2019-01-16

Designing Camera Controls for Map Environments

Danyluk, Kurtis

UNIVERSITY OF CALGARY

Designing Camera Controls for Map Environments

by

Kurtis Danyluk

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

JANUARY, 2019

Abstract

We present an exploration of two classes of navigation techniques designed for representations of real-world terrain. The first introduces look-from camera controls, a new style of camera control for touch devices designed with representations of real world-terrain in mind and provides an evaluation of three different implementations of this style of control. The second looks to virtual reality and compares the effectiveness of four existing and common camera control techniques within the context of a representations of real world-terrain. Effective camera controls greatly increase a user's ability to engage with a virtual environment, and virtual map environments are no different. However, current camera controls are difficult to use within map-like environments, requiring burdensome sequences of interactions or performing poorly within ragged terrain. To examine the effectiveness of different camera controls in this space we conducted two studies in which we asked participants to perform map reading and interaction tasks. In both studies the camera control technique greatly influenced participant engagement and enjoyment within a scene. The first study highlights the effectiveness of look-from camera controls as light-weight additions to direct manipulation controls and provides design guidelines for the construction of look-from camera controls. The second study highlights which existing common navigation techniques are most appropriate within a map-like environment presented in immersive virtual reality and how combinations of these controls can combine the strengths of the controls to cover for the weaknesses of others.

Research Acknowledgements

Some figures and material in this thesis are drawn from prior work and used with permission:

Danyluk, Kurtis Thorvald, Bernhard Jenny, and Wesley Willett. 2019. "Look-From Camera Control for 3D Terrain Maps." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 10. https://doi.org/10.1145/3290605.3300594. (In Press)

Danyluk, Kurtis and Wesley Willett. 2018. "Evaluating the Performance of Navigation Techniques in Large Scale Open Virtual Environments." University of Calgary. (In Submission)

To represent shared contributons of my collaborators, and the collaborative nature of research, I use the plural (we) throughout this thesis. Collaborative content is reproduced within Chapters 3 and 4.

Acknowledgements

I would like to thank my family and friends for their continued and consistent support. I would also like to thank the members of the Interactions Lab, especially those that helped to test the many different camera control techniques that appear in this thesis.

Table of Contents

List of Tables and Sidebars

List of Figures

List of Symbols, Abbreviations, and Nomenclature

Symbol	**Definition**
VR	Virtual Reality
RQ	Research Question

Chapter 1 : Introduction

This thesis explores how to improve user interactions with representations of real-world terrain through the choice and design of camera controls. Within this thesis we explore how light-weight controls can be added to existing touch control schemes to enrich the viewer experience and what controls are most appropriate for Immersive Virtual Reality (VR) map applications (an emerging new platform). Digital terrain tools are useful because they allow viewers to smoothly navigate complex virtual terrain. Those designed for traditional 2D displays, such as Google Earth, allow viewers to transition from top-down and overview perspectives of a terrain to ground-level and oblique perspectives. Those designed for VR, such as Google Earth VR, allow viewers to immerse themselves directly within the terrain, like standing in a miniature world. Thus, this thesis addresses: **How we can improve camera controls specialized for representations of real-world terrain.** In this chapter, we provide background on existing camera control options for desktops, touch devices, and VR. We then provide information on several challenges that viewers face with these existing controls and that designers face when choosing controls as motivation to inform our thesis questions. We then discuss the methodological approach we took to address these questions and the scope of this approach before outlining our research contributions. Finally, we summarize the structure of this thesis document by describing the remaining chapters.

