as foreground pixels. Histogram equalization was used to normalize the range of values between the two images before differencing. The histogram equalization was tested both in the grayscale space, and in the Value channel (of HSV) before conversion to grayscale. By performing the equalization on Value alone, the hope was that the Hue would remain the same; something that cannot

be guaranteed if the equalization is done in grayscale. We show in Figure 3.2(d) the result of both types of equalization to a sample image. Median filtering was done after the difference was computed in an attempt to reduce the number of spuriously labeled foreground pixels. A result of this method is shown in Figure 3.3. This figure shows the image from the rendered 3D model of the room along with the input video frame, and the virtual environment into which the user will be composited . The results show the gray mask before thresholding, and the result after thresholding the gray mask and composition. Notice that the wrist is missing from the final composition because a threshold low enough to include the wrist would also include large sections of the door since the door appears brighter in the gray mask than the wrist area.



(a) Rendered room model in grayscale (b) Captured video in grayscale (c) Captured video with grayscale equalization (d) Captured video with Value equalization

Figure 3.2: The results of histogram equalization for the purposes of contrast and brightness correction. The histogram equalization is shown here performed in grayscale (c) and on the Value channel of an HSV image (d). In both cases, the equalized video more closely matches the rendered image of the room model.

(a) Input video (b) Input IVE (c) Result gray mask (d) Result

Figure 3.3: The results of simple frame differencing when histogram equalization is performed on the Value channel of the input video. The hand and wrist are lost because the intensity difference between the hand and the wall is too low relative to the other differences in the image due to lighting inconsistencies, as well as the fact that the comparison is done in grayscale instead of full RGB colorspace.

The results of this were promising, as the hand and several objects in the room, such as chairs, which were not present in the 3D model, were roughly segmented correctly. However, areas where the lighting in the 3D model differed significantly from the real room proved troublesome as well as colors that would map to similar grayscale values. Another problem with this method, and with any method relying on our 3D model of the room, is the alignment between the rendering of the model and the live video. Since the assumption of image differencing is that the same object will appear in the same place in the image it requires perfect alignment that is unachievable due to camera video latency. Currently, we are just relying on the hardware design of the VSTHMD to align the video, and are not performing any rectification or additional alignment in software. There is also the issue of temporal alignment to deal with. Since the tracker updates at close to 1000fps, and the camera only updates at 20–30fps, the position where we render the room is not synchronized with when we capture the frame from the camera. Because of the difference in update rates the quality of the segmentation will suffer when the viewing direction is changing rapidly. Managing this temporal alignment by delaying or queuing the tracker data would help alleviate this problem to some extent. When investigating the potential of this approach we did not make such an adjustment. Based on these observations, we decided that frame differencing is only a partially successful solution to the problem that is why we needed to turn to the more involved approach inspired by the work of Bruder *et al.* [64] discussed in the next section.

3.2.3 Color Classification — Single Background Histogram

Using our COTS VSTHMD we next investigated segmentation based on skin color. We first implemented the algorithm of Jones & Rehg [56] that computes RGB histograms of the foreground (skin) and background (everything else) in an offline training phase. The foreground and back histograms are normalized to give estimates of the probability distributions $P(x|skin)$ and $P(x|\neg skin)$. These distributions, along the with probability of skin or not skin in the training data, are then used at run-time to compute the probability that a given pixel x corresponds to skin using,

$$P(skin) = \frac{P(x|skin)P(skin)}{P(x|skin)P(skin) + P(x|\neg skin)P(\neg skin)}.$$

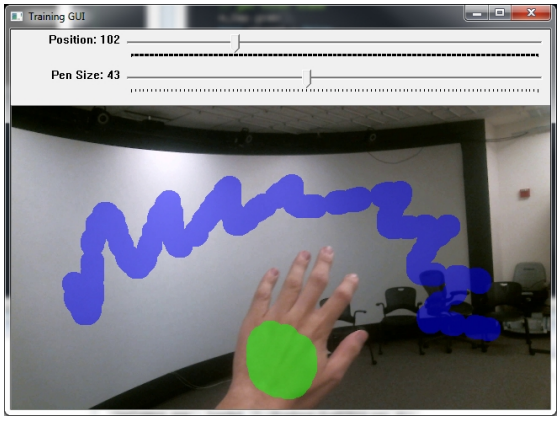
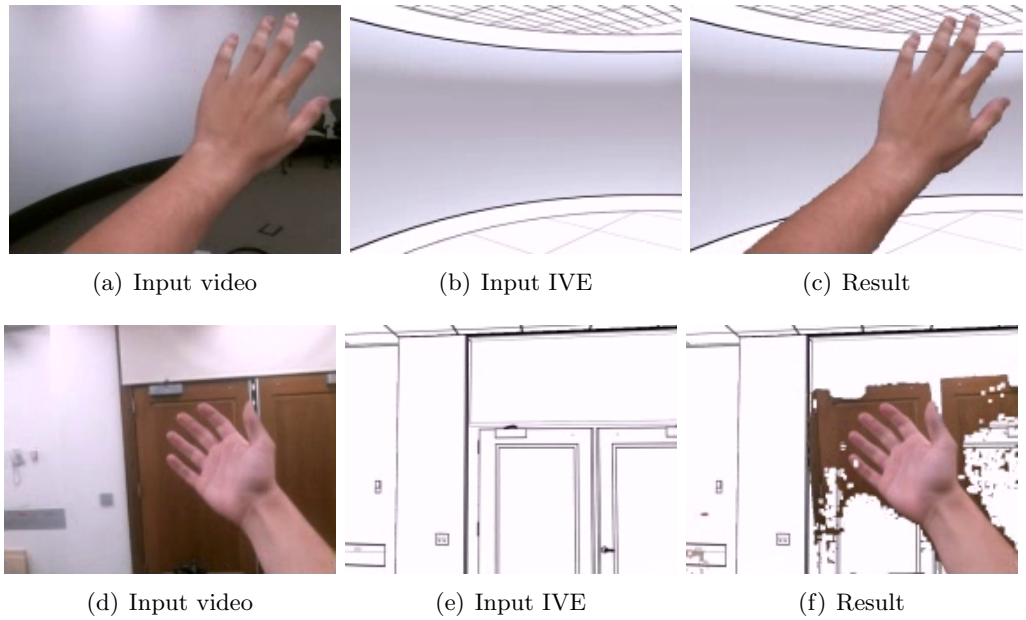


Figure 3.4: Foreground and background color calibration for a single background histogram and single foreground histogram model. The background is labeled as blue pixels, and the foreground as green pixels.

Pixels with a probability above a threshold θ are marked as being skin. We also looked at Hue/Saturation (HS) histograms, as well as Hue-only (H) histograms for classification. The algorithm remained the same in these cases, only the color and histogram space were changed.

The first step is to obtain training pixels that can be used to create the histograms needed for classification. To this end we developed a graphical user interface, shown in Figure 3.4, that allows a user to scan through a video file and highlight skin and background pixels by drawing on them with the mouse. These pixel labelings are then used to create the foreground and background histograms.

Results from this color based segmentation for two different frames of video are shown in Figure 3.5. These figures show the raw masks for a threshold of $\theta = 0.4$. Also shown is the result of applying a morphological hole closing operation to the mask, in order to fill in any gaps that may be present in the mask in an attempt to create a nicer looking mask. In our trials, RGB histograms performed the best with HS histograms a close second. The reduction to only Hue appeared to lose too much information, as background pixels were very often misclassified as skin. The HS histograms use less memory than the RGB histograms, yet appear to be very similar in classification accuracy. This might be an important fact to consider if we attempt GPU optimization in the future, and want to store and access these histograms on the GPU itself.



(a) Input video (b) Input IVE (c) Result

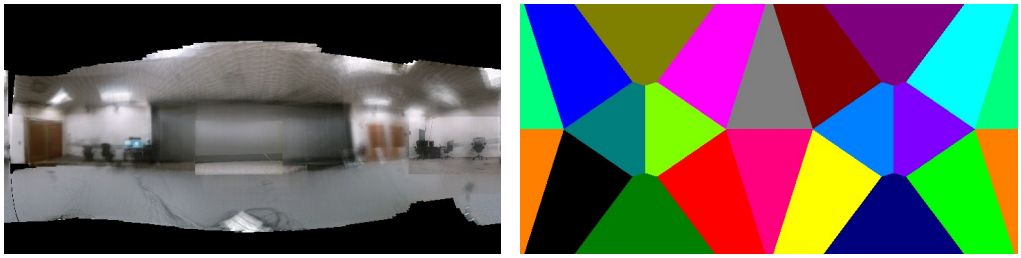
(d) Input video (e) Input IVE (f) Result

Figure 3.5: Results of segmentation using the single background histogram and single foreground histogram model. Morphological closure was performed on the binary masks in order to reduce holes in the segmented regions. The result (b,d,f) shows numerous false positives due to the wooden door in the background.

3.2.4 Color Classification — Multiple Background Histograms

The case shown on the right of Figure 3.5 shows when the user looks at the doors to our lab. Since the doors are wooden, they match closely the color of the user's hand and as a result the segmentation contains many false foreground positives. In an attempt to eliminate this issue we investigated the use of multiple background histograms each stored according to the direction the user is facing. The run-time background histogram is then interpolated from the stored background histograms using the user's current facing direction to determine the interpolation weights. In order to learn these background histograms, we use a captured video of the entire room. We also capture and store the tracker data simultaneously during this process. In order to make sure each area of the room is equally weighted, we first build a spherical environment map from the captured video and tracker data as shown in Figure 3.6(a). The environment map is then sectioned into multiple regions, one for each of the background histograms.

The current sectioning we are using is based on the vertices of a regular dodecahedron and is shown in Figure 3.6(b), however, other sectioning is possible. One other potential sectioning is to give more regions along the 0 degree elevation since that is where the majority of viewing will take place. As can be seen in Figure 3.7, using multiple histograms produces superior results to the single histogram case when run on the same input.



(a) (b)

Figure 3.6: (a) Spherical environment map of our lab created from a tracked video capture. No video frames were captured for the regions at $\pm 90^{\circ}$ elevation in this particular. (b) Regions of the environment map. Each region corresponds to a different histogram in the background model.

(a) Input video 1 (b) Result 1 (c) Input video 2 (d) Result 2

Figure 3.7: Results of using a multiple background histogram model on the same input video as Figure 3.5. No IVE was rendered in this test, to better show the segmented regions.

3.3 Video-based self-avatars using fixed external RGB+D sensors

Up until this point our investigations had all been using the images from the HMD cameras and segmenting them in 2D. This next section discusses our work on performing the segmentation using information gathered from a Microsoft Kinect sensor about the user in world coordinates. This information is then processed in order to determine which sections of the image possibly contain the user's body as a way to filter spuriously segmented pixels.

3.3.1 Setup & Calibration

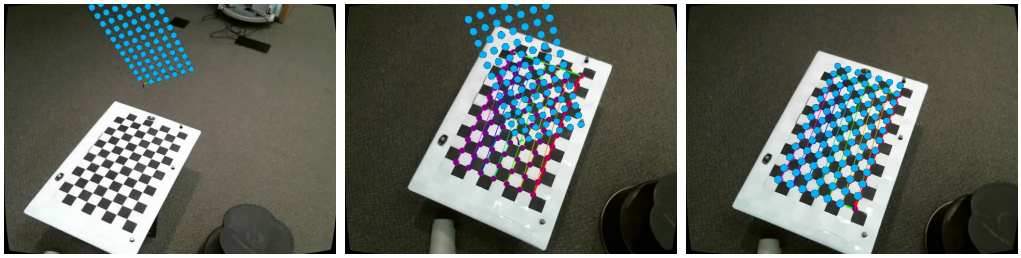
For this series of experiments we have used the Vicon tracking system in our lab. It consists of 12 Vicon MX cameras that provide six degree-of-freedom tracking of the objects needed for the experiment. We track the HMD, the Kinect, and a large chessboard pattern, which can be seen in Figure 3.8. We used Microsoft Kinect SDK v1.5 for these experiments, which gave us skeletal and depth data in meters with respect to the Kinect's coordinate frame. The tracking via the Vicon allowed us to transform this Kinect data into our virtual environment's world space.

The critical step to making the Kinect, Vicon, HMD, and cameras perform well together is to calibrate everything into a global world frame. The Vicon tracker space and the room were calibrated together upon installation, so we did not need to worry about this. The transformation from the Kinect to the world was manually chosen by selecting a point near the Kinect's depth camera origin. For the cameras mounted on the HMD we need to calibrate the camera intrinsic parameters (focal length, principal point, aspect ratio) and extrinsic parameters (rotation and translation in world frame). The intrinsic parameters were calibrated using multiple images of a chessboard pattern that was processed using the OpenCV library. The chessboard corners were re-projected into the image as a sanity check to ensure the intrinsic parameters found were valid. This was done for each camera separately. As a result of the calibration, OpenCV also gives a transformation ${}^{Ci}_{Ch}T$ from the chessboard to the camera coordinate frame. Using the Vicon we also have transformations from the chessboard to the world, ${}^{W}_{Ch}T$, and from the HMD to the world, ${}^{W}_{H}T$. Together we can then find the location of the camera

origin with respect to the HMD,

$$ {}^{H}_{Ci}T = \text{inv}\left({}^{W}_{H}T\right) \cdot {}^{W}_{Ch}T \cdot \text{inv}\left({}^{Ci}_{Ch}T\right), $$

where inv$(\cdot)$ is the operation that reverses the direction of the transformation ($i.e$ inv$\left({}^{H}_{Ci}T\right) = {}^{Ci}_{H}T$). This allows us to compute the camera positions very precisely at every frame. Using these intrinsic and extrinsic parameters we can then compute OpenGL projection and worldview matrices that will render the virtual environment to match what is seen from the camera, similar to how these matrices are found in augmented reality. The calibration process of the extrinsic parameters can be seen in Figure 3.8, and results of the alignment between the camera and the virtual environment can be seen in Figure 3.9.

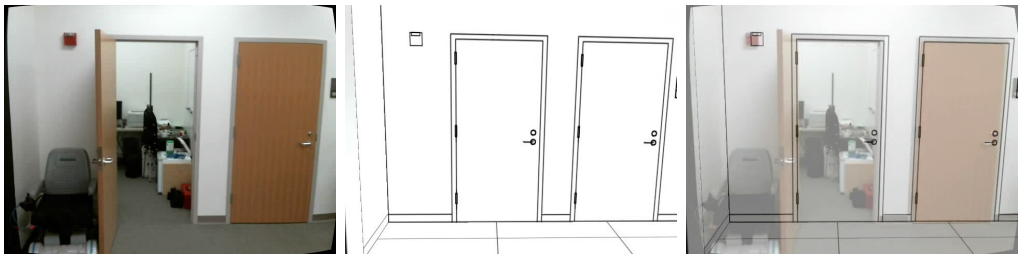


(a) Before Calibration (b) Chessboard Detection (c) After Calibration

Figure 3.8: These images show the calibration process of the HMD cameras extrinsic parameters. (a) shows what the user sees before the calibration, note that the grid of spheres does not align with the chessboard. (b) shows the overlay generated by OpenCV when it detects the chessboard pattern. The circles show the reprojected corner points as a sanity check. (c) shows the alignment after calibration that is triggered by pressing a key on the computer keyboard once OpenCV detects the chessboard. The spheres are in the world coordinate system and move with the chessboard based on the Vicon tracking data. When they are aligned the user knows that the calibration was successful from the visual feedback provided by the spheres.

3.3.2 Usage & Results

Our goal was to have multiple Kinect sensors covering the entirety of our room, but for initial testing we had a single Kinect that covered approximately a six square meter area. As such for the user's body to be tracked, the user needed to stand in front of



(a) Camera Input (b) VR Environment (c) Overlay

Figure 3.9: Here is shown (a) the image captured by one of the HMD cameras, (b) the rendering of the virtual environment, which in this case is a non-photo realistic model of our room, and (c) an alpha-blended overlay of the two images to show how well they align after the camera intrinsic and extrinsic parameters have been calibrated.

and facing the Kinect as shown in Figure 3.10. With the system properly calibrated we were able to transform the Kinect data into the world space and render it in our virtual environment. We tried rendering the skeleton as a series of cylinders, as shown in Figures 3.11 and 3.12, or rendering the raw color pixels at their appropriate depth as shown in Figure 3.13 and Figure 3.14. A third solution we tried was rendering the depth pixels and then segmenting the portion of the rendered image corresponding to the Kinect data and replacing it with the images captured from the cameras. We expanded the area rendered by the kinect data in order to create the segmentation mask to attempt capturing the entirety of the user's body. The result of this is shown is shown in Figure 3.15. We have also shown in the figures the results when the camera image data is segmented further using the color histogram methods discussed earlier in this paper. This focuses the color segmentation only onto the areas that the Kinect has identified as containing the user's body, thereby eliminating many of the spurious pixels caused by the doors. This color histogram segmentation shown in the figures was computed in a post-process offline step in a separate experimental codebranch, and as a result lacks the compositing into the immersive virtual environment.

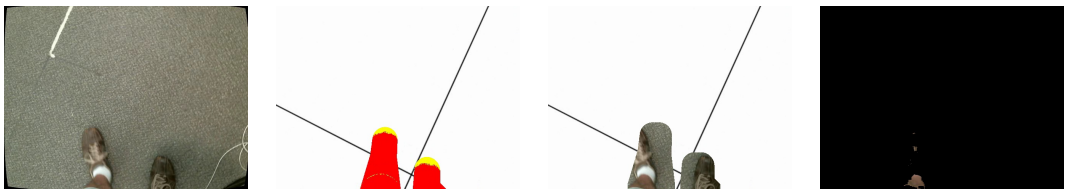
3.3.3 Discussion

The rendering of the hands and feet, since they are very close the cameras, is extremely sensitive to any errors in the calibration of the system and errors in the measurement.

(a) HMD with camera detail

(b) User facing Kinect

Figure 3.10: The user must stand within the Kinect's field of view in order for the segmentation to be effective. Currently we have a single Kinect, as shown here, but future work is to use this system with multiple Kinects that collectively cover the volume of the room.



(a) Camera Image

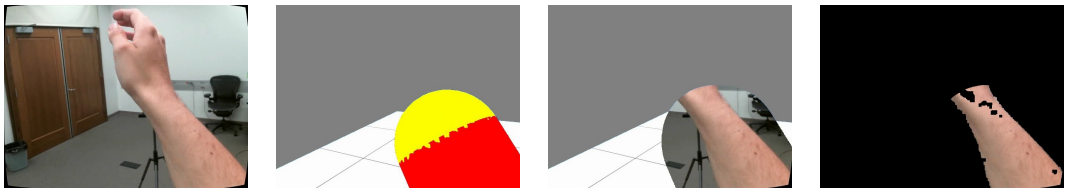
(b) VR Render

(c) Segment Kinect Only

(d) Segment Kinect & Color

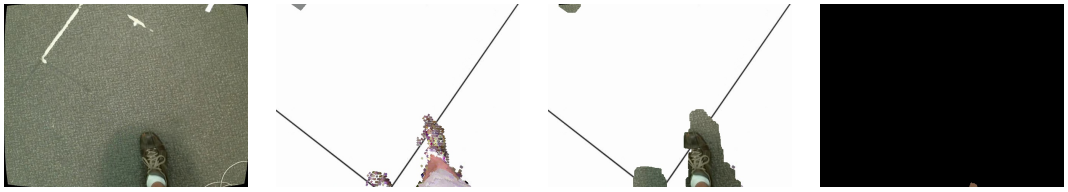
Figure 3.11: Here we show the results when the Kinect tracked skeleton is used to segment the user. In this example the user is looking at his feet. The segmentation of Kinect & color was done in an offline step for proof-of-concept and does not have the virtual environment rendered.

As can be seen in the result figures, a consequence of this is that the entire hand or foot in the camera image is never perfectly aligned with the Kinect rendering. This is most likely because of our current manual calibration of the Kinect's coordinate frame origin with respect to the Vicon. An area of future research is to develop a method similar to the one used for the HMD cameras to calibrate the Kinect precisely with respect to the Vicon coordinate frame. Another interesting area of future research is the selective enabling of the color segmentation. For instance when the user is looking at his or her feet, the Kinect data alone is almost enough to correctly segment the legs and feet without the need of color segmentation. Also, because of the numerous colors



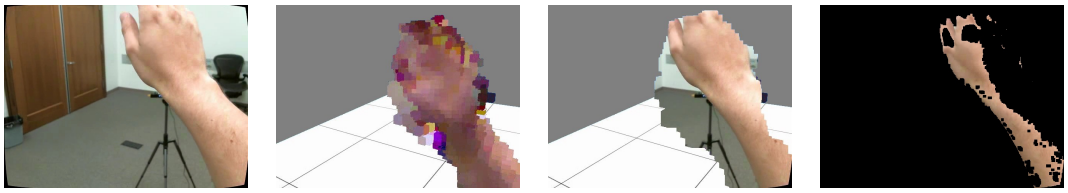
(a) Camera Image (b) VR Render (c) Segment Kinect Only (d) Segment Kinect & Color

Figure 3.12: Here we show the results when the Kinect tracked skeleton is used to segment the user. In this example the user is looking at his hand. The segmentation of Kinect & color was done in an offline step for proof-of-concept and does not have the virtual environment rendered.



(a) Camera Image (b) VR Render (c) Segment Kinect Only (d) Segment Kinect & Color

Figure 3.13: Here we show the results when the depth image is used to segment the user. In this example the user is looking at his feet. The segmentation of Kinect & color was done in an offline step for proof-of-concept and does not have the virtual environment rendered.



(a) Camera Image (b) VR Render (c) Segment Kinect Only (d) Segment Kinect & Color

Figure 3.14: Here we show the results when the depth image is used to segment the user. In this example the user is looking at his hand. The segmentation of Kinect & color was done in an offline step for proof-of-concept and does not have the virtual environment rendered

in the clothing, the color segmentation of the feet and legs would become very complex. This could be skipped by relying on the Kinect for the legs and feet.

(a) Kinect Data (b) Expanded Mask

Figure 3.15: To generate the segmentation mask from the Kinect data we take the pixels rendered by the data and expand the region by performing a dilation operation. The result of such an operation is shown here.

3.4 Video-based self-avatars using head-worn RGB+D sensors

In the previous sections we discussed our attempts at creating a video-based self-avatar using head-worn webcams and fixed external Kinect depth cameras. Each of these approaches have downsides that have prevented large scale usage and user studies. The webcams, using color segmentation alone, did not function well in arbitrary physical environments. In rooms with areas of color similar to human skin (*e.g.* wooden doors) segmentation errors occur and the real physical space intrudes on the virtual environment. This problem can be avoided by using depth cameras and performing the segmentation using depth data instead of color as we showed with the Kinect. The externally-placed, fixed-position depth cameras, however, have their own set of downfalls. The first is that covering a large physical space will require multiple depth cameras. The data from these sensors will need to be synchronized and combined using a centralized server and then communicated to the system rendering the IVE. This introduces network overheard, processing overhead and additional system complexity. Depending on the level of accuracy needed in the combined reconstruction this can easily become a non-real-time task [70]. Also, when the depth cameras are far away from the user any errors in camera rotation calibration are magnified over the long distance from the camera to the user. The effective resolution is also diminished as shown in the previous section where the user's hands and feet take a very small area of the depth image and therefore appear

pixelated and noisy both in color (RGB) and depth (D).

The solution we propose is to make use of recent advancements in infrared time-of-flight depth sensors, that have created smaller, lighter depth sensors with shorter ranges. The Kinect has an operating range of 1.2m to 3.5m making it impossible to wear on an head-mounted sensor unit and still be able to detect the user's body, particularly his or her hands. We propose to use the Intel Creative Senz3D [72] camera, which has a resolution of 1280×720 RGB, 320×240 depth and an operating range of 0.15m to 1m, making it much more applicable to mounting on a head-mounted sensor unit. Using a head-worn depth sensing camera will alleviate the need for centralized processing, in theory multiple user's could use a video-based self-avatar of this kind by wearing a backpack with a laptop running the IVE and a battery powered HMD and camera. The communication between the multiple user's IVE systems would be minimal, and only needed to synchronize state changes for the world as local depth data could be used to give self-avatars as well as avatars of other users within the camera's 1m range. Also, since the working range of the camera is much closer to the user's head and hands, the user's hands and body will occupy a larger portion of the image and appear at higher resolution and with less depth and angular noise than the Kinect based system described earlier.

The basic idea is that we will mount a depth camera and a color camera on the front of our nVisor SX60 HMD. In the case of the Senz3D camera the color and depth cameras are in a single package. We can then calibrate the cameras as a stereo pair in order to reconstruct the point cloud seen by the camera in world coordinates. By rendering this point cloud as a mesh of triangles we will be able to generate a stereoscopic view from the exact point-of-view of the person wearing the HMD and thereby avoid any appearance of a 2D cutout common to see-thru systems that have the camera origin in a different optical location than that of the user's eye.

3.4.1 Early results using pmd[vision] Camboard Nano

For early testing we used a pmd[vision] CamBoard nano [73] depth camera as the Intel camera was not readily available. This is a time-of-flight depth camera, with no in-built RGB camera, and a resolution of 160×120 pixels. Since this was an early test to see what the depth data looked like we calibrated the camera location manually using a

ruler relative to the HMD eye position. The depth data can then be converted to 3D coordinates using the camera's factory intrinsic calibration. Some typical result images from this testing can be seen in Figure 3.16.

(a) With Filtering (b) Without Filtering (c) Hole From Jewelry

Figure 3.16: These images show the early results of depth based avatars using the pmd[vision] CamBoard nano depth camera. (a) shows a good results using the simple filtering algorithm, and (b) shows without filtering. (c) shows a hole in the geometry caused by the user's watch reflecting the infrared light. Notice that in all images the boundary of the hand is not smooth due to noise in the data and the limited resolution of 160×120 of the CamBoard nano

Infrared depth cameras, like the CamBoard nano or the Senz3D, have several types of noise in the final image. Not only will there be noise in the RGB colorspace, but there will be noise in the depth data. Individual pixels, if rendered directly, will appear to randomly move small amounts towards and away from the depth camera between frames. Since the depth camera may not share an optical axis with the rendering direction, this can result in movements perpendicular to the rendering direction further exacerbating the appearance of the noise. Infrared is also easily absorbed by hair and certain types of clothing and reflected by jewelry both resulting in holes in the mesh where the depth cannot be measured. Figure 3.16 shows some examples of the type of errors we have seen in our studies with the CamBoard nano sensor.

Some noise in the image can be fixed by using Gaussian blurring as a post-processing step, and small holes can be filled using morphological operations (dilation and erosion) as shown previously with the webcam data. Errant pixel data can be removed by implementing rejection filters in the depth data that will discard polygons if, after conversion to 3D, the polygons exceed a threshold in terms of the length of the largest side. This filter has been implemented and the results shown in Figure 3.16. This filter

is not the best solution and a better variant will be discussed below with the Senz3D camera.

