Hybrid and Coordinated 3D Interaction in Immersive Virtual Environments

by

Jia Wang

A Dissertation Submitted to

The Academic Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in

Computer Science

by

April 28, 2015

APPROVED:

Professor Robert W. Lindeman
Advisor

DCBrown
Professor David C. Brown
Committee Member

Professor Craig E. Wills
Head of Department

Professor Emmanuel O. Agu
Committee Member

Professor Matthew O. Ward
Committee Member (deceased)

Professor Doug A. Bowman
External Committee Member
Virginia Polytechnic Institute and State University

ABSTRACT

Through immersive stereoscopic displays and natural user interfaces, virtual reality (VR) is capable of offering the user a sense of presence in the virtual space, and has been long expected to revolutionize how people interact with virtual content in various application scenarios. However, with many technical challenges solved over the last three decades to bring low cost and high fidelity to VR experiences, we still do not see VR technology used frequently in many seemingly suitable applications. Part of this is due to the lack of expressiveness and efficiency of traditional “simple and reality-based” 3D user interfaces (3DUIs). The challenge is especially obvious when complex interaction tasks with diverse requirements are involved, such as editing virtual objects from multiple scales, angles, perspectives, reference frames, and dimensions.

A common approach to overcome such problems is through hybrid user interface (HUI) systems that combine complementary interface elements to leverage their strengths. Based on this method, the first contribution of this dissertation is the proposal of *Force Extension*, an interaction technique that seamlessly integrates position-controlled touch and rate-controlled force input for efficient multi-touch interaction in virtual environments. Using carefully designed mapping functions, it is capable of offering fluid transitions between the two contexts, as well as simulating shear force input realistically for multi-touch gestures.

The second contribution extends the HUI concept into immersive VR by introducing a Hybrid Virtual Environment (HVE) level editing system that combines a tablet and a Head-Mounted Display (HMD). The HVE system improves user performance and experience in complex high-level world editing tasks by using a “World-In-Miniature” and 2D GUI rendered on a multi-touch tablet device to compensate for the interaction limitations of a traditional HMD- and wand-based

VR system. The concept of Interaction Context (IC) is introduced to explain the relationship between tablet interaction and the immersive interaction, and four coordination mechanisms are proposed to keep the perceptual, functional, and cognitive flow continuous during IC transitions.

To offer intuitive and realistic interaction experiences, most immersive 3DUIs are centered on the user's virtual avatar, and obey the same physics rules of the real world. However, this design paradigm also employs unnecessary limitations that hinders the performance of certain tasks, such as selecting objects in cluttered space, manipulating objects in six degrees of freedom, and inspecting remote spaces. The third contribution of this dissertation proposes the *Object Impersonation* technique, which breaks the common assumption that one can only immerse in the VE from a single avatar, and allows the user to impersonate objects in the VE and interact from their perspectives and reference frames. This hybrid solution of avatar- and object-based interaction blurs the line between travel and object selection, creating a unique cross-task interaction experience in the immersive environment.

Many traditional 3DUIs in immersive VR use simple and intuitive interaction paradigms derived from real world metaphors. But they can be just as limiting and ineffective as in the real world. Using the coordinated HUI or HVE systems presented in this dissertation, one can benefit from the complementary advantages of multiple heterogeneous interfaces (*Force Extension*), VE representations (*HVE Level Editor*), and interaction techniques (*Object Impersonation*). This advances traditional 3D interaction into the more powerful hybrid space, and allows future VR systems to be applied in more application scenarios to provide not only presence, but also improved productivity in people's everyday tasks.

DEDICATORY

To my father, who had a vision, and planted a seed, twenty years ago.

ACKNOWLEDGEMENT

In my five years of doctoral study, I have been blessed with great guidance and help from many friends. Without them, I would not have been able to reach this significant milestone in my life. Therefore, I would like to express my appreciation at this moment of academic success.

Simply put, I could not have had a better advisor than Professor Robert W. Lindeman. Rob takes real pride in me, and often introduces me as his rock-star student. But I am well aware that I did not know much about either research or programming when I first started in WPI. It is him who taught me everything about VR and 3DUI, hand by hand, with great patience and belief. It is also him who gave me the most freedom to explore any research idea I was interested in, and constantly supported me with funding and insightful advices. Rob also continuously inspired me outside of research, with his passion and dedication to teaching, his impressive presentation skills and easy-going personality, his optimistic attitude of life, and his love and responsibility to family. And above all, Rob is not only an advisor, but also a friend, one I can joke with comfortably, and consult to when facing challenges in life. For these I am immeasurably indebted and thankful to him.

I would like to thank my committee members for guiding me through this journey. Professor Matthew Ward is no longer with us, but I still remember being his TA in his last days. He showed me a professor at his best, when diligently teaching a class of eighty student, while battling with cancer. This dissertation is the minimum I can give as a tribute.

Professor Doug Bowman is the role model for anyone doing research in the field of 3DUI. It has been a real honor to have him as my external committee member, and an even bigger privilege to have him come in person to my defense. There were a few moments that I felt nervous about

presenting my ideas to Doug, because he could always pin point the weak spot instantly. But in the end, that was exactly what I needed to produce this solid work.

I would also like to thank Professor David Brown. I can easily spin out another three projects based solely on his constructive questions. And to Professor Emmanuel Agu, I will definitely miss talking to you, hearing your laugh in the hallway, and getting embarrassed by your “when are you graduating” jokes.

I would like to express my appreciation to all HIVE, ISRG, and MGRG attendees, for listening to my research presentations with great interest, and providing me with insightful feedback. These group meetings have been a major source of inspirations to my research in WPI.

I have been very fortunate to be funded for five years as a Teaching Assistant, and I would like to thank the Computer Science department for granting me the opportunity. Being a TA allowed me to consolidate my knowledge in Computer Science, develop my presentation and teaching skills, and make friends. I would like to specially thank Professor Hugh Lauer and Professor Charles Rich. It was a privilege working with them on the two best CS courses of Operating System and Game Development. And I am tremendously grateful for having their trust and support to my teaching and research endeavors.

My dearest friend, Dr. Paulo de Barros, has been so much more than “5% of an advisor” to me. Of everyone I interacted with in my life, I have the highest respect in Paulo. He showed me what it takes to be a diligent student, an honest researcher, a responsible husband, and a humble person. He has always got my back no matter what I do, even when playing Battle Tank. He made me feel that I am not in this country alone, that I have an older brother, always looking after me, taking care of me, and wishing all the best of me. May his life be forever filled with love and joy with his wife Isabella, and his three lovely sons.

My appreciation also goes to Dr. Averkin Sergey. From the moment I met him in the International House in September 2009, to my PhD defense presentation in last week, Sergey has always stood by me, giving me his best, unconditional support. With his wife Anna also being the best friend of my wife, I am sure the friendship between our families will last for a long, long time.

I would also like to thank the company of many friends in the iLab and DSRG group, including but not limited to, Qian He, Mi Feng, Zhixin Yan, Owen Leach, Adrian Boteanu, David Kent, Benar Suay, Mohammad Shayganfar, Chuan Lei, Chiying Wang, Lei Cao, Medhabi Ray, Xinbi Li, and Weijian Shang. Many of the user studies in this dissertation would not have been completed without their help, and I will certainly miss the fun lunch-time conversations about bad movies, good video games, and days of our lives in WPI.

Lastly, I would like to thank the most important people in my life, my dad Haibin Wang, my mom Xiangjun Liu, and my wife Mi Tian.

Dad, I dedicate this work to you. Twenty years ago, when you brought home that old 386 PC and that pile of MS-DOS games, you had a vision. When every parents were banning video games for their kids, you encouraged me to play and explore. Remember you used to say that the three most important skills in the 21st century were Computer, English, and Driving? You were totally right! Thank you for treating me as a man from an early age, for granting me freedom but reminding me about responsibility, and mostly, for giving me the love of your life.

Mom, your son has made you proud. Forgive me for not going home for five years, and leaving you and dad alone, missing us every day on the other side of the world. I grew up in your arm, eating the food you cook, and learning the lessons you teach. You and dad have sacrificed so much to support us over the years, wanting no rewards. If I may wish, I just hope to do the same to you two, to be with you, and take care of you until the last days of your lives.

My honey Mi, there is no word I can find, English or Chinese, to describe how thankful I am to have you in my life. When we met each other seven years ago, I was at the deepest, most miserable time of my life. I had no job, no school offer, no money, nothing. And you were this shiny, brilliant girl that I could never dream of knowing. Yet, you gave me your full love, had faith in me, and held my hands through the hardest days. At the rise of your career in China, against all noises of leaving me, you quit your job and flew to US, huddled with me in that 10 square-meter room. At one point, we had to calculate every spending, and still only had 100 dollars on the card, and I remember you comforting me, saying that you did not regret marrying me, and that it would be alright. We moved four times in the past five years, and I still have not given you a proper place to call home. But you always say, wherever I am, is home. Yet now, I am not even by your side. I owe you too much, and all I wish to pay back is by sharing the rest of my life with you. Joy or pain, we are together.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

In the past few years, a renewed interest in immersive Virtual Reality (VR) technology has emerged, thanks to a new generation of low cost, high fidelity Head-Mounted Display (HMD) devices, such as the Oculus Rift and the Sony Morpheus. Using a high resolution, wide Field-Of-View (FOV), and low latency display, these devices can grant the user a unique feeling of presence in the computer generated virtual environments (VE). The experience can be so deeply convincing that seconds after being immersed into the VE, users often reach out their hands and try to interact with virtual objects in front of them. It is evident that to make VR more successful and useful, an equally compelling set of 3D User Interfaces (3DUI) has to be provided to the user to enable them to perform real-world tasks, such as modeling a building, rehearsing surgery, or building a game level. The importance of 3DUI has also been emphasized in the fields of augmented reality (AR) [Zhou08] and mixed reality (MR) [Lok04].

In fact, research in 3DUI has been going on for decades. The tasks of 3DUI have been categorized into travel, way-finding, selection, manipulation, system control, and symbolic input [Bowman04]. There are general challenges to overcome such as spatial body tracking [Welch02], motion sickness [Kennedy10], and user fatigue [Lindeman01], and each task category also has its unique set of research questions. For example, it is difficult to manipulate a 3D object as there are too many degrees of freedom (DOF) to control simultaneously (translation, rotation, and scaling) [Hinckley94]. For travel in the virtual world, it is challenging to map actions in a limited real world space effectively and intuitively to control locomotion in a much larger VE [Suma12]. Numerous novel interfaces have been proposed under each task genre and many empirical user studies have been conducted to evaluate their usability in various, but mostly abstracted, usage scenarios, such as navigating a maze [Bowman98a], or rotating a tetrahedron [Zhai93b].

Of the 3DUIs explored, many are controlled by a single user, who perceives and interacts with a single representation of the VE from a single egocentric point of view [Poupyrev98a], similar to the system shown in Figure 1.1. By exploiting proprioception [Mine97] through a body-centered interaction paradigm [Slater94], they allow the immersed user to interact with the VE much like the way they interact in the real world.

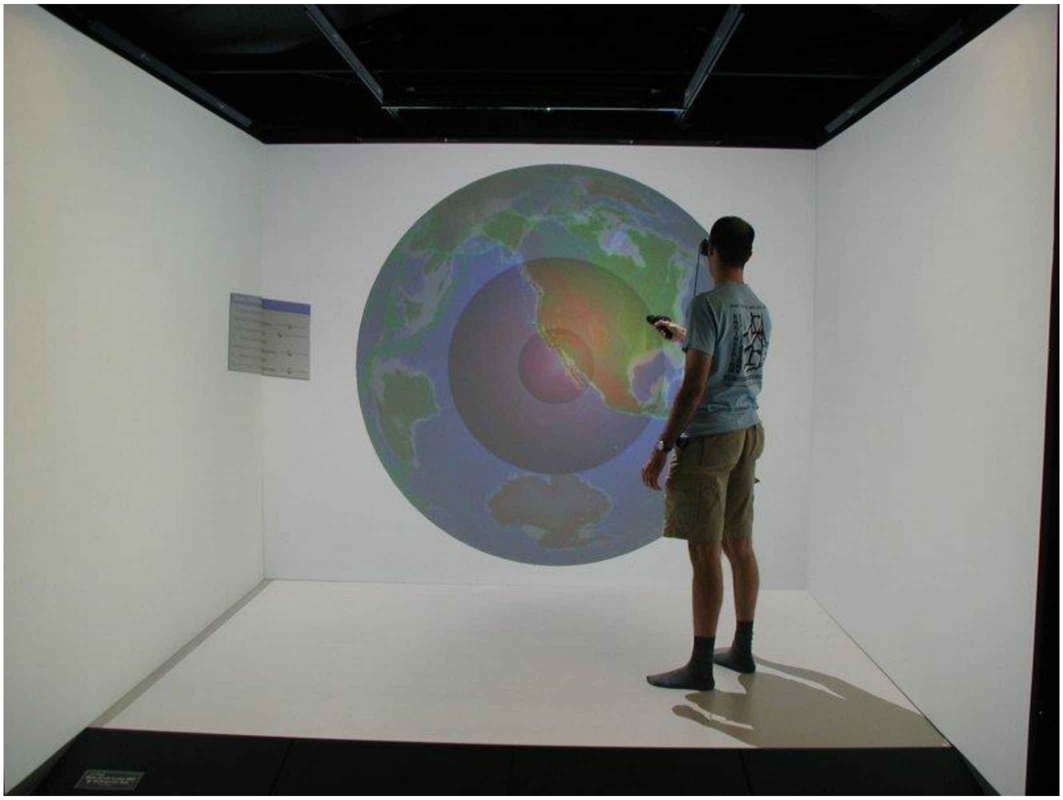


Figure 1.1: Most VR systems give the user a single 3DUI to interact with the VE from a single egocentric point of view (picture courtesy of [Kellogg06])

However, despite the simplicity and intuitiveness, this type of design paradigm can also limit the expressiveness and productivity of the user in the VE [Stoakley95]. For example, grabbing virtual objects does not offer the same level of precision for objects at a distance [Poupyrev96]; 3D spatial input devices such as a wand do not perform well for 2D menu control or symbolic input tasks [Lindeman01]; real walking from a first-person view does not support quick navigation across large landscapes [Wang12b]. There is no "silver bullet" 3DUI that fits all use scenarios, but there is also no need to limit designs to one option.

In general human computer interaction discussions, a Hybrid User Interface (HUI) is defined as a system that "combines heterogeneous display and interaction device technologies to take advantage of the strong points of each" [Feiner91]. By deliberately combining interfaces with complementary benefits, an HUI system can offer the user more and better options to perform tasks with diverse requirements. In VR, many HUIs can be more appropriately called Hybrid Virtual Environments (HVE), such as the Slice-WIM system [Coffey11] shown in Figure 1.2.

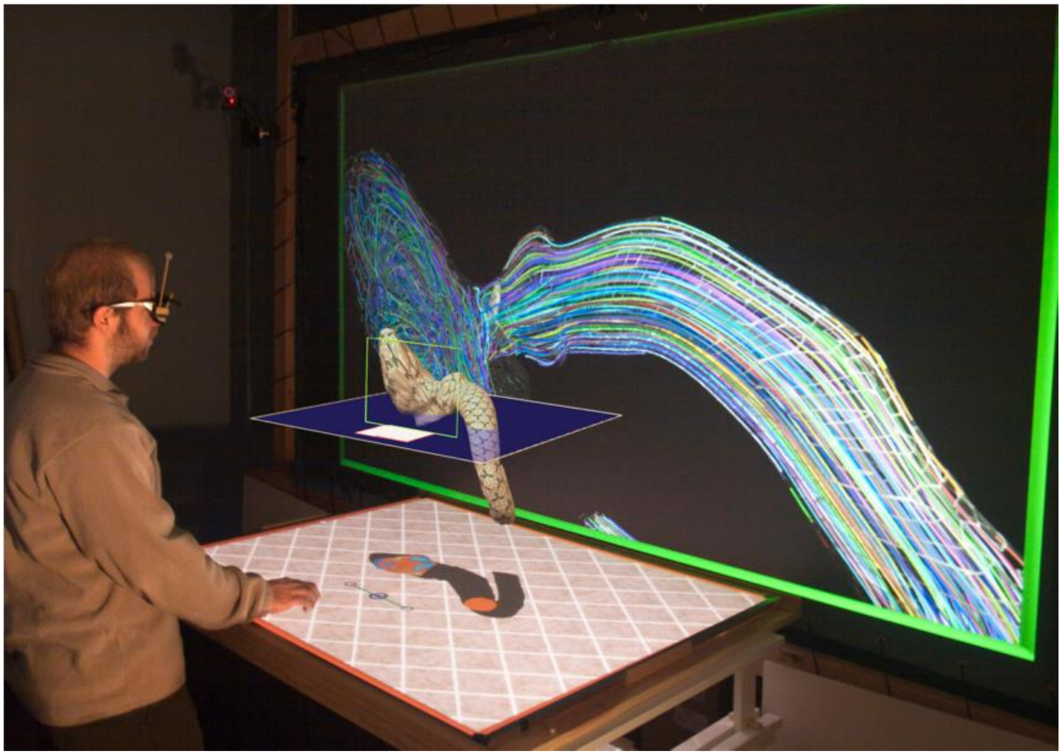


Figure 1.2: The Slice-WIM system is an HVE system that offers the user multiple interactive representations of the same virtual environment (picture courtesy of [Coffey11])

In this dissertation, we define Interaction Context (IC) in VR as “a conceptual integration of input and output devices, techniques, and parameters, which offers the immersed user one interactive representation of the virtual environment”. Using this term, HVEs can be defined as “systems that combine heterogeneous ICs to offer the immersed user with multiple heterogamous representations of the same VE, each with a set of interfaces and interaction techniques to maximize the efficiency and user experience of 3D interaction”. Some popular examples of HVE systems include World-In-Miniature (WIM) [Stoakley95], Voodoo Doll [Pierce99], Portals [Schmalstieg99b] [Kiyokawa05], and See-through Lens [Viega96] [Brown06]. Furthermore, HVE systems can also encompass Collaborative Virtual Environment (CVE) systems that assign asymmetrical ICs to different users, such as the CALVIN system [Leigh96].

An important goal of the current work is to improve the effectiveness of 3D interaction in immersive VR through new designs of HUI or HVE systems. This dissertation presents three techniques that work towards this goal from different perspectives: (1) the *Force Extension* technique blends the accuracy of position control and the large action space of rate control using a force sensing multi-touch touchpad [Wang13b]; (2) the *Tablet- and HMD-based HVE Level Editing* system uses a WIM and a 2D GUI rendered on a multi-touch tablet device to compensate for the interaction limitations of a traditional HMD- and wand-based VR system in a complex high-level world editing task [Wang14]; (3) the *Object Impersonation* technique breaks the common assumption that one can only immerse in the VE from a single avatar, and allows the user to impersonate objects in the VE and interact from their perspectives and reference frames [Wang15].

Although the designers of HUI and HVE systems expect that complementary ICs can compensate for each other’s drawbacks, the actual effective use of them is often limited by the extra cognitive overhead required to attend to, and transition between, the distinctly different interface elements [Grasset08] [Wang Baldonado00]. Therefore, another goal of my work is to reduce this cognitive overhead by employing coordination mechanisms. The *Force Extension* and the *Tablet- and HMD-based HVE system* were both designed with coordination in mind, so that the transitions between ICs can be reduced. Results of user studies are also presented to demonstrate the effectiveness of these coordination mechanisms.

1.1 Thesis Statement

Traditional 3DUIs in immersive VR use simple and intuitive interaction paradigms derived from real world metaphors. But they can be limiting and ineffective when used in complex tasks with diverse requirements. Using seamlessly coordinated HUI or HVE systems, one can benefit from the complementary advantages of multiple heterogeneous interfaces, interaction techniques, and VE representations, and achieve better task performance and user experience, without being hindered by cognitive overhead introduced during context transitions.

1.2 Technique 1: Force Extension

Position and rate controls are the two most common ways to map input data to output variables. Imagine a user uses the swiping gesture on a touchpad to rotate a virtual sandbox. The position displacement of his/her finger on the touchpad is the input data. It can be mapped directly to rotate the sandbox by a certain degree, or indirectly to control the speed of its rotation until the finger is lifted. Position control is more accurate because of the one-to-one mapping, but suffers from repeated actions (clutching) when a long output range needs to be reached [Casiez08]. Rate control, on the other hand, covers longer ranges with reduced effort. However, since the user is controlling the speed, it is much easier to overshoot the target [Zhai93b].

The first hybrid technique, *Force Extension*, strives to seamlessly combine position and rate control, so that the user can perform interaction tasks both accurately and efficiently [Wang13b], as shown in Figure 1.3. The key to achieving this goal is to coordinate the transition process between the two control mechanisms so that a seamless interaction flow can be preserved. Utilizing a force-sensing touch pad, two different algorithms, *Context Force Extension* and *Shear Force Extension*, were designed and implemented to transition fluidly between position-controlled touch input and rate-controlled force input. As the ForcePad supports separate force detection of multi-finger input, this technique can also be applied to various multi-touch gestures, such as rotation and pinch zoom. Details of this technique, including results of a preliminary user study, will be presented in Chapter 2.

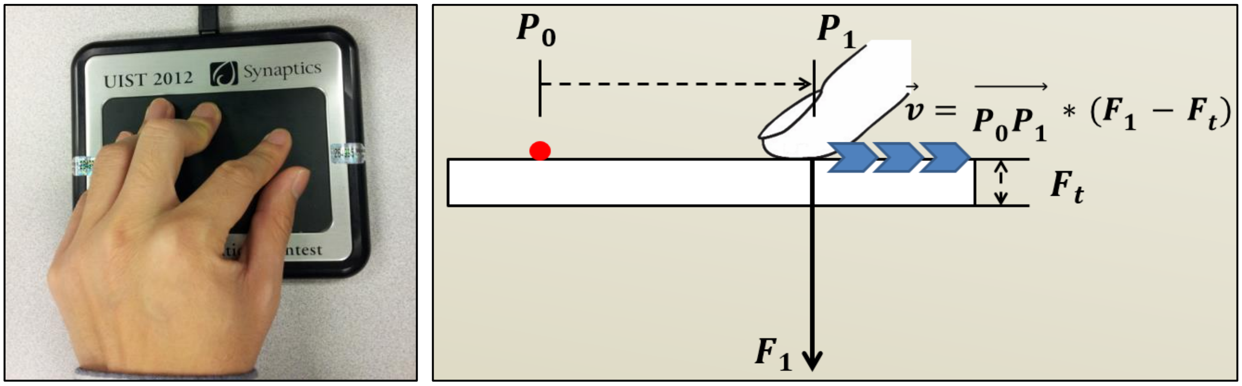


Figure 1.3: The Force Extension technique extends position-controlled touch input to rate-controlled force input by scaling the contextual position input vector with the force value.

1.3 Technique 2: Coordinated HVE Level Editor

The second technique presented in this dissertation is an HVE virtual world editing system that incorporates a smart tablet into a traditional “HMD + wand” VR setup [Wang14]. Because the HMD is non-occlusive, the user is able to glance down under the bottom edge to view and interact with the tablet placed under his/her non-dominant hand. The idea of using interactive surfaces in immersive environments was proposed in the early stages of 3DUI research, and has been developed over many years with the advancement of mobile phone/tablet techniques [Angus95][Bowman98b][Lindeman01][Wilkes12]. However, this tablet- and HMD-based HVE level editing system, as shown in Figure 1.4, has its novel contributions.

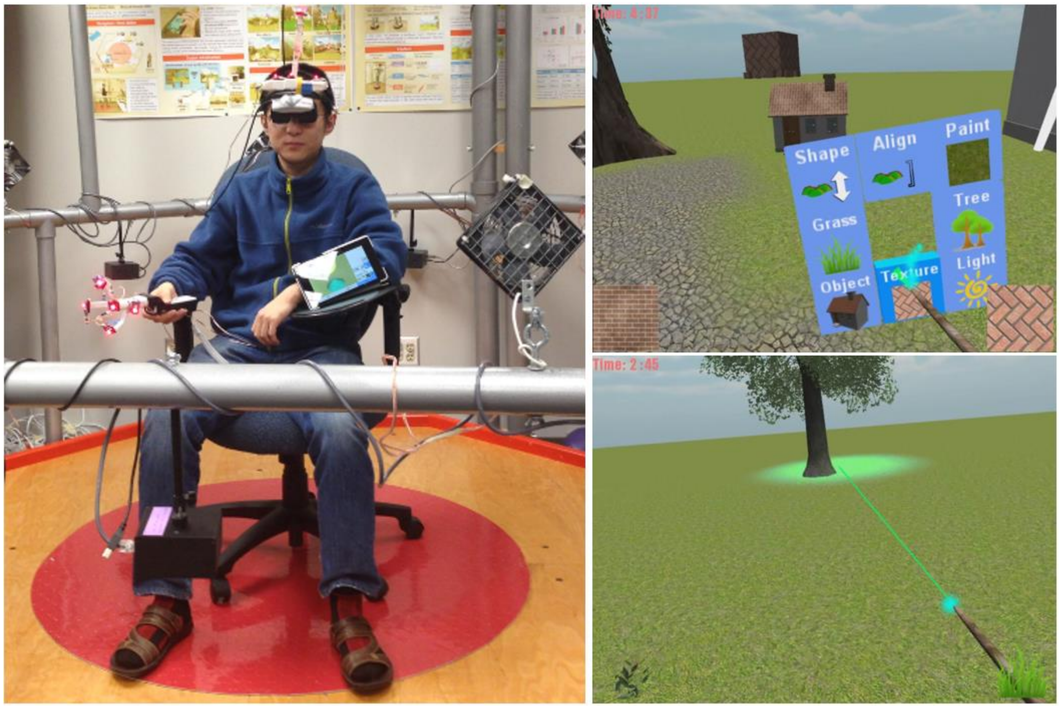


Figure 1.4: The HVE level editor uses a tablet device to complement the limitations of the immersive interfaces by enabling interaction from an above-the-world "God" view.