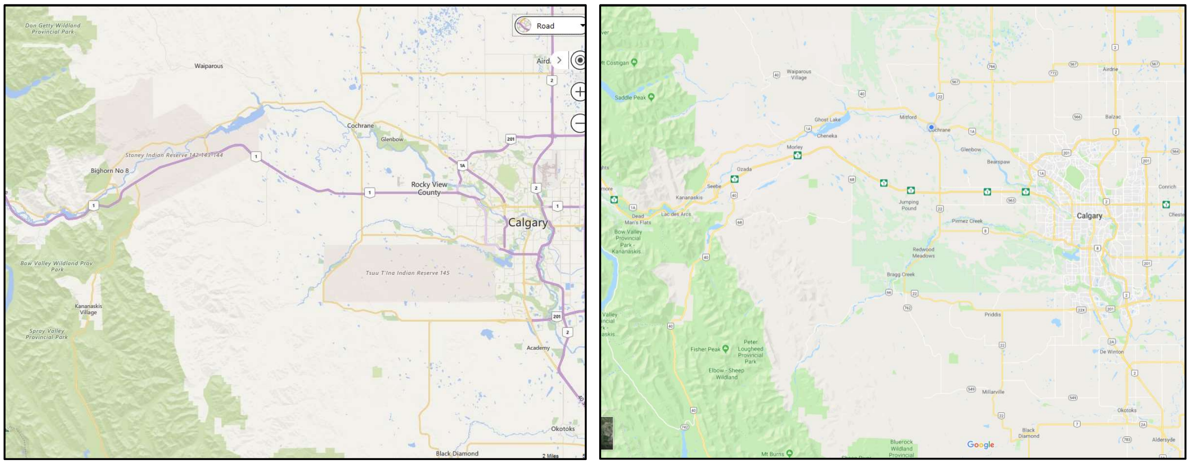


Figure 1.1. Top down view of the same area in Bing Maps (Left) and Google Maps (Right).

1.1 Background and Motivation

Camera controls for traditional 2D systems and for VR systems can vary greatly based on their display. Here we will provide background on those controls first for desktop and touch devices, and then for VR systems.

1.1.1 Interactive Camera Controls for Maps on Desktop and Touch Devices

Interacting with a digital map using a desktop or touch device is a common task: “over three-fourths (77%) of smartphone owners regularly use

Direct manipulation on desktop and mobile devices is composed of four underlying controls; pan, zoom, rotate, and pitch; which share an important feature: when the viewer grabs a point in the scene, either using touch or a mouse, that location is preserved.

- *Pan* allows the viewer to move the camera parallel to the current view plane.
- *Zoom* allows the viewer to move the camera towards or away from a zoom point. This causes objects to become bigger or smaller as per Thale’s theorem.
- *Rotate* allows the viewer to roll the camera, rotating it about the local horizontal axis.
- *Pitch* allows the viewer to tilt the camera, rotating it about the local vertical axis.

Sidebar 1.1 A summary of Direct Manipulation

navigation apps."[1] The standard for existing digital terrain tools for desktop and touch devices is to allow viewers to control their view of the map using mouse (or touch) based direct manipulation controls that pan, zoom, rotate, and pitch a camera within the virtual environment (Sidebar 1.1). For example, these controls are standard within Google Maps and Bing Maps (Figure 1.1). Using these controls, viewers can fluidly adjust their point of view, exploring the map and viewing terrain from a variety of different perspectives.

However, many common views – particularly those from ground-level and from specific vantage points in the scene – can be difficult to recreate using direct manipulation camera control techniques. In particular, navigating from a top-down view to an oblique or ground-level view often requires a complicated chain of direct manipulation operations that can be difficult to execute precisely and challenging to reverse. Good camera controls can greatly affect how users perceive a system (Yannakakis, Martínez, and Jhala 2010) and making these sequences of interactions easier for viewers is an important challenge.

1.1.2 Navigation for Maps in Immersive Virtual Reality

In VR applications there is no single standard camera control technique for navigating digital terrain. The experience within a VR application is significantly different than with an on-screen device, camera orientation is generally performed using head tracking and viewers are placed within a virtual recreation of the scene. It is for that reason that we refer to camera controls as

---

[1] https://themanifest.com/app-development/popularity-google-maps-trends-navigation-apps-2018

Figure 1.2. A promotional image of Florence presented in Google Earth VR

navigation controls within VR, as they only control a viewer's position and not orientation. Notably, because map-like environments represent seamless, large virtual environments, their sheer size rules out many familiar camera control techniques such as physical locomotion. Even ignoring navigation controls unfit for large virtual environments, navigation control techniques designed for VR have a wide range. Some common navigation techniques are: vehicle metaphors that allow the viewer to control their camera as though driving a vehicle, object manipulation that allow the viewer to grab the world and move it around, worlds-in-miniature that allow the viewer to move miniature tokens of themselves in a miniature world to move themselves in the world, and target selection techniques that allow the viewer to select a specific location and go there instantly.

While a variety of techniques for navigating large spaces exists in the research literature and in commercial applications, the trade-offs between them remain poorly understood and there does not exist any clear optimal technique for navigating digital maps. Existing research discusses each camera control technique in a vacuum, generally discussing only what a given technique can do and is designed for, and not their trade offs when compared to alternatives. Those evaluations that do exist consider a limited set of techniques (Kopper et al. 2006; Jankowski and Hachet 2015). Moreover, contemporary navigation techniques like 3D cone drag (Käser et al. 2017) which have been designed with large map environments and modern VR hardware in mind, remain largely unexamined in the research literature. As a result, designers and developers creating applications that require navigation in large map virtual environments have relatively little guidance as to which techniques are the best.