3.4.2 Replacement camera: Creative / Intel Senz3D

The Senz3D Camera is a joint collaboration between Intel and Creative. It offers two cameras, one infrared time-of-flight and one RGB, in a single enclosure along with stereo microphones. It has a higher resolution than the CamBoard nano and with the addition of RGB sensing can also capture the color texture along with the depth information. It also claims to have better noise characteristics than the CamBoard nano.

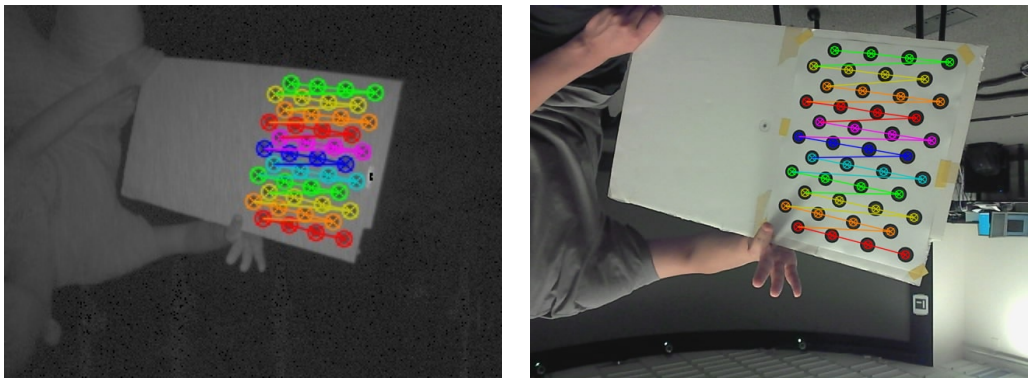
We chose a very simple technique to mount the camera to the HMD. On the bottom of the camera is a clip that is designed to hold the camera to the top of a flat-panel LCD monitor. However, fully extended the clip becomes a flat arm that we used to attach the camera to the HMD by way of double-sided tape. We also mounted the camera upside down on the HMD in order to place it as close as possible to the optical axis of the wearer's eyes. The final attachment to the HMD can be seen in Figure 3.17.

Figure 3.17: This figure shows the final mounting of the Senz3D camera to the NVIS HMD. The camera is attached using double-sided tape and is upside down in order to place it as close as possible to the wearer's line of sight.

3.4.3 Camera Calibration

The Senz3D camera consists of two cameras paired together: a color (RGB) camera, and a infrared depth camera. The system as a whole can be calibrated as a stereo camera

pair using the OpenCV library. Care must be taken to ensure that the calibration is done allowing for different focal lengths, principal axis, and image resolutions as the two cameras are physically and optically different from one another. We performed the calibration using the OpenCV asymmetric circle grid calibration pattern, because we found it to give better solutions, and was easier to detect in the infrared image, than the popular checkerboard pattern. Once the calibration was complete we had camera matrices (M_c and M_d) and distortion models for both color and depth cameras, as well as rotation and translation (R and t) from the depth to the color camera. Extrinsic calibration of the depth camera location was performed in the same manner as used with the webcams in Section 3.3.1. The location of the color camera could be computed from the extrinsic location of the depth camera and the rotation and translation matrices of the stereo pair. Figure 3.18 shows the result of finding the circle grid pattern during calibration.



(a) IR image (b) Color image

Figure 3.18: Finding the calibration pattern in the video from the depth camera stereo pair.

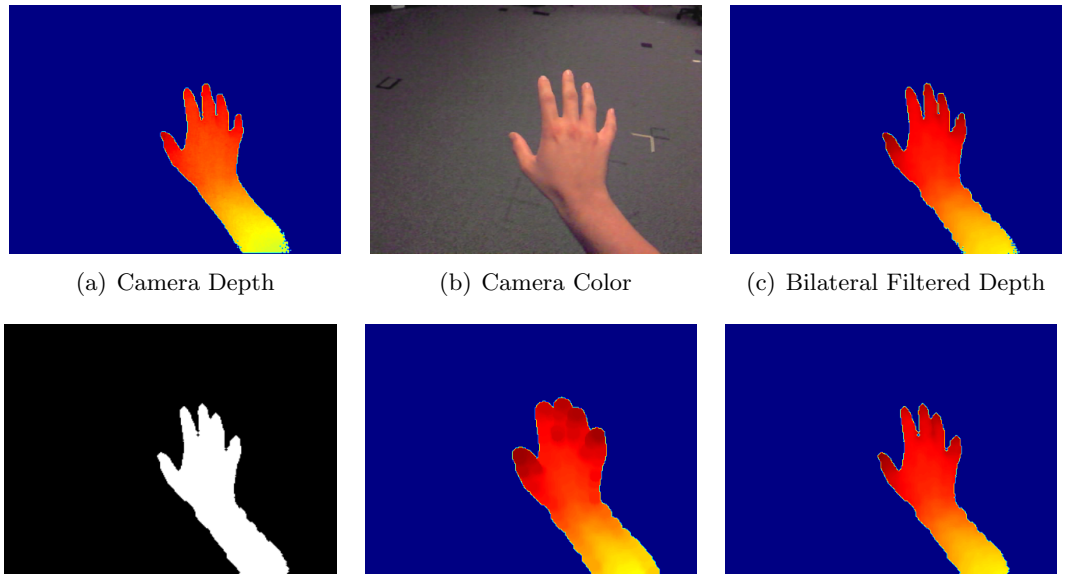
3.4.4 Depth data pipeline

Given a calibration we can then process the camera frames to compute the mesh geometry and texture to be rendering in the immersive virtual environment. The first step of the pipeline is a bilateral filtering of the depth data. We found that the depth data tends to have a small amount of noise, less than a few centimeters at each location, and

when this noise is rendered onto the hands it can cause a very strange appearance of the skin of the hand moving. The bilateral filtering smooths out this noise and significantly reduces its effect. A second error we need to remove is missing depth data, due to reflective surfaces or Vicon interference, and salt-and-pepper noise in the depth data from objects that are just out of range or have specular reflections. Some objects that we observed causing these types of errors were particular types of carpet (as seen in Figure 3.20(a)), smoothly painted walls, and jewelry. These errors are removed through the use of morphological operations on the thresholded depth mask, as shown in 3.20(d). Because this clean-up may fill holes and cause the mask to contain points that have no depth data, the depth data itself is interpolated using dilation in order to approximate the missing depth at these points. The filtered mask and interpolated depth are then combined to give the final depth result used for segmentation and mesh construction. The mesh itself is computed using the camera calibration, cleaned depth data, and camera color image. It is rendered as an actual mesh in the virtual environment from the point of view of the user and virtual camera so as to compensate for any offset between the camera optical axis and the optical axis of the user's gaze. Examples of intermediary results from this pipeline can be seen in Figures 3.19 and 3.20.

There were two types of errors that we were unable to remove in the pipeline and had to remedy in another fashion. The first stems from the requirement that the users be able to see their own feet. For depth segmentation we use a single threshold value that is computed automatically. However, this didn't work when viewing the feet as the single value would always leave part of the floor in the foreground or part of the user's feet in the background. Neither situation is desirable. A practical solution of placing black felt cloth on the floor that absorbs all infrared light was found to hide the floor and allow us to focus on the hand-segmentation problem.

The second issue was the camera's use of infrared light and the interaction with the Vicon tracking system used to track the user's head location. If looking directly at the Vicon cameras, as shown in Figure 3.21, the Senz3D camera would be overwhelmed by the infrared from the Vicon and show points of depth that would randomly move from dark (far away) to bright (close). Also, if the user's hands were angled in a particular way this effect would also happen with the reflection of the Vicon camera IR off the user's skin even without the Vicon cameras in the view of the Senz3D. We found that reducing



(a) Camera Depth (b) Camera Color (c) Bilateral Filtered Depth

(d) Depth Mask after Morphological Operations (e) Dilated Depth (f) Final Depth Result

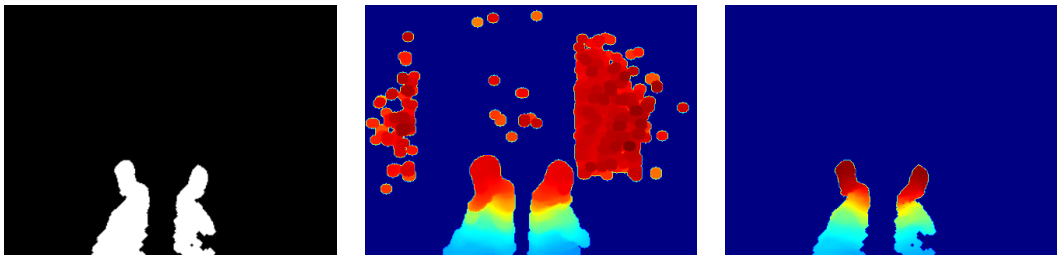
Figure 3.19: Example of processing pipeline for a single view of the hand.

the brightness of the Vicon camera IR to 10% of normal eliminated the problems from reflection. We also aimed the Senz3D camera towards the floor instead of level with the horizon to limit the occurrence of direct views of the Vicon camera, which are mounted towards the ceiling of our lab space.

3.5 Evaluation: Do video-based self-avatars affect egocentric distance perception?

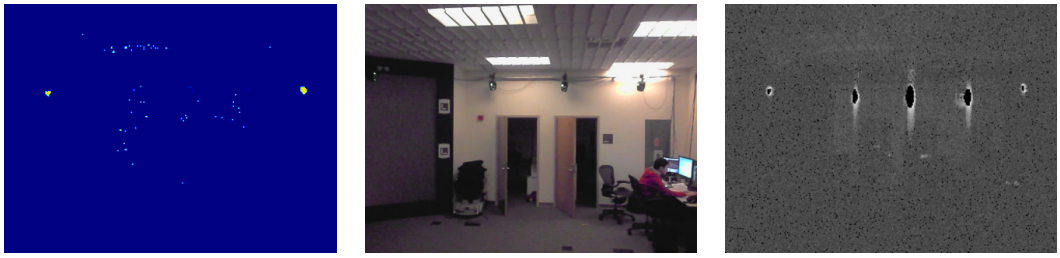
Now that we have a method that can reliably create a video-based self-avatar we would like to investigate the usefulness of such an avatar. In previous experiments at the University of Minnesota it has been found that having an avatar reduces the amount of egocentric distance compression present in virtual reality compared to the same situation with no avatar [74]. What we want to know is whether having a video-based avatar, which will be of higher fidelity than a pre-modeled generic avatar, reduces the amount of egocentric distance compression even further.

(a) Camera Depth (b) Camera Color (c) Bilateral Filtered Depth



(d) Depth Mask after Morphological Operations (e) Dilated Depth (f) Final Depth Result

Figure 3.20: Example of processing pipeline for single view of the feet.



(a) Camera Depth (b) Camera Color (c) Camera IR

Figure 3.21: Example frame demonstrating the interference of the Vicon tracking system and the Time-of-Flight depth camera, notice the "holes" in the camera IR data, and corresponding bright spots of random depth, where the sensor is overwhelmed by the IR from the Vicon cameras.

Previous experiments at the University of Minnesota have shown that egocentric distance compression can be alleviated if the IVE matches the lab space in which the study is performed [75]. Because of this we chose to have this user study take place in a virtual hallway environment that does not match our lab space, this is shown in Figure 3.22(b). The user's head is tracked in our lab space by means of a group of 12

Vicon MX40 infrared optical tracking cameras. Their position in the room is mapped directly to a position in the virtual hallway, such that translations and rotations in the virtual hallway match those in the real room.

(a) Empty Desk Scene (b) Empty Hallway Scene

Figure 3.22: Example views of the two immersive virtual environments used in the user study.

To test the effect of the avatar we performed two experiments involving distance perception at long range (walking) and short range (reaching). Each experiment was run with three conditions: no avatar, Vicon motion-capture controlled generic avatar, and our video-based self-avatar. Both experiments shared the same set of subjects and each subject performed both reaching and walking tasks during the same session in the IVE and the real-world. It is our hypothesis that the trials using the video-based self-avatar, which more accurately matches the participant's own body than that of the pre-modeled generic avatar, will exhibit smaller amounts of egocentric distance compression than either the generic avatar or no avatar conditions.

3.5.1 Participants

Twenty-four subjects were recruited from friends, colleagues, and students at the University of Minnesota. The participants ranged in age from 19 to 35 (mean = 24), and had normal or corrected-to-normal stereo vision. There were 17 men and 7 women, and 11 subjects had prior experience with VR either through tours or participation in previous, unrelated, studies conducted at our lab.

3.5.2 Methods

Equipment

In this experiment we used the NVIS nVisor SX60 HMD with 1280×1024 pixel resolution for each eye and a complete stereo overlap of 60 degrees diagonally. The experiment was implemented using the Unreal 4 [29] game engine, and ran on a computer with a 6-core Intel i7 processor and nVidia Titan Black graphics card. We implemented the Senz3D camera algorithm described above in Section 3.4.4 as a plugin to the Unreal engine in C++. OpenCV is used for handling the common image operations such as erosion and dilation. The entire program runs at 30fps which is the maximum framerate of the Senz3D camera.

Procedure — Walking

This experiment was a blind walking task. Upon arrival the participant was asked to stand in a predetermined area of the lab space, common to all participants, and put on the HMD. If they were to experience the motion captured avatar they would also be asked to wear the required motion capture suit and perform a short calibration routine prior to putting on the HMD. Once wearing the HMD they would see the hallway environment as shown in Figure 3.22(b).

A virtual piece of masking tape would then appear 8–12 inches in front of their current position and they would be tasked with lining their feet up to this mark. In all three conditions the tape would turn green if the participant was within 10 centimeters. This helped with lining up in the no avatar case. Once lined at the start the participant would look straight ahead and signal that they were ready.

At this time the experimenter would press a key making an end tape mark appear a predetermined distance away from the start mark. The participant could then take as long as he or she required to look at the end mark and around the IVE (see Figures 3.23, 3.24), and then signal they were ready to being blind walking. The participant would then close their eyes and attempt to walk to the end tape. At the same time the experimenter would press a key on the computer to blank the HMD screen and ensure that no optical flow cues could be used by the participant during blind walking.

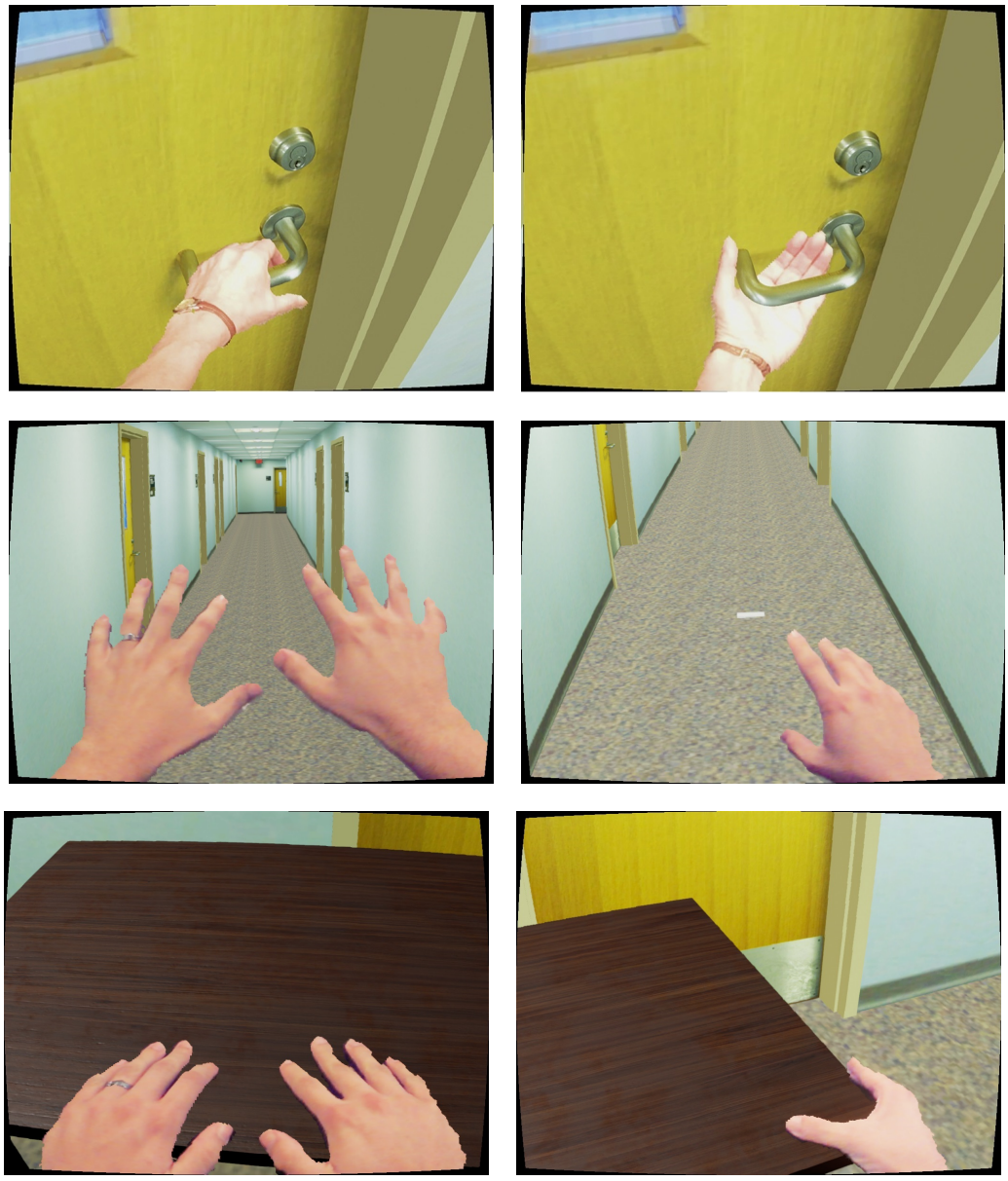
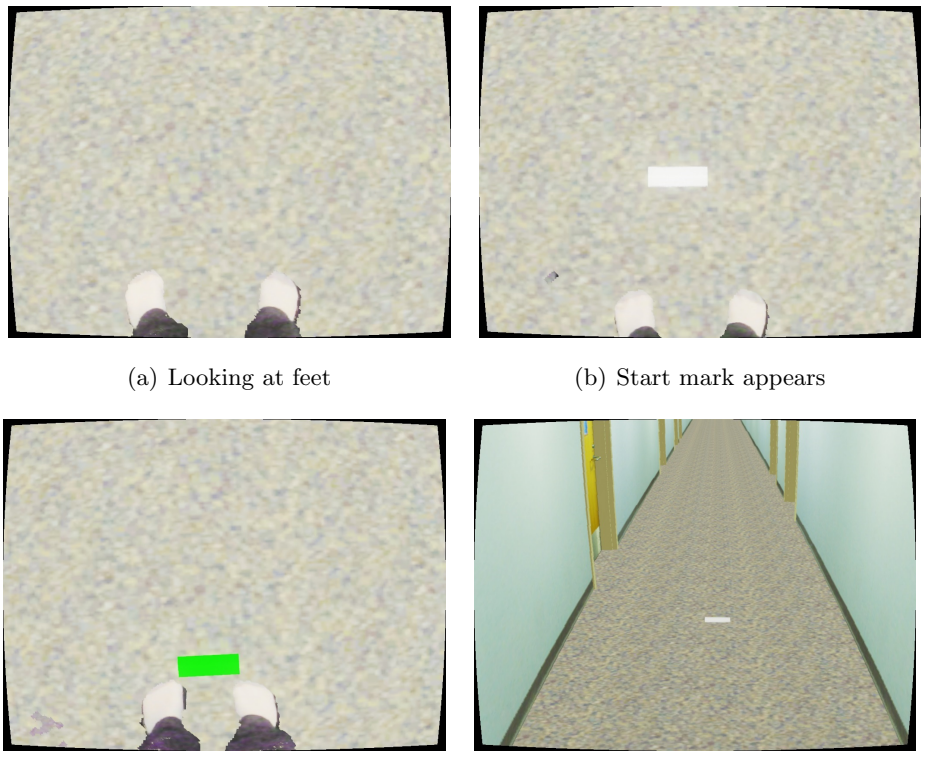


Figure 3.23: Example possible interactions within the IVE, when using an RGB+D video avatar, during the time when the HMD is displaying a view. It is important to notice that the video avatar renders with correct z-ordering with VR objects such as the table and door handle.

When the participant believed he or she had reached the end tape they would tell the experimenter who would then press a key to record the stopping point. The experimenter would then lead the participant back to the starting area, taking a circuitous route, before re-enabling the HMD display. This circuitous route limited the amount of learning of the virtual environment that could be done as the trials progressed by the participant.

There were five different distances that were the same for all participants and each was repeated three times in a random order. A throwaway trial at the mean distance was done first for all participants, bringing the total number of walking trials to 16.

Participants were given Kennedy-Lane Simulator Sickness Questionnaires (SSQs) [76] before and after the walking trials. After the walking trials in the IVE the participant would perform the second reaching trials in the virtual environment, before performing the equivalent walking and reaching tasks in the real world.



(a) Looking at feet

(b) Start mark appears

(c) At start mark

(d) End mark appears

Figure 3.24: Blind walking experiment procedure.

Procedure — Reaching

While the first experiment dealt with medium to far distance perception, we also want to test near-field distance perception. To do that we developed a second experiment that would test blind-reaching accuracy.

In this experiment the participant was seated at a desk in the lab space and their shoulders strapped back to a chair to ensure that they cannot lean forward and thus have consistent range of reach (see Figure 3.25). When they put on the HMD they are shown the same hallway environment, however this time there is a table in the hallway exactly matched with the table in the real-world. In this way they can place their hands on the virtual table and perceive passive haptic feedback of the real-world table. This is shown in Figure 3.25 for the case of the video-based avatar. The table was covered in black felt to absorb infrared radiation and render it invisible to the depth camera, since segmenting between the table and hand when they are touching is a challenging area of future research.

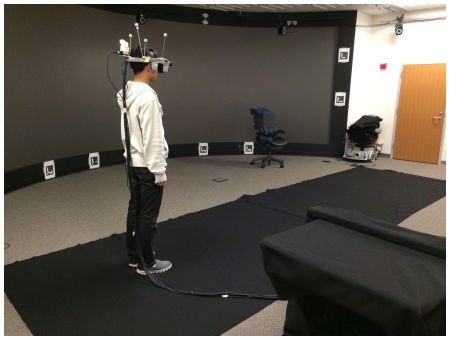
The participant was given time to look around the environment (see Figures 3.23, 3.26), and when they signaled they were ready the experimenter would press a key and a target tape mark would appear on the virtual table. Similar to the walking experiment, they were asked to look at the mark for as long as they needed, say "ready", close their eyes and attempt to touch the tape mark. When they thought they were touching the mark they needed to keep their finger on the table long enough for the experimenter to measure with a tape measure their finger position. The manual measurement was required since the finger was not tracked in any of the three avatar conditions.

There were five different distances, 50%, 60%, 70%, 80% and 90% of the participant's maximum reaching distance. Maximum reach distance was measured for each participant prior to putting on the HMD and entered into the IVE program. Each distance was repeated three times in a random order, and a throwaway trial at the mean distance (70%) was done first for all participants, bringing the total number of reaching trials to 16.

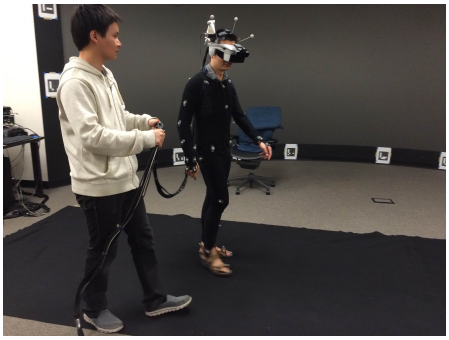
At the end of the trials the participant was given a second SSQ to fill out, and then performed the real-world equivalent trials.

(a) Participant at Desk (Vicon avatar)

(b) Participant reaching (Vicon avatar)



(c) Participant in Hallway (RGB+D avatar)



(d) Participant and Experimenter walking along hallway (Vicon avatar)

Figure 3.25: Photographs showing a participant in the video-based self-avatar user study.

Figure 3.26: Blind reaching experiment procedure.

Measures

For each trial either the computer system (for the virtual walking) or the experimenter (for all others) would record the target or viewed distance and the actual distance moved by the participant. These were then used to compute the relative error for each trial, defined as

$$e = \frac{A - V}{V}$$

where V is the viewed distance and A is the actual distance. For each subject and environment (IVE or real-world) relative errors lying more than 2 standard deviations from the mean were discarded. The mean relative error was then computed for real-world or VR cases and used to examine the hypothesis.

3.5.3 Results

The complete table of measured relative errors for these studies is available in Appendix A.

Subjective Evaluation

Upon completion of the experiment, participants were asked to fill out an exit survey asking to rate several questions on a 7 point scale from strongly agree at 1 to strongly disagree at 7. The survey in its entirety, along with results, can be found in the appendix. Among the responses, only two had near statistically significant differences in the answers among the three avatar conditions. When asked if it was as easy to perform blind reaching in the IVE as in the real-world, RGB+D avatar and no avatar on average responded "agree" while Vicon avatar was only "slightly agree". In the conditions with an avatar, participants were more likely to agree that they were "completely comfortable with the appearance" of their body in the IVE when using the RGB+D avatar as opposed to the Vicon avatar.

Simulator Sickness

Participants filled out simulator sickness questionnaires before starting the trials in the IVE and after completing both the blind walking and blind reaching trials. The total

scores showed a near significant result ($p = 0.07$) when examined using a pre/post-test ANCOVA, that the increase in total simulator sickness was the smallest with the RGB+D avatar, next the Vicon avatar, and most with no avatar. A plot of total simulator sickness score across all participants is shown in Figure 3.27.

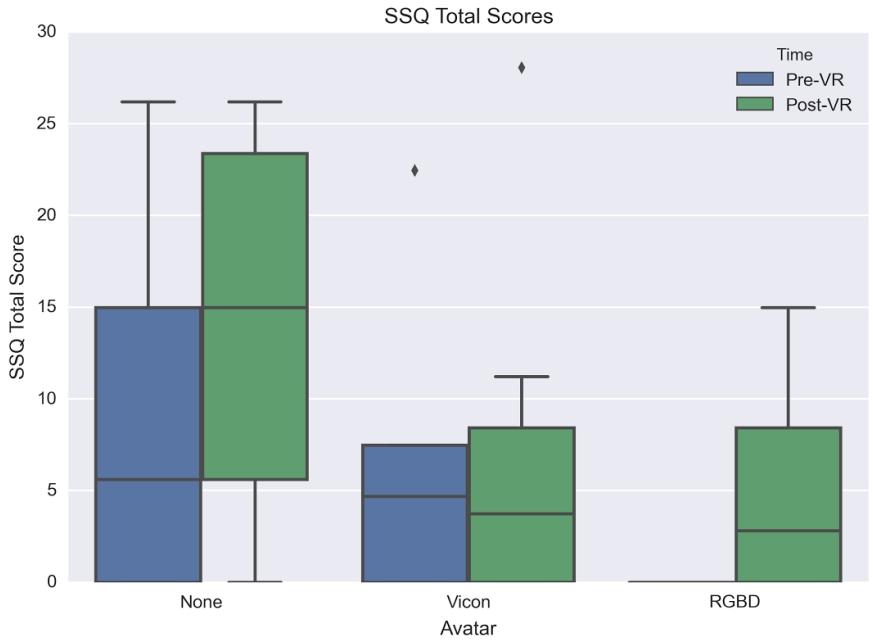


Figure 3.27: Boxplots of total simulator sickness scores for all participants between avatar condition and pre/post exposure to IVE. Notice that the increase in simulator sickness is smaller in both avatar conditions than in the no avatar condition. An ANCOVA of the data shows that this result has a near significance at $p = 0.07$.