First of all, instead of being an auxiliary interaction tool, the tablet device is treated as a full-blown IC on the same level as the HMD and wand setup in the HVE system. It renders the virtual world completely on its own, and offers rich multi-touch gestures and a GUI interface to support effective level editing from a "God" view, which complements the limitations of the first-person interaction paradigm of traditional VR. In other words, the tablet IC can be a powerful standalone level editing tool even without the VR hardware. Secondly, with distinctly different IC setups comes high transitional cognitive overhead. To overcome this challenge and make the HVE level editor an effective tool, four different coordination mechanisms are introduced, including task synchronization, mutual awareness, interface sharing, and IC blend-in. The first two are implemented in the current system, with task synchronization shown effective through a formal user study of virtual world editing tasks. Lastly, based on the iterative design and development process of the HVE level editor, a four-step process is proposed for building effective HVE systems. These support validation through task analysis, metaphor selection, IC component specification, and coordination mechanism implementation. The details of this work will be presented in Chapter 3.

1.4 Technique 3: Object Impersonation

To preserve an experience similar to being in the real world, most VR systems grant the user an avatar as his/her virtual self, and use it as a basis for VE presentation and 3D interaction. Based on this body-centric interaction paradigm [Slater94], the user can exploit proprioception [Mine97] to realistically and effectively navigate the virtual world through locomotion of the avatar [Usoh99], or select and manipulate virtual objects by simply grabbing them with his/her virtual hands [Poupyrev98a]. Normally, the user will not switch avatars, as drastic changes in spatial settings can cause confusion and disorientation, which in turn can hurt task performance and user experience in various applications [Lopez14]. However, there are still task scenarios that can benefit from interaction from an out-of-avatar perspective. For example, to center a spotlight precisely on a target requires delicate spatial rotation of the light source. If the user can somehow "impersonate" the spotlight, the task can then be completed easily by turning and looking at the target object himself/herself. Similarly, a user can select objects occluded by a wall by impersonating the wall and looking at the VE behind it. The third technique presented is therefore called *object impersonation*, to indicate that the immersed user does not have to stay attached to his/her virtual avatar, but can select and impersonate virtual objects, and view and interact with the VE from their perspectives and reference frames, as shown in Figure 1.5.

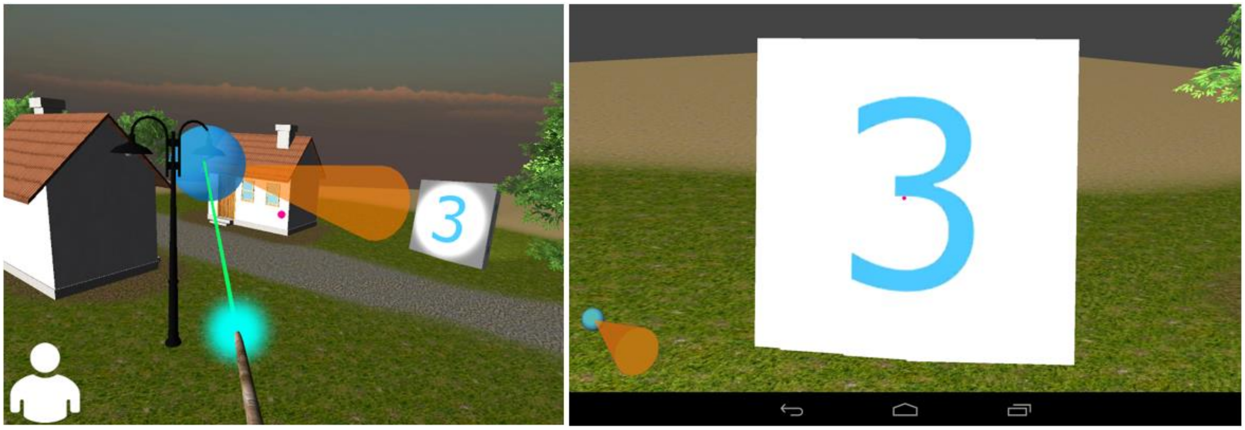


Figure 1.5: The object impersonation technique allows an immersed user to select and become an object to manipulate it from the inside, such as orienting a spotlight on a target by looking at the target from the light's own point of view. [Wang15].

Because directly replacing the virtual avatar with the object's view can be confusing and disorienting, the object impersonation technique implemented in this dissertation chose to present both views in the form of two ICs. Specifically, it builds on the hardware system infrastructure of a tablet- and HMD-based HVE system as shown in Figure 1.5, and defines a new metaphor to correlate the two ICs. In other words, a user can use the wand to select a virtual object from the HMD view, and see and interact with the VE from its "impersonated" view on the tablet using multi-touch gestures. An alternative implementation was also experimented with by having the user select the object on the tablet from an exocentric perspective, to get a more immersive impersonation experience in the HMD IC. To validate the benefits of this new HVE metaphor, a formal user study was conducted to compare the two object impersonation implementations with a standard immersive VR setup for a six degree-of-freedom (DOF) manipulation task [Zhai93b]. An in depth discussion of study results and research findings will be presented in Chapter 4.

1.5 Summary

VR is a technology with great potential, but has never been truly adopted by mainstream users into everyday life. Recent advances in HMD devices have made a good push to bring VR back into the sight of the public. Many people hope that these high-fidelity and low-cost HMDs will make this tide of VR ready for prime time. However, this goal can hardly be achieved if a practical, effective, and user friendly 3DUI still is still missing from the picture. Real world applications are filled with diverse and complicated tasks, and despite decades of research, it is still difficult today to find one 3DUI to fulfill all needs. Yet, we do have many options that can deliver satisfactory performance and experiences in different sets of task scenarios. Therefore, I set the goal of this dissertation to *joining complementary strengths of different interfaces and interaction techniques into more powerful hybrid interface systems*, and successfully achieved it through designing, implementing, and evaluating four novel HUI and HVE techniques. The next three chapters of this thesis will present and discuss the details of each technique respectively.

Chapter 2: Force Extension

Unlike the mouse and keyboard combination in desktop computing environments, immersive VR has not established a universal input device that can be used in most application scenarios. However, 3DUI designers do have a good arsenal of sensors and input devices to choose from when facing a specific set of task requirements. For example, leaning-based flying in 3D virtual spaces can be realized by either sensing the tilt angle of a board, or calculating the user's center of gravity on it [Wang12b]. Similarly, to translate a virtual object in space, the user can move a motion sensor, drag a string-attached sphere, or push the outside of a fixed trackball [Zhai95].

Based on the type of muscular feedback provided to the user, input devices can be categorized into isometric, isotonic, and elastic devices. The terms isometric and isotonic come from exercise physiology. An *isometric* contraction happens when there is tension on the muscle but no movement is made, causing the length of the muscle to remain the same [Zhai93b]. Most force-sensing devices, such as the Wii Fit Balance Board, or the IBM TrackPoint, are isometric devices. On the other hand, in *isotonic* contraction, tension remains unchanged but the muscle's length changes. Many free space input devices, such as the mouse, or the Polhemus magnetic tracker, are isotonic devices. Lastly, *elastic* feedback stands in between isometric and isotonic, giving the user increasing resistive force as the device is pushed further. Traditional joysticks, the Reebok Core Tilt Board [Wang12b], and a string-attached spherical sensor device [Zhai93b] both fall into this category.

The efficiency of isometric, isotonic, and elastic devices depends a lot on the mapping mechanism being used. Based on the study results of a 6-DOF object manipulation task, Zhai summarized that isotonic position control, isometric rate control, and elastic rate control have significantly superior performance than other combinations [Zhai93b]. Even so, the respective range and precision advantages of rate and position control are still separated, and thus hindering user performance in many interaction tasks. For example, recent 3D interfaces often integrate touchscreens or touchpads to support rich expressiveness of multi-touch gestures such as swipe, pinch, and rotation. But multi-touch input is isotonic, and the dominant mapping is for position control (*e.g.*, swiping or pinch zoom). Like the mouse, multi-touch interfaces face the same challenges of greater user fatigue and degraded user performance when frequent clutching (the temporary recalibration to extend limited input space) is required. On the other hand, rate-control devices such as the IBM TrackPoint eliminate clutching, but tend to lead to overshooting of targets.

Based on Zhai's findings, research efforts have recently been investigating hybrid approaches, tapping the advantages of both position and rate control by augmenting the normal touchpad with a rate-controlled elastic edge [Dominjon05] [Kulik12]. Empirical studies of 2D pointing tasks have shown improved performance and user experience of such hybrid solutions in comparison to position control alone. However, these interfaces still have limitations. For example, to switch between position and rate control, the finger needs to constantly move back and forth between the center and the edge of the input space to activate different sensors. This discontinuous transition may break the interaction flow of the user and hurt performance even when complementary subtasks are assigned to both sensors. Additionally, as these devices were originally designed to expand the effect range of position-controlled pointing, they are not capable of combining position and rate control for multi-touch gestures such as pinch, rotation, or multi-finger swipe.

Recent advances in touchpad development have enabled simultaneous position and force sensing of multiple touch points. For example, the Synaptics ForcePad in Figure 2.1 is a USB-connected multi-touch force-sensing touchpad that detects up to five fingers of variable pressure, at 6-bit resolution and up to 1000g of force [Wang13b]. In essence, these devices provide a third input dimension that can be used to augment existing touch-based interaction techniques. And since force input is isometric, it can be ideally coupled with rate control to *extend* the output space of position-controlled multi-touch gestures.

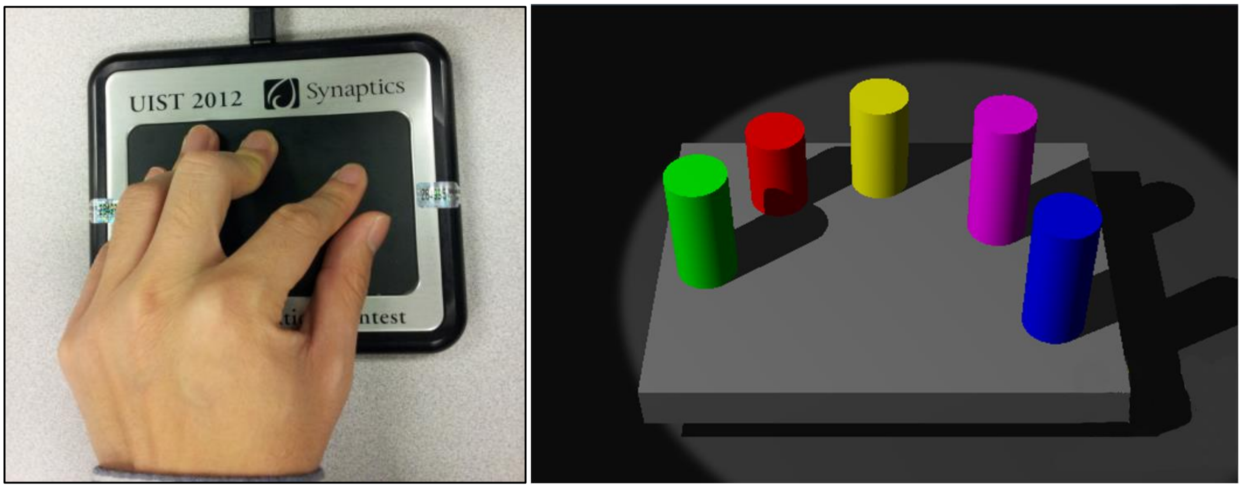


Figure 2.1: The Synaptics ForcePad detects variable pressure of up to five touch points (right picture taken from the Synaptics ForcePad demo program)

Based on this idea, I developed two novel methods to seamlessly combine the two into a more efficient interface. The first approach, *context-force extension*, uses the finger movement in the previous position control mode as the context of the extension, preserving the user's flow of interaction through a smooth transition between the two modes. On the other hand, by scaling micro-finger displacements with pressure input, the second approach, *shear-force extension*, successfully simulates shear force sensing [Harrison12], allowing the user to change the direction of rate-control movement without switching back to position-control mode. The rest of this chapter will present the design and implementation of the two methods, discuss their pros and cons based on the result of a preliminary user study, and demonstrate their usage in a 3DUI application.

2.1 Related Work

The two most popular ways to map input to output are position control (zero order) and rate control (first order) [Zhai95]. Most pointing devices such as the mouse use position control. Previous studies comparing a mouse and a finger-controlled isometric joystick (the IBM TrackPoint) revealed the movement microstructure of both devices and concluded that the random variations in the velocity of the joystick make it harder to control [Mithal96]. However, when the input space of position-control devices is limited compared to the much-larger screen space, frequent "clutching" (*i.e.*, the action to rewind the input space by lifting one's hand to a previous input location) can cause low efficiency and high fatigue. Increasing the scaling factor of position-controlled input (*i.e.*, Control-Display gain, or CD gain) can reduce clutching, but a high CD gain can hurt performance [Casiez08]. Alternatively, the CD gain can be dynamically adjusted based on the velocity (*i.e.*, the pointer acceleration technique [Jellinek90]) or the range (*e.g.*, the Go-go interaction technique [Poupyrev96]) of the input. However, as far as we know, there is no published research showing the performance benefit of such techniques in comparison to standard position control with clutching [Jellinek90]. Zhai classified input devices into isotonic, isometric, and elastic. Through the study results of a 3D object manipulation task, it was found that isotonic devices were better suited to position control, and isometric and elastic devices should be used for rate control because of their self-centering properties [Zhai95].

Instead of clutching, the effect range of position control can be extended by rate control when position input reaches the limit of the input space. Examples include the Bubble technique [Dominjon05], the RubberEdge [Casiez07], and the GroovePad [Kulik12]. Based on the findings of Zhai, these interfaces all use isotonic devices for position input and switch to elastic rate control at the edge of the input space. The Bubble technique simulates a spherical volume in physical space and visualizes it as a transparent sphere on the display. The movement of the input point is by position control within the volume and by rate control beyond the volume with elastic feedback. The RubberEdge technique identified a flaw in the mapping functions of the Bubble technique that created trajectory and velocity discontinuities at the transition point, and proposed a novel solution inspired by the physical movement of a dinner plate when pulled at the edge with a string. A user study was conducted for a 2D pointing task and results showed that RubberEdge outperformed position control by 20% when there was significant clutching [Casiez07]. The GroovePad [Kulik12] provides a hardware implementation of RubberEdge and studied its usability in pointing, panning, and dragging tasks. Although results indicated using GroovePad reduced clutching compared to a standard touchpad, performance failed to increase as users spent extra time deciding which mode to activate when using two input sensors for the same functionality. Hybrid position/rate-control techniques have also been used for other purposes. For example, the Magic Barrier Tape uses rate-controlled navigation to extend the walk-able space in VR [Cirio09].

There are two intrinsic limitations of these “push the edge” techniques that hindered them from delivering the full potential of hybrid position and rate control. Firstly, as shown in Figure 2.2a, since the rate-control region is on the edge of the input space, the user cannot smoothly transition to rate control immediately at point *B* when he/she realizes the target is out of reach. Instead, an unnecessary finger movement to point *C* is required. Secondly, as shown in Figure 2.2b, assuming the user has performed enough rate-controlled movement in the direction of *BC* to approach the aforementioned target without overshoot, he/she still has to do at least one clutch to point exactly to the target in order to use the position-controlled input space. These limitations create a transition gap between the two modes, and impose high cognitive overhead on the user in comparison to simple position- or rate-controlled solutions.

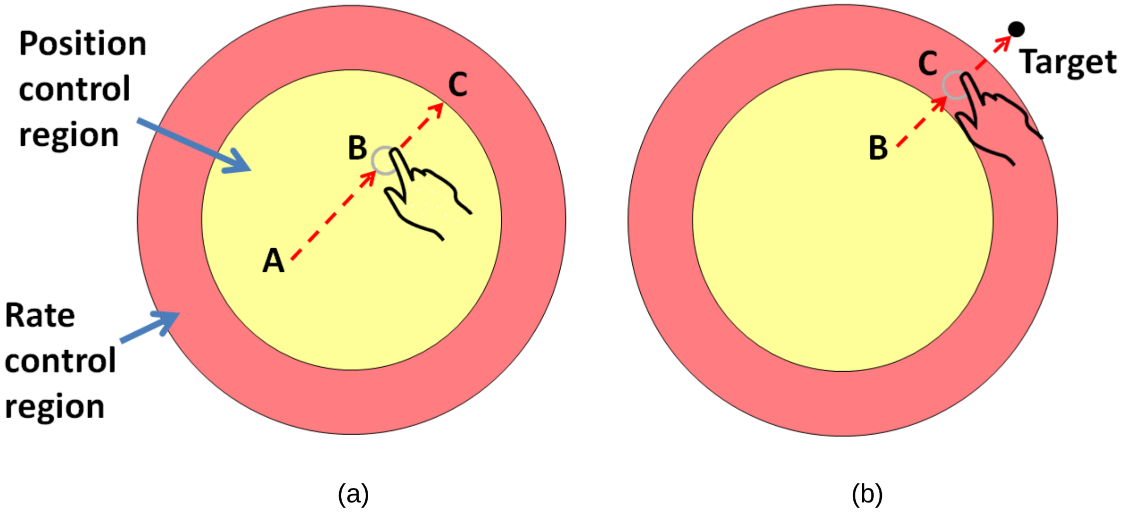


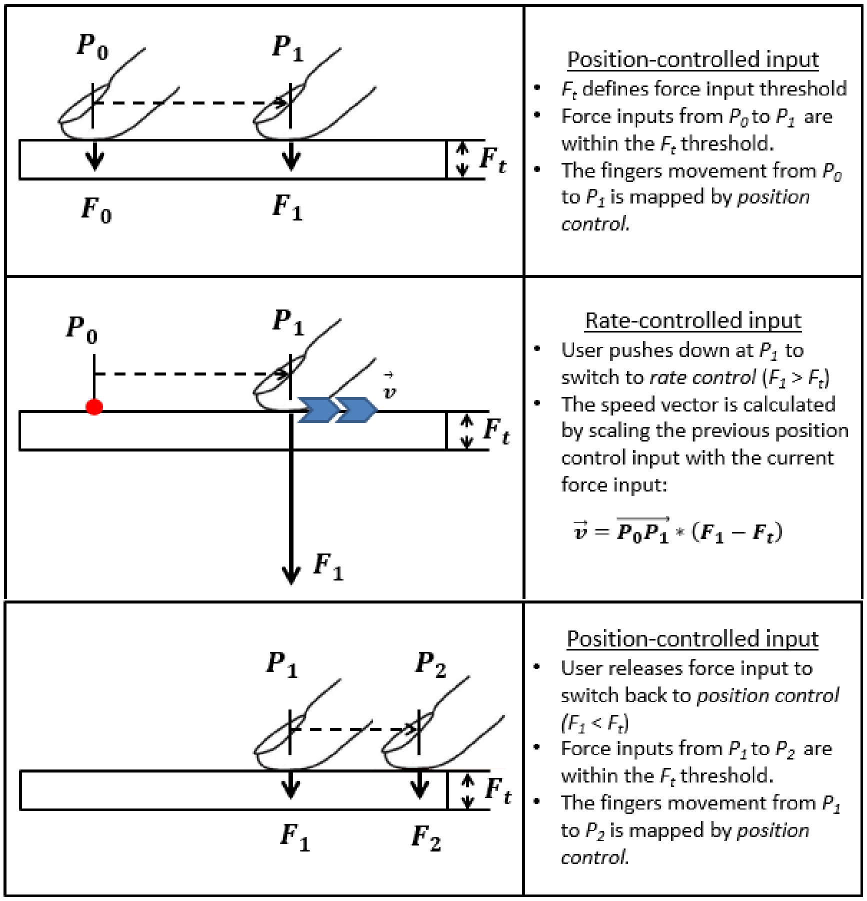
Figure 2.2: Limits of existing hybrid position/rate-control techniques: (a) they can only trigger rate control at edges of position control; (b) the user needs to clutch to use position-control for precise control.

Both limitations can be resolved using a force-sensing touchpad like the Synaptics ForcePad shown in Figure 2.1. As pressure-based rate control can be triggered at any time and any place on the touchpad, the user can transition from position to rate control immediately at point *B*. Also, because rate control started at point *B*, the extra space around point *B* can be used to finish the final touch when the target is approached, without having to lift the finger and rewind the input space.

In general, pressure sensors can be added to any position-control device to provide an extra degree of freedom (DOF) for input. Ramos & Balakrishnan studied the human ability to perform discrete target selection tasks by varying a stylus' continuous pressure, with full or partial visual feedback and different ways to confirm selection once the target is acquired [Ramos14]. One of the challenges of using pressure input on a touch surface, either through finger or stylus, is the potential interference between spatial x-y movement and pressure channels. In contrast, the movement of the mouse is much more stable and orthogonal to the control of pressure sensors attached to the side of the mouse, allowing users to comfortably control up to 64 modes with a dual-pressure augmented mouse [Cechanowicz07]. Shear (tangential) force can also be sensed by pressure sensors attached to the four corners of a touch surface. When applying shear force, the finger does not perceptibly move but the skin of the finger pad shifts position slightly and provides the user with viscoelastic feedback [Lee12]. The potential of shear-force input has been demonstrated for mobile multi-touch devices through single-touch force gestures [Harrison12] [Heo11]. A recent user study investigated the user controllability of shear force to reach and maintain target force levels with regard to hand pose and direction of force input, and found that target-acquisition tasks using shear-force input follow Fitts' law [Fitts54] and that users have more physical and perceived loads when applying shear force in the lateral direction [Lee12]. Small thumb-rolls have been found to be discernable from swipes in the definition of a gesture set for mobile input [Roudaut09]. However, to the best of my knowledge, there is no published solution that senses *shear forces of multiple touch points*, and the use of rate-controlled shear force to extend the effect range of position control is rarely discussed.

2.2 Context-Force Extension

To seamlessly transition from position-controlled touch to rate-controlled push, the context-force extension technique uses the touch input vector as the *directional and scalar context* to calculate the speed vector of rate-controlled movement. The transition between position and rate control is triggered by the force input crossing a threshold, which is set to 20 percent (200g) of the maximum force in my implementation. Taking single-finger cursor control as an example, Figure 2.3 and Equation 2.1 illustrate the transfer functions and the position/rate-control transitions of the context-force extension mode step by step.



Position-controlled input

- F_t defines force input threshold
- Force inputs from P_0 to P_1 are within the F_t threshold.
- The fingers movement from P_0 to P_1 is mapped by *position control.*

Rate-controlled input

- User pushes down at P_1 to switch to *rate control* ($F_1 > F_t$)
- The speed vector is calculated by scaling the previous position control input with the current force input:

$$\vec{v} = \overrightarrow{P_0P_1} * (F_1 - F_t)$$

Position-controlled input

- User releases force input to switch back to *position control* ($F_1 < F_t$)
- Force inputs from P_1 to P_2 are within the F_t threshold.
- The fingers movement from P_1 to P_2 is mapped by *position control.*

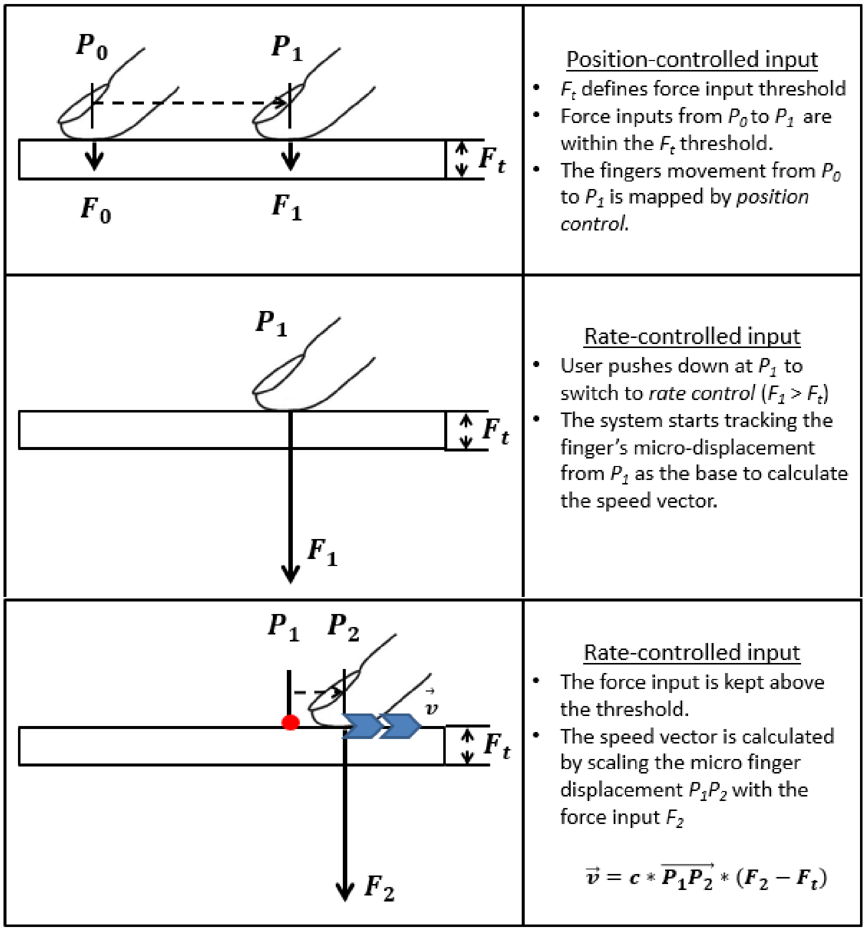
Figure 2.3: Step by step illustration of context-force extension

$$\begin{aligned} \overrightarrow{\Delta P} &= c * (P_i - P_0), when\ F_i < F_t \\ \vec{v} &= \overrightarrow{\Delta P} * (F_i - F_t), when\ F_i \geq F_t \end{aligned} \qquad \text{(Equation 2.1)}$$

When a touch-down gesture is detected, the base point is updated to P_0, and all subsequent movement of the finger (the vector from the base point to the current point, P_0P_1) is scaled by a constant CD gain (c in Equation 2.1) to move the cursor by position control as long as the current force F_1 is below the threshold F_t. When F_1 exceeds F_t, rate control is triggered and the speed vector is calculated by scaling the previous position control vector (P_0P_1, as the *context*) with the force input beyond the threshold ($F_1 – F_t$). The transition back to position control is triggered when F_1 decreases to below F_t, while the base point gets updated to P_1 so that the direction of future rate-controlled movement can be changed without lifting the finger, allowing smooth transitions between position and rate control within one touch session.