1.2 Thesis Question

With the above concerns in mind, the guiding question of this thesis is: **How can we improve camera controls specialized for representations of real world-terrain?** To better address this guiding question, we provide two more specific research questions:

Research Question 1 (RQ1): Many applications that rely on touch controls currently use direct manipulation camera control. How can we add to and improve, these touch based systems so that viewers can recreate views and better understand terrain?

> This question arises from desktop and touch systems where many applications use direct manipulation camera control. By designing around the already existing standard interactive camera controls we can directly address many of its flaws to allow for better overall user experiences, without losing the familiarity and effectiveness of the existing standard camera controls.

Research Question 2 (RQ2): Immersive virtual reality systems have several options for camera control. Given these many options, which option(s) are most appropriate for representations of real-world terrain?

This question arises from the uncertainty of camera controls in VR, where there are many possible choices. By comparing existing options against each other we can develop a firmer understanding of their strengths and weaknesses and determine under what circumstances they are appropriate to use.

1.3 Methodology

To address the guiding thesis question, we developed a testbed that enabled us to create representations of real world-terrain at varying scales from regions all around the world and to develop and explore specific camera control techniques.

Using this testbed, we designed several new camera control techniques, and recreated many existing ones. Using those implementations, we performed two within-subjects laboratory studies. One study focused on evaluating the new techniques, and one focused on comparing the existing ones.

To answer research question one (RQ1) we show how common camera movements can instead be described as *look-from* operations and integrated on top of traditional *direct manipulation* controls in an application that uses touch controls. *Look-from* camera controls reformulate camera movements as declarations from the user that they want to look from one point in a scene at or towards another.

We present three variants – *discrete look-from-at, continuous look-from-forward,* and *continuous look-from-towards* – which each allow viewers to specify ground-level views using a single touch gesture and that can be used in conjunction with existing direct manipulation controls. While these techniques vary in terms of their constraints, all three complement the existing vocabulary of direct manipulation techniques for digital maps (Pan, Zoom, Rotate, Pitch). We implemented these three *look-from* camera controls within our testbed, along with a full suite of *direct manipulation* interactions. Afterwards, we performed a study in which we asked participants to perform line-of-sight and elevation comparison tasks using each *look-from* variation and *direct manipulation* controls. We did this using a touch screen device where standard *direct manipulation* controls are already ubiquitous.

To answer research question two (RQ2) we present a direct comparison of four prominent navigation techniques for camera control in large VR environments: *flight*, *teleportation*, *world-in-miniature*, and *3D cone drag.* To compare these controls, we implemented each camera control technique and conducted two directly-linked studies in which participants performed navigation and search tasks within a mountainous environment. We sought to understand when each technique performed best, which techniques viewers preferred to use, and how their navigation behavior changed when using each technique, especially when they had access to multiple. We performed this study in VR because unlike desktop and 2D touch devices, there does not already exist a ubiquitous standard control scheme. By working within this less rigidly defined environment we could compare a wider variety of potential options.

1.5 Research Contribution

This thesis has four main contributions in the domain of designing interactive camera controls for use in representations of real world-terrain:

- *Contribution 1:* A side-by-side comparison of *direct manipulation* camera controls and three variations of *look-from* camera controls at line-of-sight and elevation comparison tasks. We performed this on representations of real world-terrain, presented on a tablet sized touch device. This shows how *look-from* camera controls can be integrated smoothly on top of *direct manipulation* controls without impacting performance but improving the user experience.
- *Contribution 2:* Design considerations for future *look-from* navigation tools for digital maps based on our observations of and experience with these initial variants. These considerations highlight many non-obvious features necessary for a clean introduction of *look-from* camera controls to system.
- *Contribution 3:* A side-by-side comparison of *flight, teleportation, world-in-miniature,* and *3D cone drag* at navigation and search tasks on a representation of real world-terrain presented in an immersive virtual environment. This shows the effectiveness of *flight* as a general-purpose navigation technique for map-like environments in VR.
- *Contribution 4:* An examination of the five specific ways in which participants combined *flight, teleportation, world-in-miniature,* and *3D cone drag* to more effectively complete search and navigation tasks by compensating for the weaknesses of some techniques with the strengths of others.