Distance Estimation

Figures 3.28(a) and 3.29(a) show a boxplot of the mean error for each avatar condition for the blind walking and blind reaching experiments, respectively. In both cases, but particularly in the blind reaching case, the effect of outliers in the data can be seen. To alleviate this, we performed outlier selection and removal by computing the interquartile distance for the measures of each user and removing points outside this distance before computing the mean. We then performed outlier detection again on entire users, instead

of individual measures, before analyzing the data further using ANCOVA. An example of this outlier removal for individual measures is shown in Figure 3.30. The effect of this outlier removal on the boxplots can be seen in Figures 3.28 and 3.29 at each stage of the removal. The effect on the distribution of all samples is shown in Figures 3.31 and 3.32.

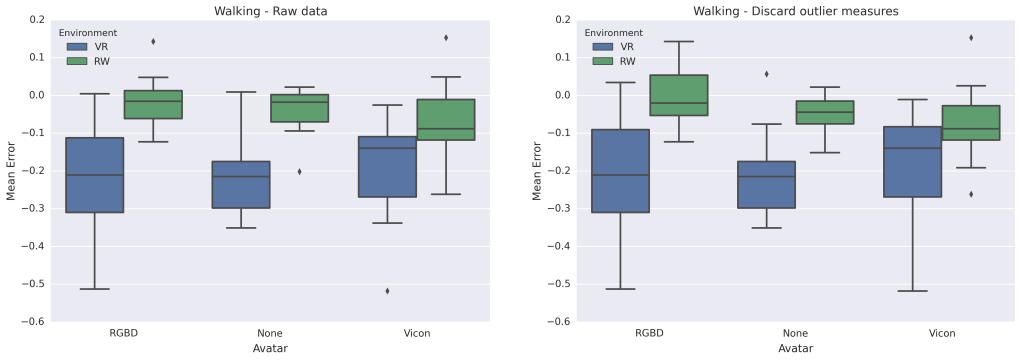
We performed an ANCOVA on the mean relative errors to test for statistical significance of differences between the avatar conditions. ANCOVA was chosen instead of an ANOVA because we were able to treat the real-world performance as a covariate of the experiment and remove the effect of differences between the users in base distance estimation ability. For walking, the ANCOVA gave a non-statistically significant p-value score of 0.887665 ($F_{(2,17)} = 0.12$). For blind reaching we found a a near-significant p-value of 0.051484 ($F_{(2,17)} = 3.55$).

3.5.4 Discussion & Conclusion

Our hypothesis before conducting these experiments was that a video-based avatar would have two effects on the user of the immersive virtual environment. First, we hypothesized that the difference between egocentric distance estimates in the virtual environment versus the real environment would decrease, and to a greater extent than when using the traditional Vicon motion-capture avatar. This hypothesis was motivated by the earlier research of Ries *et al.* [74] that showed the use of avatars to decrease the amount of egocentric distance underestimation. The second hypothesized effect was more subjective, as we believed users would feel more comfortable using the video-based self avatar than the alternatives.

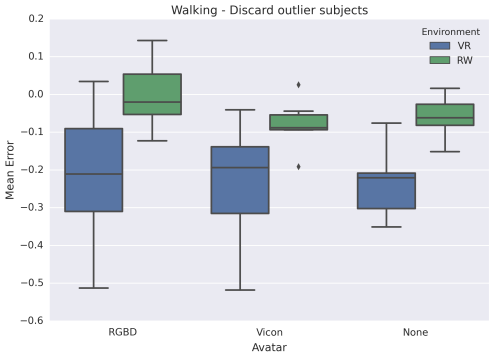
As discussed in the previous section, what we actually found was that there was no statistically significant difference in the amount of underestimation between the three avatar conditions when performing blind walking. When testing the near distance effect in the reaching study, we found that both Vicon avatar and RGB+D avatar reduced estimation error (p-value of 0.05) compared to no avatar, and Vicon reduces the error more than RGB+D avatar. However, it should be noted that the difference between the Vicon and RGB+D cases was not statistically significant.

These results support the theory that there are multiple factors influencing and causing the distance underestimation in virtual reality. They also suggest, at least in



(a)

(b)



(c)

Figure 3.28: Boxplots of blind walking mean relative error before and after each step of outlier removal.

the reaching case, that the effect of an avatar may not be a continuous linear parameter. It may be that there is a minimal needed avatar realism that triggers the improvement, and once that is reached more realistic avatars add little to the effect.

We found that our second hypothesis, that users prefer the RGB avatar, to be confirmed with statistical significance by analyzing the exit survey responses. The exact reason for this is not known, as there were no follow up questions in the exit survey. In the future, leaving open-ended questions on the exit survey would let us dive deeper into the reasons.

A surprising finding, especially given the higher latency of the RGB+D camera as compared to the high-speed Vicon cameras, is that we found that the user's SSQ scores measuring cybersickness increased the least when using the RGB+D avatar compared

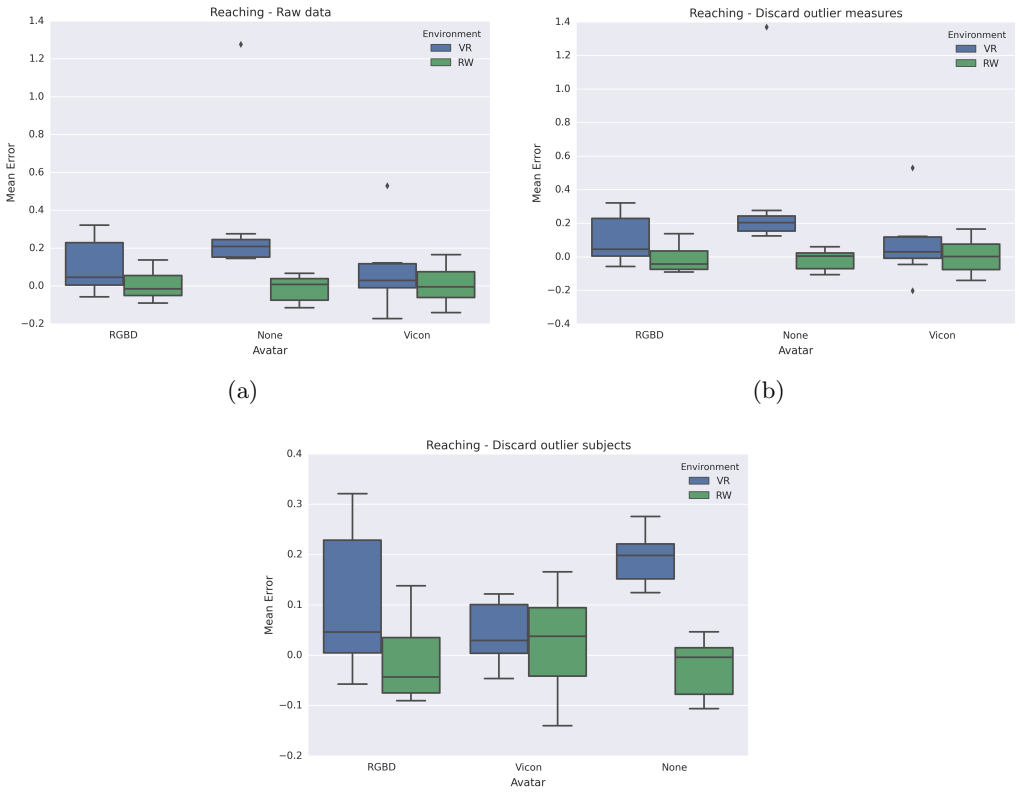
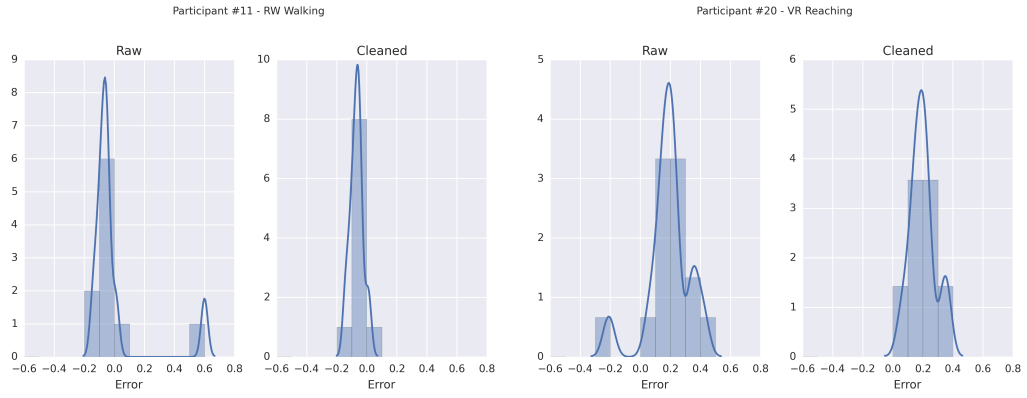


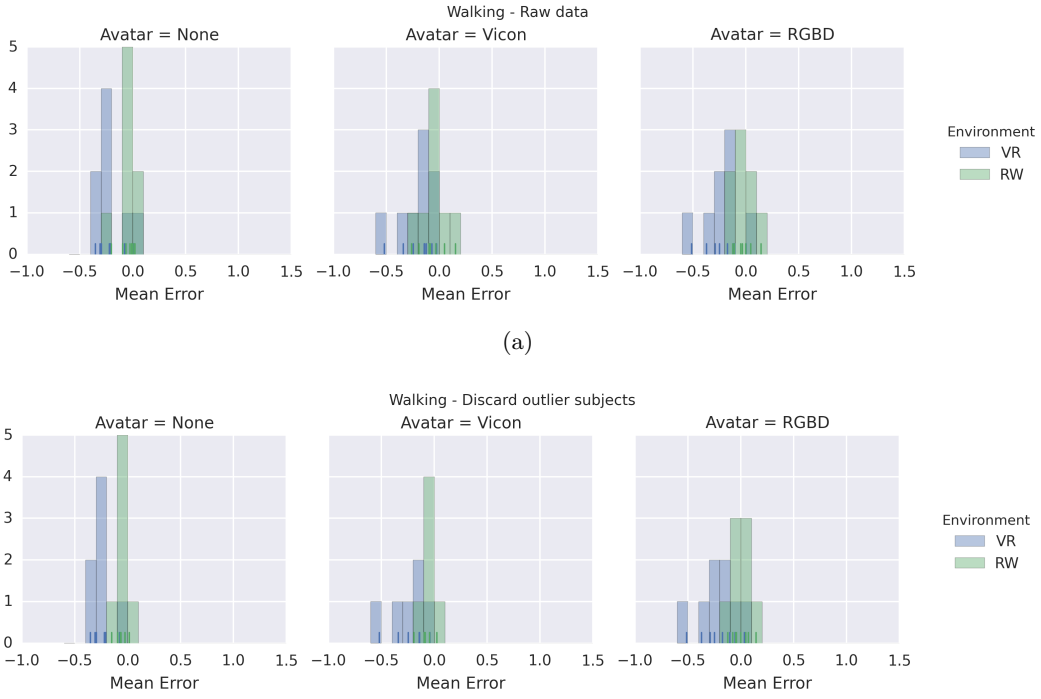
Figure 3.29: Boxplots of blind reaching mean relative error before and after each step of the outlier removal.

to baseline SSQ. A boxplot of the total SSQ scores for all participants is shown in Figure 3.27. We performed an ANCOVA in order to account for variability in the before VR responses and found that the difference in the means to be statistically significant. The avatar case with the greatest increase was no-avatar, followed by Vicon avatar and then RGB+D avatar with the lowest increase in cybersickness. This result suggests that latency is not solely responsible for physical cybersickness. It even suggests that the extra latency of the RGB+D camera, as compared to the no avatar condition, is somehow ignored by the user because of the effect of the realistic avatar. Further investigation into this result is warranted for future research, as it could lead to the development of methods to limit cybersickness.



(a) (b)

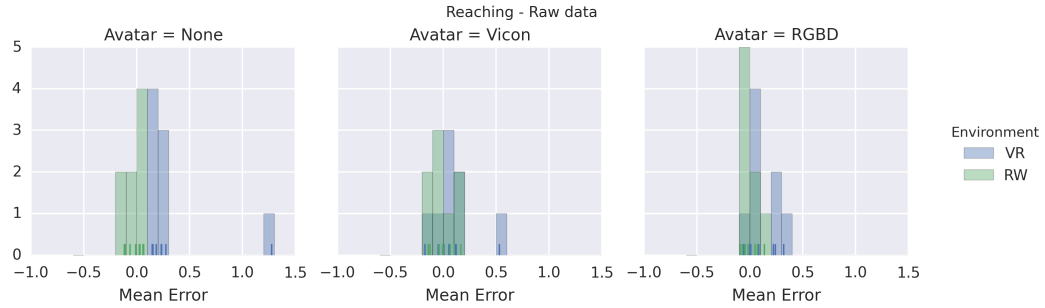
Figure 3.30: Results of removing outlier distance measures from the set of a user's measures before averaging into mean relative error used in the analysis.



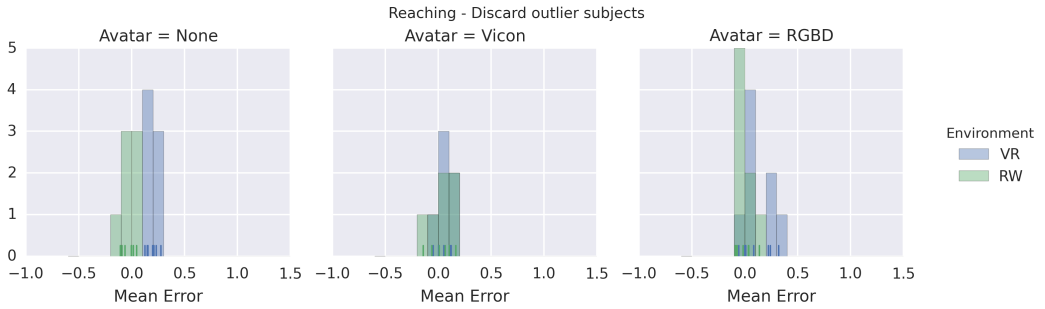
(a)

(b)

Figure 3.31: Distribution of the blind walking data before and after outlier subject removal.



(a)



(b)

Figure 3.32: Distribution of the blind reaching data before and after outlier subject removal.

3.6 General Discussion & Conclusion

The goal of this avenue of research was the creation of realistic first-person self-avatars using video cameras. We chose to investigate the use of video cameras because we believed it would allow for self-avatars to be constructed without the need to wear motion capture suits or other tracking markers, and allow for realistic avatars that exactly match the appearance of the user. This is possible since the VR depiction would be generated from photographs, the individual frames of video, of the user. We first investigated purely 2D methods using single RGB cameras but soon found those approaches too limiting given the generalized nature of the problem. We next investigated the use of the structured light camera in the Kinect 1 sensor. This was promising and we were able to achieve accurate segmentation using the depth mesh provided by the Kinect, but at a low resolution and over a physical volume. In order to make this work over

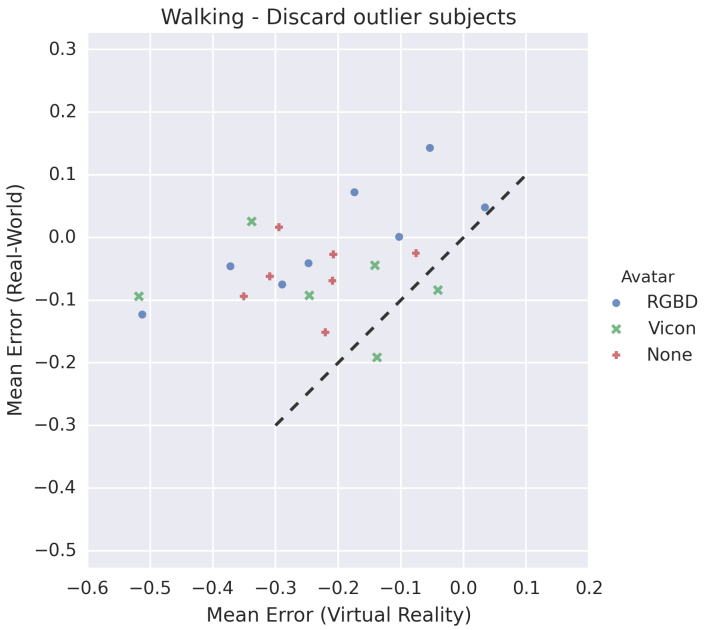


Figure 3.33: Scatterplot of blind walking mean relative error for each participant, real-world versus virtual reality. If there were no distance estimation errors we would expect each point to be on the line $x = y$.

a larger volume multiple Kinect sensors are needed. This brings with it a large set of other problems relating to interference between the Kinect sensors, since they work on the same IR frequencies, and centralized real-time reconstruction and combination of the data from multiple cameras. Finally, we investigated the use of smaller time-of-flight RGB+D sensors mounted directly to the HMD in-line with the user's view direction. This solution allows a consistent resolution image of the user's body, regardless of where they are in the physical space, and a virtually occlusion free image since the view position is very near the user's eyes.

Since this research was conducted in 2014, with user studies in early 2015, there have been several similarly intentioned alternatives published by other researchers. In [77], Tecchia *et al.* describe their system for creating video self-avatars. It differs from our approach in that it uses a PrimeSense depth camera mounted above the head, whereas our system uses a smaller and lighter camera that can be mounted to the front of the HMD. The approach of Tecchia *et al.* has the potential to create larger rendering errors

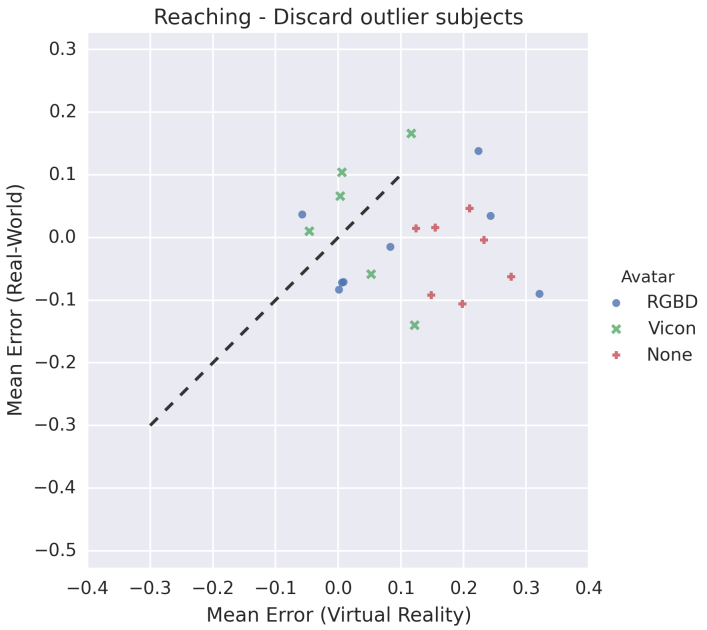


Figure 3.34: Scatterplot of blind reaching mean relative error for each participant, real-world versus virtual reality. If there were no distance estimation errors we would expect each point to be on the line $x = y$.

since the reprojection of the texture onto the mesh is more likely to contain visible self-shadowing from the perspective of the user's line of sight. Also, the PrimeSense camera appears to have a smaller FOV than the HMD to which it was attached, resulting in the appearance of floating hands since the forearm is not visible to the camera. It is also unclear how calibration is handled for the PrimeSense camera. A similar system designed by Ha *et al.* for augmented reality called the WeARHand, is discussed in [78]. This paper focuses not on the creation of a self-avatar, since the application is augmented reality where the hand is already visible, but on the interaction between the user and virtual objects. A semi-transparent proxy hand is provided, but it's appearance is not intended to be photo-realistic, and serves as a feedback to the user into the workings of the system. The WeARHand system also uses morphological operations similar to our approach, such as hole filling, in order to compensate for the noise in the depth image as well as the different resolutions between the depth and color images.

The video-based self-avatars described in this section open the door to many interesting areas of future research. Among them is the possibility of hand tracking and interaction with the immersive virtual environment without the need for wands or marker-based tracking of the hands. Since the hands themselves are based on video images, it is conceivable that image-based rendering techniques such as in-painting could be used to alter the appearance, and allow for the appearance of holding or touching virtual objects that do not exist. This could be further enhanced through the use of passive haptic props. Another future avenue is the exploration of video-based avatars in multi-user scenarios. Since the range of the RGB+D depth sensor is limited any multi-user approach will require an algorithm capable of segmenting other people in the scene at near distances where depth is available and at far distances where depth information will be missing or inaccurate.

4

Enhanced Collaboration

A popular display modality in virtual reality is the Large Screen Immersive Display (LSID). These displays are usually on the scale of 2 meters or more in height, and 2 meters or more in width and are sometimes referred to as *display walls* or *virtual windows*. A LSID may have a single screen surface that subtends a large portion of the viewers field of view, or multiple surrounding screen surfaces such as in a CAVE system. An advantage of these large immersive displays is that they are large enough for multiple users to view the scene while at the same time being able to interact with each other in a natural way. This is often useful in design review or data visualization tasks.

Traditionally a LSID capable of stereo display can display a single stereo view (one left eye and one right eye) by alternating frames between eyes and having the user wear synchronized shutter glasses. Recent technological advances have allowed the creation of high frame rate displays that can generate multiple, in some cases up to 6, separate stereo views for groups of multiple users [79]. However, even with these newer screens for large groups there will be a subset of users that are seeing a correct stereo view from their vantage points and a separate subset that will see a view that is incorrect for their vantage point. Normally in stereo vision, feature points between the images seen by both eyes can be matched and triangulated to obtain depth information about a scene. When the rendered views do not match the vantage point then the solution of geometric triangulation will not be correct, or may not even exist [11].

A critical factor in whether the information shown on the LSID is interpreted correctly is the accuracy of the rendering. In order to compute the correct stereo view from a vantage point two pieces of information are needed. First, the vantage point itself must be known. This is usually accomplished by tracking the shutter glasses worn by the viewer. The second piece, and the focus of this chapter, is a mapping from 2D display pixels to the 3D point in the world where the pixel will be seen. The process of determining this mapping is known as screen calibration.

In the simplest case the LSID is a large planar display, such as a commercially available LCD television. The mapping from 2D display coordinates to world coordinate then only requires the equation of the plane of the display, pixel size, and the location of the (0, 0) top-left pixel. For more complex displays the parametric approach involves a similar strategy of describing the screen shape via a geometric equation (e.g. a sphere, cylinder, parabola) and then measuring the specific location of a few pixels in order to solve the equation for the entire surface. Non-parametric approaches attempt to approximate the display surface as the combination of multiple parametric screens. Usually small planar sections. As the size of each piece decreases and the number of total pieces increases the approximation becomes more exact.

In this chapter we describe the design, implementation, and testing of an open source software package for automatic non-parametric LSID calibration. Using a readily available off-the-shelf webcamera and some printed fiducial markers attached to the screen bezel the software is able to reconstruct a piece-wise planar approximation of the display surface. The software can handle the case of multiple projectors displaying onto a shared surface and compute projector overlap to be used for blending and dithering at render-time. The entire software package is available online, open source, at Github (`https://github.com/lpuchallafiore/EasyCalibrateVr`) in the hope that it will prove valuable to other groups in the community and reduce the tedium and repeated engineering and software development that often occurs during display calibration.

4.1 Background

The goal of immersive virtual reality is to replace a user's default sensory input with input that is computationally controlled, thereby immersing the user into the virtual environment. Visual immersion in a computer generated virtual environment is achieved using real-time computer graphics rendering and an immersive display technology such as a head-mounted display (HMD) or large screen immersive display (LSID). The goal is to evoke a sense of *presence* in the virtual environment by rendering and presenting the virtual views in such a way that they can be interpreted as views from the user's own vantage point. However, previous studies of egocentric distance perception in immersive virtual environments have found equivalent performance to real world viewing only under rare circumstances, whereas in most cases egocentric distances are compressed when using a head-mounted display to view the IVE [74].

Vision science research has previously shown that when people view a 2D image of a 3D scene from a vantage point that is offset by a rotation from the position of the virtual camera they can correctly perceive the 3D structure of the scene as long as they can perceive the orientation of the display surface and correct for their own oblique viewing angle [80]. However, if they same study is repeated using a stereo 3D image of a 3D scene then people tend to be incapable of performing the same correction [81]. In addition, it has been shown that in cases where a geometric solution to the stereo reconstruction is not possible because the vantage point does not match the viewpoint that people can still infer some information about a 3D scene, and that geometric errors do not always match the user's perception of the errors [11]. Pollock *et al.* in [82] examine the effects of viewing a scene with improper depth cues by examining a collaborative scenario wherein both participants are shown the same stereo view that is correct for only one of them. They find that collaboration times increased as the depth discrepancy between the two locations was increased.

Despite potential problems relating to the perception of errors with immersive virtual reality, large screen immersive displays have been used in numerous applications for design, visualization and teleconferencing. Cruz-Niera *et al.*, in [20, 21], develop the CAVE immersive virtual display. This display consists of four large displays surfaces, three walls and the floor, creating a surround-screen virtual reality environment. In [83],

Tredinnick *et al.* demonstrate a hybrid system using a tablet and a large CAVE-like projection display to explore architectural designs. The tablet displayed traditional blueprint and model designs and controlled the larger life-size rendering shown on the projection display. LSIDs are also often used for teleconferencing, the goal is the create the experience of being in the same room with the other individuals in the meeting. Gross *et al.*, in [84], present the *blue-c* teleconferencing immersive display. It is a 3-sided CAVE-like environment that uses advanced glass LCD panels as a projection surface. This allows the surfaces to switch very quickly between transparent and opaque, allowing cameras to capture views of the interior of the space. The images from the cameras are used to reconstruct a 3D representation of the user in real-time. In [71], a 6-user wall-size auto-stereoscopic LSID is presented that uses multiple Microsoft Kinect 1 sensors to capture stereo information to be displayed to the display on the other end of the connection. The Kinect sensors are also responsible for tracking the viewpoint of each of the 6 users to allow for correct rendering. Fuchs *et al.* also experiment with the use of multiple Kinect 1 sensors to capture the room geometry in [85].