2.3 Shear-Force Extension

During the iterative implementation of the context-force extension mode, it was discovered that when applying shear force on the touchpad, the detected position of the fingers shift slightly with the force variance. This "micro-finger displacement" provides a direction vector which can be combined with pressure input as the magnitude to simulate shear-force sensing. Figure 2.4 shows a step-by-step illustration of how shear force extension was implemented.



Position-controlled input

- F_t defines force input threshold
- Force inputs from P_0 to P_1 are within the F_t threshold.
- The fingers movement from P_0 to P_1 is mapped by *position control*.

Rate-controlled input

- User pushes down at P_1 to switch to *rate control* ($F_1 > F_t$)
- The system starts tracking the finger's micro-displacement from P_1 as the base to calculate the speed vector.

Rate-controlled input

- The force input is kept above the threshold.
- The speed vector is calculated by scaling the micro finger displacement P_1P_2 with the force input F_2

$$\vec{v} = c * \overrightarrow{P_1P_2} * (F_2 - F_t)$$

Figure 2.4: Step by step illustration of shear-force extension

As shown in Figure 2.4, the base point is set to P_1 when F_1 exceeds F_t. Instead of scaling the previous position-control movement (P_0P_1) as the context of rate control, the shear-force extension mode tracks the micro-finger displacement (P_1P_2) and scales it with the force input beyond the threshold ($F_2 - F_t$) and a constant factor c (to scale up the micro-finger displacement, this was set to 10.0 in my implementation) to calculate the speed vector.

Preliminary tests confirmed that this mechanism was able to realistically simulate shear-force sensing of multiple active fingers in all directions. Because the base point is updated every time F_i exceeds F_t, the mechanism is also very tolerant of different use patterns, as some users tend to release the force when changing the shear-force direction while others do not. Nevertheless, it should also be mentioned that a potential problem exists at the third step. If the finger movement from P_1 to P_2 is inadvertently more than a micro displacement, the transfer function could produce a velocity much greater than expected. The speed vector can be clamped at a maximum but in future work, per finger calibration may be necessary to sample the possible range of the user's micro-finger displacement.

2.4 Application in Multi-Touch 3DUI

Both force-extension modes can be applied to multi-touch gestures. For example, by averaging the pressure of multiple fingers and replacing the single finger position (P) with two-finger centroid, separation, or rotation data, multi-touch controls such as camera pan, pinch zoom, or camera orbit can be augmented using the same mechanism demonstrated in Figures 2.3 and 2.4.

As a proof of concept, an application called the "Full Force Terrain Editor" was developed in which a single ForcePad is used to perform terrain surface editing as well as 3D camera manipulation as shown in Figure 2.5. The gesture definitions are listed in Figure 2.6. The movement of the terrain brush, *i.e.*, the cursor, is position controlled by single-finger touch. Single-finger pressure controls how fast the terrain surface under the brush gets raised or lowered (if another finger is holding the bottom left corner of the touchpad). Force Extension is applied to camera orbit, pinch zoom, and camera pan for efficient camera manipulation. A gameplay mechanism was added to the application requiring the user to raise/lower the terrain surface to herd physics-based animal objects back to the farm. The application was developed in the Unity3D game engine using TUIO [Kaltenbrunner05] to communicate to the ForcePad driver. The game was demonstrated publicly at the student contest of ACM User Interface Software and Technology (UIST) in 2012, and received highly positive feedback. Participants commented that the terrain editing application made good use of the force-sensing feature of the touchpad, that the hybrid position and rate control was very easy to learn, and that it was very intuitive and efficient for 3D camera manipulation.

Figure 2.5: “Full Force Terrain Editor” uses the Synaptics ForcePad for terrain-surface editing as well as effective 3D camera manipulation with multi-touch “Force Extension” gesutures.

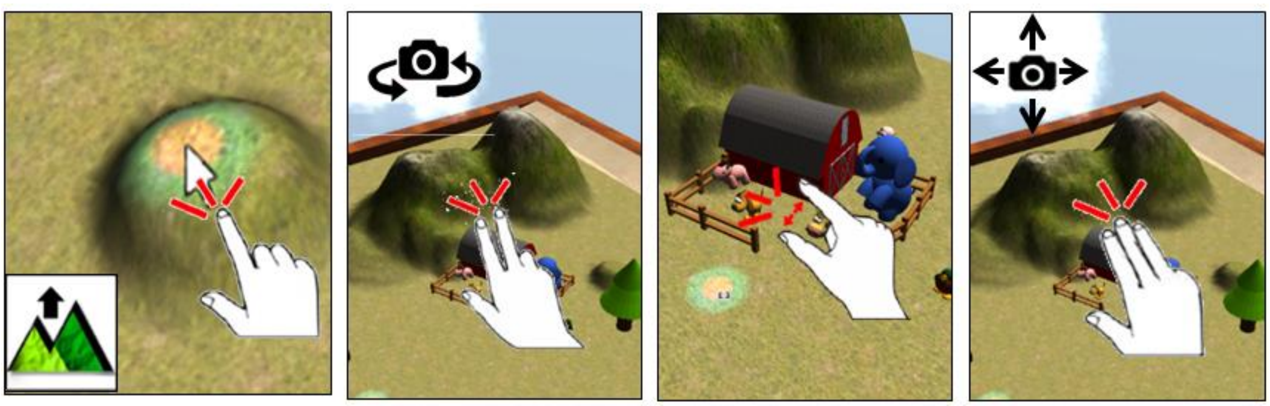

(a) (b) (c) (d)

Figure 2.6: Gesture definitions: (a) single finger raise/lower terrain surface; (b) two-finger movement orbits camera; (c) two-finger pinch zoom; (d) three-finger movement pans camera.

2.5 User Evaluation

A preliminary user evaluation was conducted with six users to collect their subjective feedback about using a single ForcePad for 3D camera manipulation tasks. A rough terrain was constructed with eight objects scattered around, each one with a three-digit number attached to it (Figure 2.7). The task was for the user to move to each of the objects and report the number verbally.

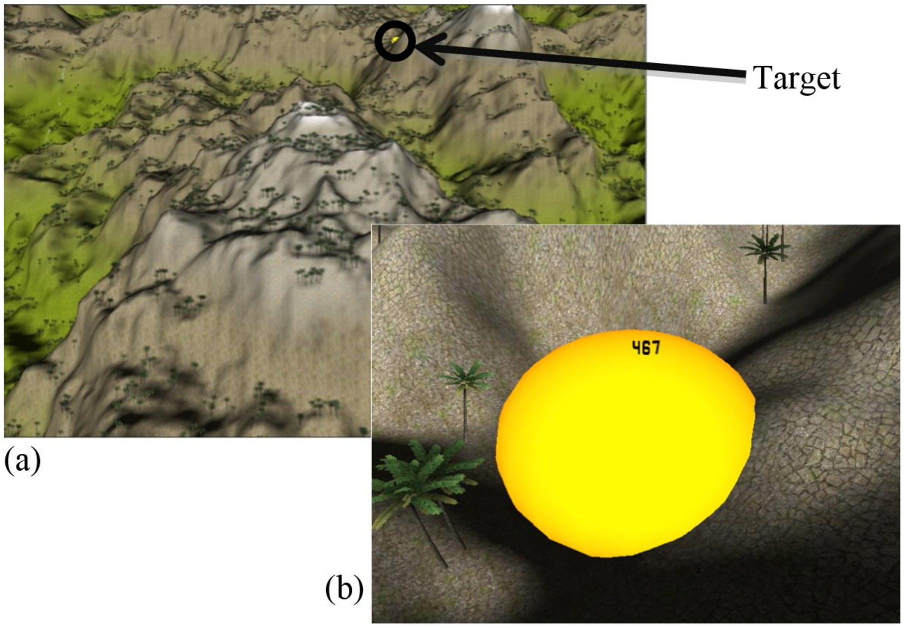


Figure 2.7: (a) Terrain scene and (b) target object close up.

Five interface variations were constructed: three position-control-only interfaces with *c* (CD gain) empirically set to 200, 400, or 800 (for small, medium, and high CD gain, and referred to as "p200", "p400", and "p800" in later discussions), context-force extension with $c = 200$ ("c200") and shear-force extension with $c = 200$ ("s200"). In a tutorial session, the experimenter demonstrated all interface variations, and encouraged the subjects to try out each mode and ask questions freely. In the study session, the subject completed the same task five times using the five different interface variations. No time limit was imposed, and the users were encouraged to think aloud as they searched. After each trial, users were asked for any comments, and after all the trials, they were asked to rank the interfaces. Of the position-control-only approaches, p400 was preferred most often, with p200 reported as requiring too much clutching, and p800 leading to too much overshoot. For all five interfaces, two users preferred c200, three preferred p400, and one preferred p800. Two users ranked c200 as the least preferable, but these users also commented that more practice might change their answers. Two users commented that they thought the rate-controlled movement was counter to what they expected, meaning that they expected the *scene* to move in the direction of finger movement instead of the *camera* to move in the direction of finger movement. This is the well-known cognitive problem of viewport vs. content scrolling that is present in many tablet interfaces [Wang13b]. Providing a settable user preference for this is one solution used in many applications. Three users suggested the strategy of using force extension modes for large scale navigation, and low (200) to medium (400) gain position control when close to a target, and commented that Force Extension was most efficient when zooming out to the highest level. Finally, two users suggested that the c200 should have a higher gain (*c*) value, and that the force should be scaled using a fixed rate, rather than a rate based on the length of the recent position-control gesture. This might remove some of the confusion, and improve usability.

2.6 Summary

With the advance of multi-touch technology, more and more 2D and 3D input devices will consist of multi-touch touchpads that can accurately sense the pressure of each individual finger. To take advantage of this new technology, this chapter presented two novel approaches to smoothly combine position and rate control for multi-touch gesture input. The transition to rate control is triggered whenever the averaged force input of all active fingers exceeds a threshold. The context-force extension approach extends the most recent position-control movement using rate control, and the shear-force extension approach utilizes the fingers' micro displacement after passing the threshold to simulate shear-force based rate control. By granting position control context to the rate-control extension, the former approach intensifies the user's recent memory of the current interaction state, promising not only physically but also cognitively smooth transitions between position and rate control. On the other hand, the latter approach builds on the intuitiveness and rich expressiveness of shear-force input, allowing the direction of rate-control extension to be altered without switching back to position control. Although a carefully designed and formally conducted user study is still needed to draw definitive conclusions, the feedback from the preliminary user study already showed promising advantages of the Force Extension techniques.

Thinking beyond multi-touch, what Force Extension enables is a way to fluidly combine two complementary control mechanisms to achieve improved user performance and experiences. Its success as a HUI technique has two important factors. First of all, the matches of isotonic touch input with position control, and isometric force input with rate control, have both been shown as optimal combinations [Zhai95]. However, their "optimal" aspects are different and conveniently complementary to each other’s drawbacks. This not only makes a legitimate motivation to combine the two, but also sets up a good basis for the hybrid solution to create an even better result.

Secondly, the two extension methods both preserve some connections between the two control mechanisms after transitions are made. The context-force extension technique does this by taking the previous position control vector into the rate calculation, and the shear-force extension technique scales the micro-finger displacement (small change in position) by the pressure value to simulate the shear force vector. Because the two modes are not clearly separated, the user is able to carry over his/her cognitive memory of the previous interaction state, and gradually adopt the new control mechanism through a continuous interaction flow [Wang Baldonado00]. These two guidelines of *complementarity* and *coordination* can be widely adopted in the design process of hybrid systems to build effective HUIs and HVEs. As a main theme of this dissertation, they will be emphasized again in the discussions of Hybrid and Coordinated Virtual Environments (Chapter 3) and Object Impersonation (Chapter 4).

Chapter 3: Hybrid and Coordinated Virtual Environments

As mentioned in Chapter 1, traditional 3DUIs in immersive environments are centered on the user's virtual body and based on real world metaphors [Slater94]. Despite the realistic experience of grabbing and manipulating a virtual object using your hand [Poupyrev98a], or real walking [Zanbaka05] and flying [Wang12a] in the VE, they can still be just as confusing, limiting, and inefficient as in the real world [Stoakley95]. These limitations are especially evident when it comes to tasks with diverse requirements [Wang14]. For example, it is difficult to select and manipulate objects of different sizes, from multiple angles, and at different distances, without spending significant time and effort on navigation. Due to these limitations, successful application of VR in people's everyday lives is still far from commonplace.

One way to overcome the limitations of traditional 3DUIs is through HVE systems that offer multiple and complementary interactive representations of the same VE, each appropriate for a set of tasks. An example of an HVE system with multiple virtual world representations is the World-In-Miniature (WIM) interaction technique, which renders an interactive miniature world in the non-dominant hand of the user to complement the immersive context with quick teleportation, range-less object selection, and large scale object translation [Stoakley95]. In addition to combining multiple virtual interaction elements, HVE systems can also consist of different physical interfaces. An often adopted approach is the combination of a tracked surface and a spatial input pen. The physical surface offers good affordance of passive haptic feedback [Lindeman99] and bimanual interaction [Guiard87], leading to enhanced performance of 2D tasks such as system control [Watsen98], symbolic input [Poupyrev98b], and map-based way-finding [Bowman98b].

The rapid progress of mobile device technology has inspired a recent research trend of offloading 3DUI tasks to mobile phone and tablet devices, to take advantage of their growing computing power, high resolution, multi-touch touch screens, and various built-in motion sensors [Bornik06] [Song11] [Roberts12] [Wilkes12]. However, most of these techniques have been focused on very simple scenarios, where only one or two UI functions are assigned to the tablet to aid the primary spatial interface used in the immersive environment. And few studies have been conducted to investigate the overhead involved in transitioning between the multiple interface elements [Grasset08].

This chapter offers an in-depth presentation of an HVE level-editing system that aims to join the strengths of a tablet device and an HMD-and-wand-based immersive VR setup [Wang14]. Unlike previous research on tablet-based VR interfaces, in which a tablet is used as a tool supplementary to the primary spatial input device, the HVE level editor targets the tablet as a complete IC, which renders the entire virtual world on its own, and supports all 3DUI tasks through multi-touch gestures and 2D GUI elements. In addition, to reduce the perceptual, cognitive, and functional overhead [Dubois02] caused by complex 3DUI transitions across multiple ICs, a set of *coordination mechanisms*, featuring mutual awareness cues, input sharing, display blend-in, and 3DUI task synchronization, is proposed.

A user study was conducted to evaluate the effectiveness of task synchronization. The results suggest that task synchronization can lead to smoother transitions across ICs, and that user performance can be increased by using multiple *complementary* ICs in an HVE system. Finally, to summarize and extend the research contribution, a four-step design process is presented, which can be used to aid the design and implementation process of HVE systems, as well as their application in various appropriate task scenarios.

3.1 Related Work

3.1.1 Tablet-Based 3D Interfaces

Interactive tablets have been demonstrated as powerful tools for interaction in VR. The Virtual Notepad enabled handwriting using a stylus on a touchpad, providing an intuitive and effective interface to input symbols in immersive virtual worlds [Poupyrev98]. By displaying an interactive 2D map on a tracked touchpad, early pen-and-tablet prototypes also made way-finding and travel efficient in cluttered indoor spaces [Angus95], as well as in large-scale outdoor scenes [Bowman98b]. The Personal-Interaction-Panel (PIP) proposed concepts of a hybrid approach for object selection and manipulation, system control, and interaction with volumetric data [Szalavári97]. The main idea was to augment virtual objects with 3D widgets and 2D GUI elements on the tablet, both of which could be interacted with using a stylus. A pen and a semi-transparent pad were combined to enable Through-The-Lens (TTL) interaction with the virtual contents displayed on a tabletop [Schmalstieg99a]. From a usability point of view, an empirical study of a UI manipulation task has shown that the bimanual interaction and passive haptic feedback offered by a physical surface held in the non-dominant hand can significantly increase precision and efficiency, as well as reduce fatigue [Lindeman99]. Based on these advantages, the design guideline of "dimensional congruence" was proposed, which advocates matching the dimensionality of the 3DUI tasks to that of the input devices [Darken05].

With no tethers attached, mobile phone and tablet devices can provide more flexibility than traditional pen-and-tablet interfaces. The use of mobile devices in VR has grown with the advancement of mobile technologies. Early work by Watsen et al. demonstrated a handheld computer used as an interaction device, which only contained simple 2D GUI widgets to aid system control tasks in the VE [Watsen98]. As the computing power increased, researchers started to experiment with rendering interactive virtual objects on the screen of mobile devices, based on PIP [Bornik06] or TTL [Miguel07] metaphors. Recently, many mobile devices contain high-performance, multi-touch touchscreens. To take advantage of this, various 3D interfaces have been proposed that combine multi-touch gestures with spatial tracking of mobile phones or tablets for object manipulation [Wilkes12], volume data annotation [Song11], and textual data visualization [Roberts12]. Furthering this trend, a different perspective was taken in the design of the HVE level editor, which treated the mobile device not as a supplementary tool, but a complete interaction system, with computing power, display technology, and interaction richness comparable to that of an HMD-based, immersive VR system [Kin11]. Compared to using mobile tablets as a simple input device, this new approach can inspire new design possibilities of HVE systems to handle complex and highly diverse interaction tasks more effectively in 3D spaces.

3.1.2 Hybrid Virtual Environments

The early seminal work of Feiner & Shamash defined the term HUI as interface systems that combine heterogeneous display and interaction devices in a complementary way to compensate for the limitations of the individual devices [Feiner91]. Rekimoto & Saitoh adopted the idea and developed the Augmented Surfaces system, which used several computers and display devices to form a continuous hybrid work environment [Rekimoto99]. Just as the Force Extension technique attempted to smooth transitions between different control mechanisms (see Chapter 2), HVE systems also strive to *seamlessly* integrate multiple representations of the same VE, in order to facilitate 3D interactions from different angles, scales, distances, reference frames, and dimensions. The multiple VE representations in HVE systems are often related based on some natural metaphor. For example, the WIM technique combines an egocentric and an exocentric view of the virtual world through a “handheld miniature world” metaphor [Stoakley95], and has been adopted to unify the multiple ICs in the HVE level editor presented in this chapter. The Voodoo Dolls technique creates a second instance of a remote object in the local space following a well-known fictional metaphor [Pierce99]. The SEAMs technique defines a portal which can be traveled through, or reached into, to translate objects across two distinct spaces [Schmalstieg99b]. The Magic Lens technique adopts an x-ray see-through metaphor to offer different visualizations of the same virtual content side by side [Viega96].

HVE systems can also incorporate different physical interface components alongside the VE representations. As an example, the HVE level editing system presented in this chapter coordinates two VE representations contained in two ICs: a tablet device with multi-touch input and a 2D GUI, and an HMD-based VR system with wand input. Two closely related works are the HybridDesk, which surrounds a traditional desktop computer with a desktop VE display [Carvalho12], and SCAPE, which puts a see-through workbench display in the center of a room with projection walls [Brown06]. However, the former limited its ICs to exclusive 3DUI tasks, forcing the user to make unnecessary switches, and the latter mainly focused on view management, instead of rich 3D interactions.

Much research work in transitional user interfaces and CVEs is also closely related to HVEs. Transitional user interface systems present multiple representations of the virtual world in a linear, time-multiplexed way [Grasset08]. The MagicBook is a classic demonstration of a transitional experience between an exocentric view of the VE in Augmented Reality (AR) to an egocentric view represented in immersive VR [Billinghurst01]. Many CVEs can be considered as HVEs with their multiple VEs assigned to different users. A well-known metaphor is the combination of a God-user and a Hero-user, who possess complementary views and reference frames in the shared

VE to aid each other towards a common goal [Holm02] [Wang13a]. The unique challenge of designing CVE systems is to ensure the collaborators are well aware of each other's viewpoints and interaction intentions as tasks are carried out. Avatars and artificial cues have been found to be effective for this [Churchill98]. Finally, it is also possible to merge hybrid, transitional, and collaborative virtual environments together into a hybrid collaborative system, such as the VITA system [Benko04].

3.1.3 Cross-Context Transitions

Compared to traditional VR, one main challenge for HVE systems is the perceptual, cognitive, and functional overhead induced by transitions across multiple virtual and physical components [Dubois02]. The challenge is also present in coordinated multiple view systems, where multiple views of the same dataset are generated and displayed to help the data analyst discover unforeseen patterns. The key to reducing the transition gap in these systems is to coordinate the visualizations of, and the interactions with, the multiple views [Wang Baldonado00]. For example, multiple views can be "snapped together" to better reveal their relationships and ease the gap between transitions [North00]. Multiple views of 3D data can also be linked [Plumlee03], or integrated through frame-of-reference interaction [Plumlee03]. Guidelines for view management have been provided to minimize the cognitive overhead of context switching [Wang Baldonado00]. Applications and study results have demonstrated improvements in user performance when coordination mechanisms are implemented [Ryu03] [Steinicke06]. Inspired by these findings, the remainder of this chapter presents four coordination mechanisms that can keep the complex 3D interaction transitions simple and smooth in the proposed HVE system.

3.2 Immersive Level Editing

The early design and development of the presented HVE system was heavily influenced by the short concept film *World Builder* by Bruce Branit, as shown in Figure 3.1. The film centers on a man who builds a virtual world for his partner who is in a coma. The hospital where she is staying supports neuro-holographic input which allows patients to experience full-sensory virtual worlds. The building process begins with the man standing in an empty space, and building up a rich city scene by combining primitive geometric objects, positioning, scaling, and rotating them, applying textures, cutting and pasting objects, applying lighting effects, and more, all from within the world itself. During the world building process, the man uses many interaction techniques, such as hand gestures, image-based group selection, bimanual interaction, and hybrid hand + tablet selection techniques.

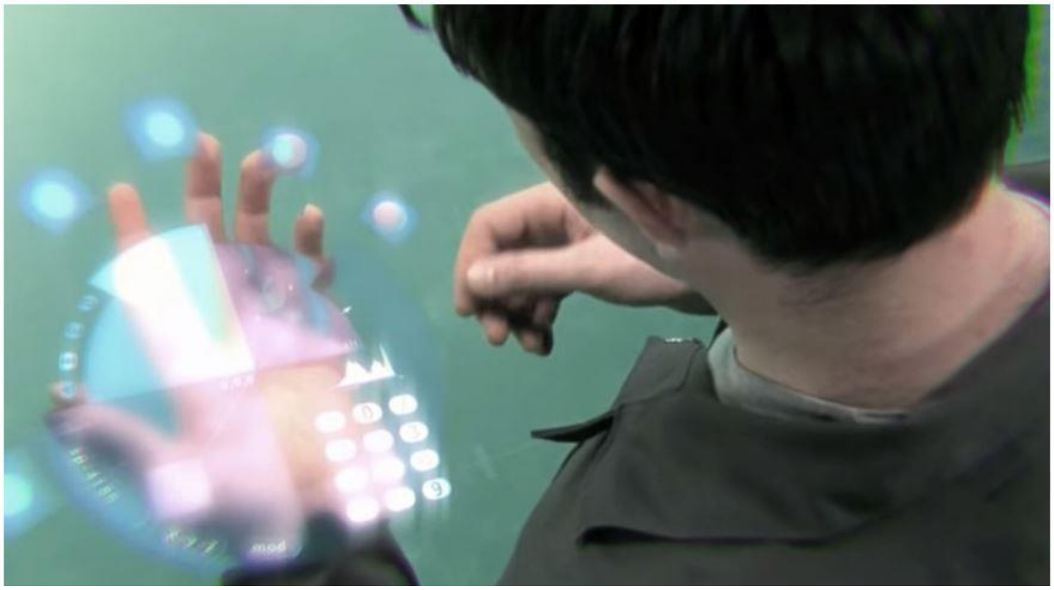


Figure 3.1: The concept film *World Builder* (*courtesy of* Bruce Branit)

Influenced by the rich and effective interaction metaphor demonstrated in the film, a goal was set to simulate the same experience using off-the-shelf technology. A prototype system called the

“DIY World Builder” was then designed, developed, and successfully demonstrated in the annual IEEE 3D User Interfaces Contest in 2013 [Wang13a]. As shown in Figure 3.2, my solution took advantage of the multi-touch input, advanced computing power, and high-resolution display of a modern phone/tablet device to render a highly interactive God-view representation of the VE that complements the traditional wand-based 3D interaction in the immersive virtual world.