1.6 Document Overview

Chapter One : Introduction provides background into some camera control techniques used in representations of real-world terrain, a guiding question for this thesis, and accompanying research questions.

Chapter Two: **Literature Review** presents an overview of previous work within the domains of interactive camera controls and the design of digital maps. We discuss early work on interactive camera controls before expanding upon more recent work on interactive camera controls appropriate for large and multiscale environments. We also discuss a sampling of works that improve upon the design of digital maps.

Chapter Three: Observations of Digital Map Use and Developing a Testbed discusses a pilot study in which we attempted to identify and describe many of the issues with current camera control techniques. we also describe the testbed that we use for later studies. This chapter informs many of the choices we make in the following studies.

Chapter Four: Look-From Camera Control presents the design of *look-from* camera controls. We provide an in-depth description of the design and implementation of each camera control. Additionally, we present the design and results of a study comparing them against *direct manipulation* camera control. We then discuss design considerations for *look-from* camera controls drawn from these explorations.

Chapter Five: Navigating Maps in VR presents the design and results of our second study in which we provide a direct comparison of *flight, teleportation, world-in-miniature,* and

3D cone drag within VR. We use that study to ground a discussion on the design and implementation of each of these camera controls and the design of future VR map systems.

Chapter Six: Conclusion closes this thesis by revisiting the thesis questions and contributions then answers them within the context of the results of Chapter Four and Chapter Five.

Chapter 2 : Literature Review

In this chapter, we provide an overview of past research related to this thesis and discuss how it relates to our work. We discuss interactive camera controls (both for traditional and VR applications), multiscale camera controls, and interaction techniques for improving user experience with digital maps.

We begin by discussing general movement and targeted movement camera controls, as defined by (Jankowski and Hachet 2015). We start by describing common general and targeted movement options before detailing those in the set of interactive camera controls that relate very closely to the *look-from* camera control metaphor that we later explore in Chapter 4.

Afterwards, we discuss work on interactive camera controls designed specifically for VR, constrained to camera controls appropriate for large scale virtual environments. These relate to the navigation techniques we explore in Chapter 5.

Next, we briefly discuss multiscale camera controls, detailing work that applies to both traditional desktop and touch screen systems, as well as VR systems.

Finally, we discuss interaction techniques for terrain navigation. This section focuses on interactions other than camera controls that allow users to interact with their environment.

2.1 Interactive Camera Control

Within this thesis we focus primarily on interactive camera control, that is, camera control that is primarily controlled by the viewer. Interactive camera control stands opposite to automatic camera control, where the camera is controlled primarily by the system. With this distinction in mind we do not discuss works that focus on automatic camera control and instead limit our discussion on interactive camera control. Interactive camera control can be broadly categorized as either general movement or targeted movement. General movement controls allow the viewer to freely explore a scene, while targeted movement controls move the viewer relative to a target.

2.1.1 General Movement

General movement camera controls allow the user to freely explore a virtual environment (Jankowski and Hachet 2015). Rotate-pan-zoom is the primary camera control option used in many virtual environments (such as Maya, Unity, Blender) and represents a standard way to interact with objects within a desktop environment. Rotate, pan, and zoom can all be represented as a two-dimensional operation and interaction is performed by swapping between rotate, pan, and zoom modes. Different from the *direct manipulation* controls described earlier, Rotate-pan-zoom controls generally work off of a virtual sphere (Chen et al. 1988). Pan moves up, down, left or right along the camera's view plane, zoom moves towards or away from the center of the virtual sphere, and rotate moves around the virtual sphere.

Direct manipulation acts similarly to Rotate-pan-zoom, but, inspired by "through the lens" (Gleicher and Witkin 1992) camera control, considers the screen space of a scene. By considering screen space, the geometry of the scene itself replaces the virtual sphere used in

Rotate-pan-zoom. This allows pan, zoom, rotate, and pitch operations to control the camera as though the viewer was directly moving, or manipulating, the underlying scene. As a reminder:

direct manipulation on desktop and mobile devices is typically composed of four underlying controls; pan, zoom, rotate, and pitch. These interactions share an important feature: when the viewer grabs a point in the scene, either using touch or a mouse, that location within the scene remains underneath the mouse or touch point (Sidebar 1.1).

This set of *direct manipulation* controls is the one that we consider in Chapter Four where we study the effects of adding light weight *look-from* controls on top of them.

Another major option for general movement in virtual environments is to allow the viewer to walk, drive, or fly through the space (Ware and Osborne 1990). This option uses vehicle or walking metaphors of camera control. On desktop or touch devices these metaphors are very commonly used in video games. They see less use in map-like applications on desktop devices but are more common in virtual reality – where we discuss them in more depth.