LSID are often made of several display panels (LCD, OLED, Plasma) or projectors because the cost to drive the display using a single device is prohibitively high and often devices with sufficient resolution for the entire LSID simply do not exist. For this reason, Humphreys *et al.* have developed WireGL [86], and its successor Chromium [87], distributed PC-cluster rendering architectures. These architectures allow for a group of inexpensive PCs to handle rendering for the multiple projectors or display panels that comprise the LSID. Geometric misalignment, color variation, and individual projector distortion are compensated for, if properly calibrated, by these rendering architectures. In [88], Bierbaum *et al.* describe the VRJuggler platform that performs distributed cluster rendering for LSID as well as handling I/O with common VR equipment such as trackers, wands, and gloves. One limitation of the VRJuggler platform is that each node in cluster must run the same program, and the programs are synchronized after each draw call. Schaeffer *et al.*, in [89], describe the Syzygy platform that allows the VR program to split responsibilities heterogeneously across the cluster while remaining synchronized.

Before a stereo LSID can be used it must be calibrated. That is to say its location in the physical space must be measured so that stereo views can be correctly projected

onto the display. Calibration is often performed manually in many existing systems [12,18, 34], or by way of specialized mechanical and optical adjustments in the display itself. Manual calibration is time consuming and tedious, and does not scale well to larger displays.

Recently, researchers have begun to develop automated methods of calibration, driven by the falling cost of digital cameras and the processing power needed for computer vision. In [31], Brown *et al.* survey the different methods available for automatic display calibration. They separate the present research into two categories, parametric screen calibration (planar, sphere, cylinder, etc) or non-parametric arbitrarily shaped screen calibration. In this chapter we focus on the second category because the goal of this research is to calibrate the saddle shaped LSID, described in more detail in a subsequent section, at the Digital Design Consortium lab of the University of Minnesota.

Both Garcia *et al.* [90], and Sajadi *et al.* [91] present methods for calibration of LSID using uncalibrated cameras. Sajadi's method applies to swept projector surfaces such as 4- and 5-wall CAVEs. The method described by Garcia is similar to the one we implement later in this chapter for EasyCalibrateVr in that it uses paper fiducial markers placed around the display surface and binary gray-codes in order to identify and reconstruct the display surface. Most calibration methods for planar displays compute a linear homography from projector space in the world space. In [92], Bhasker and Majumder describe a technique that uses rational Bezier patches in order to compensate for projector radial distortion. Numerous other camera based techniques exist to calibrate the geometry and photometric properties of a LSIDs [31–39, 93–95].

4.2 EasyCalibrateVr: Arbitrary screen calibration and rendering

In the DDC laboratory at the University of Minnesota we have a unique display that is made from a stretched fabric surface, back-projected by three 1280×1024 projectors. Because the fabric is stretched across a cylindrical frame to form a semi-circle, it is pulled towards the center of the room near its midpoint. It is also affected by gravity, creating a non-symmetrical saddle shape where it is curving towards and around the viewer along the horizontal axis but away from the viewer vertically. This is shown in

Figure 4.1: Photograph of the display in DDC laboratory at the University of Minnesota used during development of EasyCalibrateVr. The screen forms a semi-circle around the viewer, but curves away from the viewer vertically so that the top and bottom are further away than the center.

better detail in Figure 4.1.

Because of this unique shape, most commercial screen calibration programs are unsuitable for our use as they assume a predefined parametric description of the screen (e.g. cylinder, sphere, parabola). We decided instead to investigate non-parametric methods of screen calibration, but were unable to find any that were open source or readily available.

We therefore set out to develop our own calibration software package. This effort had several goals. First and foremost was to support the calibration of arbitrarily shaped projection displays, like the one we have in our lab. The second was to be easy to use, so that re-calibration could be done in an afternoon and easily explained to newly incoming members of the lab. The last goal was to release the software as open-source and encourage other groups to use and contribute to its development so that they do not need to reinvent the software.

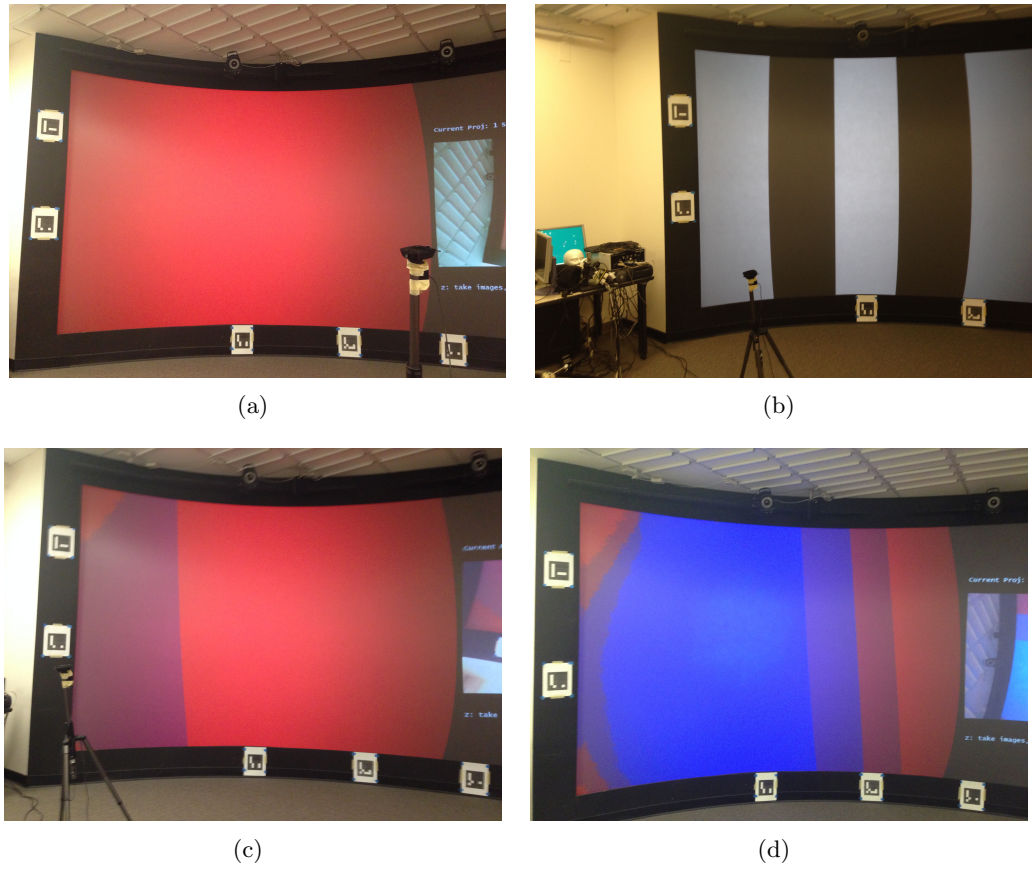
The software works by reconstructing the shape of the projection surface from a series of camera images taken from multiple viewpoints within the laboratory space. Each area of the display must be visible in at least 2 of these images for the reconstruction to complete. In practice 3 or more images of each section is preferable. The core algorithm used to perform this reconstruction is that of Structure-from-Motion (SfM) specifically the variant described by Snavely, Seitz, and Szeliski in [96] and implemented in the SfM-Toy project [97] from which we forked the reconstruction routines.

4.2.1 Data Capture

The first step in the program is the capture of camera images to be used in the reconstruction. We do this by means of a USB webcamera connected to the same computer that is driving the LSID. An on-screen interface guides the user through the capture procedure, notifying him or her when to move the camera to a new location. The process is completely automated apart from moving the camera. When the camera is placed at a new position within the room, each projector, in sequence, displays a gray-code pattern (see Figure 4.3). This pattern allows each projector pixel to be uniquely identified in the camera image and used as the feature points during the Structure-from-Motion reconstruction. Because of this we do not need to rely on computed feature points that typically rely on texture or corner features of which the uniform surface of the projector screen contains none. At each point when the user is asked to move the camera the projectors display a heatmap showing the number of times each pixel has been viewed, red for 0, blue for five times. This heatmap allows the user to easily position the camera in places to view the least viewed sections of the screen. An example of this heatmap, and the data capture procedure, can be seen in Figure 4.2.

4.2.2 Reconstruction

As mentioned earlier, our reconstruction routines based on the SfM-Toy application [97] implementing the algorithm of Snavely *et al.* from [96], although the code-bases have diverged sufficiently that submitting our patches upstream is impossible. The reconstruction can proceed with or without a camera calibration of the webcam. If no calibration is provided the program will attempt to solve for it during the reconstruction bundle adjustment. In practice we have found that better results are found more quickly if a rough calibration is provided. Since SfM only provides the reconstructed mesh up to scale, we have also place augmented reality markers [98] on the walls of the lab surrounding the LSID. These markers have been registered using the Hiball 3100 tracker so that we know their exact position within the laboratory space. We can then use a rigid alignment method, such as Horn's method [99], to transform the arbitrarily scaled SfM mesh into a mesh that is registered with the respect to the tracked physical space. As



(a) (b)

(c) (d)

Figure 4.2: Photographs showing the interactive webcamera capture process. As parts of the screen are viewed by capture images, the display surface changes from red to blue. Once the entire screen has been captured from an adequate number of images, i.e. the screen is entirely blue, then the structure-from-motion reconstruction and calibration can take place.

(a) (b) (c)

Figure 4.3: Here are some partial images from the graycode pattern shown by one projector. Notice the first frame is an entirely green screen on the projector. This is done so that changing lighting in the room outside the projector screen is not mistakenly identified as part of the graycode pattern.

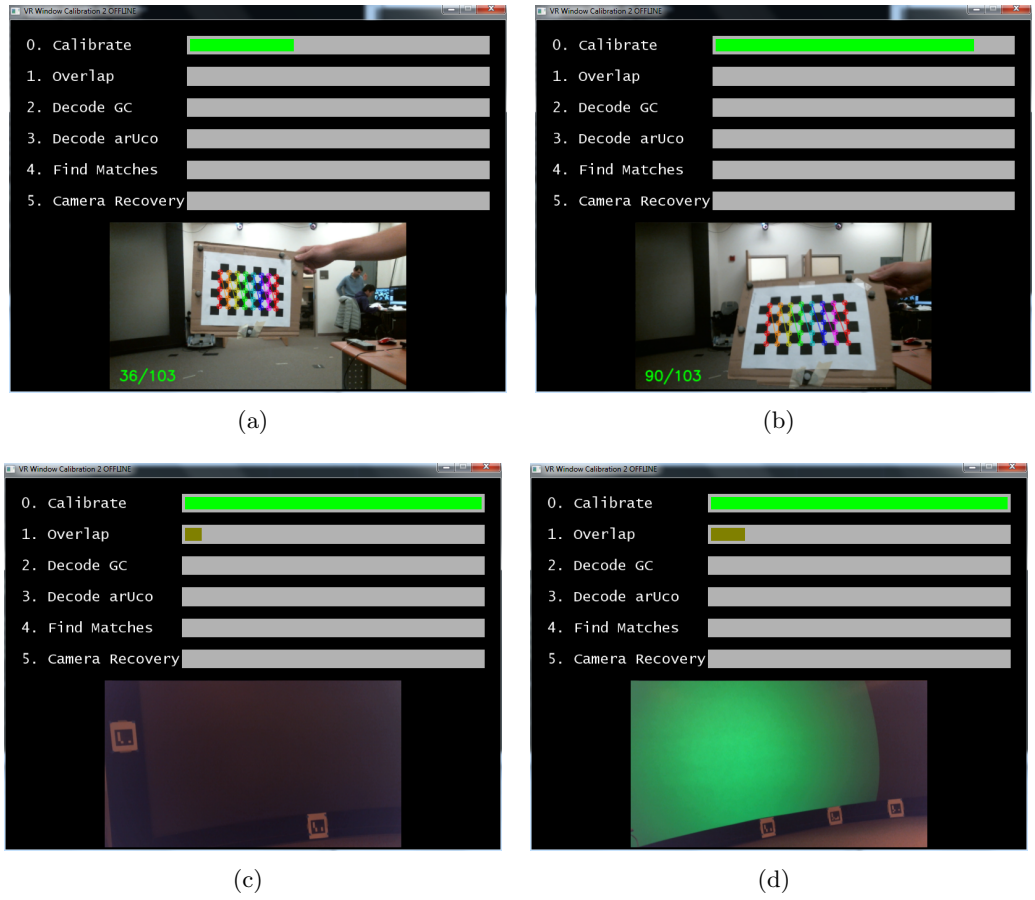


Figure 4.4: Screenshots showing the calibration progress GUI of EasyCalibrateVr. This screen contains progress bars for each phase of the calibration process. Input images, if processed in the current step, are displayed below the progress bars.

the captured images from the webcamera are being processed the EasyCalibrateVr application displays several progress bars showing the current state of the reconstruction. If any images are currently being analyzed they are also shown in this progress GUI, as seen in Figure 4.4. An example result for reconstruction of the saddle shaped screen in the DDC laboratory can be seen in Figure 4.5. Also, since each pixel location can be identified uniquely via the graycode the calibration program also generates masks for each projector identifying visible pixels as well as overlapping pixels between projectors. An example of these masks can be seen in Figure 4.6.

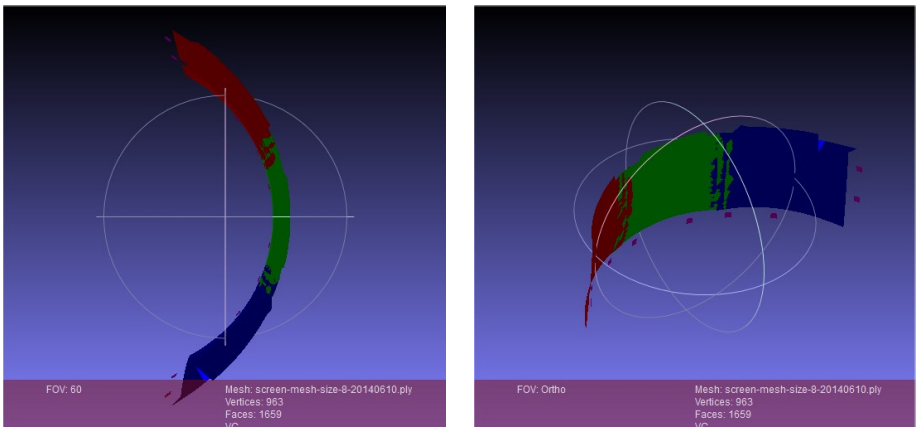


Figure 4.5: The results from our Structure-from-Motion based calibration of the large screen immersive display. Each portion of the mesh corresponding to a projector is colored a unique color. Notice the overlap between the adjacent projectors and the saddle shaped curvature of the screen.

(a) Visible pixels for projector 1 (b) Visible pixels for projector 2

(c) Overlapping pixels of projector 1 onto projector 2 (d) Overlapping pixels of projector 2 onto projector 1

Figure 4.6: Masks of visible pixels, as well as projector overlap computed during the calibration process. The three projectors are numbered from left to right, 1 to 3.

Figure 4.7: Photograph showing the rendering to a calibrated saddle-shaped screen.

4.2.3 Real-Time Rendering

The mesh and the pixel correspondences observed via the gray-codes can then be used to render 3D stereo virtual scenes onto the screen from the perspective of a head-tracked observer. To do this we have implemented the 2-pass rendering method of Raskar *et al.* from [37]. In this method the virtual scene is first rendered using the tracked observer's location with a gaze direction towards the LSID surface and the result saved to a texture buffer. In a second pass the texture is warped onto the LSID by means of a projective texturing operation from a virtual image plane of the virtual camera to the reconstructed mesh of the LSID. A result of this rendering on the LSID in the DDC laboratory can be seen in Figure 4.7.

4.3 Conclusion & Future Work

In this chapter we have described why the calibration of large-screen immersive displays is a difficult and important task. However, it is often done in a mechanical and manual fashion requiring costly maintenance over the entire operational lifetime of a LSID. Several researches have investigated ways to automatically calibration LSID using cameras and other sensing devices, however as far as we could find these software packages are unavailable to the general public.

In an attempt to remedy this lack of calibration software we have developed the EasyCalibrateVr project (at `github.com/lpuchallafiore/EasyCalibrateVr`). This

project aims to be an open-source automatic camera-based LSID calibration tool. It supports arbitrary display configurations of multiple panels or projectors and requires only a single USB webcamera for sensing input. The camera can be calibrated using OpenCV or uncalibrated. We have demonstrated that the software is effective at reconstructing the display surface mesh and can be used to render virtual scenes onto a LSID.

In the future it would be useful if EasyCalibrateVr could output the final calibration into a format that can be read by one of the more developed rendering packages such as Chromium or Syzygy. A key aspect of calibration is color correction. Currently this is not supported by EasyCalibrateVr, and would be a challenging avenue of future research because of the view-dependent nature of light transfer through the back projected display screen. Another avenue of future research is to use the package to calibrate a second display with different characteristics from the one in the DDC laboratory, and to encourage other research groups to use and contribute to EasyCalibrateVr.

5

Enhanced Locomotion

Natural self-motion in immersive virtual environments (IVE) is one of the most fundamental problems in the field of virtual reality [12]. While tracking technologies allow users to perform motions in a virtual world that match their movements in a tracked physical space, many setups provide such natural interaction only over a range of a few meters either because of limitations of the tracking hardware or the size of the available physical space. To cover longer distances, users of immersive virtual environments often have to revert to indirect forms of traveling such as virtual steering or flying using a joystick or wand input device. These systems allow users to travel in large IVE but often create greater difficulty for users attempting to keep track of their location when compared to natural self-motion forms of locomotion such as walking [3, 13, 14].

Natural walking redirection techniques have shown great potential for enabling users to travel in large-scale virtual environments while their physical movements are limited to a much smaller physical space. Traditional redirection approaches introduce a subliminal discrepancy between real and virtual motions of the user by subtle manipulations, which are thus highly dependent on the user and on the virtual scene. In the worst case such approaches result in failure cases that must be resolved by obvious intervention on the part of the experimenter or VR system, for example, when a user faces a physical obstacle and tries to move forward.

In the real world, we make use of *walking* as our primary locomotion technique over short distances, but we usually make use of various methods of physical transportation to cover longer distances. While many researchers have proposed different solutions to

enable natural walking over large distances in the virtual world while remaining within a relatively small physical space in the real world, fewer works have focused on techniques that can provide users of physical transportation devices (*e.g.* wheelchairs, scooters, bicycles, etc) with the ability to use such devices for natural navigation in immersive virtual environments.

In this chapter we introduce a novel approach to redirection with physical traveling devices. We show for a motorized electric wheelchair that it is possible to redirect users onto different paths in the real world and in the virtual environment by introducing both undetectable virtual rotations, and undetectable physical rotations of the wheelchair platform. We show in a psychophysical experiment that this approach has significant potential for making redirection of transportation devices more applicable in virtual reality laboratories than approaches relying on redirection of natural walking. Moreover, we provide evidence that the detectability of manipulations depends on the speed of self-motions, which has implications for practical implementations of redirection techniques. We also investigate results from a study seeking to elucidate the extent to which spatial understanding in immersive virtual environments is facilitated when visually-indicated translation and rotational movements are felt as well as seen, in order to comparatively assess the benefits of redirected driving over other forms of virtual self-movement.

5.1 Background

Moving about the scene is a key interaction activity in all immersive virtual environments. Various techniques have been suggested for virtual self-motion, which can be divided into *indirect* methods, which use a joystick or wand to move the virtual world about the user (*e.g.* flying), and *natural* methods such as walking which move the user through the virtual scene by means of active self-propulsion. What has been shown in multiple studies is that natural forms of motion allow users a better spatial understanding of the virtual scene when compared to indirect forms.

5.1.1 Spatial Cognition

When considering spatial understanding, researchers use several different terms to classify different aspects of space [100]. *Vista space* refers to an area that can be seen all at

once from a given vantage point. *Environmental space* refers to an area that requires significant locomotion to explore in full, such as a neighborhood or campus. *Geographical spaces* are those that are too large to be apprehended through direct experience [100]. Current research in spatial cognition [101, 102] casts doubt on the notion that, during real world travel through environmental space, people actually incrementally construct a global, allocentric, metrically accurate Euclidean cognitive map of their surrounding environment. Converging evidence suggests, instead, that when navigating in environmental space people rely on multiple, local, egocentric encodings of vista space that are interconnected in a topological graph structure [103, 104]. In this paper we focus on the question of enabling improved spatial understanding in virtual environments within vista space.

5.1.2 Spatial Updating

As we move about through a large, open environment in the real world, we keep track of where we are with respect to where we have been through a process called *spatial updating*. Under typical conditions, this process is both automatic and obligatory. For example, if people are asked to close their eyes and quickly point to the locations of learned fixed landmarks in a static surrounding environment, their performance does not decline if they are required to physically turn before responding; however they have difficulty responding, after turning, as if they were still facing in the original direction [105]. Also, people find it easier to point to the locations of learned landmarks after imagining that they have turned within a stationary external environment than after imagining that the external environment has turned around them [106]. When we use a technologically-mediated locomotion interface to enable the virtual exploration of large remote places, however, the natural process of spatial updating can be disrupted.

In seeking to understand how to best support the spatial updating process in conditions where natural walking is not possible, it is useful to briefly review what is known from previous work. Much research has been done in the field of psychology to elucidate both the cognitive processes involved in spatial updating and the perceptual mechanisms that trigger it (*e.g.* [107], etc.). We focus here on work that seeks to explore the impact of cues derived from physical (as opposed to purely visual or imagined) movement.

There is strong evidence that some of the various cues (proprioceptive, vestibular,

and efference copy) provided by real physical movement can play a valuable role in facilitating the process of spatial updating and enhancing spatial understanding. Kennedy *et al.*, tested participants ability to update their heading after travelling along a 2-segment path that contained an intermediate turn [76]. Of the five stimulus conditions: real walking while blindfolded, imagined walking from a verbal description, imagined walking from watching someone else walk, optic flow only, and optic flow with a passive physical turn, they found that performance on a point-to-origin task was significantly better in the walk and passive turn cases than in the other three. Likewise, Loomis *et al.*, found that peoples performance on a triangle completion task was enhanced when they were passively moved [108]. Chance *et al.*, found that participants performed significantly more accurately on a point-to-unseen-target task after exploring a virtual environment using real walking than when using a joystick to control their motion virtually [109]. They found that performance was intermediate in a condition that involved real turns but used the joystick to translate. Zanbaka *et al.*, similarly found that real walking enabled improved performance on a variety of spatial cognition tasks compared with the use of visual-only and rotation-only virtual locomotion methods [110]. Also, in [3], Ruddle and Lessels, found that participants who used a real walking interface performed better on several measures of performance in a hidden target search task than did participants who used a visual-only or rotation-only method.

However, in contrast to the findings of Kennedy *et al.* [76], Sigurdarson *et al.* [111], found that when people were asked to point to their starting location after virtually traversing a curved path in a highly detailed and realistically-rendered virtual environment, their performance was not improved when they were also passively turned. In contrast to the findings of Zanbaka *et al.* [110], Suma *et al.* [14], found no significant difference in participants performance on a variety of measures of spatial memory and cognition after exploring a photo-realistically rendered virtual branching maze using real walking versus gaze-directed virtual travel. Also, Riecke *et al.*, found that performance on 6 of 8 dependent measures in a hidden target search task was equivalent when participants used a joystick-based virtual locomotion method with real turns as when they used real walking [5]. And, in [112], Riecke *et al.*, demonstrated that in the absence of visual cues, participants performed as well on a spatial updating task when they experienced the *illusion* of rotational motion as when they actually turned.

Most investigations of the impact of physical motion on spatial updating have focused on the rotational component of the movement. In tests of spatial updating after an imagined change in vantage point, Rieser, found a significant increase in errors and latencies when the change involved a rotation, but not when it involved a translation, suggesting that the rotational component of physical motion is more important to the spatial updating process than the translational component [107]. However, in [113], Ruddle *et al.*, found that participants who explored a large virtual environment were able to more accurately estimate distances between landmarks when a linear treadmill locomotion interface was used in conjunction with virtually-executed turns, while allowing real turns in conjunction with virtual translational movement did not offer similar benefits.

In light of these disparate findings, it is clear that more work is needed to fully elucidate the importance and impact of different types of physical motion cues, including translation in particular, as highlighted by Ruddle in [114], to the spatial updating process.

5.1.3 Redirection

A range of different methods have been developed to allow people to use physical actions within a confined real space to control their virtual travel over unbounded distances in IVEs. Many such interfaces seek to evoke physical sensations related to walking, which may increase the subjective realism of the locomotion experience. Examples of the different types of approaches include: hardware-based solutions such as the omni-directional treadmill [45]; gesture-based solutions such as walking-in-place [115]; and software-based solutions such as redirected walking [15]. The approach considered in this paper falls most closely into the category of motion simulation platforms [116], albeit at a very low level of sophistication.

Redirected walking, introduced by Razzaque *et al.* in [15], seeks to allow the illusion of unbounded free walking in an immersive virtual environment while physically walking within a finite real space. It works by introducing subtle dissociations between the users actual movement and the associated change in their viewpoint in the virtual environment. Through small, carefully orchestrated manipulations of the visual input, the user is prompted to make small "corrective" adjustments in the direction of

their heading [117], by which means it becomes (theoretically) possible to keep them walking in circles and never actually reaching the boundaries of the tracked area. In practice, however, "failure" situations do occur, in which the user finds himself in a state where he cannot move forward. Considerable effort has gone into the development of methods to assuage the cognitive disruption and spatial disorientation associated with such events [118, 119]. In [48], Steinicke *et al.*, have determined thresholds for the detection of different types of redirection during real walking, including rotational and translational gains. They found that one would need a lab space of over 40m $\times$ 40m to imperceptibly guide people on a curved path of infinite length. However, Hodgson *et al.*, found that even suprathreshold amounts of rotational redirection do not cause significant spatial interference [120]. Peck *et al.*, in [121], found that people performed better on several navigational measures when exploring a large IVE using redirected walking with distractor-based re-orientation rather than walking-in-place, or using a joystick to translate.

Redirection implicitly requires the introduction of some visual/vestibular and visual/proprioceptive conflict. Even when such conflict is not overtly noticed, it can still have an impact on peoples perception of their motion. In [122], Campos and Bülthoff, provide a comprehensive review of recent work in multi-sensory self-motion perception under conditions of cue conflict. They report that peoples responses generally reflect a combination of the information available from all sources, though body-based cues tend to be given more weight than visual cues when subjects are walking. Studies have so far not shown large differences in peoples sensitivity to redirection when driving in a motorized wheelchair versus walking [123]. Nevertheless, as the distances that people want to traverse in a virtual environment get increasingly large, we believe that driving may become an attractive complement to walking. We know of few studies that have explicitly compared the effect of active locomotion mode (*e.g.* walking versus biking or driving) on the accuracy of the spatial understanding that people tend to accrue through the active exploration of large, environmental space areas.