Figure 3.2: The *DIY World Builder* system uses a phone/tablet device with a “wand + HMD” VR setup.

Based on the success of *DIY World Builder* [Wang13a], virtual world editing was finalized as the test bed to drive the design and study of the HVE system. It was selected for several reasons. First, level editing plays a key role in many real world applications, such as video game design, animation production, and urban planning. Second, many level-editing tasks feature diverse and complementary requirements, which makes them good candidates to adopt HVE approaches [Bowman98b] [Steinicke06]. Third, unlike the simple and monotonous tasks most VR studies have been designed for (*e.g.*, travel from A to B [Zanbaka05]), level editing actually involves all 3DUI tasks (*i.e.*, navigation, selection, manipulation, system control, and symbolic input) and combines them in various ways. This grants an opportunity to study complex *3D interaction transitions* *across multiple ICs, and the overhead involved in the process*. The following specific level-editing tasks were defined and implemented in the HVE level editing system:

- **Terrain height editing:** The height of the terrain surface can be raised, lowered, or aligned to a pre-sampled height within the range defined using a circular terrain brush.
- **Terrain texture editing:** The terrain surface can be painted with a selected texture using the terrain brush.
- **Foliage editing:** Trees and grass can be planted on the terrain surface using the terrain brush.
- **Object geometry editing:** Objects in the virtual world can be created, selected, manipulated, duplicated, and deleted. Manipulation includes translation on the terrain surface, rotation around the up-axis, and isometric scaling.
- **Object texture editing:** The subparts of the objects, such as the roof of a house, can also be painted with different textures in customizable scales.
- **Time-of-day editing:** Users can change the time of day, which affects the sunlight and the textures of the skybox.

3.3 Interaction Context

This section underscores the notion of an interaction context as an important concept in HVE systems. As discussed in Chapter 1, an Interaction Context (IC) is “a conceptual integration of input and output devices, techniques, and parameters, which offers the immersed user one interactive representation of the virtual environment”. HVE systems are formed by relating multiple ICs under a unified metaphor. The metaphor defines the conceptual relationship between the ICs, making it more likely for the user to consider the overall HVE system as an integrated whole. Common HVE metaphors include WIM [Stoakley95], Portal [Schmalstieg99b] [Kiyokawa05], Voodoo Doll [Pierce99], See-Through Lens [Viega96] [Brown06], and Information Surround [Feiner91]. For the HVE level editor, WIM was selected as the metaphor to combine the exocentric "God" view with the egocentric first-person "Hero" view. An IC can be formed by specifying the following components:

- **Medium:** The type of medium adopted by the IC on the reality-virtuality continuum [Milgram95], such as VR, AR, or mixed reality.
- **Display device:** The multi-sensorial devices used to display the virtual world to the user's sensory organs, such as HMD, CAVE [Cruz-Neira92], headphones, haptic stylus, etc.
- **Rendering technique:** The technique used to render the virtual content to the display device (*e.g.,* shaders for visual display).
- **Input device:** The device(s) used to express commands, such as a data glove or a multi-touch touch pad.
- **Interaction technique:** The software that maps the input data to control parameters in the virtual world. For example, wand input devices usually use ray-casting based interaction techniques [Poupyrev98a].
- **Perspective:** The position, orientation, and other parameters of a virtual camera that determine the IC's view of the virtual world. Immersive VR systems usually offer an in-the-world, first-person perspective.

- **Reference frame:** The coordinate system that determines the perception of the virtual world and the effect of interaction. Egocentric (body-centered) and exocentric (object-centered) are two reference frames commonly discussed in VR [Plumlee03a].

This list of components defines a taxonomy that can be used to categorize HVE systems. For example, the original WIM interaction technique includes two ICs [Stoakley95]. Both ICs use VR as the medium, and render their views of the VE in the same HMD, using a photorealistic shader. In addition, a buttonball prop is used in both ICs to interact with virtual objects, using a collision-based pick-and-drop technique. However, the two ICs are different in their perspectives and reference frames. The immersive IC has an in-the-world, first-person view where all interactions are based on the user's egocentric body, while the miniature IC adopts an above-the-world, God view with object-centered exocentric reference frame.

Another example, the HybridDesk system [Carvalho12], features two ICs with more differences in hardware components. The immersive IC uses a desktop CAVE system as the display device, and a Wii Remote controller-based wand interface as the input device to do travel, object selection and manipulation in the virtual world. The desktop IC can be activated by bringing its display device, a regular desktop monitor, to the front of the user. It uses the standard mouse and keyboard as input devices to support efficient performance of system control tasks, such as selecting models from a hierarchical repository, or entering text to annotate 3D objects. The desktop IC is also used to bring selected virtual objects close to the user (*i.e.*, creating a Voodoo Doll), to enable effective editing of small details from an exocentric reference frame. Both ICs use VR as the medium, and render the VE photo-realistically. The HVE level editor presented in this chapter incorporates an immersive IC and a tablet IC, whose components are specified in Table 3.1.

Table 3.1: The IC components of the HVE level editor

Components	**Immersive IC**	**Tablet IC**
Medium	Virtual reality	Virtual reality
Display device	HMD, fans	Tablet screen
Rendering technique	Photorealistic	Photorealistic
Input device	6-DOF wand & head tracker	Touch screen
Interaction technique	Ray-casting & button based	2D GUI & multi-touch gestures
Perspective	In the world	Above the world
Reference frame	Egocentric (body-centered)	Exocentric (object-centered)

3.4 Immersive IC

As shown in Figure 3.3a, an eMagin Z800 HMD is used to display a first-person, in-the-world view of a photorealistic VE, with a 60-degree horizontal field-of-view (FOV). The HMD utilizes two 800x600 OLED screens to render mono-scopic images to both eyes with a 40-degree diagonal FOV. It is tracked in six degrees of freedom (DOF) using the PhaseSpace motion capture system. A constellation of four active LED markers is attached to the top of the HMD and tracked by sixteen cameras surrounding an octagon-shaped cage space, with the user seated in a swivel chair in the center. Since the HMD is non-occlusive, the user is able to see the display in the center of his/her field of view, as well as look at the screen of the tablet by gazing down below the bottom edge of the HMD. While the user is traveling, a group of fans corresponding to the direction of the locomotion is turned on, and blows wind at a constant speed to enhance the sense of motion in the virtual world.

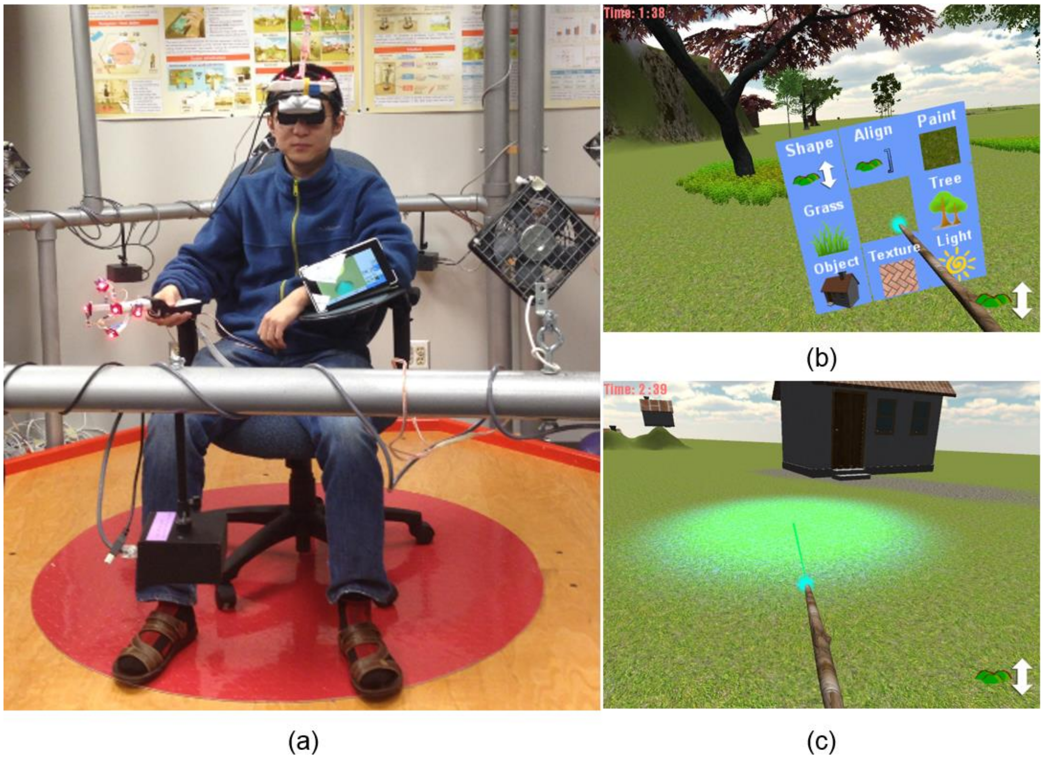


Figure 3.3: The hardware setup (a), the floating menu (b) and terrain brush (c) of the immersive IC in the HVE level editing system.

A wand interface is provided to the dominant hand of the user to enable 3D interaction in the immersive VE. The wand is made by attaching a 6-DOF tracking constellation to a Wii Remote controller. 3DUI tasks are performed by pointing the wand and pressing buttons to issue commands. To navigate within the VE, the user can point the wand in different directions, and press down the D-pad buttons to travel in that direction at a constant speed. To preserve a realistic feeling, virtual locomotion is always constrained to the ground, but the swivel chair gives extra flexibility to point the wand easily in all directions.

As listed in Section 3.2, the user can perform a set of level editing tasks in the VE. To select an editing mode, the user can call out a floating menu as shown in Figure 3.3b, by holding down the “home” button on the Wii Remote controller. The tile pointed to by the wand is highlighted, and the corresponding editing mode is selected upon release of the “home” button. In the modes of terrain shape, texture, grass, or tree editing, a ray is cast from the tip of the wand to the intersection on the terrain surface, and a circular terrain brush is visualized to indicate the effective range. The size of the terrain brush can be changed using the "+" and "-" buttons on the wand controller. The "A" and "B" buttons have opposite effects. The former is used to raise, align, and plant trees and grass, while the latter is used to lower, sample, and remove trees and grass. In object editing mode, the objects in the VE, such as houses, can be selected by ray-casting and pressing the "A" button, or deselected by pressing the "B" button. Objects are highlighted in light blue when being pointed at, and in bright blue when actually selected. Once selected, the user can drag the object on the terrain surface by holding the "A" button, rotate it around the up-axis by pressing the left and right buttons on the D-pad, or scale it by pressing the "+" and "-" buttons. Lastly, the user can paint subparts of the virtual objects with different textures, as well as change the scale of each texture.

3.5 Tablet IC

Figure 3.3a shows a user wearing a Google Nexus-7 tablet on his left forearm, and resting it on an armrest to reduce fatigue. To leverage bimanual interaction [Lindeman99], the user is asked to hold the wand interface temporarily in the left hand, or place it between the legs, and use the right hand to apply multi-touch gestures to the touch screen. The interface on the tablet is illustrated in Figure 3.4. It consists of a three-tier GUI menu, a WIM view of the VE, and a shortcut bar. The top tier (1) is a tool bar for switching between the general editing modes. The tool bar at the second tier (2) displays further sub-modes, such as height, texture, grass, and trees for terrain editing. Based on the selection in the first two tiers, the third tier (3) shows specific GUI elements that can be used to perform the current task, such as a slider to resize the terrain brush, a selection grid to choose a type of grass to plant, and a broom button to clean grass from the terrain. Note that the immersive IC and the tablet IC each have their own terrain brush, so that terrain editing can be performed at different scales.

Figure 3.4: The tablet IC used to edit the VE from the God view

To the right of the third-tier panel, an above-the-world, photorealistic, third person view of the VE is presented (4), whose camera has a 60-degree horizontal FOV in the VE, and can be manipulated using multi-touch gestures. These include a pinch gesture for zooming, a rotate gesture for orbiting, a two-finger all-direction swipe gesture for panning, and a three-finger up-and-down swipe gesture for pitch control. The one finger tap and swipe gestures are reserved for level editing, such as painting the terrain, or dragging an object on the terrain surface. The functionality of the shortcut buttons (5) will be discussed later.

Regarding the software implementation, the HVE system was developed using the Unity game engine as a multi-player game running separately on the desktop and the tablet platforms. The hardware devices of the immersive IC are connected to the desktop computer through USB and Bluetooth connections. The input data from the PhaseSpace motion capture system and the Wii

Remote controller are collected and streamed to the game process through VRPN [Taylor01] and the Unity Indie VRPN Adapter (UIVA). Both the desktop and the tablet simulate the VE locally, and keep each other synchronized by sending UDP data streams and RPC calls over a local WiFi network. This way, both ICs can run the game at a steady 30 frames per second, and editing performed in one IC can be propagated to the other IC in real time, giving the user a convincing experience that they are viewing and interacting with the same virtual world, only from two different perspectives.

3.6 Coordination Mechanisms

The advantages of the two ICs can complement each other to support diverse tasks efficiently. For example, a fast way of moving a small object across a long distance in the VE is to select the object in the local space using the wand, and drag it to the destination using the tablet. However, such processes involve frequent switches between the ICs, and the mental overhead of adapting to different IC components cannot be overlooked. The challenges to create smooth transition experiences in the HVE level editor are further illustrated in Figure 3.5, in which each level-editing task is decomposed into a set of basic 3DUI tasks.

As shown in Figure 2.5, the user's virtual world editing workflow may start with any low-level 3DUI task in one IC and end with another task in a different IC. During transitions, the user needs to understand the relationship between the two VE representations, and adapt to distinctly different display devices, input devices, interaction techniques, reference frames, and perspectives. To reduce this transition gap, I propose the following four different coordination mechanisms.

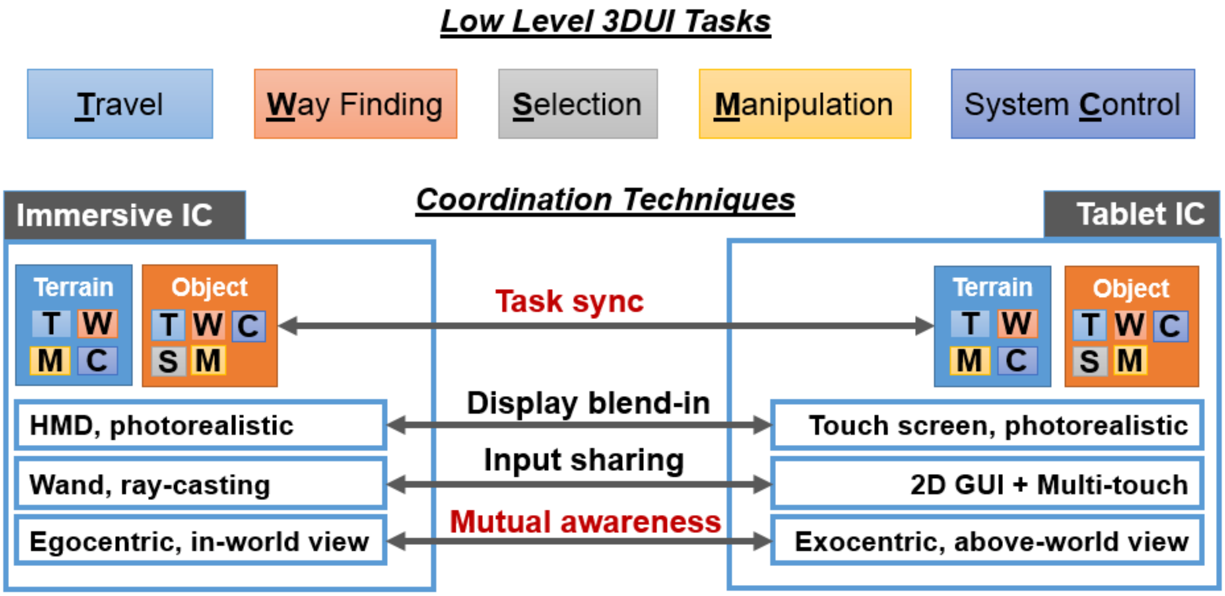


Figure 3.5: The four different coordination mechanisms aimed to smooth the complex cross-task, cross-IC transitions in the HVE level editor.

- **Task synchronization:** In coordinated multiple view systems, the multiple data views are often implemented to be consistent during user interaction [North00] [Plumlee03b] [Wang Baldonado00]. Similarly, the effect of low-level 3D interaction (*e.g.*, travel, way-finding, selection, manipulation, system control, and symbolic input [Bowman04]) in one IC should also be propagated to all other ICs, to avoid interrupting the user's workflow during transitions. For example, when a user changes to object editing mode (*i.e.*, system control) and selects an object using the wand (*i.e.*, selection), the tablet should also update to the same mode and select the same object, so that the user can directly continue to manipulate this object after changing the IC. Without synchronization of the low-level 3DUI tasks, the user's work would be interrupted, forcing her to repeat actions.

- **Display blend-in:** The change of display device can cause perceptual discrepancies between ICs due to differences in screen size, resolution, brightness, and other parameters. Using mixed reality technology [Bruder10], the image of one IC's display device can be embedded into another IC's view to reduce this discrepancy. For example, compared to viewing the tablet screen from the peripheral vision, a better experience may be provided by tracking and rendering a virtual tablet in the HMD view, in place of the physical tablet itself.

- **Input sharing:** Some generic input devices, such as the mouse and keyboard, can be effectively used in multiple ICs [Benko04]. For example, a similar HVE system can be formed using a desktop computer and a tablet. In this situation, the mouse and keyboard could be efficient tools for controlling both the first-person view on the monitor and the God view on the tablet. The system could even incorporate an eye tracking device to detect the user's gaze, so that the mouse and keyboard input events could be directed to control the IC being currently looked at. Sharing input among ICs may not only reduce the mental overhead of transitions between interfaces, but also the physical effort of switching between devices.

- **Mutual awareness:** Research in CVE systems has stressed mutual awareness as the key to efficient human collaborations in VR [Churchill98] [Holm02]. This rule can also be applied to HVE systems where different views are assigned to the same user. By knowing the whereabouts of the other view and the status of its interfaces, the user can better determine when to make the IC transition, and be more prepared to adapt to the new IC once the transition is made. Examples of effective mutual awareness cues include avatars, viewing frusta, pointing rays, and editing brushes.

Of the four coordination mechanisms, task synchronization and mutual awareness cues have been implemented in the current version of the HVE level editor. Figure 3.6 shows an example of the implementation in object-editing mode. The ultimate goal of this mode is to properly arrange virtual objects in the scene, through manipulation of the objects' positions, orientations, and scales. Manipulation is preceded by enabling object-editing mode (system control), moving to an appropriate spot (travel), and selecting the object (selection).

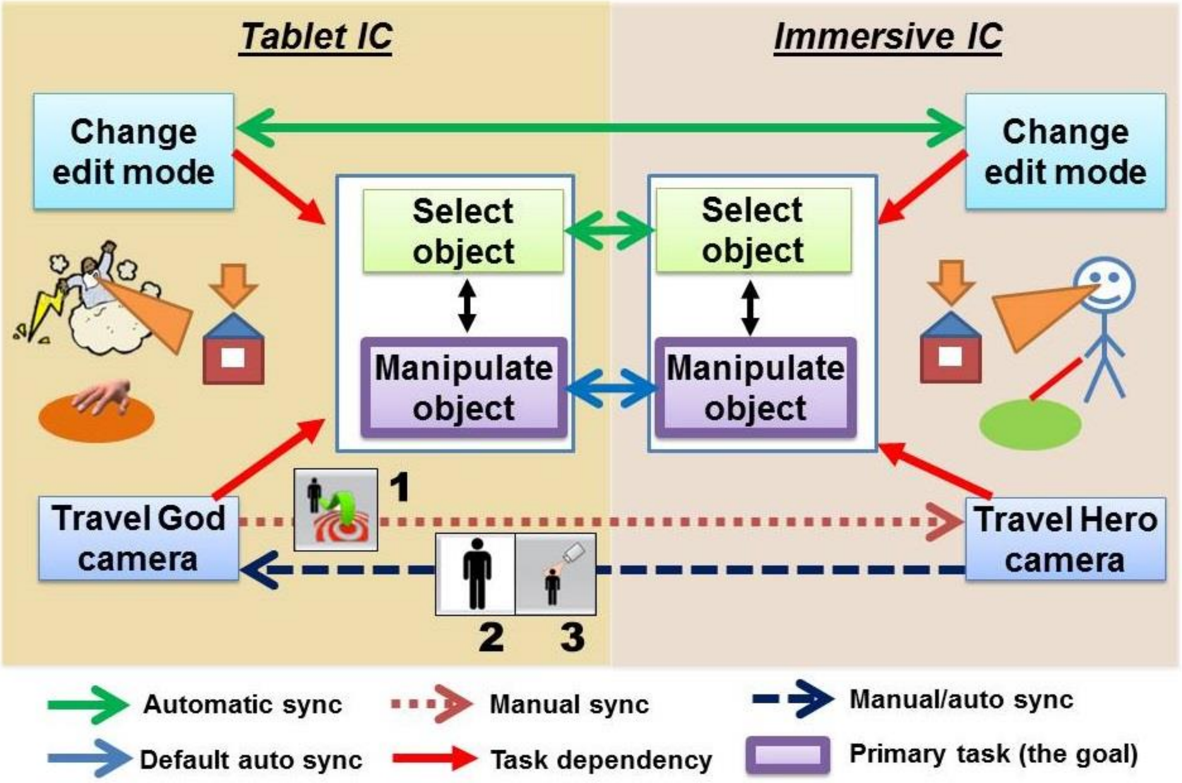


Figure 3.6: An example of task synchronization and mutual awareness cues implemented in the HVE level editor to achieve seamless IC transitions in object selection and manipulation tasks.

By default, the effect of object manipulation is synchronized between the two ICs, as the VE needs to look the same on both displays. However, synchronization of the preceding steps is optional, and very much dependent on the level of multi-tasking a hybrid system aims to support. I hypothesize that by synchronizing the effects of all 3DUI basic tasks, the working-memory demands required to keep track of the status of 3D interactions across ICs can be effectively reduced, leading to better task performance and user experience. Thus, task synchronization was implemented, with the goal of minimizing the mental overhead during transitions between the ICs. As illustrated in Figure 3.6, changing the editing mode or selecting a virtual object in one IC is always automatically synchronized to the other IC.

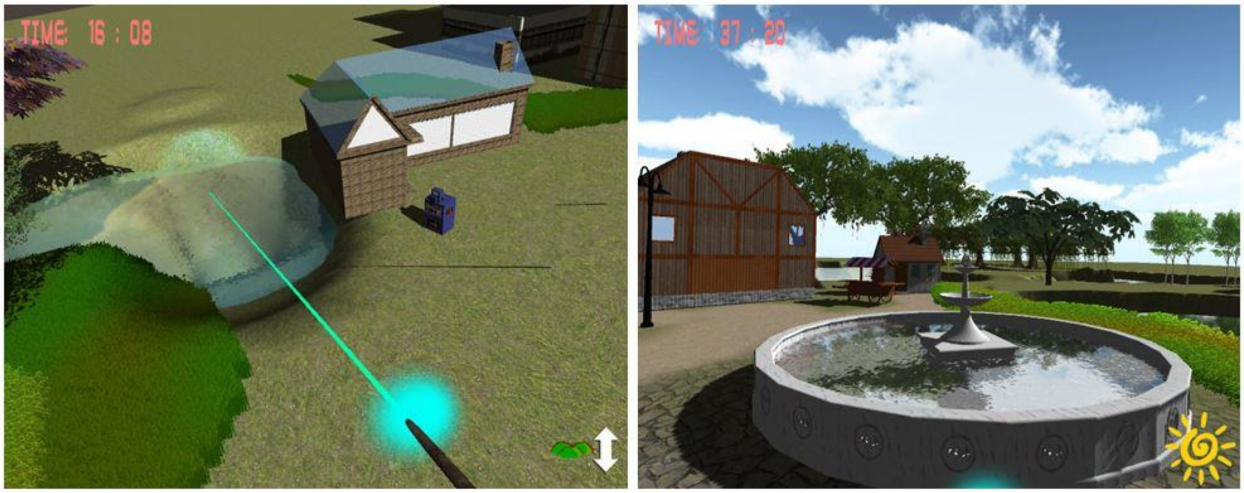
The virtual location of the God and Hero viewpoints are also synchronized by enabling teleporting of the Hero avatar. Teleporting the user's Hero avatar to the field of the God view is done *manually* with the tap of a shortcut button (1) on the tablet, as previous research has indicated that constantly changing an immersive view can cause disorientation and even motion sickness symptoms [Stoakley95]. To synchronize the God view with the space surrounding the Hero avatar, the user can either tap a button (2) to focus the God camera on the Hero avatar, or switch a toggle (3) to enable/disable the God camera to follow the Hero avatar as the user travels around the VE in the immersive IC.