2.1.2 Targeted Movement

Targeted movement allows a viewer to control their camera with respect to a target (Jankowski and Hachet 2015). Perhaps the simplest form of this is go-to movement (Card, Mackinlay, and Robertson 1991) that simply moves the camera towards an indicated point of interest. This can be expanded upon to allow for more complicated operations such as the interactions shown by Unicam (Zeleznik and Forsberg 1999) that allow a viewer to focus their camera around a point, or in speed-coupled flying with orbiting (Tan, Robertson, and Czerwinski 2001) that allows the camera to orbit around the point of interest.

Once again taking inspiration from "through the lens" (Gleicher and Witkin 1992) camera controls, targeted movement can be expanded to consider the screen space of a scene. For example Navidget (Hachet et al. 2008) allows viewers to choose a navigation endpoint and transition towards it and Hagedorn and Döllner (2008) consider screen space to allow viewers to control their camera through sketches. We consider screen space in the form of *look-from* camera controls where we allow viewers to specify two points of interest in a scene to automatically transition the camera so that it is looking from one point and at another. We expand more closely upon these screen space targeted movement camera controls in Chapter Four by examining them within the context of representations of real world-terrain.

2.1.3 Navigation in VR

As a form of interactive camera control, camera controls designed for VR can be described in terms of general movement and targeted movement. However, because of how different the viewer experience can be within VR we provide a more focused overview of navigation in VR.

Early work on navigation in virtual reality by Ware and Osborne (1990) introduced and evaluated several navigation metaphors, including techniques they referred to as "flying vehicle control" and "scene-in-hand", both of these are forms of general movement. In the former viewers control their position and orientation as though they were controlling a flying vehicle. With the latter viewers move the entire scene as though they physically had the scene in their hand. Due to the limitations of these early systems, scenes were limited to single objects and camera orientation was entirely controlled as part of the navigation technique, rather than by the viewer's head motion. Later work, including Stoakley et al.'s (1995) "Virtual reality on a WIM" and Pausch et al.'s (1995) "hand-held miniatures", introduced the notion of *worlds-in-*

miniature, which allowed viewers to navigate by manipulating a token placed in a miniature recreation of the scene. Much like earlier work, Stoakley et al.'s original system still relied on orientation of the token rather than viewer's head position to provide camera orientation.

In contrast, most contemporary VR systems use the viewer's head position (typically tracked via sensors on head-mounted displays) to provide camera orientation and rely on navigation techniques like flight and teleportation to change just the viewer's position in the environment. Other techniques like *3D cone drag* (Käser et al. 2017) also extend the direct-manipulation metaphor of scene-in-hand to modern VR systems, allowing viewers to grab, manipulate, and move the scene around them.

With the addition of tracking, physical locomotion for navigation in VR is another popular method of navigating VR space, with many implementations of walking (Slater, Usoh, and Steed 1995; Yan, Lindeman, and Dey 2016). However, because of the large scale of map-like environments most simple realizations of locomotion in VR are inadequate in large scenes. More complex realizations, such as redirected walking (Peck, Fuchs, and Whitton 2012) or redirected teleportation (Liu et al. 2018), allow locomotion to function within large virtual spaces, but still don't reduce the physical effort required to walk across a potentially kilometers long scene. Other locomotion alternatives, such as virtual surfboards (Jia Wang and Lindeman 2012) or finger-walking (Yan, Lindeman, and Dey 2016) translate locomotion into a method to control another navigation technique, normally flight.

Early comparative work examined basic *flight* and *teleportation* in relatively small environments and showed that *teleportation* can be faster than *flight* but causes greater disorientation (Bowman, Koller, and Hodges 1997). Moreover, a large body of work suggests that constant velocity *flight* is less disorienting than accelerated *flight*, regardless of velocity (Bowman, Koller, and Hodges 1997). However, the current state of the art still reveals little about the relative effectiveness of techniques other than *teleportation* and *flight* (Jankowski and Hachet 2015). In Chapter Five we expand upon these comparisons, providing a comparison of *flight, teleportation, world-in-miniature,* and *3D cone drag* within the context of a large-scale map environment. We access the strengths and weaknesses of each, as well as evaluating how they can be combined to address their weaknesses.