While motorized wheelchair travel engages the proprioceptive system to a much lesser extent than do non-motorized travel modes such as walking or cycling, it affords strong vestibular cues to motion. The commercial success of motion simulators attests to some of the benefits that might be expected from such feedback [124]. Active wheelchair

driving also engages similar cognitive processes as other types of active locomotion. Furthermore, hardware-assisted redirection methods also afford the potential to manage reorientation in a less overtly intrusive manner. Traditional approaches to redirection require the user to subconsciously adjust his physical actions in accordance with the altered visual feedback he receives in order to maintain his locomotor objectives. With redirected driving, we have the ability to provide the user with kinesthetic feedback that is consistent with his visual stimulus as redirection is being applied (as shown in Section 5.4. Furthermore, with redirected driving we have the potential ability to automatically execute evasive maneuvers such as an exaggerated turn of the chair when required to avoid a failure state. This could avoid the cognitive disruption associated with having to explicitly notify the user of the need to stop and take corrective action.

5.1.4 Cybersickness

Redirection involves deliberately introducing variable amounts of sensory conflict between the body-based sensations associated with a persons actual motion (or lack of motion) and their concomitant visual stimulus. It is well-known that noticeable levels of visual/vestibular conflict can lead to cybersickness [125], even if they are infrequent and transient [126]. Such conflict arises when visual motion is immersively experienced in the absence of physical motion, or when there is even a small amount of latency between the onset of visual and vestibular cues to motion. However, cybersickness is less frequently associated with the introduction of a moderate gain in the magnitude of concurrently experienced visual and physical rotational or translational movements. Also, it has been reported by Stanney and Hash, that higher levels of proprioceptive engagement are associated with decreased severity of cybersickness symptoms [127]. Individuals have been found to vary widely in their propensity to become cybersick, and studies by Kolasinski, have identified a variety of factors such as gender, age, and prior experience in cybersickness-inducing situations that can co-vary with susceptibility [128].

5.2 Prior Experiments By Our Group

In the real world we make use of walking as our main form of self-motion for short to medium distances, but other long distances we use mechanical devices such as wheelchairs, scooters, bicycles and automobiles. The motorized wheelchair allows for the spatial updating benefits of natural locomotion while at the same time widening the possibilities for active and passive redirection enabling virtual scenes of a larger scale than the tracked physical space.

5.2.1 Redirected Walking-and-Driving

Because of the drawbacks of redirected walking when applied to longer distances, in [123], Bruder *et al.* introduce the concept of redirected driving by applying existing redirection techniques to a motorized electric wheelchair. A traveling machine such as a motorized wheelchair will give different self-motion cues compared to walking and thus it has to be carefully analyzed to determine the extent to which redirection techniques can be applied to such devices. Towards this end, the authors ran three experiments comparing walking locomotion and driving locomotion, shown in Figure 5.1.

All of the experiments were performed in a 11m $\times$ 9.5m darkened empty room. The subjects wore an nVisor SX60 HMD (1280 $\times$ 1024 @ 60Hz, 60° diagonal FOV) to view the IVE, and a Hiball 3100 tracking system tracked the subject's, and wheelchair's, position in the room. In each experiment the subject performed a number of motions in the virtual scene and was asked the following question: "Was the virtual movement *smaller* or *larger* than the physical movement?". Responses such as "I cannot tell" were not allowed. The first experiment focused on rotations alone, the second on translations, and the third on movement along a curvature.

The hypothesis was that since driving a wheelchair lacks many of the muscle proprioceptive feedbacks of walking that subjects would be less sensitive to redirection while driving. The results of the experiment show that this is only the case in the third experiment when moving along a curvature. When dealing with rotations and translations the limits of redirection are similar to those of redirected walking. An interesting finding was an overestimation of virtual translation compared to physical translation, this indicates that virtual translations may have to be scaled up in the wheelchair condition

in order to match the perceived real-world physical translation.

Both redirected walking and redirected driving provide the users with near-natural vestibular and proprioceptive feedback of actually moving in the real world. The user interface and algorithm can be easily implemented in head-tracked VR laboratories without extensive hardware and software changes. The only addition being the tracking of an electric wheelchair to provide a colocated virtual wheelchair. Comparing the efficacy of the redirection shows promise as the limits of redirection are no worse, and in some cases better, than those for redirected walking.

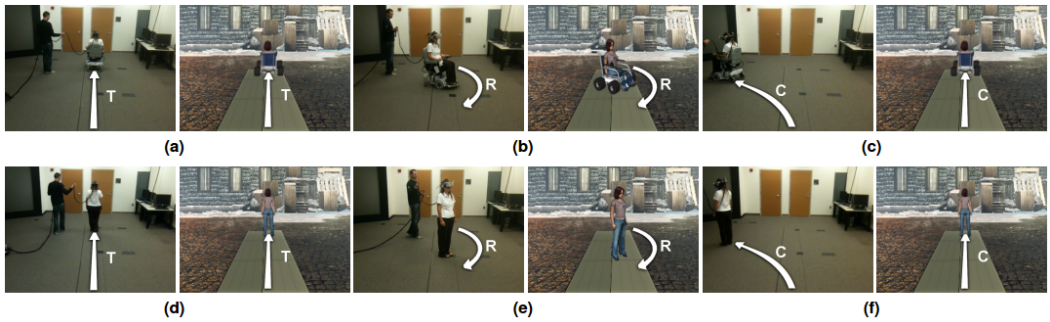


Figure 5.1: Redirected walking-and-driving in immersive environments: (a)-(c) user steering an electric wheelchair with head-mounted display, and (d)-(f) natural walking counterparts. Figure from [123].

5.2.2 The Benefits of Active Locomotion

The previous work showed that redirection of physical locomotion devices (*e.g.* wheelchairs) is possible and has the potential for stronger redirection than walking alone. However, the practical benefits of pursuing further research of redirected driving depend upon the extent to which presence, engagement, and spatial awareness can be augmented by allowing people to experience physical movement while traveling in an immersive virtual environment. To this end, Nybakke *et al.*, in [129], ran a series of experiments seeking to comparatively assess the extent to which people are able to maintain spatial awareness while driving an electric wheelchair versus walking.

The experiments focused on the incorporation of real *translational* movement into the locomotion interface. Specifically, spatial updating performance was compared between real walking, joystick control of translation but not rotation, and wheelchair

locomotion. The task is based off the methodology of Riecke *et al.*, from [5]. A head-mounted display was used to immerse participants in a realistically rendered virtual environment that was devoid of notable landmarks, as shown in Figure 5.7. The participants were asked to search and find 8 targets randomly hidden in 16 possible locations, here shown as the boxes atop the pillars. A box would change color when touched depending on if it contained a target (red) or not (blue). If a participant revisited the same box a set number of times because they have become disoriented the trial would end without finding all of the targets.

Each participant performed the search using four different locomotion methods, shown pictorially in Figure 5.2. In method **R** participants moved by freely and naturally walking in the lab space. In **S** they stood on a 1" high platform to discourage walking, and used a joystick to move forward and backwards. In **J** participants sat in a swivel chair and used a joystick attached to the chair arm to control translation. For both **J** and **S** rotation was done by physically turning in place. Finally, in case **W** participants moved by driving around the environment using a motorized electric wheelchair.

A key performance metric for these trials were the number of *perfect* trials, in which all of the targets were found before the end of the trial. Participants had the greatest number of perfect trials (33.3%) when using the wheelchair **W**, followed closely by walking **R** (31.3%). The number of perfect trails was considerably less with sitting **J** and standing **S** (14.6% and 18.8%, respectively). Another factor that was important was average trial time, which was significantly faster with walking than with the other methods, despite the virtual movement speed similarity between the conditions. The trial time was slower with the wheelchair because of the artificially imposed travel rate of the device.

Overall, participants performed best when walking. However, performance was also better when using the wheelchair when compared to the conditions with the joystick. This suggests that the experience of physical motion facilitates the process of spatial updating and awareness when exploring the virtual environment. These results suggest that there is merit in exploring the use of redirected driving in a motorized wheelchair as an alternative to purely virtual movement when exploring immersive virtual environments.

So far we have investigated redirected driving in the context of similar algorithms to

redirected walking. That is to say that by causing imperceptible changes in the *virtual* rotation experienced by the participant we are able to redirect their path due to the subconscious corrections done by the user's own motor control. However, since we have a motorized electric wheelchair we are not limited to only virtual modifications. If we were able to command the wheelchair to move without user input we could introduce imperceptible *physical* rotations which would redirect the user around the room regardless of their subconscious motor control actions. The following section discusses how we can modify an electric wheelchair to allow these types of physical movements, and after that we look into the limits of this new type of redirection and apply it to a similar box search task.

Figure 5.2: Photographs of the different locomotion methods: real walking (**R**), joystick translation while standing (**S**), motorized wheelchair (**W**), and joystick translation while sitting (**J**). Figure from [129].

5.3 Motorized Electric Wheelchair Platform

For the next studies reported, we developed a novel locomotion control system in which virtual changes in viewpoint, actively controlled by a joystick interface and presented

via a head-mounted display, can be seamlessly accompanied by either fully- or partially-corresponding physical movement while the user is seated in a computer-controlled power wheelchair. The wheelchair can also move independently of the joystick interface either automatically or manually from the attached PC.

Our hardware platform was constructed by augmenting a Hoveround MPV5 electric motorized wheelchair with an Arduino microcontroller board. The microcontroller intercepts the voltages output by the wheelchair's joystick and sends them to the computer running the IVE simulation, where they can be modified and returned before they are passed on to the wheelchair's motor controllers. The components of this system are shown schematically in Figure 5.3. Thus we are able to define the participants virtual viewpoint based on the raw joystick signals while retaining independent control over the corresponding physical movement of the wheelchair. This gives us the ability to redirect the wheelchairs motion while allowing the user to retain the illusion that they are controlling the wheelchair directly.

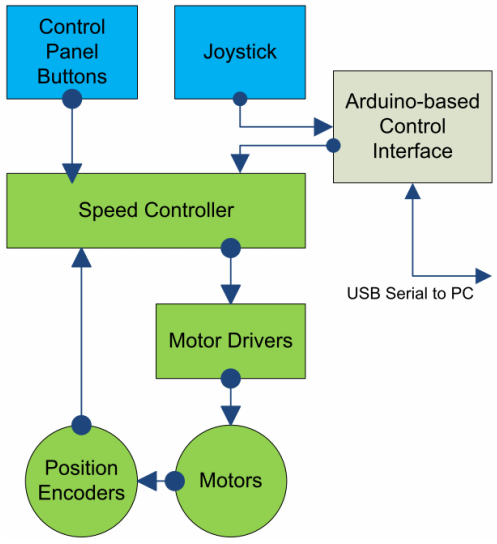


Figure 5.3: A generic electric wheelchair will consist of a speed controller, motor drivers, batteries, motors and position encoders which form the speed control loop. The target speed are specified via a joystick and potentially other input buttons on a control panel. Our modification is to place a hardware interface between the joystick and speed controller that allows us to monitor the user's joystick inputs and, if needed, substitute them with our own during the redirection.

The Hoverround wheelchair joystick has balanced voltages for each axis. What this means is there is a reference voltage signal (between 0v and 5v), a positive voltage signal (between 0v and 2.5v), and a negative voltage signal (between 0v and -2.5v) for each axis which gives a total of six values which we need to read from the joystick. The Arduino supplies 5v to power the joystick independently of the wheelchair, and reads the six input voltages using its six analog-to-digital (A2D) input pins. Six output signals are then created each using one of the Arduino's pulse-width-modulation (PWM) outputs connected to a low-pass filter for each signal. The low-pass filter smooths the square-wave PWM output to form a digital-to-analog (D2A) circuit to match the input. The input voltage values are broadcast over a bidirectional serial USB connection to the host PC which is running the IVE simulation that is responsible for sending commands to control the output voltages.

The input and output signals, as well as supply voltage for the joystick, are carried over Cat5e cables between the wheelchair and the Arduino. These cables can be disconnected allowing the wheelchair to return to a stock configuration by using a bypass cable. A protective plastic case mounted beneath the seat of the wheelchair holds the Arduino. Originally, the Arduino received power over USB which worked well during the development of the firmware. However, when we switched to using a 32 foot USB cable for the experiment the Arduino could no longer reliably get 5 volts supply voltage. This caused the wheelchair speed controller to go into an inactive fail-safe mode because it believe the joystick was damaged. To fix this we added a battery pack to the bottom of the Arduino case along with an on-off switch to supply adequate power.

The wheelchair is tracked via 2 Hiball 3100 sensors, one attached to the chair and on to the HMD.This allows the virtual chair position and physical chair position to differ while maintaining the correct participant viewpoint with respect to the wheelchair. The particpant uses the wheelchair joystick to control the virtual position of the wheelchair and the physical movement is controlled by the IVE program running on the host PC.A Wiimote is used by the participant during one of the experiments discussed in Section 5.4.1 to select virtual objects. The entire hardware setup can be seen in Figure 5.4.

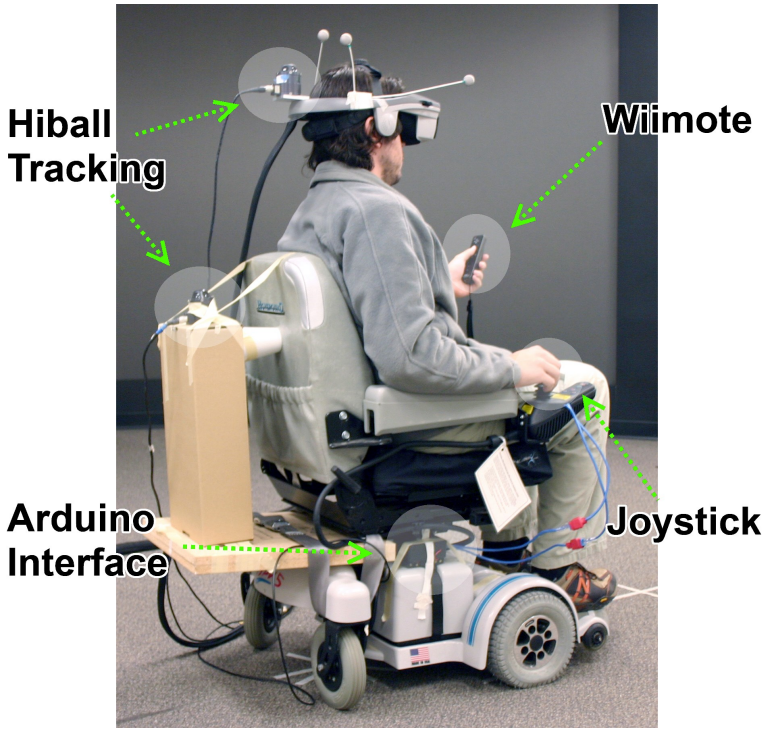


Figure 5.4: Here is shown the dual tracker setup used to differentiate the subject's head motion from the motion of the wheelchair. The joystick is used by the participant to control virtual movement, while the real movement is controlled by the IVE via the Arduino interface. A Wiimote can be used for additional user controls depending on the experiment.

5.4 Thresholds of redirection perception

With the motorized electric wheelchair now able to be controlled remotely by the IVE simulation, we want to know how the thresholds for *physical* redirection compare to those for *virtual* redirection. To this end we ran a series of experiments similar to the work in [123] where participants were moved along a curvature and asked if they thought the virtual motion matched the physical motion. This work originally appeared in [130].

5.4.1 Experiment Design

We evaluated computer-controlled redirected driving in an experiment with 10 participants. Participants experienced redirected driving at two speeds and randomized

curvatures. Participants responded in a *two-alternative forced-choice* (2AFC) task to identify the direction of the curvature.

Eight males and two females participated in the study. They were recruited from the department of computer science and the authors acquaintances. Participants ages ranged from 18 to 51 years, with the average age being 28 years. All participants had normal or corrected-to-normal vision, and none reported any problems with stereo vision or balance disorders. Six participants reported having much experience with 3D games, although only two were regular players. Seven participants had experience with HMD virtual environments, including three participants who are authors on this paper. The experiment lasted approximately an hour, and participants were compensated with a $10 gift card to a national retail chain.

The IVE was presented to the participant through an NVIS Nvisor SX 60 HMD, which has a manufacturer-specified 60° field of view, and a 1280-by-1024 pixel resolution. We attached the cable to the back of the wheelchair seat to relieve its weight from the participants head and to prevent any positioning feedback from the cable tension. Tracking of the participants head and the wheelchair was provided by a Hiball 3100 optical ceiling tracker. A veil of two layers of black felt was attached to the HMD to prevent participants from seeing any part of the environment beyond their own torso. Brownian noise was also played through the HMD headphones to mask auditory positioning cues.

The IVE (see Figure 5.5) was modeled from our DDC laboratory using Google Sketchup. A realistic appearance was achieved by using photographs of the lab interior as texture maps. We implemented the experiment using the G3D rendering engine [27], which ran on an Intel computer with Core i7 processors, 6 GB of RAM and Nvidia Quadro FX 4500 graphics card. We compensated for the pincushion distortion of the HMD (see Figure 5.5).

The IVE included a path on the ground marked with two strips of tape and a circular indicator centered at 80% eye height on the door at the end of the path. The indicator showed the participants speed through color. Green meant “go faster”, red meant “slow down”, and yellow corresponded to the correct speed. The color of the indicator was tied to the participants forward joystick input. Since we were keeping the speed of the wheelchair at a set value, the indicator was there to encourage the participant to keep

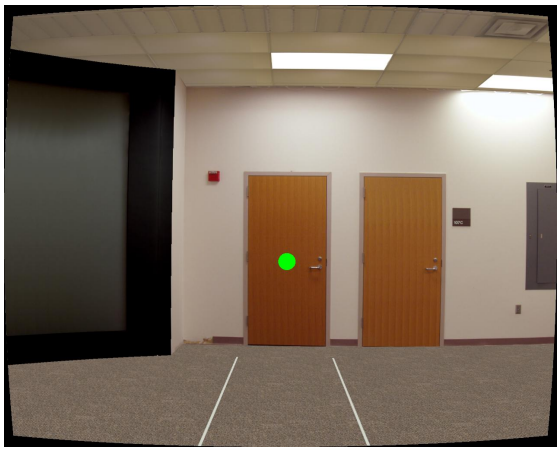


Figure 5.5: A participant's view at the start of a trial.

the joystick pushed forward and give the illusion that they controlled the speed.

Participants began by signing a consent form and filling out the Kennedy-Lane simulator sickness questionnaire (SSQ) [76] and a demographics questionnaire. Participants then read printed instructions for the experiment.

Participants performed 72 trials in four blocks of 18. While the participant saw text on the screen instructing him or her to wait, an experimenter used a joystick to position the wheelchair at one end of the room. The experimenter then pressed a button to begin the trial, making the IVE visible and enabling the joystick on the wheelchair. Every trial started at the same point in the IVE.Participants were required to drive down the path in the IVE while pressing forward on the joystick enough that the indicator was yellow.

When the participant pushed forward on the joystick, one of two controller modes was activated: computer-controlled or human-controlled. In the computer-controlled trials the participant drove on a straight path in the IVE (the participants steering did not affect the virtual motion) while the wheelchair was moved on a curved path in the real environment. The real world movement was driven by a PID controller implemented in the IVE software that could compensate for bumps in the floor or wheel spin to keep the real movement on the circular path. In the human-controlled trials, the participants view of the virtual world was rotated as they moved forward, and the participant had to steer the wheelchair to stay on a straight path. The participants steering also controlled

the physical wheelchair. Participants completed equal numbers of computer-controlled and human-controlled trials. The wheelchair also moved at one of two possible maximum speeds, 0.33 m/s or 0.54 m/s. The speed was limited by clamping the joystick input to a maximum value.

For each mode and speed combination, participants saw each of six curvatures three times. The curvatures corresponded to following circular paths of these radii: 10 meters to the left, 20 meters to the left, 30 meters to the left, 30 meters to the right, 20 meters to the right, and 10 meters to the right. The fast and slow trials were grouped into two blocks, and half of the participants saw the fast trials first, while the other half saw the slow trials first. Within those blocks the mode-curvature combinations were presented in randomized order. By interleaving the trials so that the participant could not anticipate the controller mode, we felt that the participant would be more attentive to the task of driving and would be less likely to use a different strategy for detecting the curvature or become complacent during a block of computer-controlled trials. Although we collected experiment data in all conditions, we observed an implementation error in the logs of the human-controlled trials, and decided to exclude those conditions from further evaluation.

After the wheelchair had traveled 3 meters it stopped, and instructions on the screen asked the participant on what side of the *real* room, right or left, did he or she end up. The participant indicated the answer by pushing on the joystick.

After the participant answered the question, the display showed instructions asking them to wait while the experimenter moved the wheelchair into position for the next trial. After every 18 trials the participant was required to take a five-minute break. We wanted to prevent fatigue, and we were concerned that if the breaks were optional, then participants might choose not to take them.

After completing 72 trials, participants then completed another SSQ and a short questionnaire about the experiment and were paid their gift card.

5.4.2 Results

We had to reject one participant for always answering that he was on the right side of the room. Figure 5.6 shows the pooled results for the tested curvature radii on the x-axis, with negative values referring to physical paths bent to the left, and positive

values referring to physical paths bent to the right. The y-axis shows the probability for estimating the physical path as bent to the left while moving straight in the IVE.We represented the discrimination performance via a sigmoid psychometric function of the form

$$f(x) = \frac{1}{1 + e^{a \cdot x + b}}$$

with fitted real numbers a and b. The gray psychometric function shows the results for the slow trials, and the black function for the fast trials.

The curvature radii at which subjects answered that they were redirected towards the left side of the room in 50% of the trials is taken as the *point of subjective equality* (PSE), at which subjects judge the virtual motion to match the physical movement. From the psychometric functions we determined PSEs at a radius of -57m for the slow trials, and 595m for the fast trials, i. e., the responses indicate that subjects on average judged straight movements in the real world as straight. As the radii decrease or increase from the PSE the ability of subjects to detect the difference between physical and virtual motion increases. A practically applicable range of manipulations is given by the smaller (i. e., conservative) detection threshold of 75% correct judgments, i. e., the middle between the 50% chance level and 100% certainty of subjects that they have been manipulated, which we determined from the psychometric functions as radii larger or equal to approximately 5.76m for slow movements, and approximately 16.52m for fast movements with the electric wheelchair.

5.4.3 Discussion

The results plotted in Figure 5.6 show an impact of the wheelchair speed on responses. The results show that the subjects were less accurate at detecting manipulations of physical driving directions when they were driving slowly compared to the trials with the faster driving speed. The data suggests that the detection threshold may be reached at a circular path radius of less than 5.76m in case subjects move slowly, whereas for faster movements subjects were able to detect manipulations up to a circular path radius of approximately 16.52m, which indicates a surprisingly strong effect of movement speed on direction estimates. Effects of movement velocity on redirection techniques have

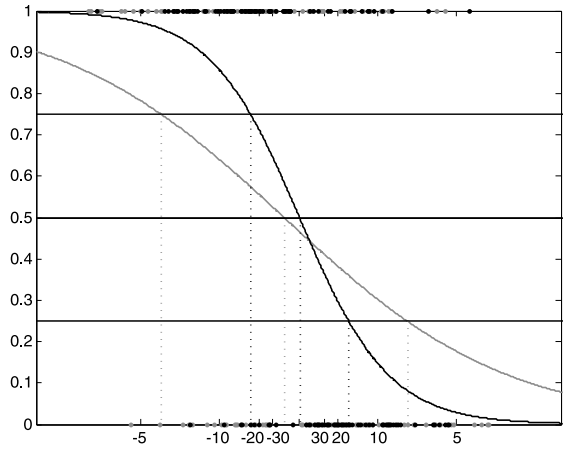


Figure 5.6: Pooled results of curvatures for the fast trials (black function) and slow trials (gray function). The x-axis show the circular path radii in the real world, with negative values referring to paths bent to the left, and positive values referring to rightward paths. The y-axis shows the probability of estimating the physical movement path as bent to the left.

first been observed by Neth *et al.*, in an experiment on redirected walking, in which subjects were significantly better at judging walking directions if they were walking at a higher velocity [131]. The results shown in Figure 5.6 are interesting, since they suggest that this observation also holds when driving a wheelchair, and, moreover, that it seems not to be caused by the fact that traditional redirection techniques require users to adapt to visual rotations. Since subjects in the present experiment were passively reoriented without the requirement to actively compensate for virtual rotations, the increased discrimination performance in the experiments may be related to less ambiguous proprioceptive and vestibular physical self-motion cues during redirection.

The detection thresholds for the trials with fast movements are in line with results for redirected walking in previous experiments. Bruder *et al.* [123], observed a radius of 14.92m as detection threshold, whereas Steinicke *et al.* [48], observed a radius of 22.03m for walking subjects. The differences in the results may be caused by the different redirection techniques, *i.e.*, walking versus driving, and different visual stimuli used in the experiments. The detection thresholds for the trials with slow movements indicate that passive redirection as used in the present experiment can result in less observable manipulations than using traditional redirection techniques when driving a wheelchair,

for which an experiment with similar motion speed has suggested a detection thresholds of 8.97m [123].

5.5 Comparison of redirection methodologies

The steering controller we developed for the motorized electric wheelchair allows complete control over the wheelchair motion, irrespective of the user's joystick inputs. This opens up many possibilities for redirection of rotations, translations, and combinations of the two. The principal goal of the experiment presented in this section was to investigate the extent to which passively-experienced physical cues to translational and rotational motion might facilitate peoples ability to keep track of where they have been in a richly detailed but landmark-free, vista-space virtual environment. This work was originally published as [132] at Joint Virtual Reality Conference (JVRC)) 2013.

5.5.1 Experiment Design

Following [3, 5, 6, 129] we used a hidden target search task in a realistically-rendered immersive virtual environment to assess spatial awareness under four different active locomotion conditions, each of which used the same joystick interface to control the virtual viewpoint. Specifically, the conditions differed only in the nature of the physical feedback that accompanied the visually-indicated movement. All participants experienced all conditions, in counter-balanced order. In condition **V** (visual only) the wheelchair did not physically move; in condition **R** (rotation only), the wheelchair was allowed to rotate, but not to translate; in condition **P** (partial) the rotational and translational displacement of the chair was dampened to approximately half that of the visually-indicated movement; and in condition **T** (full) the physical movement of the chair matched its visually-indicated movement. In all cases, subjects were head-tracked and could freely look around in the virtual environment while driving through it.

We collected data from a total of 35 participants (26 male, 9 female; aged 18 to 55, $\mu = 24.6 \pm 8.8$), recruited from personal contacts and from passersby to the building housing our lab, and including three of the authors. Two participants were unable to complete the experiment due to technical difficulties with the equipment, and another 13 participants were unable to complete trials in all of the locomotion methods due to

cybersickness. The 20 participants who completed all trials were 16 male, 4 female, aged 18 to 51, $\mu = 23.2 \pm 7.1$. The four tested conditions were presented in counterbalanced order among these 20 participants.

Each participant experienced each condition using the same hardware, shown in Figure 5.4. We presented the visual stimulus on an NVIS nVisor SX 60 HMD, which uses twin Liquid Crystal on Silicon (LCOS) displays to provide a 1280×1024 image to each eye over a manufacturer-specified 60° diagonal field of view with 100% stereo overlap. The HMD is equipped with foam blinders that restrict peripheral vision; however to guarantee complete immersion during testing we dimmed the room lights and draped a large veil of heavy black felt over the front of the HMD.Participants sat in the motorized wheelchair and used the wheelchairs built-in joystick to control their movement through the virtual environment. We used a HiBall 3100 6DOF optical ceiling tracker to separately determine the position and orientation of the wheelchair and the participant's head.