3.7 User Study

3.7.1 Preliminary User Study

The prototype "DIY World Builder" system was tested in a preliminary user study. The subjects were introduced into a blank virtual world with only a flat terrain, and asked to build a level to their liking. They were free to ask questions and give comments during the exercise. Task synchronization and mutual awareness were always enabled, and the subjects were able to try out all level editing tasks listed in Section 3.2. Four users, including one expert, participated in the study. The subjects had no prior experience using immersive VR technology, but were all everyday smart phone users with significant experience in multi-touch gestures. The expert user was a college student majoring in game design, and used the Unity3D game engine to do level design on a daily basis.

Two sample worlds built by the users are shown in Figure 3.7. The duration of immersion ranged from 45 to 90 minutes, and feedback was generally positive. All of the users said they would have liked to continue, and that the system was fun to use. Interestingly, fatigue was not mentioned as an issue during the sessions. The user comments were separated into two categories: feature requests and system improvements. In terms of feature requests, the ability to undo and redo actions, and save and load created worlds were the most common system control features that were requested. One user asked for a way to smooth the terrain better, and the ability to resize trees was requested by three users. Two users asked for more objects, such as leaves, rain, and fish, and the ability to change the water color.



(a) (b)

Figure 3.7: Virtual worlds created by novice (a) and expert (b) users in the preliminary user study.

Regarding system improvements, two users commented that changing the focus from the tablet to the HMD breaks presence and interrupts the flow of interaction, which motivated the four coordination mechanisms proposed in Section 3.6. One user also requested a “sample” tool, whereby objects, colors, or textures currently used in the scene could be sampled, and then applied, which would speed creation of similar objects. The user feedback informed the next iteration (see

Section 3.4, 3.5, and 3.6), which was formally evaluated in a larger scale user study, with hypotheses focused on transition continuity between the immersive and tablet ICs.

3.7.2 Hypotheses

Our HVE system aims to combine the strengths of an immersive VR setup and a multi-touch tablet device. Being inside the virtual world, the user can better understand the space, judge scales of objects, and do manipulation of finer details [Holm02]. Meanwhile, from the God view, the user can better navigate the VE, investigate the overall layout, and perform large-scale manipulations [Stoakley95]. The two ICs are unified under the WIM metaphor, and coordinated through mutual awareness cues and task synchronization. Based on these analyses, the following hypotheses were made:

- ***H1*****:** Having the effects of basic 3DUI tasks synchronized between the ICs can make the transitions more continuous, and lead to better task performance and user experience.
- ***H2*****:** The users are able to learn the HVE system despite its complexity, and use both ICs to handle tasks with diverse requirements.
- ***H3*****:** The users are able to decompose a complex, high-level task into a series of basic 3DUI tasks, and find step-by-step strategies to efficiently leverage the complementary benefits of both ICs.

H2 and ***H3*** are trying to capture higher-level processes, such as user behavior, as opposed to low-level, performance-based claims. These unconventional research hypotheses were made with a strong belief that 3DUI research is now at a stage where this level of assessment is possible and necessary. As will be shown in Section 3.7.4 and 3.7.5, various approaches were used to assess these unconventional hypotheses, including post-questionnaires, subject interviews, and video capture of task performance.

3.7.3 Formal User Study

An initial task design was first experimented with in a pilot study, where subjects were asked to duplicate a pre-built virtual scene in an empty space from scratch. This design was rejected after testing with two subjects, who got lost quickly with too much freedom and too many options to build a relatively complex virtual world. Therefore, a different approach was taken in the actual study. Instead of building a virtual world from scratch, the study presented the subjects an unfinished virtual world (see Figure 3.8), and asked them to find and fix five different types of design flaws in the VE as quickly and precisely as possible:

- **Fix mountain textures:** Paint the mountain sides with rock, the mountain top with snow, the platform with dirt, and the ground with grass textures.
- **Match trees and flowers:** Plant flowers under each tree, and plant a tree on each patch of flowers.
- **Clear foliage in rivers:** Remove trees and grass in the rivers.
- **Correct houses:** Edit the terrain surface to make the houses stand on flat ground. Scale the house and the roof textures to realistic sizes.
- **Collect cubes:** Collect small cubes spread around the virtual world, and bring them close to a large cube.

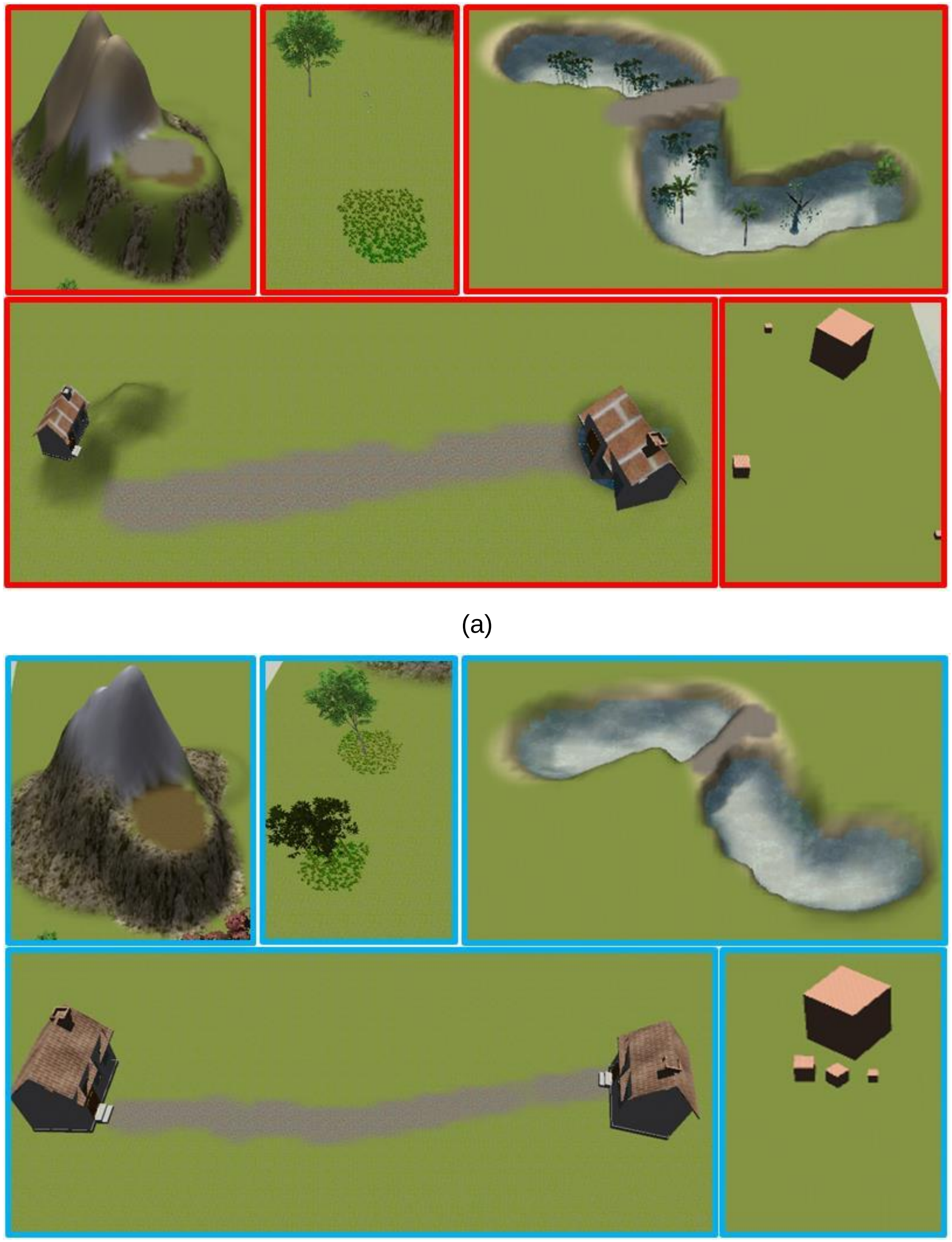
This applied task approach was chosen for several reasons. First of all, based on natural metaphors, the design flaws were clear to identify, and the goals easy to understand and remember. Secondly, compared to building a VE from scratch, fixing existing design flaws takes less time to complete, making the threats such as user fatigue and motion sickness much more manageable. Thirdly, to complete the tasks efficiently, the subject needed to take different angles, interact at different scales and reference frames, and use different interfaces. This encouraged the subjects to learn both ICs, and explore different ways to use their complementary advantages. Finally, the tasks made it possible to study both between-task and within-task IC transitions. As an example of *between-task transitions*, the user could plant flowers and trees using the wand, followed by riverbed foliage cleanup using the tablet. As an example of *within-task transitions*, the cube collecting task could be done by searching for the small cubes on the tablet, teleporting the Hero avatar close to the cube, selecting it with the wand, and dragging it to the destination using the tablet again.

Figure 3.8: The study task is to fix five types of design flaws in an unfinished VE.

The study employed a within-subjects approach to compare the HVE level editor with and without task synchronization (indicated by green lines in Figure 3.6). The study began with the subject reading and signing the consent form approved by the institutional review board (IRB). The subject then answered a demographic questionnaire that asked about gender, age, and handedness, as well as experiences with immersive VR, multi-touch devices, multi-screen devices (*e.g.*, the Nintendo WiiU), real-time strategy games, first-person shooter (FPS) games, first-person world building games (*e.g.*, Minecraft), and 3D modeling software, in a range from one to six (one for “Never” and six for “Everyday”).

The subject was then introduced to the hardware used in the study, including the HMD, the wand, the tablet, and the fans. While having the freedom to swivel the chair, the subject was asked to stay in the center of the cage, to keep the best tracking quality of the motion capture cameras. The experimenter also explained the five world-fixing tasks as illustrated in Figure 3.9. The subject then put on the equipment, and learned the interfaces and the tasks in a 20-minute training session. Because the system supported multiple ways to perform the same tasks, the experimenter first explained all approaches, using either the wand or the tablet, in a training session. The VE used in this session had the five types of design flaws and the goals shown side by side as in Figure 3.9.

After the training session, the subject took a five-minute break, and then continued through two experimental conditions, each of which had one trial of world editing tasks. The conditions were presented to the subject in counterbalanced order, and only one of them had task synchronization enabled. To get used to the HVE system with different configurations, the subject spent eight minutes in a practice scene prior to each trial. In each trial, the subject had up to 15 minutes to fix the virtual world, but could end the trial early if they felt all design flaws had been addressed.



(a)

(b)

Figure 3.9: The five types of design flaws were shown side-by-side to the subjects. Images with Red borders (a) need fixing, and those in Blue (b) show them fixed.

After completing both conditions, the subject was asked to fill in a questionnaire to compare the HVE level editor with and without task synchronizations enabled, and to rate them on a one to six scale regarding questions such as efficiency, ease of learning, ease of use, smoothness of IC transitions, understanding of spatial relationship between ICs, time and mental effort to adapt to new ICs, and level of integration. At the end, the subject was interviewed to indicate any perceived perceptual, cognitive, and functional disconnections between ICs, and to give comments about the benefits and drawbacks of having multiple ICs and the effectiveness of task synchronization.

With approval from the IRB, 25 WPI students were recruited with no remuneration. Only one subject dropped out of the study because of motion sickness, and commented that the visual change from spinning the chair made him uncomfortable. Of the 24 subjects who successfully completed the experiment, 18 were males and six were females. Only one subject was left-handed. Their ages ranged from 18 to 34 years (mean = 22.8, SD = 4.2). Their experiences with VR ranged from 1 to 6 (mean = 3.9, SD = 1.5), multi-touch devices from 2 to 6 (mean = 5.5, SD = 1.2), multi-screen devices from 1 to 4 (mean = 2.1, SD = 1.2), real-time strategy games from 1 to 6 (mean = 2.9, SD = 1.4), first-person shooter games from 1 to 5 (mean = 3.2, SD = 1.4), first-person world building games from 1 to 5 (mean = 1.9, SD = 1.2), and 3D modeling software from 1 to 4 (mean = 1.9, SD = 1.0). These demographic data were used in correlation analyses with the subjects' task performance and questionnaire responses, and will be presented in the next section.

3.7.4 Results

At the end of each trial, the system recorded the total time spent, and saved the edited VE into a data file. All VE data files were then reloaded and independently rated by the experimenter and an external anonymous reviewer, to evaluate the editing quality and the level of completeness of the end-result VE. The rating process involved comparing the completed VE with a snapshot of the sample VE, and grading each subtask following the rubric detailed below, adding up to a maximum possible score of 100 points.

- **Fix mountain textures (25 points):** Paint the mountain sides with rock (7 points), the mountain top with snow (7 points), the platform with dirt (4 points), and the ground with grass textures (7 points).

- **Match trees and flowers (18 points):** Plant flowers under each tree, and plant a tree on each patch of flowers. Each of the six trees in the VE was worth 3 points.

- **Clear foliage in rivers (12 points):** Remove trees (6 points) and grass (6 points) in the rivers.

- **Correct houses (25 points):** Edit the terrain surface to make the houses stand on flat ground (8 points). Scale the house (8 points) and the roof textures to realistic sizes (9 points).

- **Collect cubes (20 points):** Collect small cubes spread around the virtual world, and bring them close to a large cube. Each of the five cubes in the VE was worth 4 points.

The inter-rater reliability was evaluated using Pearson's correlation analysis and the result showed high agreement (R = 0.92). As indicators of task performance, the task time, task score, and score-per-minute of the two conditions were compared using two-sided, paired t-test, with a threshold of 0.05 for significance. Score-per-minute was calculated by dividing score by time, and used as a measure of user efficiency. As indicated in Figure 3.10, subjects spent less time, and achieved higher task completeness, with task synchronization. The results are statistically significant for score-per-minute (p = 0.02), and showed trends for task time (p = 0.08) and score (p = 0.07).

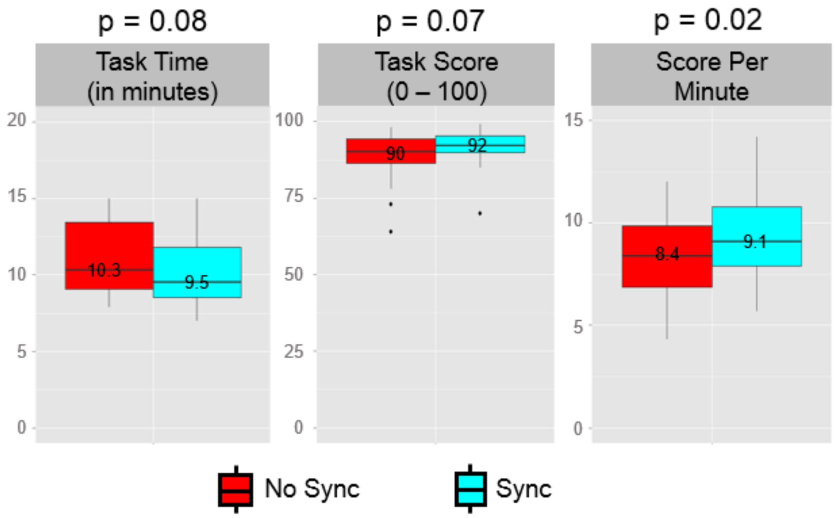


Figure 3.10: The analysis results of task performance indicators.

A Pearson correlation analysis was performed between these task performance measurements and the subjects' demographic information, with a special interest in their prior experience with related hardware and software applications. Although no very strong correlation was discovered (all correlation coefficient values were below 0.7), an interesting trend was found, showing correlations between the performance measures and the subjects' prior experience in video games, FPS games, and world building games, were much stronger when task synchronization was enabled. Specifically, the inverse correlation coefficient values in "Sync" and "No-Sync" modes were -0.61 and -0.25 respectively between video game experience and task time, -0.53 and -0.19 between FPS games and task time, and -0.51 and -0.16 between world building games and task time. Similarly, the coefficient values in "Sync" and "No-Sync" modes were 0.55 and 0.27 respectively between video game experience and score-per-minute, 0.51 and 0.23 between FPS games and score-per-minute, and 0.56 and 0.67 between world building games and score-per-minute. In other words, subjects' expertise predicted their performance with the "Sync" condition but not the "No-Sync" condition. With more experiences dealing with complex spatial and multi-view tasks, expert users can benefit more from the "Sync" condition than non-expert users.

The six-point rating scores of the two conditions were analyzed using two-sided Wilcoxon signed-rank tests with a threshold of 0.05 for significance on all questions. As indicated in Figure 3.11, the HVE system with task synchronization was considered to be more efficient, easier to learn, and easier to use, and the transitions between ICs smoother, and less time and mental effort demanding.

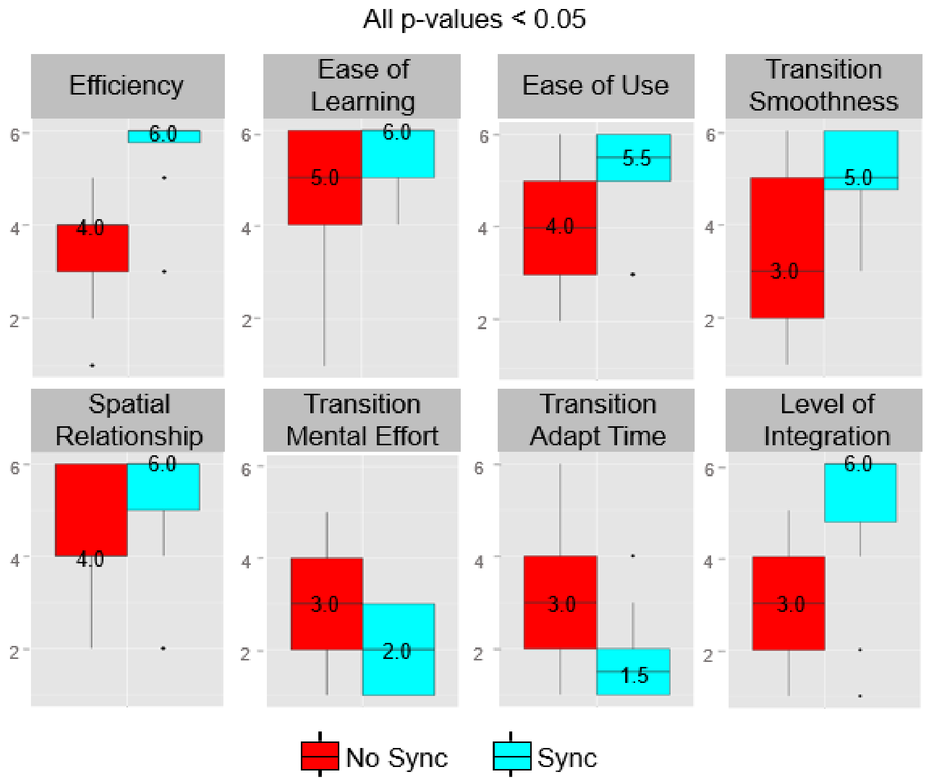


Figure 3.11: The analysis results of subjective rating scores.

In addition, the subjects felt the task synchronization mechanisms made it easier to understand the spatial relationship between the two VE representations, and the ICs were better integrated in the HVE system. All results were statistically significant ($p < 0.01$). Pearson correlation analyses between the rating scores and the subjects' prior experiences were also performed, however, no strong correlation coefficient was identified.

3.7.5 User Feedback

In the interview, the experimenter explained the definitions of perceptual, cognitive, and functional transition overhead [Dubois02], and asked the subjects whether they experienced any of such disconnections between the ICs when the transitions were made. The summary of their answers indicated better transitional continuity when task synchronization was enabled. The number of subjects who reported disconnected experiences, comparing "Sync" with "No-Sync," were six and 11 for perceptual disconnection, one and seven for cognitive disconnection, and two and 16 for functional disconnection.

Table 3.2: The summary of comments on "disconnected interaction experience"

Sync	**No-Sync**	**Comment about "disconnection"**
6	11	*Perceptually* disconnected experience.
1	7	*Cognitively* disconnected experience.
2	16	*Functionally* disconnected experience.

For the "Sync" condition, eight subjects complimented the synchronization of the editing mode, for emphasizing strong connection between the ICs, and making sure the non-active IC always kept up with the user's workflow in the active IC. The travel synchronization buttons on the tablet (teleport, focus, and follow) also had significant contributions to the smooth transition experiences, according to eight subjects who claimed that "the two views were spatially connected with these buttons" and that "the appropriate camera view was always available at hand when I tapped these buttons." Synchronization of selected objects was also liked by four subjects, as it enabled effortless within-task transitions, such as picking up a small cube using the wand and dragging it across the virtual world on the tablet screen.

Table 3.3: The summary of comments on about the “Sync” mode (“+” indicates positive comment)

Sync	Comment
8	+ *System control* synchronization ensures smooth workflow.
8	+ *Travel* synchronization makes transition smooth.
4	+ *Object selection* makes within-task transition effortless.

For the “No-Sync” condition, seven subjects felt the ICs were disconnected, and the overall HVE system was confusing and awkward to learn and use. Because the editing mode and the selected object did not get updated in both ICs, the subjects had to keep track of their individual status, and repeat actions they already took before the transitions. Four subjects even gave up using both ICs, and stayed with one interface throughout the trial. Six subjects complained about the absence of travel synchronization, which required more manual navigation, and separated the perceptions of space in the two VE representations. However, because the interaction states are different between the two ICs in the “No-Sync” mode, four subjects did point out its advantage to enable simultaneous performance on two different tasks and/or in two different spaces.

Table 3.4: The summary of comments on about the “No-Sync” mode (“-” indicates negative comment)

No-Sync	Comment
4	+ Can work on different tasks or in different spaces simultaneously.
7	- Confusing and awkward to learn and use.
4	- I gave up using both ICs and just stayed with one.
6	- Too much manual navigation without travel synchronization.

When asked about preference of ICs in “Sync” mode, 22 subjects preferred to use both ICs, two subjects preferred tablet only, and no subject selected VR only. Different answers were given in the “No-Sync” mode, with nine for both ICs, four for tablet only, and 11 for VR only. In other words, subjects preferred using both ICs with task synchronization, but staying with one IC without it.

Table 3.5: The comparison of IC preferences in “Sync” and “No-Sync” modes

	Both	**Tablet-only**	**VR-only**
Sync	22	2	0
No-Sync	9	4	11

The subjects were also asked to give general comments about the HVE level editor. Eleven subjects appreciated the complementary benefits offered by the heterogeneous views and interfaces. They suggested 2D tasks (*e.g.*, painting and menu control), long distance navigation, and large scale manipulation to be performed on the tablet, and 3D tasks (*e.g.*, object selection and scaling), local space locomotion, and small scale adjustment to be performed using immersive VR. Having redundant functionality on both ICs was acknowledged by two subjects, for it granted them freedom to perform the tasks differently in different situations. Lastly, suggestions to improve the HVE level editor were given in the interviews, such as undo and redo (three subjects), ambient sound and sound effects (two subjects), teleport in VR (three subjects), flying in VR (two subjects), showing a virtual tablet in the HMD (one subject), and combining the wand and tablet into a single interface like the Nintendo WiiU controller (one subject).

3.7.6 Video Analysis

To understand how the subjects used the two ICs, video footage of the experiment trials was captured from three sources. A web camera was mounted on the ceiling to capture the subject from the top, and screen capture software was installed on the desktop computer and the tablet to capture from both screens. The three streams of video footage for each trial were then merged, timelinesynchronized, and analyzed. The videos showed that subjects were able to connect the two views in the shared 3D space, and take advantage of both ICs for different tasks. For example, after painting the mountain with the wand, many subjects immediately switched to the tablet, located the river near the mountain, and continued to clean the foliage in it. With task synchronization, the subjects did not need much time to plan such sequences of transitional actions, and were able to execute them smoothly. On the other hand, although all subjects eventually adapted to the absence of task synchronization, many of them expressed confusion and awkwardness at repeating actions that had already been done, and some even made a few mistakes when they lost track of the ICs' individual statuses. The videos also showed that subjects made fewer transitions without task synchronization. They grouped all appropriate tasks for one IC, and completed them before changing to the other IC.

There was also no within-task transition for the cube collecting task in “No Sync” mode. Many subjects chose to stay with the wand, and traveled long distances to carry the cubes to their destinations. This is probably because they had to reselect the same cube on the tablet, which was why the wand was used in the first place. In contrast, several subjects were able to discover some efficient strategies to leverage both ICs with task synchronization enabled. For example, three subjects completed the cube collecting task quickly by using the tablet to teleport the Hero avatar near a small cube, selecting it with the wand, teleporting with the tablet again near the destination, and dropping the cube. Another interesting approach was taken by two subjects, who positioned the Hero avatar near the destination, and used the wand to drop cubes that had been selected using the tablet from a zoomed-in view.

It is also evident that subjects only made transitions when there were clear advantages. No one switched ICs just to change the editing mode, as they could do so in both ICs. When a cube was selected, and the destination was available in the current view, many subjects tended to finish the manipulation within the current IC, even though it might be more efficient to do in the other IC. In fact, one subject praised the redundant functionalities of the two ICs in the interview, for it did not force him to waste time on unnecessary transitions.

The “teleport” and “focus” buttons were used a lot in the experiment. Using these two buttons, one subject demonstrated an interesting strategy to speed up multi-scale navigation on the tablet. Instead of panning and zooming in the God camera, the subject teleported his Hero avatar, and tapped the focused button. This allowed him to instantly navigate to an area of interest. However, the “follow” toggle was not used as much, probably because the test bed did not include any “focus + context” task [Baudisch01].

Lastly, the video analysis offered insight about how the interfaces were used for the five test bed tasks. In general, the tablet was mainly used for 2D tasks that needed to be done from different angles, and at large scales, such as painting textures on the terrain, clearing foliage in the rivers, and moving cubes across the VE. In contrast, the wand and HMD were used to edit details of objects in 3D spaces, such as selecting cubes, smoothing terrain surfaces, scaling houses, and planting flowers under trees. These interaction patterns agreed with the subjects’ comments in the interview, and clearly indicated the complementary benefits of the two ICs for 3D interaction tasks with diverse requirements.

3.7.7 Discussion

The user study results provided strong evidence that is consistent with all three hypotheses. Similar interaction patterns were discovered in the interview feedback and the video analysis, proving that the subjects were able to connect the Hero and God views in the shared virtual space, and learn to use both ICs effectively to perform tasks with diverse and complementary requirements (***H2***). However, the transitions between ICs were much more continuous with task synchronization enabled, as suggested by comparative ratings, user comments in the interview, and video analysis of the experiment trials (***H1***). First of all, the synchronization of system control parameters, such as the editing mode, saved not only the mental overhead of tracking the ICs' individual status, but also the physical effort to repeat actions previously done in other ICs. In addition, the synchronization of object selection enabled and inspired various within-task transition strategies to perform the cube-collecting task efficiently (***H3***). Furthermore, the travel synchronization techniques, especially the "teleport" and "focus" buttons, were used effectively to aid the connection between the two VE presentations at the perceptual and cognitive levels.

In comparison, the HVE system without task synchronization was perceived to be confusing, awkward, and inefficient to learn and use in a hybrid way. In essence, the absence of task synchronization broke the hybrid system into two separate tools. Although it was still beneficial to use both ICs for complementary task requirements, subjects tended to avoid transitions as much as possible. The video analysis showed them doing so by dividing the tasks into two groups, and finishing all tasks in one IC before transitioning to a different one. And when some subjects attempted to add more transitional interactions to their workflows, mistakes were made, because they forgot to constantly invest more working memory to keep track of the status of both systems. The synchronization of travel and object selection also enabled and inspired various within-task transition strategies to perform the cube-collecting task efficiently (***H3***). In comparison, these strategies were abandoned when task synchronizations were absent, because subjects had to reselect the cubes in the second IC, which was the reason why it was not used in the first place.

Due to all these disadvantages, many subjects had worse task performance in the “No-Sync” condition, preferring to stay in one IC, the inefficient way, to complete the tasks.

Another interesting finding is that in both the “Sync” and “No-Sync” modes, the subjects only made transitions when there were clear benefits in making the task performance more efficient. In other words, subjects would not spend the time and effort to switch contexts, just to change the editing mode, or to select an object that could already be picked up using the interface at hand. This goes against some previous research findings, such as the rule of dimensional congruence [Darken05], or the methodology to assign 3DUI tasks exclusively to ICs [Carvolho12]. However, this may also change given an HVE system that has fewer different components between the ICs, which would demand less time and effort to switch interfaces.

3.8 HVE Design Process

Based on the design and development of the HVE level editor, and the research findings from the user study, a four-step design process for HVE systems, as illustrated in Figure 3.12, is proposed in this section.

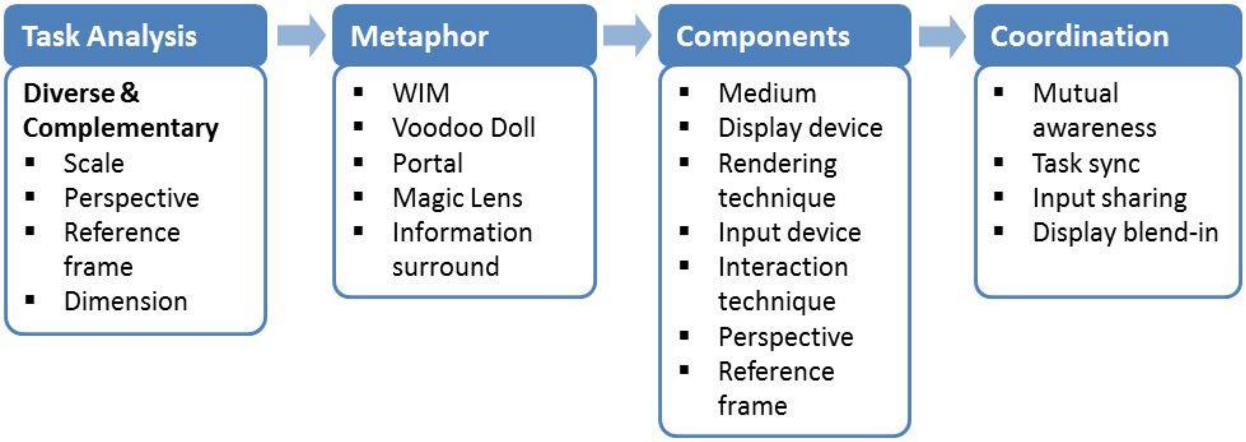


Figure 3.12: Four-step design process of an HVE system

In comparison to traditional, single IC VR systems, HVE systems can be much more complex for the developer to build, and for the user to learn and use. Therefore, a designer should first conduct a *task analysis* to determine the necessity of involving such complexity. Following similar rules suggested for the design of coordinated multiple view systems [Wang Baldonado00], one can decompose the 3DUI tasks in the target application, and look at their degree of diversity and complementarity, regarding the scales, perspectives, reference frames, and dimensionality. High diversity could indicate the need for multiple ICs, while high complementarity promises the benefits of implementing an HVE system. As an example, when analyzing the tasks in 3D level editing, I found that a level designer would constantly change his perspective between an above-the-world God view and an in-the-world Hero view, in order to validate his design from different perspectives, as well as edit objects at different scales. This observation in task diversity and complementarity served as a motivation to design an HVE system that could provide the designers with two sets of tools, allowing them to do both tasks simultaneously and efficiently.

With the motivation justified, the next step is to choose appropriate *metaphors* to conceptually unify the multiple ICs. Depending on the application, the HVE designer can choose from WIM [Stoakley95], Voodoo Doll [Pierce99], Portal [Schmalstieg99] [Kiyokawa05], Magic Lens [Viega96] [Brown06], Information Surround [Feiner91], or even invent his/her own metaphor (as will be seen in Chapter 4). An appropriate metaphor, such as the WIM metaphor used in the HVE level editor, can enable the user to immediately understand the purpose of having multiple interface components, and the functional relationship between the multiple ICs. Contrarily, a badly designed metaphor can confuse users, giving them an impression that the ICs are unrelated parts with no benefits of being put together.

From the designer's point of view, the metaphor also helps to visualize a framework of the final HVE system, which can be effectively referred to in the specification of the *IC components*. As elaborated in Section 3.3, each IC needs to be specified by its medium, display device, rendering technique, input device, interaction technique, perspective, and reference frame. Provided the requirements of the application are met, the number of heterogeneous components between different ICs should be minimized to reduce the cognitive overhead of IC transitions.

The final step of the process is to implement the HVE system, by simulating the VE on each IC, and programming the communications between ICs to keep the multiple VE representations synchronized in real time. As an example, the HVE level editor presented in this chapter adopted a peer-to-peer approach over a WiFi network. During the implementation, the designer is suggested to program *coordination mechanisms*, such as mutual awareness, task synchronization, input sharing, and display blend-in, to effectively reduce the transition gap between the heterogeneous ICs to achieve a seamless interaction experience.

3.9 Summary

To summarize, this chapter presented a novel HVE system to overcome the limitations of traditional immersive VR systems, in task scenarios that involve diverse scales, angles, perspectives, reference frames, or dimensions. The system leveraged the power and rich interactivity of a tablet device to complement the natural yet limiting 3D interfaces in a traditional HMD and wand-based immersive VR setup. The definition of IC was given, and a taxonomy of IC components was presented to describe the immersive and tablet ICs based on their mediums, display devices, display techniques, input devices, input techniques, viewing perspectives, and reference frames. Based on research findings in related fields, four coordination mechanisms were proposed to increase the transition continuity between the ICs. Two of these, namely, mutual awareness and task synchronization, were implemented in the current version of the HVE system. Lastly, a user study was conducted based on five level-editing tasks, to validate the benefits of multiple ICs, and compare the transition experience with and without task synchronization enabled. The study results confirmed that complex HVE systems can be learnt and used to perform diverse 3D tasks efficiently, and suggested that task synchronization is necessary to keep continuous and effortless transitions across ICs. Based on these research findings, a four-step design process was proposed to aid the design and development of HVE systems, as well as the coordination of multiple ICs for seamless transitions.

In the future, the HVE system can be continuously improved from many aspects. For example, the formal user study had the user seated in a swivel chair with an armrest to reduce fatigue. Alternatively, the system could replace the tablet with a lighter phone device, and allow the user to stand up and really walk around the tracked space. As shown in previous research, real walking can significantly improve the level of immersion and engagement, and has the potential to lead to a better sense of presence in the virtual environment [Usoh99]. Another possible change to the system is to combine the tablet and the wand into a single interface device, similar to the approach adopted by the Nintendo WiiU controller, the Go‘Then’Tag point cloud annotation interface [Veit14], or the tracked touch surface from Disney Imagineering [Mine14]. These approaches can better unify the input devices of the two ICs, and reduce the physical effort and cognitive overhead of IC transitions. Although, as discussed in previous research, reaching a proper ergonomic design for such a hybrid “pointing + touching” input device is difficult, and demands much effort in designing and testing iterations [Mine14].

Following the theme laid out by the previous chapters, the guidelines of *diversity*, *complementarity*, and *coordination* for building effective hybrid systems were extensively discussed again in this chapter. The four-step design process presented in Section 3.8 provided concrete examples of how to apply these guidelines in the actual design workflow. It also served as a good summary of many contributions in this research work, from the beginning task analysis to the final evaluation of coordination mechanisms.

While the proposed design process could help designers create new HVE systems, the underlying research methodology could also serve as inspiration to researchers working in virtual reality, augmented reality, and 3D user interfaces. For example, the systematic research approach suggests any interface innovation to start with a careful validation through task analysis. Only after problems or new requirements have been identified should researchers proceed to design, develop, and evaluate their proposed solutions. Furthermore, decomposing an interface system into components not only helps effectively characterize it in terms of prior work, but also enables researchers to explore the design space more thoroughly before finalizing the design. Lastly, as the interaction paradigms in immersive environments tend to follow those used in the real world rather closely, the naturalness of the interaction metaphor can be a critical factor in the success of the interface design. However, a good interaction technique does not have to strictly follow reality, but could instead come completely from imagination, or, more likely, relaxing or bending the rules of the real world [Pierce07]. As an example, a new HVE metaphor, object impersonation, will be introduced in the next chapter.

Chapter 4: Object Impersonation

HVE systems provide the immersed user with multiple interactive representations of the virtual world, and can be effectively used for 3D interaction tasks with highly diverse requirements. In the last chapter, a HVE level editor was introduced that combined the traditional wand- and HMD-based immersive system with a multi-touch tablet. Based on the WIM metaphor, it successfully joined the realistic experience from the egocentric Hero point of view with the efficient and effortless interaction capability from the exocentric God perspective. In addition to WIM, various metaphors can also be used to form HVE systems, such as Voodoo Doll [Pierce99], Portal [Schmalstieg99b] [Kiyokawa05], See-through Lens [Viega96] [Brown06], and Information Surround [Feiner91]. A common characteristic of these metaphors is that they were all invented by breaking real world assumptions [Pierce07], such as "one object cannot exist in two places" (Voodoo Doll), "no object can travel across disconnected spaces" (Portals), and so on. Therefore, a question is naturally raised about whether one can come up with new and useful HVE metaphors by simply identifying and breaking assumptions in current immersive systems and interaction paradigms. The answer is "yes," and this chapter demonstrates an exemplary idea I experimented with using the system infrastructure of the HVE level editor.

I propose Object Impersonation, an immersive interaction technique that allows the user to not only manipulate a virtual object from outside, but also *become* the object, and maneuver from inside. For example, by impersonating a virtual spotlight, one can efficiently change its location by travelling around the space, and precisely illuminate a target area by turning and looking at it. Similarly, being inside the head of a train, one can pave railroads on a terrain surface, or drill a tunnel through a mountain, simply by traveling through the path from the first-person view. In other words, object impersonation has the potential to turn complex object manipulation tasks into intuitive travel tasks. This approach blurs the line between travel from the object's view and manipulation of the object, leading to efficient *cross-task interaction* in various task scenarios. To the best of my knowledge, cross-task interaction was only briefly suggested as an alternative approach to design 3DUIs [Bowman99a] [Bowman99b], but has not been formally implemented or studied, particularly in the context of HVEs.

Figure 4.1: The system infrastructure of the HVE level editor was used to realize and study object impersonation in object alignment tasks.

After explaining the general methodology and presenting six use cases, the rest of this chapter will focus on two different implementations of object impersonation in the tablet- and HMD-based HVE system (see Figure 4.1), as well as a user study that comparatively evaluated their efficiency and user experience in three object-target alignment tasks. The study results indicate improved task performance and enhanced user experience with the added orientation control from the object's point of view, in comparison to a traditional, non-hybrid interface. However, they also revealed higher perceptual and cognitive overhead to attend to both ICs, especially without sufficient reference cues in the virtual environment.

4.1 Methodology

Object impersonation is formally defined here as an interaction technique that enables an immersed user to select an object in the virtual world as his/her virtual self, and view, move, and interact with other objects from its point of view. Generally speaking, impersonating a different virtual object can cause various changes in view point location, orientation, field of view, body scale, reference frame, and mappings between the user's body motion and his/her avatar actions. As a first exploration of this paradigm, the discussion in this chapter is limited solely to view point position and orientation changes, as well as reference frame changes of the virtual object.

Object impersonation can be implemented as a transitional user interface by allowing the user to jump in and out of the first-person view of his/her virtual avatar [Billinghurst01], or used as a metaphor to define the relationship between ICs in an HVE system [Wang14]. The discussion in this chapter is focused on the latter scenario, where a user is given two ICs, one using the traditional, avatar-based approach, while the other is based on the perspective and reference frame of the selected object. Being the object, the user is still able to perform the same 3DUI tasks (*i.e.*, travel, way-finding, selection, manipulation, etc.), thereby supporting effectiveness in the following application scenarios:

- **Remote space inspection (way-finding):** Using object impersonation, an enhanced version of Worldlets [Elvins01] can be implemented, which allows a user to navigate and inspect remote spaces without having to travel there first. By jumping between objects at different geo-locations, the views of each object's surrounding environment can be connected to form survey knowledge [Chen05] of a large VE relatively quickly.

- **Avatar transportation (travel):** From the object's point of view, the user can also drag and drop his/her virtual avatar to locations in nearby space. This enables quick and accurate transportation, and can be helpful for collaboration, or tasks with distributed goals (*e.g.*, annotation of landmarks). However, certain awareness cues may be necessary to highlight the spatial relationship between the multiple views [Wang14], as seeing one's previous avatar in his/her current view may cause disorientation.

- **Occlusion-free object selection (selection):** Selecting objects in cluttered virtual space can be difficult due to the large amount of occlusion in the scene. Applying object impersonation can alleviate this challenge in two different ways. First, the user can select and impersonate an object to the side of the occluded space, offering an orthographic view to complement the limited selection angle from the current perspective [Pinho08]. Second, the user can even become the occluding object itself, and use its perspective as a see-through lens [Miguel07] to select the objects behind it. Using these approaches, the amount of travel needed to gain different viewing angles of the VE can be effectively reduced to one click of a button.

- **Multi-perspective object manipulation (manipulation):** Like the previous use case, object impersonation can also be utilized to enable object manipulation from two orthographic perspectives. Similar approaches have been shown to be effective in collaborative virtual environments for a variety of cooperative object-manipulation tasks [Pinho08]. In addition, objects at high elevation can be impersonated to gain a God view of the VE, offering the user a WIM-like interface [Stoakley95] to ease large-scale manipulation tasks.

- **Object-target alignment (manipulation):** The previous use cases all focused on what the user can do to *other objects* from the impersonated object's perspective. The user can also affect the impersonated object *itself*, by simply looking, turning, and moving around within its frame of reference. This approach can be used to simplify object manipulation tasks where the goals of the tasks are related to the object's view. For example, the user can impersonate a spotlight, and simply look at the target to accurately illuminate its surrounding area. This crosses the 3DUI tasks of travel and manipulation, implying an interesting "What-I-See-Is-What-I-Do" (WISIWID) metaphor.
- **Path editing (manipulation):** In addition to looking around, the user can also travel around the VE using the object as his/her virtual self. Opposite to the path-drawing technique used for navigation [Igarashi98], a "Where-I-Go-Is-What-I-Do" (WIGIWID) metaphor can be implemented, letting the user impersonate a brush to draw a 3D spline, or the front of a train to lay out a roller coaster in the VE. Compared to a traditional interface such as a 3D stylus, this object-egocentric approach can make it easier to draw a spline across multiple anchor points, especially through cluttered or enclosed spaces.

It should be mentioned that despite the advantages listed above, object impersonation also has its limitations, so one should not rely on it completely for all 3DUI tasks. For example, directing the orientation of a spotlight may be easier from its own point of view, but setting its position can be difficult without seeing it from a third-person view. Similarly, a third-person view is necessary to keep track of the overall structure of a spline, even though passing through the anchor points can be easily done by impersonating the brush itself. Fortunately, the advantages of object impersonation and traditional avatar-based approaches appear to complement each other's drawbacks in many aspects. Therefore, I propose a hybrid solution based on an HVE system, and expect it to combine the strengths of both techniques, to offer effective cross-task 3D interaction in immersive VR. The research presented in this chapter specifically studies this methodology using the object-target alignment task as a test bed.

4.2 Test Bed Tasks

As discussed previously, the object-target alignment task was selected as the test bed to evaluate object impersonation for cross-task 3D interaction in HVEs. To gain an in-depth understanding of all task scenarios, three different object-target alignment tasks were implemented. As shown in Figure 4.2, the spotlight task asked the user to translate and rotate a spotlight, in order to have it placed in the position of a street lamp, and oriented to illuminate a text plate. Taking advantage of object impersonation, one efficient hybrid strategy is to first drag the spotlight to its destination using an avatar-based third-person view interface, and then to impersonate the spotlight, and illuminate the text plate by simply looking at it.

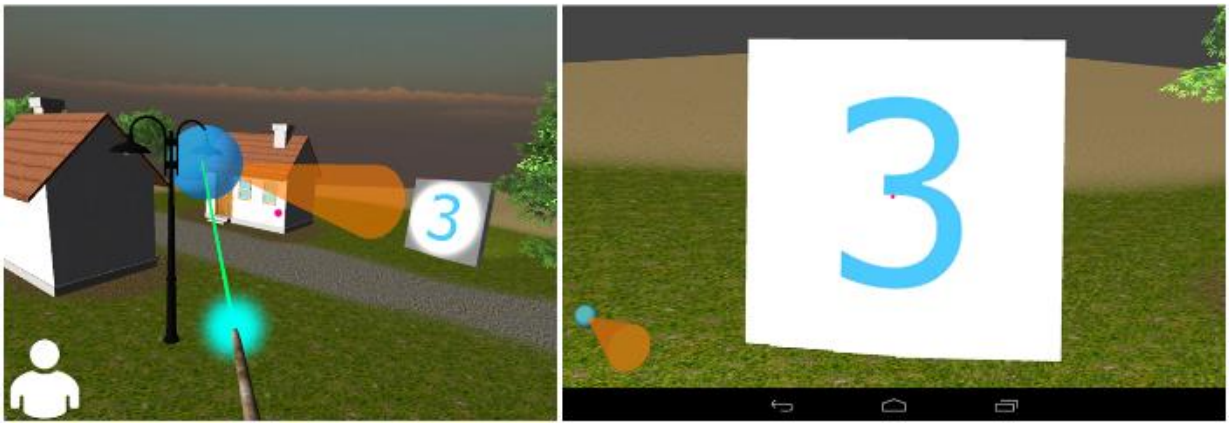


Figure 4.2: The spotlight task can be effectively done by dragging the light bulb to the right location, and then impersonating it to look at (orient to) the target number plate.

The spotlight task presents a special case of object manipulation in VR. More generally, the impersonated object may neither feature a shape similar to the viewing frustum, nor afford a visual indicator (the light) to naturally connect the goal of the task to the style of the first-person view. Therefore, a second task is illustrated in Figure 4.3, which asks the user to translate and rotate a house in 6-DOF, in order to have it stand on the ground, and face another house door to door. Without the visual cues, the advantages of object impersonation in this task may not be as significant as in the spotlight task. However, the user may still find it helpful to level the house on the ground, or determine its alignment with the other house.

Figure 4.3: The house alignment task is a more general use case, in which the impersonation view can be used to aid the leveling and alignment of the house.

To facilitate controlled comparison with traditional 3D interfaces, a further generalized object-docking task was developed, following the classic object manipulation task proposed by Zhai [Zhai95]. As shown in Figure 4.4, this task requires the user to manipulate a tetrahedron in 6-DOF, and match it with another tetrahedron with arbitrary position and orientation. To avoid ambiguity of the orientation matching, a uniquely colored sphere is attached to each vertex of the tetrahedron. Using object impersonation, the user can become the tetrahedron, and change its position and orientation by moving and looking around the VE, respectively. To make the task goal visible in the object's view, a crosshair was added to both tetrahedra, which can be matched to align their orientations. This approach reduces the overhead of mentally rotating the tetrahedron by separating the interrelated 3-DOF object rotation control to the combination of a 2-DOF looking action (*i.e.*, crosshair translation) and a 1-DOF rolling action (*i.e.*, crosshair rotation), and is expected to enhance user performance and experience in comparison to traditional, non-hybrid spatial input interfaces.

Figure 4.4: The tetrahedron alignment task is a classic 6-DOF object manipulation task. Using object impersonation, the 3-DOF rotation control can be decomposed to simpler looking and rolling actions.

4.3 Interface Design

Based on the hardware and software infrastructure of the HVE level editor, two different object impersonation modes, namely, VIEW and DRIVE modes, were realized and studied. The main differences between these two modes, from a user's standpoint, are the depth of immersion in, and the degree of control over, the impersonated object. As shown previously in Figure 4.4, the Object View Impersonation (VIEW) mode displays the view of the impersonated object on the screen of the tablet, leaving the HMD to the traditional avatar-based immersive model. Object translation in the immersive IC is realized using a combination of virtual hand and ray-casting techniques [Bowman99a]. Ray-casting-based translation is triggered when the user points the wand at the tetrahedron and presses down the "B" button on the Wii Remote controller. The tetrahedron will follow the movement of the wand at the original hit point, while two fishing rod buttons can be used to move it further or closer along the direction of the ray [Bowman97]. Virtual-hand-based translation starts when the user points at the tetrahedron and presses down the "Home" button. The tetrahedron will then follow the position change of the user's hand, allowing more accurate position control over a much smaller range. This hybrid control approach combines two modes of position-controlled wand movement and rate-controlled button pressing, allowing the user to match the targets both quickly and precisely [Wang13b]. In the tablet IC, the user can see the tetrahedron's first-person view, and look around using a single-finger swipe gesture, or roll the view using a two-finger rotation gesture. Since the tetrahedron's viewing frustum is fixed to its body, changing its first-person view will also affect the object's orientation. Therefore, by moving the tetrahedron onto the target, and matching the two crosshairs, the orientation of the objects can be roughly matched. The result can then be perfected by micro-adjusting the tetrahedron's position using the wand interface, until a "Right There!" text is shown on the screen to indicate the completion of the task.

Figure 4.5 illustrates the Object Drive Impersonation (DRIVE) mode. In this mode, the tablet screen is used to display a third-person view looking towards one vertex of the tetrahedron from behind, allowing the user to use the HMD to gain a first-person view immersion in the tetrahedron. The experience in the immersive IC is similar to driving a spacecraft from the inside, with the tetrahedron being the spacecraft, and following pointing-directed locomotion of the user. Rotations around the up- and right-axes (*i.e.*, yaw and pitch) are realized by turning the head. To avoid straining the neck, rolling can also be done by pressing down two buttons on the wand, with one being clockwise and the other counter-clockwise. It should also be mentioned that pressing these two buttons will only rotate the tetrahedron; the immersive view is always kept upright, in order to prevent disorientation and motion sickness induced by looking at the VE upside-down [Vidal04].