In contrast to the studies by [129], we used this wheelchair motion simulator to control the location of the viewpoint based on the output of the joystick controls. Our goal was to preserve the visual illusion of wheelchair driving while dampening or eliminating various components of the physical movement of the chair. We defined the simulation model by measuring the wheelchairs maximum speed and rate of acceleration from a complete stop for several positional settings of the joystick, considering both linear and rotational motions. This calibration produced a piece-wise linear mapping from joystick input, case, and in the rotational, x-axis input, case. At each frame, we can then compute the expected wheelchair linear speed and rotational speed based on the calibration data and the current joystick position. Given a joystick x and y, the calibrated values for maximum speed, s_m, and maximum acceleration, a_m, are found from the piecewise linear curve for both linear and rotational velocities. Given these values and the current speed s_i, the next speed, s_{i+1}, is computed as follows,

$$s_{i+1} = \begin{cases} \max\left(s_m, s_i + a_m dt\right) & \text{if } s_m > s_i, \\ \min\left(s_m, s_i + a_d dt\right) & \text{otherwise,} \end{cases}$$

where dt is the time since the last frame and ad is a constant deceleration speed, which for this simulation was set to 1 m/s. The movement of the viewpoint is then

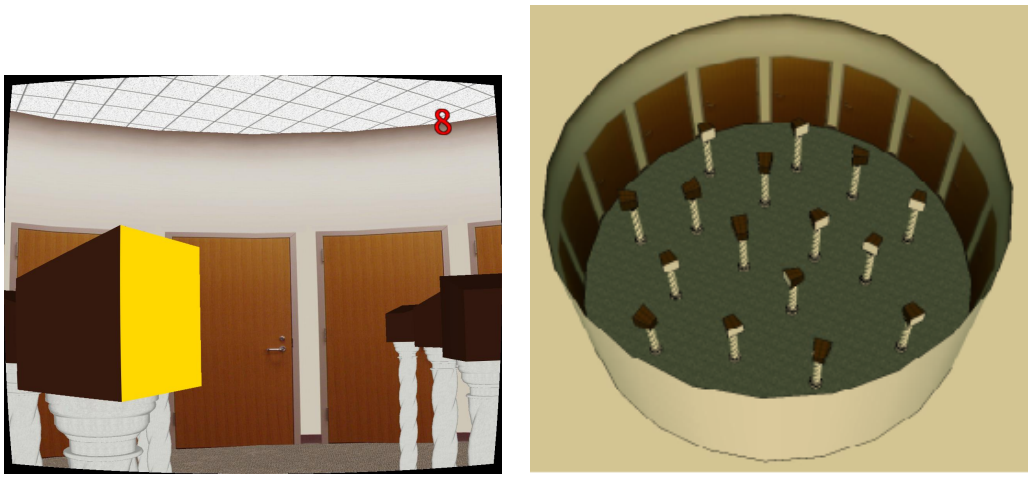
updated based on these quantities.

Target selection was accomplished using two different approaches. The first fourteen participants used a third HiBall sensor affixed to a hand-held wand, while the next 21 participants used a Wiimote. With three HiBall sensors running simultaneously, each sensor operated at 200–400Hz, but when only two sensors were used the update rate increased to approximately 600Hz. However, we found that this change had no discernible effect on the end-to-end system latency, which we measured as 53ms in the 3-sensor case using the method described by [133]. During the trials, two experimenters managed the cables attached to the HMD so that they would neither pull on the participant's head nor be run over by the wheelchair. We played an ambient soundtrack of tropical birds and flowing water through the HMDs built-in headphones to mask any subtle auditory cues from the outside world.

The virtual environment was modeled using Google Sketchup as a circularly symmetrical room, 10 feet tall and 24 feet in diameter, matching the height and slightly smaller than the narrowest dimension of our DDC laboratory physical space. We achieved a realistic appearance by texture mapping the walls, ceiling and floor of the model with excerpts from photographs of our lab space. The experiment was implemented in C++ using the G3D game engine [27], which ran on a Windows XP computer with an Intel Core i7 processor, 6GB of RAM and Nvidia Quadro FX 1500 graphics. We compensated for the pincushion distortion of the HMD when rendering and generated the stereo views using a constant inter-pupillary distance (IPD) for all participants.

Participants began by filling out a demographics questionnaire and a baseline simulator sickness questionnaire (SSQ) [76]. They then read printed instructions explaining the task and the use of the equipment. The main experiment consisted of a total of 12 hidden target search trials, split into in 4 separate blocks, one for each locomotion method. The first trial in each block was considered as training and discarded; performance measures on the other two trials were averaged to yield one data point per participant, per measure, in each condition for subsequent analysis. To avoid inadvertently biasing participants interpretation of the different locomotion methods, we took care to refer to each of them solely by their generic, single letter identifiers and we refrained from providing any descriptive information about any of the them. We counterbalanced the presentation order of the methods between participants, and enforced

a 5 minute break after each block of 3 trials, during which time the participant was offered water and cookies and asked to fill out a SSQ. After the last trial, participants completed a short exit survey in addition to the final SSQ.The survey forms were customized for each participant, so that each participant was asked to provide feedback about each locomotion method in the same order as he or she had experienced them.



(a) Participant Viewpoint (b) Aerial View of IVE

Figure 5.7: An example of what participants saw during the experiment, along with an overhead view of the virtual environment.

At the start of each trial, participants found themselves in the center of a realistically rendered, circularly symmetric virtual room containing 16 pillars, each of which held an asymmetrically-shaped box whose largest face was colored white. The pillar positions, box orientations, and target locations were all randomly defined at the start of each trial. For the participants who used a third HiBall sensor, this sensor allowed them to directly control a rigidly tracked virtual hand that they could search a box by reaching out to touch its white side. Participants who used the Wiimote to execute the searches had a very similar user experience. To select a box they had to approach it to within arms length and turn to face its white side. At that point the box face would turn yellow to indicate its availability to be searched via a button press. Half of the boxes contained targets, and would turn red when searched for the first time; the other half, when selected, would turn blue. After being searched, the face color would return to white. All boxes would appear blue if re-searched, requiring participants to remember

which boxes they had already visited. A counter was continuously displayed in the upper corner of the display to indicate the number of targets remaining to be found. The trial ended either when the participant successfully found the last of the hidden targets, or after they had made eight consecutive re-searches of previously searched locations. Participants were not allowed to see the color (red or blue) of the last searched box, so that they would not know if a trial had ended in failure.

Figure 5.8 illustrates the key differences between the four locomotion methods tested. Each image shows a plan view of the virtual environment with its random boxes. The red paths trace the movement of the virtual viewpoint, while the blue paths show the actual head position. Please note that these images were created for explanatory purposes only, and were made after the completion of the experiment. During testing we did not save the participants actual head positions for subsequent analysis, just their virtual viewpoint.

5.5.2 Results

A total of 20 participants successfully completed the entire experiment, giving us data from a total of 40 test trials in each of the 4 conditions. Eight of these 20 participants had used the HiBall interface and 12 had used the Wiimote to query the box contents. No significant differences due to the selection mechanism were observed. We computed seven performance measures on the pooled data. Figure 5.9 plots the results of two of these measures: percent perfect trials and percent failed trials. We can see that there were more perfect searches (with 0 boxes revisited), and fewer failed trials (with 8 consecutive re-searches of previously-searched boxes) in the full motion condition than in the three other conditions. Statistical tests of significance on this data found that $\chi^2(3, N = 160) = 3.53$, $p = 0.32$ for the incidence of perfect trials and $\chi^2(3, N = 160) = 6.25$, $p = 0.10$ for the incidence of failed trials, however, which is not sufficient to reject the null hypothesis.

Figure 5.10 plots the results of the five other measures: total number of boxes revisited per trial; total distance traveled per trial; number of boxes searched before the first revisit; average distance between successive box searches (a measure of the economy of movement); average speed of movement (total traversed distance/total time taken per trial); and change in cybersickness score between the pre-test and post-test

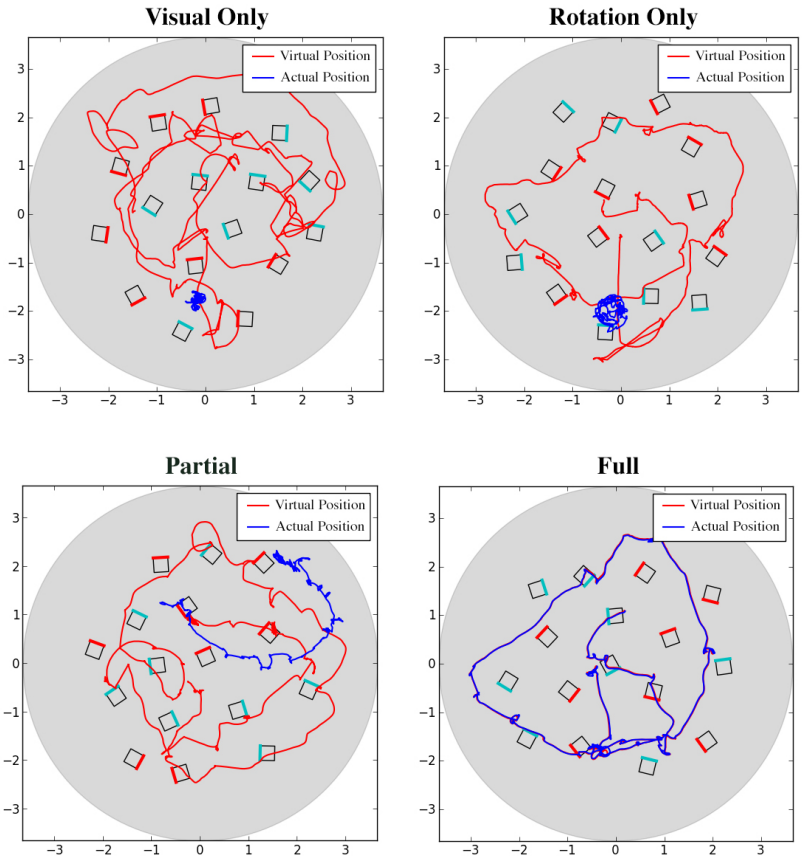


Figure 5.8: An illustration of the virtual (red) and actual (blue) head positions during the search task for each method.

questionnaires for each method. We can see that participants in the full motion condition tended towards having fewer revisits overall, and searching a greater number of novel locations before re-checking a previously searched box, than did participants in the other three conditions. However, an ANOVA analysis found that the only measure on which there was a statistically significant difference in performance between the methods was in participants self-selected speed of travel during the search task ($F_{3,76} = 11.15$, $p < 0.001$). Outliers were not removed before doing the ANOVA.

In the exit surveys of the twenty people who completed the experiment, the majority (7.5) indicated a preference for the partial motion method, followed closely by the full motion method (5.5) and visual-only (5). However the majority (9) of the 13 participants

who had to discontinue their participation due to cybersickness fell ill during the partial motion trials, and we did not collect preference data from the participants who did not finish the experiment. Again considering only the 20 participants who completed all trials, ten of them made comments either to the effect that the full-motion method felt too fast or that the partial-motion method felt more realistic. However, eight others made comments to the effect that the full-motion method felt most realistic, and/or less nauseating than the partial-motion method. Two participants did not comment on any notable differences between these two methods. In other feedback, ten participants cited the lack of movement in the visual-only case as a problem, along with complaints of nausea or ease of getting lost. However, six other participants either expressed a preference for the lack of movement in this case, or explicitly said that they did not see it as a problem.

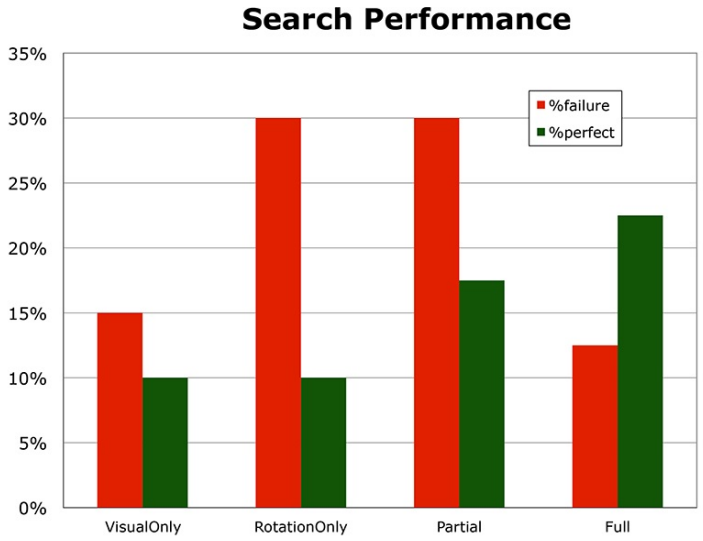


Figure 5.9: A plot of the rate of perfect searches (green bars) and failed searches (red bars) in each of the four conditions.

Finally, we noted remarkable similarity in the qualitative nature of the paths traversed between the four locomotion methods examined in this experiment. Comparing histograms of both speed and curvature, following the methods of [113], we found that the characteristics of participants virtual movement using our wheelchair motion simulator in the visual-only, as well as the **R** and **P** conditions, were qualitatively similar

to when the joystick controls were directly driving the wheelchair. This stands in stark contrast to the quality of motion we observed in earlier experiments [129] where participants used the joystick to control their translational movement while physically using their bodies to turn. Figure 5.11 compares a trace of the *worst* (maximum distance) path over all trials in our visual-only condition and in the joystick/sitting condition from [129].

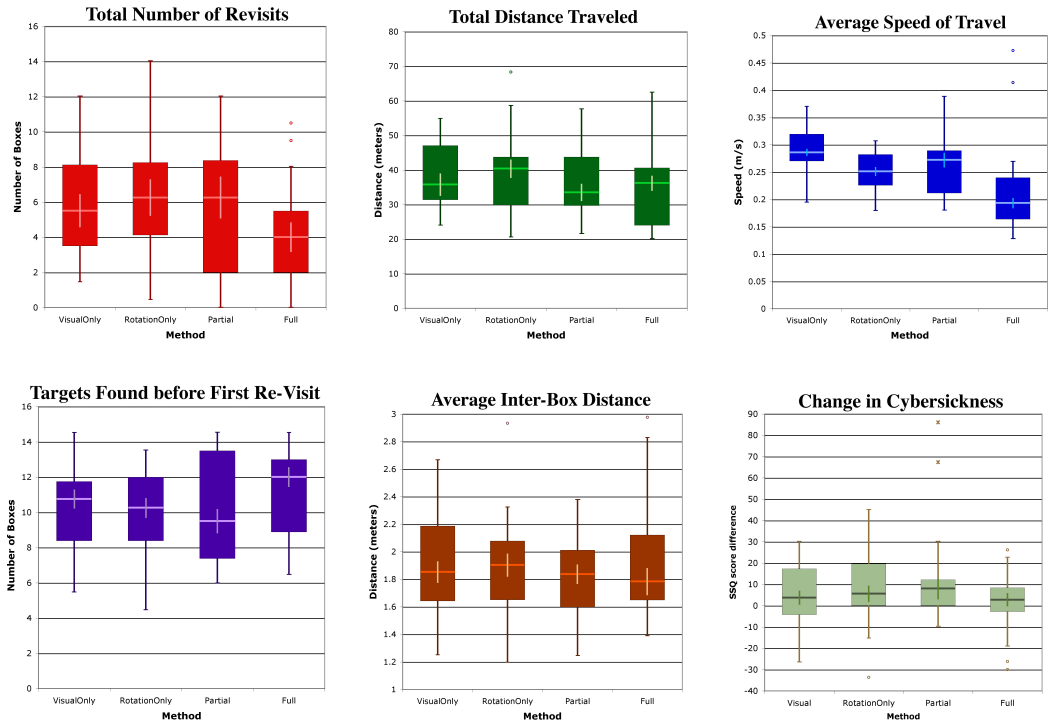


Figure 5.10: Box and whiskers plots of several performance measures computed for the 20 participants who successfully completed the experiment. The boxes enclose the range between the 25^{th} to 75^{th} percentile, and the horizontal lines inside the box indicate the median. The whiskers extend from the minimum to maximum values within 150% of the inter-quartile range (IQR). Outlying results are indicated by small circles, or by stars if beyond 300% of the IQR.The smaller vertical lines insides each box bound the 95% confidence interval around the mean. Although we can see a slight trend in the revisit data, only the differences in speed were statistically significant.

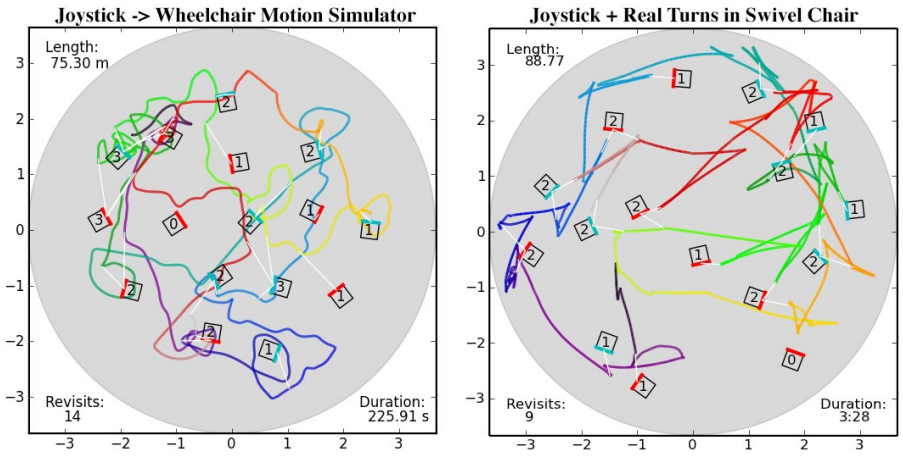


Figure 5.11: Left: a trace of the longest traversed path in the Visual-Only (**V**) condition in our present experiment where movements of the joystick controlled a wheelchair motion simulator; Right: a trace of the longest path in an earlier experiment where participants controlled their orientation in a swivel chair with their feed while using the joystick to translate.

5.5.3 Discussion

Several factors complicated our ability to compare the four locomotion methods as effectively as we had intended. The first major problem we ran into was with cybersickness. Three of the four locomotion methods we tested (**V**, **R**, and **P**) involved a dissociation between the visual and vestibular feedback that people received, and we knew that with any such dissociation cybersickness could be a concern. We encouraged participants to discontinue the experiment if they felt cybersick, but were surprised when a total of 13 people terminated early for this reason. Nine these 13 discontinued because of sickness in the partial-movement condition and four because of sickness in the rotation-only case. Of the 20 participants who completed all trials, most did not show significant signs of cybersickness, though there were several cases where people had high SSQ scores in one or more conditions but then felt recovered enough after the mandatory break to continue with the other methods.

Extensive diagnostic testing subsequently revealed a shortcoming in the design of our virtual motion simulator that was likely to have been the aggravating factor. Essentially, the model we used did not properly account for the inertial forces acting upon the chair,

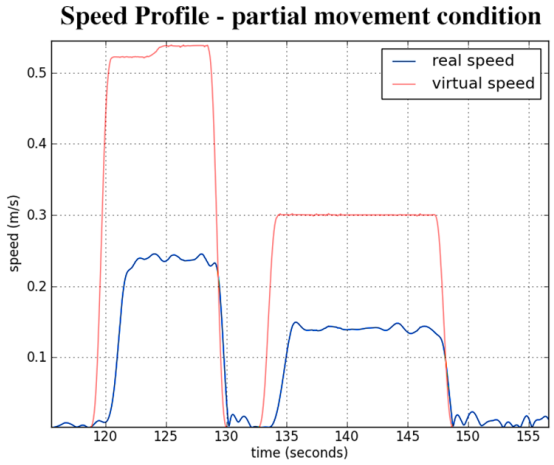


Figure 5.12: Comparison of chair velocities in real world (blue) and virtual (red) while being driven forwards and backwards in a straight line. This recording shows an inaccuracy in the virtual simulation whereby virtual speeds accelerate faster than the real speeds by a small amount. The amplitude differences are by the design of the partial movement condition.

particularly when it was accelerating from a resting position, and when the alignment of the passive rear wheels was not consistent with the direction of the intended forward motion of the chair. Figure 5.12 shows a close-in view of the velocities of motion of the chair (green) and the virtual viewpoint (red) in the **P** (dampened movement) condition while the wheelchair was being driven forwards and backwards in a straight line. One can see in this example that the virtual viewpoint is starting to move ahead of the physical movement of the chair (a *reverse* lag), although the timing of the deceleration is well-matched. The amplitude differences are of course by design, and consistent with our aims. The second complication we encountered was that in the two conditions that involved substantial physical movement, subjects occasionally ran the wheelchair too close to the walls of the lab, necessitating an adjustment of their physical position. Interruptions also occurred at times due to problems with loose contacts in the cables. Our system log recorded a total of 10 interruptions during the full motion trials and 14 during the partial motion trials, as well as 7 in the rotation-only condition and 1 in the visual-only condition. Post-hoc testing found that two of the performance measures, distance and time, were significantly worse in the interrupted than in the

non-interrupted trials, but probably because the interruptions were relatively broadly distributed when we re-analyzed the data considering only the uninterrupted trials we found no qualitative difference in the pattern of results.

Despite these shortcomings, we believe that the results of our experiment are able to contribute informative new insights on several key points. First of all, we see a performance trend that supports the idea that people are somewhat better able to keep track of where they are while immersively exploring a visually-detailed but landmark-free vista-space virtual environment when they receive reliable physical feedback in addition to visual feedback about their motion, even when their physical motion is not controlled by walking but rather by sitting passively and driving. This observation extends the results of earlier experiments [3, 5, 129] and suggests potential benefit in the continued investigation of locomotion control methods that involve actual physical movement as opposed to purely simulated travel. Secondly, our experience provides a cautionary example of the potentially deleterious impact of providing physical motion feedback that is not adequately representative of the concomitantly indicated virtual motion. In the two locomotion conditions where the correspondence between the visually-indicated and physically-felt motions was close in some ways but imperfect in others, a lot of people got sick despite the fact that others were perfectly content. Accompanying a visual signal with conflicting or lagging vestibular cues to motion can cause cybersickness even if such incidences are both sporadic and brief, and the problem may even be worse under these conditions than when no motion cues are provided at all. Thirdly, in this work we have developed a model for locomotion control via joystick movement that evokes more natural traversal behaviors through a virtual environment than we have observed in the past using a simpler and more direct mapping from joystick position to changes in virtual viewpoint. By constraining the joystick to control participants virtual movement in an identical fashion as it would control the actual movement of a wheelchair, we created affordances that encouraged people to travel in ways that more closely matched the characteristics of the travel paths they naturally chose when walking. Our observations are resonant with the findings of [134], who observed the importance of providing people with properly-integrated control over both the rotational and translational components of motion in order to facilitate natural engagement and support effective user interaction.

5.6 General Discussion & Conclusion

In the course of this research we wished to determine if redirection techniques could be applied to wheeled motion platforms, in what we are referring to as "redirected driving". Redirected driving allows for exploration of virtual environments that are larger than the available physical space. We have shown that passive, visual-only, redirection can be applied to driving VR and can achieve similar levels of redirection as that of redirected walking. We have also demonstrated remote control of a wheeled motion platform that enables active redirection wherein user input is modified or discarded and platform motion is computer controlled. Because this setup allows for precise control of physical velocities we have been able to show a connection between the speed of travel and the level of perception of the redirection. Specifically, when movements are slower the radius of redirection can be larger at the same rate of correct perception.

Active redirection methods also open the door for many interesting areas of future research. Among them is the possibility of failure-free redirection. In this scenario the system could completely uncouple the user-controlled virtual motion from the motion of the wheeled platform in order to avoid physical obstacles such as walls. The key point that will need to be investigated to make this a reality is to accurately, and precisely, describe the conditions under which this uncoupling leads to cybersickness. In our experiments we saw that a slight decoupling can lead to increased rates of cybersickness, however a complete decoupling, as in the case of visual-only and no movement, produces somewhat lower rates of cybersickness. Finding the exact stimuli that cause cybersickness in these IVEs is critical to the future adoption of this technology.

6

Conclusion

In this dissertation we have presented the research contributions of three different but complementary research avenues. In this chapter we summarize the contributions of this dissertation and discuss possible avenues for future research.

6.1 Summary of Contributions

The goal for this dissertation was to increase the effectiveness of immersive virtual environments by developing novel techniques and technologies to overcome the problems still facing virtual reality. Specifically, the dissertation focused on showing that immersive virtual environments can be made more effective through the development of enhanced locomotion and interaction methods that provide more accurate visual and vestibular feedback to the user. The three key areas investigated were immersion, collaboration, and locomotion.

6.1.1 Immersion

In Chapter 3 we investigated the use if video cameras to create self-avatars for users exploring immersive virtual environments. Prior research had shown the benefit of avatars in virtual reality with respect to immersion and task performance in the virtual environment. An example is the reduction in egocentric distance underestimation when an avatar is present versus no avatar. However, despite the benefits of avatars they're not the default in IVE applications. The construction of avatar models takes time

and requires artistic talent. Generally, a handful of generic avatars are created and the selected from prior to using the IVE. Also, motion captured avatars require the wearing of a motion capture suit with reflective markers and a time-consuming, per-user calibration procedure. Our goal was to leverage the recent advancements in video camera technology, such as RGB+D cameras, to create avatars that are quick to setup and match the user's physical appearance far better than a pre-made generic avatar.

We first investigated using simple off-the-shelf RGB webcams attached to our existing HMD in order to capture color video from the perspective of the wearer. Several segmentation and compositing algorithms were developed and evaluated to combine the video signal with the renderings of the virtual environment. From these early results we arrived at the conclusion that two-dimensional methods alone would not be enough to accurately create video self-avatars. Two-dimensional compositing has several limitations. First, is the problem of segmenting the hands from the background environment. In an arbitrary environment there will be situations where segmentation will fail and background pixels in the image will appear indistinguishable from the pixels corresponding to the user's body. Also, the camera capturing the view must be precisely oriented so as to share the optical axis with the head-mounted display, otherwise the composition of two images rendered with different perspectives will make the hands appear to float over the virtual scene instead of being a part of it. Finally, interaction with the virtual environment becomes difficult as we do not know the 3D location of the user's hands. For example, the hands may physically be positioned behind a virtual object but the composition will be unable to take this z-ordering into account and instead will render the hands above the virtual object.