(a) (b)

Figure 4.5: The tetrahedron docking task in the object drive impersonation (DRIVE) mode, from (a) the avatar's view on the tablet and (b) the tetrahedron's view on the HMD.

To match the tetrahedron with the target in DRIVE mode, a three-step procedure is suggested. The first step is to drive the tetrahedron to the center of the target object, which sets up a base point to align the crosshairs. The user can then hold down the "B" button, and turn his/her head to find and match the reference crosshair, which will match the orientation of the tetrahedrons as well. Finally, the user switches to the tablet, and uses one-finger swipe and two-finger pinch gestures to precisely match the positions of the tetrahedrons. It should be mentioned that the last two steps may need to be repeated, depending on the precision of the initial position match in the first step.

To evaluate object impersonation, a traditional, non-hybrid 6-DOF manipulation interface was implemented as a control condition, using the wand device only (WAND). The interaction technique adopted is similar to HOMER [Bowman99a]. Translation control is the same as in VIEW mode, with the "B" button dedicated to enabling ray-casting-based object dragging, and the

"Home" button used to trigger virtual hand-based accurate position adjustment. Instead of turning the object's first-person view, object rotation in WAND mode is done by holding down the "A" button, and directly rotating the wand device, as shown in Figure 4.6. Clutching is supported by releasing and repressing the "A" button. Furthermore, since there is only one object to control in the tetrahedron task, the user can start rotating it as soon as the button is pressed down, without having to point the wand at the object first. These two settings compensate for the physical constraint of the wrist, giving the user more freedom and flexibility to operate the wand interface effectively [Hinckley97b].

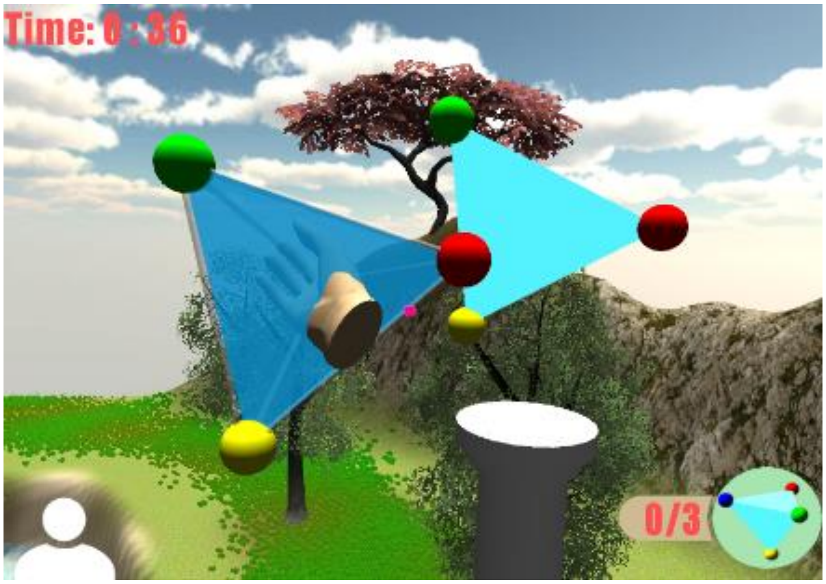


Figure 4.6: The WAND mode allows tetrahedron manipulation from a single immersive view.

4.4 User Study

4.4.1 Hypotheses

Object impersonation offers the user a cross-task approach to perform 3D manipulation tasks from the point of view of the target object itself. As proposed in Section 4.1, this metaphor can benefit many task scenarios where traditional 3DUIs fall short, such as the object-target alignment task selected as the study test bed. However, it is also believed that despite its advantages, object impersonation also has its limitations, and should not be used to *replace* traditional, avatar-based 3DUI techniques, but rather to *supplement* them. The HVE system thus offers a hybrid solution to combining the benefits of both approaches, allowing the user to select and drag the object from the outside, as well as to change its orientation from the inside. I feel that the WAND interface does offer a more realistic simulation of object rotation, and integrates all interaction in one single IC. Based on these analyses, the following hypotheses were made:

- ***H1*****:** Users will spend less time completing the tetrahedron docking task in the VIEW and DRIVE modes compared to the traditional WAND interface.
- ***H2*****:** Users will feel the WAND interface to be more intuitive and natural to understand and learn than the VIEW or DRIVE modes.
- ***H3*****:** Users will find the VIEW and DRIVE modes to be more efficient and precise, and easier and less tiring to use compared to the traditional WAND interface.
- ***H4*****:** The mental rotation skill required to manipulate the object in 6-DOF will be lower in the VIEW and DRIVE modes compared to the traditional WAND interface.
- ***H5*****:** Higher cognitive overhead will be required by users when multiple ICs are involved in the VIEW and DRIVE modes compared to the traditional WAND interface.

4.4.2 Procedure

To validate these hypotheses, a within-subjects user study was designed and conducted. The study was approved by the institutional review board (IRB), and 26 WPI students were recruited with no remuneration. Each session began with the subject reading and signing a consent form, followed by a demographic questionnaire that asked about gender, age, and handedness, as well as experiences with video games, 3D modeling software (*e.g.*, Maya, SketchUp), immersive VR, multi-touch devices, and multi-screen devices (*e.g.*, the Nintendo WiiU). The subject was then asked to complete Peters' redrawing of Vandenber & Kuse Mental Rotation Test (MRT), which presented 24 questions with a time limit of 10 minutes [Peters95]. After the MRT, the experimenter gave the subject a brief introduction to the hardware used in the study, including the HMD, the wand, and the tablet. The experimenter also explained the details of the three object-target alignment tasks, especially the tetrahedron docking task, which served as the primary task to compare the efficiency of the WAND, VIEW, and DRIVE interfaces.

After the introduction, the subject put on the equipment, and completed the tetrahedron docking task using each of the three interfaces, following a counterbalanced order based on a Latin square. Each of the three conditions included a training session and an experiment session, in which the same VE was used. As shown in Figure 4.7, the VE included three tetrahedral targets in different positions and orientations. The subject was asked to practice the specific interface in each training session, by matching the three targets one after another. In the experiment sessions, the subject was asked to match up to three rounds (nine trials) of the same targets, within a time limit of 10 minutes.

At the beginning of a session, the subject's avatar was spawned in the center of the VE shown in Figure 4.7, together with a semi-transparent tetrahedron object floating right in front of him/her. The subject could then use this tetrahedron to match the targets one by one, as quickly as possible. The distances between each pair of the colored spheres were calculated, and were compared to a threshold variable *d* to determine whether the tetrahedrons had been matched. When the threshold was reached, a "Right There!" text would show up on both screens to indicate a match. The subject could then let go of the control, and wait for the current target to disappear, and the next target to appear, in three seconds.

Figure 4.7: The task VE of the tetrahedron docking task.

This process was repeated three times in training (one round, with *d* = 0.8m and tetrahedron's edge length = 5m), and up to nine times (three rounds, with *d* = 0.8m, 0.4m, and 0.2m), or 10 minutes in the experiment sessions. The experiment sessions had increased precision requirements with each round, so that the effects of the interfaces on task precision could be inspected. During the experiment, a timer was displayed in the top-left corner, and a target counter was shown in the bottom right, on both screens. The crosshair plates accompanying each target were only made visible in the VIEW and DRIVE conditions. They indicated the target's first-person views, and were used to aid rotation alignment from the impersonated object's perspective. After completing all three conditions, the subject was asked to fill in a questionnaire to compare the WAND, VIEW, and DRIVE interfaces, and to rate them on a one to six scale regarding six different questions (see Figure 4.9, discussed later). The subject was also asked to indicate his/her general preference for the three interfaces, and provide comments on what they liked and disliked about each of them. Lastly, to expand the investigation to real world applications, the house and spotlight tasks were also included in the study. However, instead of being formally evaluated, they were only tested in a short session after the tetrahedron experiment. The subjects casually selected the houses and spotlights in a VE, and tried each aforementioned interface to align them with their targets. During the process, the experimenter kept an active conversion with the subject, so that he/she could give comments on the go about the advantages and drawbacks of each interface for the two tasks.

Of the 26 participants, 14 were male and 12 were female. All subjects were right handed. Their ages ranged from 19 to 31 years (mean = 23.9, SD = 3.1). With 1 being “Never” and 6 being “Every day,” their experiences with video games ranged from 1 to 6 (mean = 3.2, SD = 1.5), 3D modeling software ranged from 1 to 4 (mean = 1.9, SD = 1.0), VR ranged from 1 to 3 (mean = 1.4, SD = 0.6), multi-touch devices from 1 to 6 (mean = 5.7, SD = 1.0), multi-screen devices from 1 to 6 (mean = 2.8, SD = 1.7). Their responses to the MRT were also graded. With 24 being the maximum score, their answers ranged from 7 to 24 (mean = 14.9, SD = 4.9).

4.4.3 Results

For each experiment session, the system recorded how many targets were successfully completed in 10 minutes, as well as the exact time stamp when each target was matched. The numbers of completed targets of the three interface conditions were compared using a Friedman test, however the results were not significant. Since many subjects were able to match all nine targets before the time expired, a more accurate indicator of task efficiency was needed. To do this, for each subject, the time he/she spent to match the targets was averaged, for all targets collected by the subject, as well as targets in the first round, or the second round alone (all subjects were able to complete all three targets in the first round, and at least one target in the second round). The seconds-per-target data produced from this process was analyzed using a one-way ANOVA, and the results are shown in Figure 4.8 below.

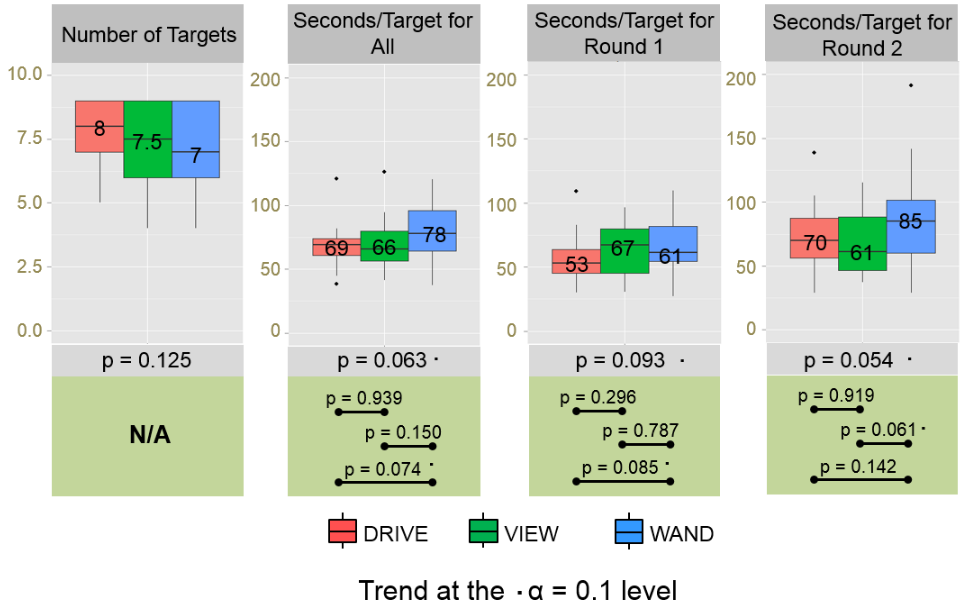


Figure 4.8: The analysis of the task performance indicators.

Although no results are strictly significant (*i.e.*, p > 0.05), statistical trends towards significance were evident in all of them (*i.e.*, p < 0.1), suggesting further post-hoc investigation. Using the Tukey HSD test, trends were identified that suggested better efficiency in DRIVE mode than the WAND interface, for all targets in general (p = 0.074), and low-precision-requirement targets in the first round (p = 0.085). Additionally, a trend was also identified indicating better efficiency in VIEW mode than using the WAND interface, for the second-round targets that required medium-precision matching (p = 0.061). Finally, a Pearson correlation analysis was performed between these task performance measurements and the subject's prior experiences and mental rotation skills. However, no strong correlation was discovered (all correlation coefficient values are below 0.7).

The six-point rating scores of the three conditions were analyzed using a Friedman test. As indicated in Figure 4.9, the differences among the three conditions were significant regarding efficiency (p = 0.032) and precision (p = 0.040), and just short of significance for ease-of-learning (p = 0.054) and fatigue (p = 0.057, lower score is better).

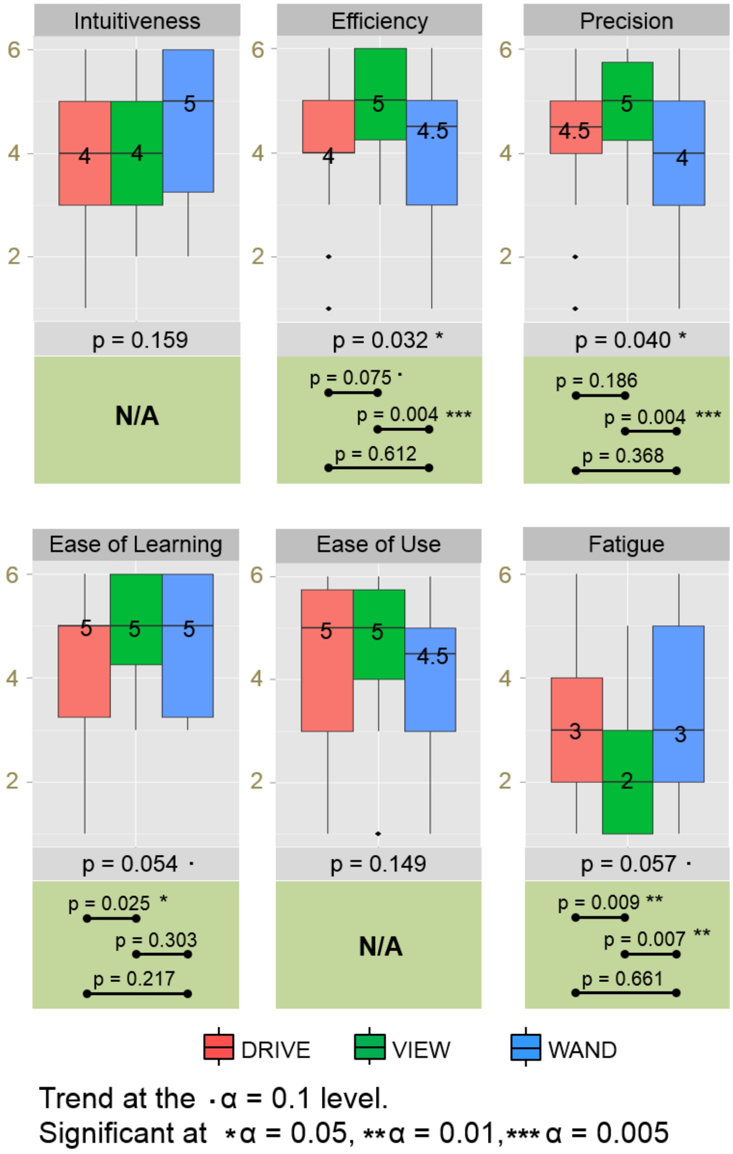


Figure 4.9: The analysis of the subjective rating scores.

Post-hoc analyses were performed using pairwise Wilcoxon signed-rank tests. The results suggest that the subjects considered VIEW mode to be more efficient, and less tiring to use than DRIVE mode (p = 0.075 and 0.009, respectively) and the WAND interface (p = 0.004 and 0.007, respectively). Additionally, VIEW mode was also considered to be more precise than the WAND interface (p = 0.004), and easier to learn than DRIVE mode (p = 0.025). Pearson correlation analyses between the rating scores and the subjects' prior experiences and MRT scores were also performed, however, no strong correlation coefficient was identified.

For VIEW mode, 14 of 26 subjects complimented it for making the target matching process easier and faster. Specifically, six subjects found the combination of the avatar's view and the object's view helpful, as third-person control from the avatar's view allowed them to translate the object efficiently, while first-person control from the object's view allowed them to match the rotation intuitively and precisely. Four subjects preferred this mode because matching the 2D crosshairs was easier than figuring out the mental rotations to match the targets in 3D space. Five subjects liked to use the tablet device, because it was more stable and precise to touch on a 2D plane than holding and manipulating a wand.

On the other hand, seven subjects disliked having another display, as it made the task more complicated, and took away the immersion and spatial orientation established in the HMD view. In addition, nine subjects pointed out that searching for the reference plates (*i.e.*, the ones that accompanied each target tetrahedron) could sometimes become very difficult to do on the tablet, partly due to the first-person view [Chen05]. Based on the experimenter's observations, some subjects attempted to alleviate this challenge during the experiments by looking at the HMD while touching the tablet. Nevertheless, many of them struggled, because the mapping of the swiping gesture was based on the object's view, and felt inverted from the avatar's view. Noticing this problem, one subject asked the experimenter if it was possible to detect his gaze change to the

HMD, and base the touch control on its perspective instead - a solution similar to the interface sharing idea proposed in Section 3.6.

The “think-aloud” feedback session presented the user with a hybrid interface that combined all three aforementioned interface conditions. The user could point at an object and hold down different modal buttons to translate and rotate it using the wand device. The first-person view of the object was displayed on the tablet upon selection, and could then be rotated using multi-touch gestures. Furthermore, pressing the “+” and “-” buttons on the Wii Remote made the user *jump in and out* of the selected object respectively, realizing DRIVE mode through a transitional user interface approach [Billinghurst01]. Due to user fatigue and other logistical reasons, only 15 of the 26 subjects participated in this session. Nonetheless, they all tried different modes for both the spotlight task and the house task, and provided oral feedback to the experimenter on the go.

Summarizing subject comments, a majority group of 12 subjects did not state a clear interface preference for either task. Instead, they liked the fact that they could switch between interfaces, and felt that having all three options was actually better than any of them alone. For example, seven subjects preferred using the wand device for positioning the house, but would rather use the object’s first-person view to align the orientation. For object-impersonation-based rotation control, eight subjects considered the jump-in DRIVE mode to be more effective than VIEW mode, as target house searching was easier by looking around, and the house orientation could quickly follow the view by pressing the “B” button. Specific to the house task, five subjects pointed out that none of the two modes had made the object’s first-person view appropriate for judging the leveling of the house; without any visual cue added to the VE, the user still had to refer to the avatar’s third-person view to place the house on the ground, using either the WAND interface or the tablet in DRIVE mode.

The three interfaces also had different and complementary advantages in the spotlight task. Nine subjects preferred using the wand device for translating the spotlight, and two subjects were willing to use it for orienting it as well, since it only involved 2-DOF rotation, in comparison to the 3-DOF tetrahedron rotation task. In addition, a majority group of 12 subjects preferred to control orientation from the spotlight's first-person view, as it was more direct, intuitive, and efficient. Four subjects even felt it *too easy to do* using the jump-in DRIVE mode, as they could simply look at the target, and press a button to accurately illuminate it.

4.4.4 Discussion

Using the object-target alignment tasks as the test bed, the user study results revealed various advantages and limitations of object impersonation in HVE systems. Although the results were not conclusive, the performance results, such as the average time spent on each round of targets, did show statistical trends that object impersonation could complement a traditional 3D wand interface to make performance of 6-DOF manipulation tasks more efficient (***H1***). Analyses of subjective measurements revealed advantages and shortcomings of each interface condition. According to the subjects' ratings, VIEW mode provided the most efficient, precise, and least-fatiguing interface of the three conditions (***H3***). The subjects' post-study comments suggested two explanations for these preferences. First, by requiring the user to align the crosshairs to match the rotation, the object impersonation techniques transformed complex and hard-to-reason 3D rotation tasks [Zhai93a] into simpler and more-intuitive 2D target-matching tasks. Second, the tablet device offered a physical surface to touch on, leading to an increase in operator effectiveness and precision, and a reduction in user fatigue confirming results of other studies [Angus95] [Lindeman99] [Marzo14], especially in comparison to spatial input devices (*i.e.*, the WAND interface).

Partially refuting ***H3***, neither the VIEW nor DRIVE mode was considered to be easier to use than the traditional WAND interface. A summary of user comments suggests they had difficulty searching for the reference plates from the object's first-person view, especially using the tablet in VIEW mode. By allowing the user to search with head and chair turning, this challenge was alleviated in DRIVE mode. However, DRIVE mode forced the user to completely immerse themselves in the object's body, without providing a maneuverable avatar camera on the tablet to adjust the position of the tetrahedron from all angles. On the other hand, refuting ***H2***, the WAND interface was not rated to be more intuitive or easier to use than the two object impersonation modes, although DRIVE mode was commented as being more difficult to understand and learn. This suggests that object impersonation may be better accepted as an augmentation to, instead of a complete replacement for, existing interaction metaphors.

The hypotheses in ***H4*** and ***H5*** were only evaluated anecdotally. Six subjects complimented VIEW and DRIVE modes for requiring less mental rotation, as the DOFs involved in the rotation alignment process were reduced from three to two (***H4***). Increased cognitive overhead of attending to two ICs was mentioned by seven subjects for VIEW mode, and six subjects for DRIVE mode (***H5***). According to them, dividing the task sequences to different ICs made it more complex to complete, and also broke the immersion established in the HMD. This issue was mainly caused by divided attention during context switching, and could be alleviated by peripheral displays [Chen05], display blend-in, and interaction coordination mechanisms [Wang14].

The "think aloud" feedback collected during the spotlight and house task sessions suggest a need to further combine the three interface conditions to form a more-advanced hybrid interface on top of the current HVE system. In other words, *such a system should not only combine the immersive and tablet ICs (the avatar and object perspectives), but also the different interface approaches, to* *counter each of their disadvantages*. In addition, this session also provided interesting insights about the applicability of object impersonation in real world application tasks. The preference of object impersonation was most evident in the spotlight task. On one hand, the cone shape of the spotlight was similar to the frustum of the first-person view, offering good visual affordance for the object impersonation metaphor. On the other hand, the goal of the task (*i.e.*, having the light illuminate the target) also had a strong similarity to the user's action of looking at a target. In contrast, the effectiveness of object impersonation fell short for leveling the houses on the ground, due to the lack of visual cues from inside the house itself, and users needed to refer to the traditional exocentric interaction paradigm for better efficiency. Similarly, the effectiveness of the object's point of view may degrade significantly without the crosshair plate indicating the proper alignment. These findings suggest that object impersonation should be used in a hybrid context, and in a task-dependent way according to the following guidelines:

- **Rule of personification:** Objects with human-like shapes or behaviors, such as a spotlight, or a train head, are more natural to impersonate. A natural impersonation not only makes the object-centered interfaces easy to learn and effective to use, but also reduces the cognitive overhead during transitions between the avatar- and object-based ICs. In contrast, objects with less human-like features, such as a house, make the attachment of a first-person view ambiguous. As a result, the user can get confused about how his/her interactions may affect manipulation of the object.

- **Rule of actionable goals:** Object impersonation can better enhance task performance when the goals of the task are more *actionable* from the first-person view, through looking, traveling, or moving different body parts. For example, the goal of the spotlight task was to illuminate a target, which closely relates to the action of looking at the target. The performance of aligning the house to the ground was not successful using object impersonation, but may become easier if the system can detect the user’s body motion, and apply the “sit” gesture to the vertical movement of the house.

- **Rule of goal indication:** Object impersonation can be used to reduce the complexity of third-person view tasks. However, this potential is constrained when the goal of the task is not clearly evident in the object’s first-person view. Various indicators can be added to clarify the task goal, such as the crosshair plate adopted in the tetrahedron matching task, or the illuminated target itself in the spotlight task. In addition to visual indicators, auditory or haptic indicators can also be explored in future work.

4.5 Summary

This chapter proposed a new interaction technique that can benefit various 3DUI task scenarios in immersive VR. By impersonating a virtual object, the user can perform 3D interaction from a different perspective, or even manipulate the impersonated object by looking and traveling around the VE. This blurs the line between basic 3DUI tasks, and can be used in HVE systems to complement the limitations of traditional 3D interfaces. As listed in Section 4.1, object impersonation can be used to enhance 3D interaction in many task cases. As a start, the presented user study explored three types of object-target alignment tasks as the test bed to investigate the task performance and user experience with two different object impersonation implementations, within a tablet-and-HMD-based HVE system. The results showed improved task performance and user experience using object impersonation together with traditional 3DUIs, but also suggested issues and limitations that make it less useful by itself. For example, alignment from the object’s view is only useful when the target is already in the view. When the target is outside the view of the impersonated object, it becomes difficult to find it because the user has no idea of which direction to look. On the other hand, this is exactly where avatar-based interaction is good. In other words, just like the tablet and immersive ICs are good complements for level editing, the two distinctly different embodiments (*i.e.*, avatar and object) offer complementary benefits at different stages of the object alignment task, and thus should be combined in a hybrid solution to maximize their efficiency.

There are certainly limitations in this research work. For instance, the impersonation studied here is still limited to view point and reference frame changes, and does not allow the user to use his/her full body motion to *act* as the impersonated virtual object. The divided attention between the tablet and HMD induces cognitive overhead in context transitions. The study results show promising performance and user experience improvements, but due to the compound effect of touch input, multiple views, and reduced task DOF (*i.e.*, the crosshair plate), it is difficult to isolate and precisely appraise the real benefits of object impersonation. Lastly, to advocate cross-task interaction as a mainstream 3DUI design, many more convincing use cases, like the spotlight alignment task, need to be discovered and tested. Nevertheless, it served as a good example to advocate the creative thinking of how we may “relax *reality* in VR”, and once again demonstrated the power of carefully designed hybrid techniques in improving 3DUI in immersive environments.

Chapter 5: Conclusions and Future Work

As the technology for VR, AR, smartphones, tablets, and wearable systems continues to boom, it is inevitable that users will be confronted more and more with strongly hybrid spaces that mix virtual and real content presented in various ICs. Therefore, I believe the contributions in this dissertation are both important and timely. Specifically, this dissertation focused on improving the usability of VR systems, through three related projects in the common theme of hybrid interaction in immersive virtual environments. The contributions include, but are not limited to, the following core innovations:

1. Created two new methods for extending position-controlled multi-touch gestures using a force-sensing touchpad.
2. Designed and evaluated a Hybrid Virtual Environment (HVE) approach to improving user performance on complex tasks in immersive virtual environments.
3. Introduced the notion of Interaction Context (IC) to define the multiple sets of interface elements used in HVE systems.
4. Designed and evaluated mechanisms to coordinate the interaction flow between ICs to enhance the transition experiences in HVE systems.
5. Explored the idea of Object Impersonation as a cross-task interaction metaphor for positioning and orienting objects in VEs.
6. Proposed a methodology for 3DUI designers to use when devising new HVEs.

In the near future, continuous advances in multi-touch technology will enable force sensing in many more 2D and 3D touch input devices. The context and shear force extension techniques can thus be used in many different application scenarios to combine the strengths of position and rate control in a seamless manner. Using Force Extension, the user can conveniently extend the effective range of his/her input actions, without sacrificing the accuracy of control. Its application to multi-touch gestures also suggest stronger use cases that allow efficient control of multiple simultaneous DOFs, such as the 3D camera manipulation example demonstrated in Chapter 2.

Regarding future work, a carefully designed and formally conducted user study is necessary to draw comparative results of the two force extension approaches and is regarded as future work. Furthermore, it has been argued that transitioning from horizontal movement to vertical pressure for rate control may not be intuitive [Casiez07], which poses a challenge to Force Extension. Therefore it is also of my interest to formally compare the context and shear force extension approaches with existing techniques such as the GroovePad [Kulik12] to explore under what circumstances each technique is preferred.

Although the non-occlusive HMD used in this work may be seen as contrary to current HMD trends towards more occlusion, the research findings about the HVE level editor and transition coordination can help inform a broad array of devices, device combinations, and usage scenarios. For example, most of the emerging glasses-based AR devices still lack efficient, natural, and non-fatiguing approaches to interacting with virtual content. Combining these devices with a smartwatch or smartphone might support such design goals, but could also lead to increased user frustration due to poor design. Therefore, the next step of this research is to explore the use of our tablet IC in CAVEs [Cruz-Neira92], occlusive VR headsets such as the Oculus Rift and Sony Morpheus, or AR headsets like the Microsoft HoloLens. It is also of interest to investigate the transition dynamics between three or more ICs in HVE systems, and experiment with further optimized IC transitions through input sharing and display blend-in techniques.

Finally, while many VR systems struggle to discover more 3D interaction opportunities based on traditional metaphors, it becomes necessary to experiment with designs that strive to break common assumptions. The object impersonation technique does this by stressing that immersive interaction does not have to center on the user’s virtual self, but can be dynamically changed to different virtual objects according to the task requirements. It blurs the line between basic 3DUI tasks, enabling new interface design opportunities in both traditional and hybrid virtual environments. Moving forward, a more comprehensive impersonation technique can be implemented to allow the user to use his/her full body motion to *act as* the impersonated virtual object. Many more convincing use cases, like the spotlight alignment task, need to be discovered and tested as well to advocate cross-task interaction as a mainstream 3D UI design. Finally, the divided attention between the tablet and HMD induces considerable cognitive overhead in the context transition process, which should be alleviated by implementing coordination mechanisms such as those proposed in Chapter 3.

BIBLIOGRAPHY

[Angus95] I. Angus and H. A. Sowizral, “Embedding the 2D interaction metaphor in a real 3D virtual environment,” In *Proc. of IS&T/SPIE’s Symposium on Electronic Imaging: Science & Technology*, pp. 282-293, 1995.

[Baudisch01] P. Baudisch, G. Nathaniel, and P. Stewart, “Focus plus context screens: combining display technology with visualization techniques,” In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 31-40, 2001.

[Benko04] H. Benko, E. W. Ishak, S. Feiner, “Collaborative mixed reality visualization of an archaeological excavation,” In *Proc. of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 132-140, 2004.

[Billinghurst01] M. Billinghurst, H. Kato, and I. Poupyrev, “The MagicBook: a transitional AR interface,” *Computers & Graphics*, vol. 25, no. 5, pp. 745-753, 2011.

[Bornik06] A. Bornik, B. Beichel, E. Kruijff, B. Reitinger, and D. Schmalstieg, “A hybrid user interface for manipulation of volumetric medical data,” In *Proc. of IEEE 3D User Interfaces*, pp. 29-36, 2006.

[Bowman97] D. Bowman and L. Hodges, “An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments,” In *Proc. of the ACM Symposium on Interactive 3D Graphics and Games*, pp. 35-ff, 1997.

[Bowman98a] D. A. Bowman, D. Koller, and L. F. Hodges, “A methodology for the evaluation of travel techniques for immersive virtual environments,” *Virtual Reality*, vol. 3, no. 2, pp. 120-131, 1998.

[Bowman98b] D. A. Bowman, J. Wineman, L. F. Hodges, and D. Allison, “Designing animal habitats within an immersive VE,” *IEEE Computer Graphics and Applications*, vol. 18, no. 5, pp. 9-13, 1998.

[Bowman99a] D. Bowman, “Interaction techniques for common tasks in immersive virtual environments,” *PhD Dissertation*, Georgia Institute of Technology, 1999.

[Bowman99b] D. Bowman, L. Hodges, “Interaction techniques for common tasks in immersive virtual environments,” *unpublished*, Georgia Institute of Technology.

[Bowman04] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, “3D user interfaces: theory and practice,” Addison-Wesley, 2004.

[Brown06] L. D. Brown and H. Hua, “Magic lenses for augmented virtual environments,” *IEEE Computer Graphics and Applications*, vol. 26, no. 4, pp. 64-73, 2006.

[Bruder10] G. Bruder, F. Steinicke, D. Valkov, and K. Hinrichs, “Augmented virtual studio for architectural exploration,” In *Proc. of Virtual Reality International Conference*, pp. 43-50, 2010.

[Calvalho12] F. G. Carvalho, D. G. Trevisan, and A. Raposo, “Toward the design of transitional interfaces: an exploratory study on a semi-immersive hybrid user interface,” *Virtual Reality*, vol. 16, no. 4, pp. 271-288, 2012.

[Casiez07] G. Casiez, D. Vogel, Q. Pan, and C. Chaillou, “RubberEdge: reducing clutching by combining position and rate control with elastic feedback,” In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 129-138, 2007.

[Casiez08] G. Casiez, D. Vogel, R. Balakrishnan, and A. Cockburn, “The impact of control-display gain on user performance in pointing tasks,” *Human-Computer Interaction*, vol. 23, no. 3, pp. 215-250, 2008.

[Cechanowicz07] J. Cechanowicz, P. Irani, and S. Subramanian, “Augmenting the mouse with pressure sensitive input,” In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1385-1394, 2007.

[Chen05] J. Chen, M. Narayan, and M. Perez-Quinones, “The use of hand-held devices for search tasks in virtual environments,” In *Proc. of IEEE Virtual Reality Workshop on New Directions in 3DUI*, pp. 15-18, 2005.

[Churchill98] E. F. Churchill and D. Snowdon, “Collaborative virtual environments: an introductory review of issues and systems,” *Virtual Reality*, vol. 3, no. 1, pp. 3-15, 1998.

[Cirio09] G. Cirio, M. Marchal, T. Regia-Corte, and A. Lécuyer, “The magic barrier tape: a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace,” In *Proc. of ACM Virtual Reality Software and Technology*, pp. 155-162, 2009.

[Coffey11] D. Coffey, N. Malbraaten, T. L. Iman Borazjani, F. Sotiropoulos, and D. F. Keefe, “Slice WIM: a multi-surface, multi-touch interface for overview+ detail exploration of volume datasets in virtual reality,” In *Proc. of the ACM Symposium on Interactive 3D Graphics and Games*, pp. 191-198, 2001.

[Cruz-Neira92] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, “The cave: audio visual experience automatic virtual environment,” *ACM Communications*, vol. 35, no. 6, pp. 64-72, 1992.

[Darken05] R. P. Darken and R. Durost, “Mixed-dimension interaction in virtual environments,” In *Proc. of the ACM Symposium on Virtual Reality Software and Technology*, pp. 38-45, 2005.

[Dominjon05] L. Dominjon, A. Lécuyer, J. Burkhardt, G. Andrade-Barroso, and S. Richir, "The "Bubble" technique: interacting with large virtual environments using haptic devices with limited workspace," In *Proc. of IEEE World Haptics*, pp. 639-640, 2005.

[Dubois02] E. Dubois, L. Nigay, and J. J. Troccaz, "Assessing continuity and compatibility in augmented reality systems." *Universal Access in the Information Society*, vol. 1, no. 4, pp. 263-273, 2002.

[Elvins01] T. Elvins, D. Nadeau, R. Schul, and D. Kirsh, "Worldlets: 3D thumbnails for wayfinding in large virtual worlds," *Presence: Teleoperators and Virtual Environments*, vol. 10, no. 6, pp. 565-582, 2001.

[Feiner91] S. Feiner and A. Shamash, "Hybrid user interfaces: Breeding virtually bigger interfaces for physically smaller computers," In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 9-17, 1991.

[Fitts54] P. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, no. 6, pp. 381-391, 1954.

[Grasset08] R. Grasset, A. Dünser, and M. Billinghurst, "Moving between contexts-a user evaluation of a transitional interface," 2008.

[Guiard87] Y. Guiard, "Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model," *Journal of Motor Behavior*, vol. 19, no. 4, pp. 486-517, 1987.

[Harrison12] C. Harrison and S. Hudson, "Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction," In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3149-3152, 2012..

[Heo11] S. Heo and G. Lee, "Force gestures: augmenting touch screen gestures with normal and tangential forces," In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 621-626, 2011.

[Hinckley94] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell, "A survey of design issues in spatial input," In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 213-222, 1994.

[Hinckley97a] K. Hinckley, R. Pausch, D. Proffitt, J. Patten, and N. Kassell, "Cooperative bimanual action," In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 27-34, 1997.

[Hinckley97b] K. Hinckley, J. Tullio, R. Pausch, D. Proffitt, and N. Kassell, "Usability analysis of 3D rotation techniques," In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 1-10, 1997.

[Holm02] R. Holm, E. Stauder, R. Wagner, M. Priglinger, J. Volkert, “A combined immersive and desktop authoring tool for virtual environments,” In *Proc. of IEEE Virtual Reality*, pp. 93-100, 2002.

[Igarashi98] T. Igarashi, R. Kadobayashi, K. Mase, and H. Tanaka, “Path drawing for 3D walkthrough”, In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 173-174, 1998.

[Jellinek90] H. D. Jellinek and S. K. Card, “Powermice and user performance,” In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 213-220, 1990.

[Kaltenbrunner05] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza, “TUIO - A protocol for table-top tangible user interfaces,” In *Proc. of the International Workshop on Gesture in Human-Computer Interaction and Simulation*, pp. 1-5, 2005.

[Kellogg06] L. H. Kellogg, G. W. Bawden, T. Bernardin, M. I. Billen, E. S. Cowgill, B. Hamann, M. Jadamec, O. Kreylos, O. G. Staadt, and D. Y. Sumner, “Interactive Visualization to Advance Earthquake Simulation,” *Technical Report TR06-04, W.M. Keck Center for Active Visualization in the Earth Sciences (KeckCAVES), University of California Davis*.

[Kennedy10] R. S. Kennedy, J. Drexler, and R. C. Kennedy, “Research in visually induced motion sickness,” *Applied Ergonomics*, vol. 41, no. 4, pp. 494-503, 2010.

[Kin11] K. Kin, T. Miller, B. Bollensdor, T. DeRose, B. Hartmann, and M. Agrawala, “Eden: a professional multitouch tool for constructing virtual organic environments,” In *Proc. of the SIGCHI Conference on Human Factors in Computing* Systems, pp. 1343-1352, 2011.

[Kiyokawa05] K. Kiyokawa and H. Takemura, “A tunnel window and its variations: seamless teleportation techniques in a virtual environment,” *HCI International*, 2005.

[Kulik12] A. Kulik, A. Kunert, A. Huckauf, and B. Froehlich, “The Groovepad: ergonomic integration of isotonic and elastic input for efficient control of complementary subtasks,” In *Proc. of ACM Asia Pacific Conference on Computer Human Interaction*, pp. 75-84, 2012.

[Leigh96] J. Leigh and A. E. Johnson, “CALVIN: an immersimedia design environment utilizing heterogeneous perspectives,” In *Proc. of IEEE Multimedia Computing and Systems*, pp. 20-23, 1996.

[Lee12] B. Lee, H. Lee, S. Lim, H. Lee, S. Han, and J. Park, “Evaluation of human tangential force input performance,” In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3121-3130, 2012.

[Lindeman99] R. W. Lindeman, J. Sibert, and J. Hahn, “Towards usable VR: an empirical study of user interfaces for immersive virtual environments,” In *Proc. of the SIGCHI Conference on Human Factors in Computing* Systems, pp. 64-71, 1999.

[Lindeman01] R. W. Lindeman, J. L. Sibert, and J. N. Templeman, “The effect of 3D widget representation and simulated surface constraints on interaction in virtual environments,” In *Proc. of IEEE Virtual Reality*, pp. 141-148, 2001.

[Lok04] B. Lok, “Toward the merging of real and virtual spaces." *ACM Communications*, vol. 47, no. 8, pp. 48-53, 2004.

[Lopez14] T. Lopez, R. Bouville, E. Loup-Escande, F. Nouviale, V. Gouranton, and B. Arnaldi, “Exchange of avatars: Toward a better perception and understanding,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 644-653, 2014.

[Marzo14] A. Marzo, B. Bossavit, and M. Hachet, “Combining multi-touch input and device movement for 3D manipulations in mobile augmented reality environments”, In *Proc. of the ACM Symposium on Spatial User Interaction*, pp. 13-16, 2014.

[Miguel07] M. M. Miguel, T. Ogawa, K. Kiyokawa, and H. Takemura, “A PDA-based see-through interface within an immersive environment,” In *Proc. of IEEE Artificial Reality and Telexistence*, pp. 113-118, 2007.

[Milgram95] P. Milgram, H. Takemura, A. Utsumi, F. Kishino, “Augmented reality: a class of displays on the reality-virtuality continuum.” In *Proc. of Photonics for Industrial Applications*, pp. 282-292, 1995.

[Mine97] M. R. Mine, F. P. Brooks, and C. H. Sequin, “Moving objects in space: exploiting proprioception in virtual-environment interaction,” In *Proc. of ACM Computer Graphics and Interactive Techniques*, pp. 19-26, 1997.

[Mine14] M. Mine, A. Yoganandan, D. Coffey, “Making VR work: building a real world immersive modeling application in the virtual world,” In *Proc. of the ACM Symposium on Spatial User Interaction*, pp. 80-89, 2014.

[Mithal96] A. K. Mithal and S. A. Douglas, “Differences in movement microstructure of the mouse and the finger-controlled isometric joystick,” In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 300-307, 1996.

[North00] C. North, and B. Shneiderman, “Snap-together visualization: a user interface for coordinating visualizations via relational schemata,” In *Proc. of the ACM Working Conference on Advanced Visual Interfaces*, pp. 128-135, 2000.

[Peters95] M. Peters, B. Laeng, K. Latham, M. Jackson, R. Zaiyouna, and C. Richardson, “A redrawn Vandenberg and Kuse mental rotations test-different versions and factors that affect performance,” *Brain and Cognition*, vol. 28, no. 1, pp. 39-58, 1995.

[Pierce07] J. S. Pierce, R. Pausch, “Generating 3D interaction techniques by identifying and breaking assumptions,” *Virtual Reality*, vol. 11, no. 1, pp. 15-21, 2007.

[Pierce99] J. S. Pierce, B. C. Stearns, and R. Pausch, “Voodoo dolls: seamless interaction at multiple scales in virtual environments,” In *Proc. of the ACM Symposium on Interactive 3D Graphics*, pp. 141-145, 1999.

[Pinho08] M. S. Pinho, D. A. Bowman, and C. M. Freitas, “Cooperative object manipulation in collaborative virtual environments,” *Journal of the Brazilian Computer Society*, vol. 14, no. 2, pp. 54-67, 2008.

[Plumnlee03a] M. Plumlee and C. Ware, “Integrating multiple 3D views through frame-of-reference interaction,” In *Proc. of IEEE Coordinated and Multiple Views in Exploratory Visualization*, pp. 34-43, 2003.

[Plumlee03b] M. Plumlee and C. Ware, “An evaluation of methods for linking 3D views,” In *Proc. of the ACM Symposium on Interactive 3D Graphics*, pp. 193-201, 2003.

[Poupyrev96] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa, “The go-go interaction technique: non-linear mapping for direct manipulation in VR,” In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 79-80, 1996.

[Poupyrev98a] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst, “Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques,” *Computer Graphics Forum*, vol. 17, no. 3, pp. 41-52, 1998.

[Poupyrev98b] I. Poupyrev, N. Tomokazu, S. Weghorst, “Virtual notepad: handwriting in immersive VR,” In *Proc. of IEEE Virtual Reality*, pp. 126-32, 1998.

[Ramos04] G. Ramos, M. Boulos, and R. Balakrishnan, “Pressure widgets,” In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 487-494, 2004.

[Razzaque05] S. Razzaque, “Redirected walking,” *PhD Dissertation*, University of North Carolina at Chapel Hill, 2005.

[Rekimoto99] J. Rekimoto, and M. Saitoh, “Augmented surfaces: a spatially continuous work space for hybrid computing environments,” In *Proc. of the SIGCHI conference on Human Factors in Computing Systems*, pp. 378-385, 1999.

[Roberts12] C. Roberts, B. Alper, J. K. Morin, T. Hollerer, “Augmented textual data viewing in 3d visualizations using tablets.” In *Proc. of IEEE 3D User Interfaces*, pp. 101-104, 2012.

[Roudaut09] A. Roudaut, E. Lecolinet, and Y. Guiard, “MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb,” In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 927-936, 2009.

[Ryu03] Y. S. Ryu, B. Yost, G. Convertino, J. Chen, and C. North, “Exploring cognitive strategies for integrating multiple-view visualizations,” In *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, vol. 47, pp. 591-595, 2003.

[Schmalstieg99a] D. Schmalstieg, L. M. Encarnacao, Z. Szalavári, “Using transparent props for interaction with the virtual table.” In *Proc. of the ACM Symposium on Interactive 3D Graphics*, pp. 147-153, 1999.

[Schmalstieg99b] D. Schmalstieg and G. Schaufler, “Sewing worlds together with SEAMS: A mechanism to construct complex virtual environments,” *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 4, pp. 449-461, 1999.

[Slater94] M. Slater and M. Usoh, “Body centered interaction in immersive virtual environments,” *Artificial Life and Virtual Reality*, vol. 1, pp. 1-22, 1994.

[Song11] P. Song, W. Goh, and C. Fu, “WYSIWYF: exploring and annotating volume data with a tangible handheld device,” In *Proc. of the SIGCHI Conference on Human Factors in Computing* Systems, pp. 1333-1342, 2011.

[Stafford06] A. Stafford, W. Piekarski, and B. Thomas, “Implementation of god-like interaction techniques for supporting collaboration between outdoor AR and indoor tabletop users,” In *Proc. of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 165-172, 2006.

[Steinicke06] F. Steinicke, T. Ropinski, K. Hinrichs, and G. Bruder, “A multiple view system for modeling building entities,” In *Proc. of IEEE Coordinated and Multiple Views in Exploratory Visualization*, pp. 69-78, 2006.

[Stoakley95] R. Stoakley, M. J. Conway, and R. Pausch, “Virtual reality on a WIM: interactive worlds in miniature,” In *Proc. of the SIGCHI Conference on Human Factors in Computing* Systems, pp. 265-272, 1995.

[Suma12] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas, “Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 555-564, 2012.

[Szalavári97] Z. Szalavári, and M. Gervautz, “The personal interaction panel - a two-handed interface for augmented reality,” *Computer Graphics Forum*, vol. 16, no. 3, C335-C346, 1997.

[Taylor01] R. M. Taylor, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, “VRPN: a device-independent, network-transparent VR peripheral system,” In *Proc. of the ACM Symposium on Virtual Reality Software and Technology*, pp. 55-61, 2001.

[Usoh99] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks, “Walking> walking-in-place> flying, in virtual environments,” In *Proc. of the ACM Computer Graphics and Interactive Techniques*, pp. 359-364, 1999.

[Veit14] M. Veit and A. Capobianco, “Go‘then’tag: a 3D point cloud annotation technique,” In *Proc. of IEEE 3D User Interfaces*, pp. 193-194, 2014.

[Vidal04] M. Vidal, M. Amorim, and A. Berthoz, “Navigating in a virtual three-dimensional maze: how do egocentric and allocentric reference frames interact?” *Cognitive Brain Research*, vol. 19, no. 3, pp. 244-258, 2004.

[Viega96] J. Viega, M. J. Conway, G. Williams, R. Pausch, “3D magic lenses,” In *Proc. of the ACM Symposium on User Interface Software and Technology*, pp. 51-58, 1996.

[Wang12b] J. Wang and R. W. Lindeman, “Comparing isometric and elastic surfboard interfaces for leaning-based travel in 3D virtual environments,” In *Proc. of IEEE 3D User Interfaces*, pp. 31-38, 2012.

[Wang12a] J. Wang, R. Budhiraja, O. Leach, R. Clifford, and D. Matsuda, “Escape from meadwyn 4: a cross-platform environment for collaborative navigation tasks,” In *Proc. of IEEE 3D User Interfaces*, pp. 179-180, 2012.

[Wang13a] J. Wang, O. Leach, R. W. Lindeman, “DIY world builder: an immersive level editing system,” In Proc. of IEEE 3D User Interfaces, pp. 195-196, 2013.

[Wang13b] J. Wang and R. W. Lindeman, “Force Extension: extending isotonic position-controlled multi-touch gestures with rate-controlled force sensing for 3D manipulation,” In *Proc. of IEEE 3D User Interfaces*, pp. 3-6, 2013.

[Wang14] J. Wang, and R. W. Lindeman, “Coordinated 3D interaction in tablet-and HMD-based hybrid virtual environments,” In *Proc. of ACM Spatial User Interaction*, pp. 70-79, 2014.

[Wang15] J. Wang and R. W. Lindeman, “Object impersonation: towards effective interaction in tablet- and HMD-based hybrid virtual environments,” In *Proc. of IEEE Virtual Reality*, 2015.

[Wang Baldonado00] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinskyk, “Guidelines for using multiple views in information visualization,” In *Proc. of ACM Advanced Visual Interfaces*, pp. 110-119, 2000.

[Watsen98] K. Watsen, R. Darken, M. A. Capps, “Handheld computer as an interaction device to a virtual environment,” *International Immersive Projection Technology Workshop*, 1998.

[Welch02] G. Welch and E. Foxlin, “Motion tracking: no silver bullet, but a respectable arsenal,” *IEEE Computer Graphics Applications*, vol. 22, no. 6, pp. 24-38, 2002.

[Wilkes12] C. B. Wilkes, D. Tilden, and D. A. Bowman, “3D user interfaces using tracked multi-touch mobile devices,” In *Joint Virtual Reality Conference*, pp. 65-72, 2012.

[Zanbaka05] C. A. Zanbaka, B. C. Lok, S. V. Babu, A. C. Ulinski, and L. F. Hodges, “Comparison of path visualizations and cognitive measures relative to travel technique in a virtual environment,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, pp 694-705, 2005.

[Zhai93a] S. Zhai and P. Milgram, “Human performance evaluation of manipulation schemes in virtual environments,” In *Proc. of IEEE Virtual Reality*, pp. 155-161, 1993.

[Zhai93b] S. Zhai, “Investigation of feel for 6DOF inputs: isometric and elastic rate control for manipulation in 3D environments,” In *Proc. of Human Factors and Ergonomics Society Annual Meeting*, vol. 37, no. 4, pp. 323-327, 1993.

[Zhai95] S. Zhai, “Human performance in six degree of freedom input control,” *Ph.D. Thesis*, University of Toronto, 1995.

[Zhou08] F. Zhou, H. Duh, and M. Billinghurst, “Trends in augmented reality tracking, interaction and display: a review of ten years of ISMAR,” In *Proc. of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 193-202, 2008.