We therefore set out to investigate the use of the recently available RGB+D color and depth cameras. Both structured light cameras, such as the Kinect 1, and time-of-flight cameras were investigated. The experiments with the Kinect 1 were promising, but due to the size of the camera it needed to be mounted in the physical space and not attached the HMD as with the earlier webcams. This led to position-dependent resolution issues, whereby the user's avatar would fluctuate in fidelity depending on where they were in the room with respect to the Kinect sensor. Also, the limited range of the Kinect 1 would require several sensors to cover the typical virtual reality lab space. In the end, we developed a system using small infrared time-of-flight cameras that could be attached

to the HMD and view the room from the perspective of the user to create a video-based self-avatar. To validate this setup, we ran user studies involving the measurement of distance in virtual reality for both walking and reaching. The experiments tested three separate conditions: no avatar, motion-captured avatar, and video-based avatar. While the long range blind-walking estimation results were statistically insignificant, the short-range case of blind-reaching was statistically significant ($p = 0.051$), showing that both motion-capture and video avatars greatly improve distance estimation in short-range estimation tasks.

6.1.2 Locomotion

In Chapter 5 we developed an electric wheelchair based motion platform by modifying a Hoveround MPV5 electric wheelchair to operate under computer control. This allowed the computer to control the physical movement of the wheelchair based on the virtual movement of the view in the virtual environment. We conducted several user studies comparing the effectiveness of this motion platform to the state-of-the-art methods of redirected walking and found it to be comparable in the amount of redirection capability. We hypothesize that such a motion platform has advantages over redirected walking because it could allow recovering from failure states in a more natural manner that does not break the user's feeling of immersion in the virtual environment, however studies to show this remain an area of future work.

6.1.3 Collaboration

In Chapter 4 we enumerated the benefits of a well-designed and well-calibrated large-screen immersive display (LSID) and discussed the challenges related to the calibration of arbitrary displays. We described the system we have developed, based on existing state-of-the-art research in structured lighting, structure from motion, and shader-based rendering that allows for easy and effective calibration of arbitrary tiled projection displays. We have used this software to calibrate the display surface in the DDC VR laboratory and have made it open-source on the world-wide web so it may benefit other research groups faced with this problem.

6.2 Future Work

The work presented in this dissertation opens several interesting avenues for future research. Many of these directions are suggested by the limitations encountered in the course of our research, and are discussed below within their respective sections.

6.2.1 Immersion

The video self-avatars we created showed promise as a way to easily and quickly allow users to see their body within the immersive virtual environment. We have shown that for some tasks the avatars are just as effective as motion-capture based avatars that require a far greater investment in equipment and setup time. One avenue for future research is to examine a range of tasks, such as architectural design review, scientific visualization, and teleconferencing, and determine the extent to which this equivalence holds.

Once users can see their real hands in the virtual environment, a natural extension is to allow interaction with the virtual environment using those hands. This interaction can be passive, such as by pointing or gesturing, or active through the grasping or touching of virtual objects. Therefore the integration of hand tracking and gesture recognition into the RGB+D avatars described in this dissertation is an ideal avenue of future research. For grasping and touching virtual objects, an anticipated problem is the physical proprioceptive mismatch that will occur when attempting to touch an object that has no physical manifestation. A possible direction of research is the use of "dummy" objects for passive haptic feedback [135, 136]. The extensive literature from the fields of psychology and neuroscience on the rubber-hand illusions [137–139] as well as the virtual reality research on redirected touching by Kohli *et al.* [140] in which user's are made to believe they are touching a curved or slanted table where in fact it is perfectly flat and level, may prove helpful in this area in order to create realistic haptic feedback for objects that do not exist, and allow a single passive haptic object to take the place of several virtual objects.

6.2.2 Collaboration

There is a wealth of knowledge on how to calibrate various types of screens and cameras in the virtual reality and computer vision communities. That being the case, however, it is often that bespoke systems are designed and created for each virtual reality installation or application. We have tried to create an open source package for calibration of projector screens that can be reused by others in the field, but this is only the first step. Extending this package to other screens and actually testing its use, is an immediate short-term future research direction.

6.2.3 Locomotion

The active redirection platform we described in this dissertation opens the door for many interesting areas of future research. Chief among them is the possibility of failure-free redirection. In such a scenario the system could completely uncouple the user-controlled visual motion from the motion of the wheel platform in order to avoid physical obstacles such as walls. The difficult problem which remains unsolved is how to perform this uncoupling in such a way that does not break presence and immersion in the virtual environment and, at a more practical level, does not leave the user incredibly cybersick. In our experiments we saw some evidence that a small uncoupling, such as rotation or translation only conditions, can lead to increased rates of cybersickness, but complete decoupling, such as visual-only condition, produces a much lower rate of cybersickness. Determining the exact cause of this differences remains and area of active future research.

Acronyms

2AFC	Two Alternative Forced Choice.
A2D	Analog to Digital.
ANCOVA	Analysis of Covariance.
ANOVA	Analysis of Variance.
CAVE	Cave Automatic Virtual Environment.
COTS	Commercial Off-The-Shelf.
CPU	Central Processing Unit.
D2A	Digital to Analog.
DDC	Digital Design Consortium.
fMRI	Functional Magnetic Resonance Imaging.
FOV	Field of View.
GPU	Graphics Processing Unit.
GUI	Graphical User Interface.
HMD	Head-Mounted Display.
HSV	Hue Saturation Value (color image).
IPD	Inter-Pupillary Distance.

IQR	Inter-Quartile Range.
IR	Infrared.
IVE	Immersive Virtual Environment.
LCD	Liquid Crystal Display.
LCOS	Liquid Crystal on Silicon.
LSID	Large-Screen Immersive Display.
PID	Proportional Integral Derivative (controller).
PSE	Point of Subjective Equality.
PTSD	Post-Traumatic Stress Disorder.
PWM	Pulse Width Modulation.
RGB	Red Green Blue (color image).
RGB+D	Color (Red-Green-Blue) and Depth.
SfM	Structure from Motion.
SSQ	Simulator Sickness Questionnaire.
VR	Virtual Reality.
VSTHMD	Video See-Thru Head-Mounted Display.

References

[1] Amir Raz, Mark G Packard, Gerianne M Alexander, Jason T Buhle, Hongtu Zhu, Shan Yu, and Bradley S Peterson. A slice of π: An exploratory neuroimaging study of digit encoding and retrieval in a superior memorist. *Neurocase*, 15(5):361–372, 2009.

[2] Eleanor A Maguire, Elizabeth R Valentine, John M Wilding, and Narinder Kapur. Routes to remembering: the brains behind superior memory. *Nature neuroscience*, 6(1):90–95, 2003.

[3] R. A. Ruddle and S. Lessels. The benefits of using a walking interface to navigate virtual environments. *ACM Transactions on Computer-Human Interaction*, 16(1):18, 2009.

[4] Roy A Ruddle, Stephen J Payne, and Dylan M Jones. Navigating large-scale virtual environments: what differences occur between helmet-mounted and desktop displays? *Presence*, 8(2):157–168, 1999.

[5] B.E. Riecke, B. Bodenheimer, T. P. McNamara, B. Williams, P. Peng, and D. Feureissen. Do we need to walk for effective virtual reality navigation? physical rotations alone may suffice. In *Proceedings of the 7th International Conference on Spatial Cognition*, pages 234–247, 2010.

[6] R.A. Ruddle and S. Lessels. For efficient navigational search, humans require full physical movement but not rich visual scene. *Psychological Science*, 17(6):460–465, 2006.

[7] Mattias Roupé, Petra Bosch-Sijtsema, and Mikael Johansson. Interactive navigation interface for virtual reality using the human body. *Computers, Environment and Urban Systems*, 43:42–50, 2014.

[8] Frederick P. Brooks. What's real about virtual reality? *IEEE Computer Graphics and Applications*, 19(6):16–27, November 1999.

[9] Lee Anderson, James Esser, and Victoria Interrante. A virtual environment for conceptual design in architecture. In *Proceedings of the workshop on Virtual environments 2003*, pages 57–63. ACM, 2003.

[10] Mel Slater and Martin Usoh. Body centred interaction in immersive virtual environments. In N Magnenat Thalmann and D. Thalmann, editors, *Artificial Life and Virtual Reality*, pages 125–148. John Wiley and Sons, 1994.

[11] Robert T Held and Martin S Banks. Misperceptions in stereoscopic displays: a vision science perspective. In *Proceedings of the 5th symposium on applied perception in graphics and visualization*, pages 23–32, 2008.

[12] M. C. Whitton, J. Cohn, P. Feasel, S. Zimmons, S. Razzaque, B. Poulton, B. McLeod, and F. Brooks. Comparing ve locomotion interfaces. In *Proceedings of IEEE Virtual Reality*, pages 123–130, March 2005.

[13] B. G. Witmer, J. H. Bailey, B. W. Knerr, and K. C. Parsons. Virtual spaces and real world places: transfer of route knowledge. *International Journal on Human-Computer Studies*, 45(4):413–428, October 1996.

[14] E. A. Suma, S. L. Finkelstein, M. Reid, S. V. Babu, A. C. Ulinski, and L. F. Hodges. Evaluation of the cognitive effects of travel technique in complex real and virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 16(4):690–702, 2010.

[15] S. Razzaque, Z. Kohn, and M. C. Whitton. Redirected walking. In *Proceedings of Eurographics Short Papers*, 2001.

[16] NVIS Incorporated. http://www.nvisinc.org/.

[17] Oculus Rift. https://www.oculus.com/rift/.

[18] Samsung Gear VR. http://www.samsung.com/global/microsite/gearvr/gearvr_features.html.

[19] Google Cardboard. https://www.google.com/get/cardboard/.

[20] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–72, 1992.

[21] 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 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.

[22] Mechdyne Corporation. Mechdyne corporation : About us. http://www.mechdyne.com/about-why-choose-mechdyne.aspx.

[23] CAVE2: Next-generation virtual-reality and visualization hybrid environment for immersive simulation and information analysis. https://www.evl.uic.edu/entry.php?id=2016.

[24] Vicon MX cameras. http://www.vicon.com/products/viconmx.html.

[25] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci. High-performance wide-area optical tracking: The hiball tracking system. *presence: teleoperators and virtual environments*, 10(1):1–21, 2001.

[26] WorldViz Vizard. http://www.worldviz.com/products/vizard.

[27] G3D Innovation Engine. http://g3d.sourceforge.net/.

[28] Unity3D. http://unity3d.com/.

[29] Unreal Engine. https://www.unrealengine.com/what-is-unreal-engine-4.

[30] Benjamin Lok, Samir Naik, Mary Whitton, and Frederick Brooks. Effects of handling real objects and self-avatar fidelity on cognitive task performance and sense of presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, 12(6):615–628, 2003.

[31] Michael Brown, Aditi Majumder, and Ruigang Yang. Camera-based calibration techniques for seamless multiprojector displays. *Visualization and Computer Graphics, IEEE Transactions on*, 11(2):193–206, 2005.

[32] Han Chen, Rahul Sukthankar, Grant Wallace, and Kai Li. Scalable alignment of large-format multi-projector displays using camera homography trees. In *Proceedings of the conference on Visualization '02*, pages 339–346. IEEE Computer Society, 2002.

[33] Yuqun Chen, Douglas W Clark, Adam Finkelstein, Timothy C Housel, and Kai Li. Automatic alignment of high-resolution multi-projector display using an uncalibrated camera. In *Proceedings of the conference on Visualization '00*, pages 125–130. IEEE Computer Society, 2000.

[34] Mark Hereld, Ivan R Judson, and Rick Stevens. Dottytoto: a measurement engine for aligning multiprojector display systems. In *Electronic Imaging 2003*, pages 73–86. International Society for Optics and Photonics, 2003.

[35] Aditi Majumder and Rick Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):177–188, 2004.

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

[37] Ramesh Raskar, Michael S Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Scales, and Henry Fuchs. Multi-projector displays using camera-based registration. In *Visualization'99. Proceedings*, pages 161–522. IEEE, 1999.

[38] Ramesh Raskar, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. ilamps: geometrically aware and self-configuring projectors. In *ACM Transactions on Graphics (TOG)*, pages 809–818. ACM, 2003.

[39] Aditi Majumder, David Jones, Matthew McCrory, Michael E Papka, and Rick Stevens. Using a camera to capture and correct spatial photometric variation in

multi-projector displays. In *IEEE International Workshop on Projector-Camera Systems*, 2003.

[40] David Waller, Eric Bachmann, Eric Hodgson, and Andrew C Beall. The hive: A huge immersive virtual environment for research in spatial cognition. *Behavior Research Methods*, 39(4):835–843, 2007.

[41] Warren Robinett and Richard Holloway. Implementation of flying, scaling and grabbing in virtual worlds. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 189–192. ACM, 1992.

[42] Mel Slater, Martin Usoh, and Anthony Steed. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):201–219, 1995.

[43] Martin Usoh, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P Brooks Jr. Walking¿ walking-in-place¿ flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 359–364. ACM Press/Addison-Wesley Publishing Co., 1999.

[44] Victoria Interrante, Brian Ries, and Lee Anderson. Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In *3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on*. IEEE, 2007.

[45] J. L. Souman, P. Robuffo Giordano, M. Schwaiger, I. Frissen, T. Thümmel, H. Ulbrich, A. De Luca, H. H. Bülthoff, and M. O. Ernst. Cyberwalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Transactions on Applied Perception*, 8(4):22, December 2008.

[46] Hiroo Iwata, Hiroaki Yano, Hiroyuki Fukushima, and Haruo Noma. Circulafloor: A locomotion interface using circulation of movable tiles. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 223–230, Washington, DC, USA, 2005. IEEE Computer Society.

[47] Eliana Medina, Ruth Fruland, and Suzanne Weghorst. Virtusphere: walking in a human size vr "hamster ball". In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 52, pages 2102–2106. SAGE Publications, 2008.

[48] Frank Steinicke, Gerd Bruder, J. Jerald, H. Fenz, and M. Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 16(1):17–27, 2010.

[49] Lane Phillips, Brian Ries, Michael Kaeding, and Victoria Interrante. Avatar self-embodiment enhances distance perception accuracy in non-photorealistic immersive virtual environments. *Proceedings of the 2010 IEEE Virtual Reality Conference (VR)*, pages 115–118, mar 2010.

[50] Betty J Mohler, Sarah H Creem-Regehr, William B Thompson, and Heinrich H Bülthoff. The effect of viewing a self-avatar on distance judgments in an hmd-based virtual environment. *Presence: Teleoperators and Virtual Environments*, 19(3):230–242, jun 2010.

[51] Brian Ries, Victoria Interrante, Michael Kaeding, and Lane Phillips. Analyzing the effect of a virtual avatar's geometric and motion fidelity on ego-centric spatial perception in immersive virtual environments. *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST '09)*, 1(212):59–66, 2009.

[52] Andrew State, Kurtis P. Keller, and Henry Fuchs. Simulation-based design and rapid prototyping of a parallax-free, orthoscopic video see-through head-mounted display. In *Proceedings of the 2005 International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 28–31, 2005.

[53] Loren Puchalla Fiore and Victoria Interrante. Towards achieving robust video self-avatars under flexible environment conditions. In *1st Workshop on Off-The-Shelf VR, IEEE VR 2011*, 2011.

[54] Loren Puchalla Fiore and Victoria Interrante. Towards achieving robust video self-avatars under flexible environment conditions. *International Journal of Virtual Reality*, 11(3):33–41, 2012.

[55] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–54, jan 2005.

[56] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 200.

[57] Daniel Henry and Tom Furness. Spatial perception in virtual environments: Evaluating an architectural application. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 33–40. IEEE, 1993.

[58] Rebekka S Renner, Boris M Velichkovsky, and Jens R Helmert. The perception of egocentric distances in virtual environments-a review. *ACM Computing Surveys (CSUR)*, 46(2):23, 2013.

[59] David Waller. Factors affecting the perception of interobject distances in virtual environments. *Presence: Teleoperators and Virtual Environments*, 8(6):657–670, 1999.

[60] M. Leyrer, S. A. Linkenauger, H. H. Bulthoff, U. Kloos, and B. Mohler. The influence of eye height and avatars on egocentric distance estimates in immersive virtual environments. In *Proceedings of ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 67–74, August 2011.

[61] Qiufeng Lin, Xianshi Xie, Aysu Erdemir, Gayathri Narasimham, Timothy P McNamara, John Rieser, and Bobby Bodenheimer. Egocentric distance perception in real and hmd-based virtual environments: the effect of limited scanning method. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 75–82. ACM, 2011.

[62] Erin A McManus, Bobby Bodenheimer, Stephan Streuber, Stephan de la Rosa, Heinrich H Bülthoff, and Betty J Mohler. The influence of avatar (self and character) animations on distance estimation, object interaction and locomotion in

immersive virtual environments. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 37–44. ACM, 2011.

[63] Benjamin Petit, Jean-Denis Lesage, Clément Menier, Jérémie Allard, Jean-Sébastien Franco, Bruno Raffin, Edmond Boyer, and François Faure. Multicamera real-time 3d modeling for telepresence and remote collaboration. *International Journal of Digital Multimedia Broadcasting*, 2010:1–12, 2010.

[64] Gerd Bruder, Frank Steinicke, Kai Rothaus, and Klaus Hinrichs. Enhancing presence in head-mounted display environments by visual body feedback using head-mounted cameras. *2009 International Conference on CyberWorlds*, pages 43–50, 2009.

[65] xxarray photobooth. http://makezine.com/2014/01/09/nikon-xxarray/.

[66] A. Weiss, D. Hirshberg, and M.J. Black. Home 3D body scans from noisy image and range data. In *Int. Conf. on Computer Vision (ICCV)*, pages 1951–1958, Barcelona, nov 2011. IEEE.

[67] Evan Suma, David M. Krum, and Mark Bolas. Sharing space in mixed and virtual reality environments using a low-cost depth sensor. In *Proceedings of the IEEE International Symposium on Virtual Reality Innovations*, 2011.

[68] Andrew Maimone and Henry Fuchs. Reducing interference between multiple structured light depth sensors using motion. In *Proceedings of IEEE VR*, pages 51–54, 2012.

[69] Andrew Maimone, Jonathan Bidwell, Kun Peng, and Henry Fuchs. Enhanced personal autostereoscopic telepresence system using commodity depth cameras. *Computers & Graphics*, 36, May 2012.

[70] Mingsong Dou and Henry Fuchs. Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In *Proceedings of IEEE VR*, 2014.

[71] S. Beck, A. Kunert, A. Kulik, and B. Froehlich. Immersive group-to-group telepresence. *IEEE transactions on visualization and computer graphics*, 19(4):616–625, 2013.

[72] Creative senz3d depth camera. http://sg.creative.com/p/web-cameras/creative-senz3d.

[73] pmd[vision] camBoard nano. https://www.cayim.com/.

[74] Brian Ries. *Facilitating effective virtual reality for architectural design.* PhD thesis, University of Minnesota, 2011.

[75] Victoria Interrante, Brian Ries, and Lee Anderson. Distance perception in immersive virtual environments, revisited. In *Proceedings of IEEE Conference on Virtual Reality*, pages 3–10. IEEE, 2006.

[76] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology*, 1993.

[77] Franco Tecchia, Giovanni Avveduto, Raffaello Brondi, Marcello Carrozzino, Massimo Bergamasco, and Leila Alem. I'm in vr!: using your own hands in a fully immersive mr system. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 73–76. ACM, 2014.

[78] Taejin Ha, Steven Feiner, and Woontack Woo. Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 219–228. IEEE, 2014.

[79] A. Kulik, A. Kunert, S. Beck, R. Reichel, R. Blach, A. Zink, and B. Froehlich. C1x6: A stereoscopic six-user-display for co-located collaboration in shared virtual environments. *ACM Transactions on Graphics*, 30(6), December 2011.

[80] D. Vishwanath, A. R. Girshick, and M. S. Banks. Why pictures look right when viewed from the wrong place. *Nature Neuroscience*, 8(10):1401–1410, 2005.

[81] M. S. Banks, R. T. Held, and A. R. Girshick. Perception of 3-d layout in stereo displays. *Information Display*, 25(1):12–16, 2009.

[82] Brice Pollock, Melissa Burton, Jonathan W Kelly, Stephen Gilbert, and Eliot Winer. The right view from the wrong location: Depth perception in stereoscopic multi-user virtual environments. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):581–588, 2012.

[83] Ross Treddinick, Lee Anderson, Brian Ries, and Victoria Interrante. A tablet based immersive architectural design tool. In *Proceedings of the 25th Annual Conference for the Association for Computer-Aided Design in Architecture*, pages 328–340, 2006.

[84] Markus Gross, Stephan Würmlin, Martin Naef, Edouard Lamboray, Christian Spagno, Andreas Kunz, Esther Koller-Meier, Tomas Svoboda, Luc Van Gool, Silke Lang, et al. blue-c: a spatially immersive display and 3d video portal for telepresence. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 819–827. ACM, 2003.

[85] Henry Fuchs, , Andrei State, and Jean-Charles Bazin. Immersive 3d telepresence. *Computer*, (7):46–52, 2014.

[86] Greg Humphreys, Matthew Eldridge, Ian Buck, Gordan Stoll, Matthew Everett, and Pat Hanrahan. WireGL: a scalable graphics system for clusters. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 129–140. ACM, 2001.

[87] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D Kirchner, and James T Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 693–702. ACM, 2002.

[88] Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, and Carolina Cruz-Neira. VR Juggler: A virtual platform for virtual reality application development. In *Proceedings of the IEEE Conference on Virtual Reality, 2001.*, pages 89–96. IEEE, 2001.

[89] Benjamin Schaeffer and Camille Goudeseune. Syzygy: native pc cluster vr. In *Proceedings of the IEEE Conference on Virtual Reality, 2003.*, pages 15–22. IEEE, 2003.

[90] Ignacio Garcia-Dorado and Jeremy Cooperstock. Fully automatic multi-projector calibration with an uncalibrated camera. In *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 29–36. IEEE, 2011.

[91] Behzad Sajadi and Aditi Majumder. Autocalibration of multiproject CAVE-like immersive environments. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(3):381–393, 2012.

[92] Ezekiel S Bhasker and Aditi Majumder. Geometric modeling and calibration of planar multi-projector displays using rational bezier patches. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. (CVPR)*, pages 1–8. IEEE, 2007.

[93] Aditi Majumder. Properties of color variation across multi-projector displays. *Proceedings of SID Eurodisplay*, 2002:3, 2002.

[94] Aditi Majumder. Contrast enhancement of multi-displays using human contrast sensitivity. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).*, volume 2, pages 377–382. IEEE, 2005.

[95] Aditi Majumder, Zhu He, Herman Towles, and Greg Welch. Achieving color uniformity across multi-projector displays. In *Proceedings of Visualization*, pages 117–124. IEEE, 2000.

[96] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2), 2007.

[97] Sfm-toy-library. https://github.com/royshil/SfM-Toy-Library.

[98] S. Garrido-Jurado, R. Muaz-Salinas, F. J. Madrid-Cuevas, and M. J. Marin-Jimenez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[99] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4, 1987.

[100] D. R. Montello. A conceptual model of the cognitive processing of environmental distance information. In *Proceedings of 9th International Conference on Spatial Information Theory*, pages 1–17, September 2009.

[101] D. B. Rothman and W. H. Warren. Wormholes in virtual reality and the geometry of cognitive maps. *Journal of Vision*, 6(6), June 2006.

[102] R. F. Wang. Theories of spatial representations and reference frames: What can configuration errors tell us? *Psychonomic Bulletin and Review*, 19(4):575–587, August 2012.

[103] T. Meilinger. The network of reference frames theory: a synthesis of graphs and cognitive maps. In *Proceedings of International Conference on Spatial Cognition VI*, pages 344–360, September 2008.

[104] J. Ericson and W. H. Warren. Rips and folds in virtual space: ordinal violations in human spatial knowledge. *Journal of Vision*, 9(8), 2009.

[105] B. E. Riecke, M. von der Heyde, and H. H. Bülthoff. Spatial updating in real and virtual environments - contribution and interaction of visual and vestibular cues. In *Proceedings of Symposium on Applied Perception in Graphics and Visualization*, pages 9–17, August 2004.

[106] M. J. Wraga, S. H. Creem-Regehr, and D. R. Proffitt. Spatial updating of virtual displays during self and display rotation. *Memory and Cognition*, 32(3):399–415, 2004.

[107] J. J. Rieser. Access to knowledge of spatial structure at novel points of observation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 15(6):1157–1165, November 1989.

[108] J. M. Loomis, R. L. Klatzky, and R. G. Golledge. Navigating without vision: basic and applied research. *Optometry and Vision Science*, 78(5):282–289, May 2011.

[109] S. S. Chance, F. Gaunet, A. C. Beall, and J. M. Loomis. Locomotion mode affects the updating of objects encountered during travel: the contribution of vestibular and proprioceptive inputs to path integration. *Presence: Teleoperators and Virtual Environments*, 7(2):168–178, April 1998.

[110] C. Zanbaka, B. Lok, S. Babu, D Xiao, A. Ulinski, and L. F. Hodges. Effects of travel technique on cognition in virtual environments. In *Proceedings of IEEE Virtual Reality*, pages 149–156, March 2004.

[111] S. Sigurdarson, A. P. Milne, D. Feuereissen, and B. E. Riecke. Can physical motions prevent disorientation in naturalistic vr? In *Proceedings of IEEE Virtual Reality*, pages 31–34, March 2012.

[112] B. E. Riecke, D. Feuereissen, J. J. Rieser, and T. P. McNamara. Self-motion illusions (vection) in vr – are they good for anything? In *Proceedings of IEEE Virtual Reality*, pages 35–38, March 2012.

[113] R.A. Ruddle, E. Volkova, and H. H. Bülthoff. Learning to walk in virtual reality. *ACM Transactions on Applied Perception*, 10(2):17, May 2013.

[114] R. A. Ruddle. *The effect of translational and rotational body-based information on navigation.* Springer, 2013.

[115] J. N. Templeman, P. S. Denbrook, and L. E. Sibert. Virtual locomotion: walking in place through virtual environments. *Presence: Teleoperators and Virtual Environments*, 8(6):598–617, December 1999.

[116] H. J. Teufel, H. G. Nusseck, K. A. Beykirch, J. S. Butler, M. Kerger, and H. H. Bülthoff. Mpi motion simulator: development and analysis of a novel motion simulator. In *Proceedings of AIAA Modeling and Simulation Technical Conference and Exhibit*, volume AIAA-2007-6476, 2007.

[117] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas. A taxonomy for deploying redirection techniques in immersive virtual environments. In *Proceedings of IEEE Virtual Reality*, pages 43–46, March 2012.

[118] T. C. Peck, H. Fuchs, and M. C. Whitton. Improved redirection with distractors: a large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In *Proceedings of IEEE Virtual Reality*, pages 35–38, March 2010.

[119] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. J. Rieser, and B. Bodenheimer. Exploring large virtual environments with an hmd when physical space is limited. In *Proceedings of Symposium on Applied Percpetion and Graphics Visualization*, pages 41–48, July 2007.

[120] E. Hodgson, E. Bachmann, and D. Waller. Redirected walking to explore virtual environments: assessing the potential for spatial interference. *ACM Transactions on Applied Perception*, 8(4):22:1–22:22, November 2011.

[121] T. C. Peck, H. Fuchs, and M. C. Whitton. An evaluation of navigational ability comparing redirected free exploration with distractors to walking-in-place and joystick locomotion interfaces. In *Proceedings of IEEE Virtual Reality*, pages 55–62, March 2011.

[122] J. L. Campos and H. H. Bülthoff. *Multimodal integration during self-motion in virtual reality.* CRC Press, 2012.

[123] Gerd Bruder, Victoria Interrante, Lane Phillips, and Frank Steinicke. Redirecting walking and driving for natural navigation in immersive virtual environments. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(4):538–545, 2012.

[124] A. Kemeny and F. Panerai. Evaluating perception in driving simulation experiments. *Trends in Cog. Sci.*, 7(1):31–37, January 2003.

[125] J. T. Reason. Motion sickness adaptation: a neural mismatch model. *Journal of Royal Society of Medicine*, 71(11):819–829, November 1978.

[126] M. H. Draper. *The adaptive effects of virtual interfaces: vestibulo-ocular reflex and simulator sickness.* PhD thesis, University of Washington, 1998.

[127] K. M. Stanney and P. Hash. Locus of user-initiated control in virtual environments: influences on cybersickness. *Presence: Teleoperators and Virtual Environments*, 7(5):447–459, October 1998.

[128] E. M. Kolasinski. Simulator sickness in virtual environments. Technical report, US Army Research Institute, May 1995.

[129] A. Nybakke, R. Ramakrishnan, and Victoria Interrante. From virtual to actual mobility: assessing the benefits of active locomotion through an immersive virtual environment using a motorized wheelchair. In *Proceedings of IEEE Symposium on 3D User Interfaces*, pages 27–30, 2012.

[130] Loren Puchalla Fiore, Lane Phillips, Gerd Bruder, Victoria Interrante, and Frank Steinicke. Redirected steering for virtual self-motion control with a motorized electric wheelchair. In *Proceedings of the Joint Virtual Reality Conference of ICAT-EGVE-EuroVR*, pages 45–48, 2012.

[131] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bülthoff, and B. J. Mohler. Velocity-dependent dynamic curvature gain for redirected walking. In *Proceedings of IEEE VR*, pages 1–8, 2011.

[132] Loren Puchalla Fiore, Ella Coben, Samantha Merritt, Peng Liu, and Victoria Interrante. Towards enabling more effective locomotion in vr using a wheelchair-based motion platform. In *Proceedings of the Joint Virtual Reality Conference of ICAT-EGVE-EuroVR*, pages 83–90, 2013.

[133] A. Steed. A simple method for estimating the latency of interactive, real-time graphics simulations. In *Proceedings Virtual Reality Software and Technology*, pages 123–129, October 2008.

[134] B. Froehlich, J. Hochstrate, V. Skuk, and A. Huckauf. The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of ACM SIGCHI*, pages 191–199, April 2006.

[135] Ken Hinckley, Randy Pausch, John C Goble, and Neal F Kassell. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 452–458. ACM, 1994.

[136] Brent Edward Insko. *Passive haptics significantly enhances virtual environments.* PhD thesis, University of North Carolina at Chapel Hill, 2001.

[137] Wijnand IJsselsteijn, Yvonne A. W. de Kort, and Antal Haans. Is this my hand i see before me? the rubber hand illusion in reality, virtual reality, and mixed reality. *Presence*, 15(4):455–464, 2006.

[138] Manos Tsakiris and Patrick Haggard. The rubber hand illusion revisited: visuotactile integration and self-attribution. *Journal of Experimental Psychology: Human Perception and Performance*, 31(1):80, 2005.

[139] Daniel Perez-Marcos, Maria V Sanchez-Vives, and Mel Slater. Is my hand connected to my body? the impact of body continuity and arm alignment on the virtual hand illusion. *Cognitive neurodynamics*, 6(4):295–305, 2012.

[140] Luv Kohli. *Redirected Touching.* PhD thesis, University of North Carolina at Chapel Hill, 2013.

Appendix A

Video-based Avatar Experiment Data

Table A.1: Relative Error - VR Blind Walking

ID	Avatar	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15
1	**RGBD**	-0.35	-0.52	-0.43	-0.41	-0.55	-0.35	-0.36	-0.17	-0.43	-0.52	-0.36	-0.23	-0.40	-0.27	-0.23
2	**None**	0.11	0.11	0.04	0.00	0.09	0.07	0.05	0.03	0.04	0.15	0.03	0.11	-0.48	-0.12	-0.09
3	**RGBD**	-0.37	-0.34	-0.19	-0.08	-0.29	-0.25	-0.23	-0.19	-0.09	-0.06	-0.17	-0.09	-0.19	-0.04	-0.03
4	**Vicon**	-0.30	-0.25	-0.15	-0.20	-0.08	-0.08	-0.19	-0.09	-0.10	-0.15	-0.08	-0.10	-0.05	-0.16	-0.14
5	**Vicon**	-0.17	-0.22	-0.25	-0.30	-0.37	-0.31	-0.33	-0.23	-0.27	-0.26	-0.26	-0.13	-0.19	-0.26	-0.14
6	**None**	-0.06	-0.21	-0.13	-0.24	-0.37	-0.30	-0.20	-0.17	-0.27	-0.24	-0.19	-0.15	-0.26	-0.15	-0.18
7	**Vicon**	-0.14	-0.16	-0.01	-0.08	-0.14	-0.11	-0.20	-0.10	-0.22	-0.25	-0.27	-0.06	-0.08	-0.10	-0.15
8	**RGBD**	-0.06	0.02	-0.05	0.17	0.05	0.04	0.05	0.07	-0.11	0.01	-0.41	-0.02	0.06	0.11	0.14
9	**RGBD**	-0.04	-0.23	-0.12	-0.11	-0.10	0.05	0.00	-0.98	-0.06	-0.09	-0.01	0.02	-0.06	0.01	-0.01
10	**Vicon**	-0.32	-0.41	-0.41	-0.44	-0.35	-0.32	-0.49	-0.25	-0.37	-0.27	-0.28	-0.27	-0.32	-0.31	-0.26
11	**None**	-0.07	-0.15	-0.20	-0.21	-0.20	-0.16	-0.11	-0.30	-0.26	-0.33	-0.38	-0.21	-0.16	-0.25	-0.15
12	**None**	-0.39	-0.37	-0.36	-0.34	-0.29	-0.38	-0.40	-0.38	-0.34	-0.39	-0.34	-0.31	-0.28	-0.30	-0.39
13	**RGBD**	-0.38	-0.35	-0.31	-0.36	-0.30	-0.40	-0.30	-0.25	-0.14	-0.25	-0.28	-0.21	-0.25	-0.30	-0.26
14	**Vicon**	-0.23	-0.13	-0.19	-0.15	-0.48	-0.17	-0.15	-0.01	-0.01	-0.04	-0.20	-0.12	0.11	-0.04	-0.03
15	**RGBD**	-0.24	-0.20	-0.16	-0.19	-0.27	-0.33	-0.23	-0.28	-0.32	-0.12	-0.32	-0.22	-0.28	-0.32	-0.23
16	**Vicon**	-0.55	-0.46	-0.50	-0.40	-0.55	-0.54	-0.58	-0.52	-0.45	-0.58	-0.48	-0.53	-0.53	-0.58	-0.52
17	**RGBD**	-0.19	-0.40	-0.36	-0.25	-0.09	-0.31	-0.03	-0.13	0.04	0.05	0.09	0.12	0.00	-0.19	0.11
18	**None**	-0.16	-0.21	-0.26	-0.25	-0.20	-0.33	-0.26	-0.36	-0.24	-0.19	-0.14	-0.16	-0.23	-0.20	-0.12
19	**None**	-0.13	-0.04	0.02	-0.02	-0.13	-0.14	-0.10	-0.18	-0.09	-0.04	-0.05	-0.12	-0.03	-0.16	0.07
20	**None**	-0.43	-0.38	-0.20	-0.37	-0.24	-0.34	-0.28	-0.20	-0.32	-0.30	-0.30	-0.37	-0.31	-0.37	-0.23
21	**None**	-0.38	-0.43	-0.20	-0.36	-0.23	-0.28	-0.26	-0.25	-0.22	-0.25	-0.28	-0.31	-0.25	-0.36	-0.36
22	**Vicon**	-0.08	-0.01	-0.13	-0.05	0.03	-0.04	0.02	-0.06	-0.01	0.19	-0.18	-0.03	0.09	0.03	-0.15
23	**RGBD**	-0.47	-0.54	-0.55	-0.47	-0.45	-0.41	-0.40	-0.59	-0.52	-0.49	-0.58	-0.59	-0.54	-0.57	-0.52
24	**Vicon**	-0.32	-0.17	-0.04	0.11	0.00	-0.47	-0.05	-0.07	0.26	0.02	-0.02	-0.42	0.07	0.12	-0.05

Table A.2: Relative Error - Real-world Blind Walking

ID	Avatar	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
1	**RGBD**	-0.14	-0.03	0.05	-0.11	-0.01	0.01	-0.14	-0.05	-0.05	0.01
2	**None**	-0.02	0.08	0.12	-0.05	0.01	-0.05	0.05	0.02	-0.02	0.08
3	**RGBD**	0.01	-0.04	-0.06	-0.66	0.08	0.08	0.07	0.17	0.15	0.19
4	**Vicon**	-0.07	-0.01	-0.08	-0.06	-0.02	0.09	0.03	-0.05	-0.05	-0.09
5	**Vicon**	-0.12	-0.20	-0.11	0.02	-0.03	-0.06	-0.14	-0.10	-0.10	-0.08
6	**None**	0.14	-0.02	-0.05	-0.09	-0.01	-0.10	0.06	-0.02	-0.02	0.01
7	**Vicon**	-0.20	-0.21	-0.19	-0.19	-0.17	-0.19	-0.09	-0.19	-0.24	-0.24
8	**RGBD**	0.05	0.06	0.07	0.07	-0.02	0.14	0.02	0.03	-0.01	0.07
9	**RGBD**	0.08	0.08	0.12	0.08	0.16	0.21	0.10	0.22	0.07	0.31
10	**Vicon**	0.00	-0.04	0.07	0.26	0.10	-0.04	0.08	0.02	-0.01	0.05
11	**None**	-0.11	-0.14	0.60	-0.08	-0.09	-0.06	-0.05	0.01	-0.06	-0.04
12	**None**	-0.10	-0.04	-0.13	-0.09	-0.06	-0.09	-0.10	-0.13	-0.07	-0.13
13	**RGBD**	-0.20	-0.07	-0.08	-0.02	-0.08	-0.13	-0.07	-0.07	-0.27	-0.08
14	**Vicon**	-0.37	-0.34	-0.26	-0.30	-0.24	-0.23	-0.28	-0.18	-0.22	-0.20
15	**RGBD**	-0.05	0.07	-0.01	-0.04	-0.04	0.00	-0.07	-0.05	-0.12	0.01
16	**Vicon**	-0.26	-0.28	-0.14	-0.19	0.02	-0.10	0.04	0.00	0.04	-0.07
17	**RGBD**	0.09	0.02	0.00	0.00	-0.05	0.01	-0.12	0.11	-0.09	0.04
18	**None**	-0.03	-0.13	-0.78	-0.14	-0.19	-0.11	-0.16	-0.19	-0.13	-0.16
19	**None**	-0.08	-0.05	-0.02	0.02	-0.03	0.00	0.03	0.01	-0.02	-0.11
20	**None**	-0.18	0.16	-0.11	0.05	0.01	-0.02	-0.11	-0.14	-0.02	-0.26
21	**None**	-0.15	-0.03	-0.02	0.05	-0.01	0.16	0.02	0.00	0.02	0.10
22	**Vicon**	-0.13	-0.15	-0.01	-0.10	-0.13	-0.04	-0.06	-0.13	-0.08	-0.01
23	**RGBD**	-0.06	-0.14	-0.13	-0.15	-0.04	-0.22	-0.11	-0.21	-0.08	-0.09
24	**Vicon**	0.16	0.09	0.10	0.16	0.08	0.12	0.28	0.09	0.22	0.23

Table A.3: Relative Error - VR Blind Reaching

ID	Avatar	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15
1	**RGBD**	0.06	-0.03	0.07	-0.05	0.04	-0.08	-0.06	0.00	-0.05	-0.13	0.05	0.05	0.05	0.13	0.08
2	**None**	1.51	1.23	1.39	1.53	1.82	1.30	1.39	1.14	0.90	0.67	0.67	1.58	1.46	1.34	1.21
3	**RGBD**	0.08	0.03	0.34	0.28	0.44	0.33	0.26	0.20	0.08	0.14	0.34	0.29	0.38	-0.01	0.18
4	**Vicon**	0.15	-0.04	0.12	0.05	0.00	0.10	-0.07	0.03	0.15	0.11	0.11	0.05	0.03	0.02	-0.02
5	**Vicon**	0.09	0.16	-0.09	0.03	-0.04	-0.03	-0.05	0.04	-0.04	-0.26	-0.23	-0.19	0.04	-0.01	-0.11
6	**None**	0.43	0.25	0.07	-0.13	0.10	0.10	-0.05	0.04	0.13	0.38	0.23	0.23	0.12	0.26	0.07
7	**Vicon**	-0.23	-0.13	-0.22	0.10	0.18	-0.05	0.29	-0.07	-0.09	0.06	0.04	0.00	-0.03	0.04	0.16
8	**RGBD**	0.03	-0.11	0.11	-0.08	0.09	0.04	0.01	-0.02	-0.04	-0.03	-0.06	-0.05	0.04	0.05	0.04
9	**RGBD**	0.16	0.10	0.32	0.35	0.18	0.17	0.29	0.47	0.13	0.40	0.25	0.20	0.18	0.19	0.26
10	**Vicon**	-0.23	-0.16	-0.26	-0.13	-0.16	-0.16	-0.15	-0.21	-0.29	-0.22	-0.21	-0.20	-0.25	0.03	0.02
11	**None**	0.11	0.26	0.07	0.02	0.07	-0.09	0.06	0.07	0.26	0.13	0.05	0.35	0.29	0.25	0.42
12	**None**	0.05	0.14	0.14	0.25	-0.09	0.09	0.17	0.45	0.22	0.18	0.11	-0.03	0.16	0.32	0.03
13	**RGBD**	0.18	-0.01	0.07	0.27	0.01	-0.09	0.04	0.21	0.14	0.10	-0.04	0.09	0.26	0.11	-0.09
14	**Vicon**	0.15	-0.05	0.33	0.12	0.20	-0.11	0.16	0.21	0.20	0.13	0.00	0.08	-0.07	0.41	-0.01
15	**RGBD**	0.18	0.07	0.02	0.09	0.18	0.03	-0.14	-0.13	-0.03	-0.17	-0.04	-0.07	0.04	-0.04	0.10
16	**Vicon**	0.18	0.07	0.02	0.09	0.18	0.03	-0.14	-0.13	-0.03	-0.17	-0.04	-0.07	0.04	-0.04	0.10
17	**RGBD**	0.14	0.38	0.20	0.18	0.24	0.18	0.17	0.24	0.28	0.40	0.42	0.30	0.55	0.68	0.46
18	**None**	0.20	0.44	0.22	0.30	0.33	0.17	0.13	0.10	0.11	0.36	0.03	0.38	0.20	0.29	0.23
19	**None**	0.59	0.35	0.47	0.28	0.28	0.20	0.14	0.14	0.27	-0.06	0.38	0.16	0.04	0.17	0.12
20	**None**	0.17	0.06	0.11	0.14	0.20	0.35	0.20	0.23	0.23	-0.21	0.42	0.17	0.15	0.22	0.35
21	**None**	-0.03	0.06	0.55	0.04	0.33	0.10	0.45	0.06	0.29	0.32	0.55	0.26	0.38	0.25	0.53
22	**Vicon**	0.50	0.72	0.57	0.30	0.50	0.70	0.36	0.56	0.75	0.43	0.47	0.57	0.61	0.54	0.36
23	**RGBD**	-0.05	-0.10	0.08	-0.08	-0.02	-0.13	0.05	-0.12	-0.06	-0.11	0.01	-0.05	-0.02	-0.21	-0.05
24	**Vicon**	0.31	0.04	0.12	0.02	0.05	0.04	0.13	0.14	0.18	0.16	-0.03	0.24	0.11	0.24	0.08

Table A.4: Relative Error - Real-world Blind Reaching

ID	Avatar	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15
1	**RGBD**	-0.13	-0.11	-0.03	-0.12	0.02	-0.15	-0.08	0.48	-0.03	-0.01	0.06	-0.13	-0.02	0.08	0.33
2	**None**	-0.10	-0.02	0.34	0.19	-0.05	-0.10	0.07	0.01	0.07	0.19	0.00	0.10	0.06	0.00	-0.01
3	**RGBD**	-0.03	0.07	0.13	0.22	0.07	0.12	0.32	0.24	0.17	0.07	0.12	0.14	0.06	0.08	0.04
4	**Vicon**	-0.06	0.20	-0.08	-0.05	-0.09	0.02	-0.05	-0.20	-0.04	-0.04	-0.07	-0.13	-0.07	-0.10	-0.15
5	**Vicon**	0.04	0.01	-0.10	-0.12	0.01	0.06	0.04	0.06	-0.02	-0.01	-0.04	0.09	0.09	0.22	0.18
6	**None**	0.02	-0.10	-0.12	-0.12	-0.06	-0.32	-0.17	-0.09	-0.02	-0.17	-0.08	0.03	-0.08	0.14	0.02
7	**Vicon**	0.07	0.09	0.14	0.00	-0.08	0.16	0.00	0.19	0.10	-0.01	0.18	-0.05	0.05	0.08	0.24
8	**RGBD**	0.00	-0.09	-0.13	-0.11	-0.05	-0.08	-0.05	0.32	-0.07	-0.17	-0.04	-0.09	-0.17	-0.07	0.03
9	**RGBD**	0.01	0.05	-0.01	0.03	0.09	0.01	0.04	0.02	0.19	0.07	0.10	0.10	0.06	0.26	0.08
10	**Vicon**	-0.07	-0.10	-0.03	0.04	-0.06	0.04	0.06	-0.02	-0.09	0.16	-0.04	-0.07	-0.09	0.02	0.12
11	**None**	0.00	-0.01	-0.04	0.05	0.18	0.03	0.05	-0.05	-0.01	0.12	0.12	0.07	0.05	0.07	0.13
12	**None**	0.06	0.06	-0.01	-0.13	-0.03	0.06	0.04	0.09	-0.01	-0.23	-0.17	-0.29	-0.03	0.04	-0.07
13	**RGBD**	-0.07	0.14	0.04	0.01	0.03	0.03	-0.13	0.01	-0.09	-0.12	-0.07	-0.03	-0.04	-0.04	0.07
14	**Vicon**	0.27	0.07	0.07	0.21	0.30	0.44	0.10	0.09	0.03	0.08	-0.04	0.03	0.37	0.09	-0.03
15	**RGBD**	0.00	0.02	0.07	0.02	-0.15	-0.21	-0.05	-0.25	0.00	-0.17	0.00	-0.06	0.02	0.01	-0.04
16	**Vicon**	0.16	0.24	0.10	0.03	0.05	0.10	0.09	0.05	0.01	0.21	0.03	0.07	0.07	0.10	0.23
17	**RGBD**	-0.04	-0.03	-0.17	-0.07	-0.19	-0.07	-0.05	-0.05	-0.07	-0.16	-0.43	-0.10	-0.12	-0.05	-0.43
18	**None**	0.03	0.01	0.14	-0.04	0.15	0.00	-0.04	-0.03	0.04	0.00	-0.09	0.08	-0.06	0.01	-0.04
19	**None**	0.25	-0.01	0.10	0.09	0.15	0.08	0.07	-0.01	0.04	-0.09	-0.06	-0.14	-0.01	-0.01	0.16
20	**None**	0.00	0.03	-0.08	-0.14	-0.04	-0.07	-0.16	-0.26	-0.06	-0.28	-0.02	-0.48	-0.36	-0.53	-0.03
21	**None**	-0.05	0.05	-0.02	-0.05	-0.07	-0.08	-0.05	-0.19	-0.10	-0.08	-0.14	-0.06	-0.40	-0.43	-0.18
22	**Vicon**	0.01	-0.03	-0.06	-0.23	-0.14	-0.10	-0.23	-0.08	-0.10	-0.30	-0.36	-0.04	-0.20	-0.07	-0.03
23	**RGBD**	-0.01	0.02	0.40	0.02	0.15	0.11	-0.02	0.08	0.02	-0.04	0.10	-0.07	0.06	0.08	-0.09
24	**Vicon**	-0.12	-0.23	-0.35	-0.04	0.00	-0.07	-0.06	-0.17	-0.13	-0.23	-0.22	-0.09	-0.27	-0.26	-0.02

Table A.5: Blind Walking - Mean Error with Outliers Removed

ID	Avatar	Virtual Reality	Real-World
1	**RGBD**	-0.372000	-0.046000
3	**RGBD**	-0.174000	0.072222
4	**Vicon**	-0.141333	-0.044444
5	**Vicon**	-0.246000	-0.092500
6	**None**	-0.208000	-0.026667
7	**Vicon**	-0.138000	-0.191429
8	**RGBD**	0.034286	0.048000
9	**RGBD**	-0.053571	0.143000
10	**Vicon**	-0.338000	0.025556
11	**None**	-0.209333	-0.068889
12	**None**	-0.350667	-0.094000
13	**RGBD**	-0.289333	-0.075000
15	**RGBD**	-0.247333	-0.041111
16	**Vicon**	-0.518000	-0.094000
17	**RGBD**	-0.102667	0.001000
18	**None**	-0.220667	-0.151250
19	**None**	-0.076000	-0.025000
20	**None**	-0.309333	-0.062000
21	**None**	-0.294667	0.016250
22	**Vicon**	-0.040714	-0.084000
23	**RGBD**	-0.512667	-0.123000

Table A.6: Blind Reaching - Mean Error with Outliers Removed

ID	Avatar	Virtual Reality	Real-World
1	**RGBD**	0.008667	-0.071111
3	**RGBD**	0.224000	0.138000
4	**Vicon**	0.052667	-0.058571
5	**Vicon**	-0.046000	0.010000
6	**None**	0.148667	-0.092222
7	**Vicon**	0.003333	0.066000
8	**RGBD**	0.001333	-0.083333
9	**RGBD**	0.243333	0.034444
11	**None**	0.154667	0.015556
12	**None**	0.124286	0.014444
13	**RGBD**	0.083333	-0.015000
14	**Vicon**	0.116667	0.166000
15	**RGBD**	0.006000	-0.072000
16	**Vicon**	0.006000	0.104000
17	**RGBD**	0.321333	-0.090000
18	**None**	0.232667	-0.003750
19	**None**	0.210000	0.046667
20	**None**	0.198462	-0.106000
21	**None**	0.276000	-0.062500
23	**RGBD**	-0.057333	0.036667
24	**Vicon**	0.122000	-0.140000

Table A.7: Simulator Sickness Questionaire Scores

		Pre-VR				Post-VR			
id	**avatar**	**N**	**O**	**D**	**TS**	**N**	**O**	**D**	**TS**
2	**None**	0	0	0	0	0	0	0	0
6	**None**	9.54	30.32	27.84	26.18	4.77	30.32	34.8	26.18
11	**None**	9.54	7.58	0	7.48	9.54	22.74	27.84	22.44
12	**None**	0	0	0	0	0	0	27.84	7.48
18	**None**	9.54	0	0	3.74	0	0	0	0
19	**None**	9.54	30.32	27.84	26.18	19.08	22.74	27.84	26.18
20	**None**	0	18.95	6.96	11.22	14.31	15.16	20.88	18.7
21	**None**	0	0	0	0	4.77	15.16	6.96	11.22
4	**Vicon**	0	7.58	0	3.74	0	15.16	0	7.48
5	**Vicon**	0	0	0	0	0	0	0	0
7	**Vicon**	0	0	0	0	0	0	0	0
10	**Vicon**	0	0	0	0	0	0	0	0
14	**Vicon**	0	7.58	13.92	7.48	4.77	15.16	6.96	11.22
16	**Vicon**	9.54	26.53	20.88	22.44	9.54	34.11	27.84	28.05
22	**Vicon**	4.77	3.79	6.96	5.61	0	0	0	0
24	**Vicon**	0	15.16	0	7.48	0	15.16	0	7.48
1	**RGBD**	0	0	0	0	0	0	0	0
3	**RGBD**	0	0	0	0	0	15.16	0	7.48
8	**RGBD**	0	0	0	0	0	7.58	27.84	11.22
9	**RGBD**	0	0	0	0	0	0	0	0
13	**RGBD**	0	0	0	0	0	15.16	27.84	14.96
15	**RGBD**	0	0	0	0	0	3.79	0	1.87
17	**RGBD**	0	0	0	0	0	0	0	0
23	**RGBD**	0	0	0	0	0	7.58	0	3.74

Spatial Perception and Presence in Virtual Environments
Exit Survey

Please answer the following questions by marking a number from 1 to 7:

1. The virtual environment felt like a rece that I was in, as opposed to a video I was watching.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

2. It was as easy to perform the blind walking task in VR as in the real world.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

3. It was as easy to perform the blind reaching task in VR as in the real world.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

4. I had complete confidence in my ability to judge distances using blind walking in VR.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

5. I had complete confidence in my ability to judge distances using blind walking in the real hallway.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

6. I was completely comfortable with the appearance of my body in VR.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

7. Seeing my virtual body made me feel comfortable in the virtual environment.

strongly agree	agree	slightly agree	neutral	slightly disagree	disagree	strongly disagree
1	2	3	4	5	6	7

Table A.8: Exit Survey Answers

id	avatar	Q1	Q2	Q3	Q4	Q5	Q6	Q7
2	**None**	4	6	5	4	2	n/a	n/a
6	**None**	6	5	2	6	5	n/a	n/a
11	**None**	3	5	3	3	1	n/a	n/a
12	**None**	2	3	2	2	3	n/a	n/a
18	**None**	3	2	2	3	3	n/a	n/a
19	**None**	2	3	3	2	2	n/a	n/a
20	**None**	2	2	1	4	2	n/a	n/a
21	**None**	3	no answer	no answer	2	2	n/a	n/a
4	**Vicon**	3	3	3	2	2	5	no answer
10	**Vicon**	4	2	2	2	2	2	no answer
14	**Vicon**	5	4	7	6	3	5	no answer
5	**Vicon**	3	2	2	2	3	2	3
7	**Vicon**	2	5	5	5	4	2	2
16	**Vicon**	2	4	2	5	4	3	3
22	**Vicon**	3	5	4	6	5	4	3
24	**Vicon**	2	2	3	4	4	3	2
1	**RGBD**	3	3	2	4	2	2	2
3	**RGBD**	2	3	3	5	3	2	2
8	**RGBD**	2	3	2	3	2	2	2
9	**RGBD**	2	3	2	4	2	2	3
13	**RGBD**	5	3	1	6	2	1	3
15	**RGBD**	5	2	2	3	3	4	4
17	**RGBD**	3	3	2	2	3	2	1
23	**RGBD**	6	5	2	6	6	3	2