2.1.4 Multi-Scale Navigation

Work on multi-scale and world-scale navigation includes adaptations that can improve speed and precision across multiple environmental scales. Trindade and Raposo (2011) and McCrae et al. (2009) both extend *flight* by using cubemaps to control the viewers' movement velocity based on their distance from scene geometry. "GiAnt" (Argelaguet and Maignant 2016) tackles this same problem by dynamically adjusting the viewers' scale to give the appearance of constant-velocity *flight* at multiple different scales. Google Earth VR (Käser et al. 2017) uses a hybrid of these approaches, adjusting the viewer's scale to facilitate both large and small movements, but also slowly increasing velocity when the viewer is close to the terrain. General comparisons of these techniques remain limited, with most research focusing on comparisons with targeted movement navigation techniques which focus views around specific objects in the world, rather than supporting general movement (Kopper et al. 2006).

2.2 Interaction Techniques for Terrain Navigation

As previously mentioned, due to large scales, camera control for digital globes and large terrain generally requires multiscale navigation and camera control techniques. As an alternative, the use of constraints applied to camera positioning, orientation and animation has been explored. Buchholz, Bohnet, and Dollner (2005) adapt physically-based navigation by Turner et al. (1991) for navigation in terrain models. Other work explores physically-based navigation where the camera is modelled as a rigid body (Silva, Santos, and Oliveira 2009) or as a mass-spring system (Buchholz, Bohnet, and Dollner 2005) with user input exerting forces on that body. Additional constraints for moving the camera are often essential for navigation in terrain (Hanson and Wernert 1997) or 3D city models (Hildebrandt and Timm 2014). Occlusion avoidance is particularly relevant for camera navigation in ragged terrain. In response, force fields (Xiao and Hubbold 1998) or distance fields (Wan, Dachille, and Kaufman 2001) have been proposed for physically-based navigation.

Alternatives to these traditional camera control techniques also exist. Hagedorn and Döllner (2008) proposed sketch-based navigation in 3D virtual environments, particularly geospatial visualization. The user draws sketches on a touch display, which are interpreted to control the camera. For example, a curve drawn on a street is converted to an animated drive along that street. Another option is to place 3D representations of distant landmarks and places in the currently visible view (Pierce and Pausch 2004). The representations are interactive and trigger a ride to the corresponding landmarks and places.

Other work has attempted to aid viewers in understanding terrain maps by adding new interaction techniques to 2D digital maps without allowing for full six degrees of freedom navigation. Lightweight Relief Shearing (Willett et al. 2015) and Elastic Terrain (Buddeberg, Jenny, and Willett 2017) both use simple touch gestures to expose the shape of the terrain in ways that are compatible with existing pan-zoom interactions, but because they reveal shape by obliquely shearing the entire map, it is not possible to create ground-level views with them. In spite of this, participants in initial evaluations of these techniques (Willett et al. 2015) often tried to stretch the terrain in order to examine the silhouettes of terrain features and recreate specific ground-level views. Other observation work such as the findings from Abend et al. (2012) show user patterns within map-like environments that can be employed when designing for them. For example, users of Google Earth most often retain the north orientation of maps and tend to quickly return to a north orientation after a rotation.

Chapter 3 : Observations of Digital Map Use and Developing a Testbed

In this chapter, we briefly discuss the results of an exploratory pilot study that informs the design of our later studies, as well as the development of a standalone testbed.

We begin by briefly discussing the pilot study and follow with a discussion of the reoccurring problems that pilot participants faced throughout. These problems, and the execution of the pilot itself, helped identify the difficulties we would expect in follow up studies and informed what measures to capture in future work.

Finally, we discuss the motivation behind developing our own standalone testbed for camera control techniques and how the pilot study influenced its design. We then detail some more specific capabilities of the testbed and the implementation of them.

3.1 Understanding Digital Map Use

Navigating between traditional map views and more realistic views such as between Google Map view and Google Street view is a complex challenge. When changing their perspective within a scene a viewer often needs to chain together a series of direct manipulation operations. Further, while performing that chain of operations they must keep track of their destination by managing scale, moderating detail, and avoiding occlusion. This challenge is exacerbated by computer devices with limited options for user input, such as touch screens. In order to better understand how people approach this problem, we ran an exploratory pilot experiment in which we tasked participants with navigating through representations of real world-terrain. The goal was to observe how viewers tackle the challenge of changing their perspective and controlling their camera. We were most interested in how participants transitioned from top-down views to ground-level oblique views, and how they acted to avoid occlusion. Due to the limited scope of this pilot study we will not discuss the results of participants in depth, but will discuss the high-level observational takeaways of the study and how it informed future work.

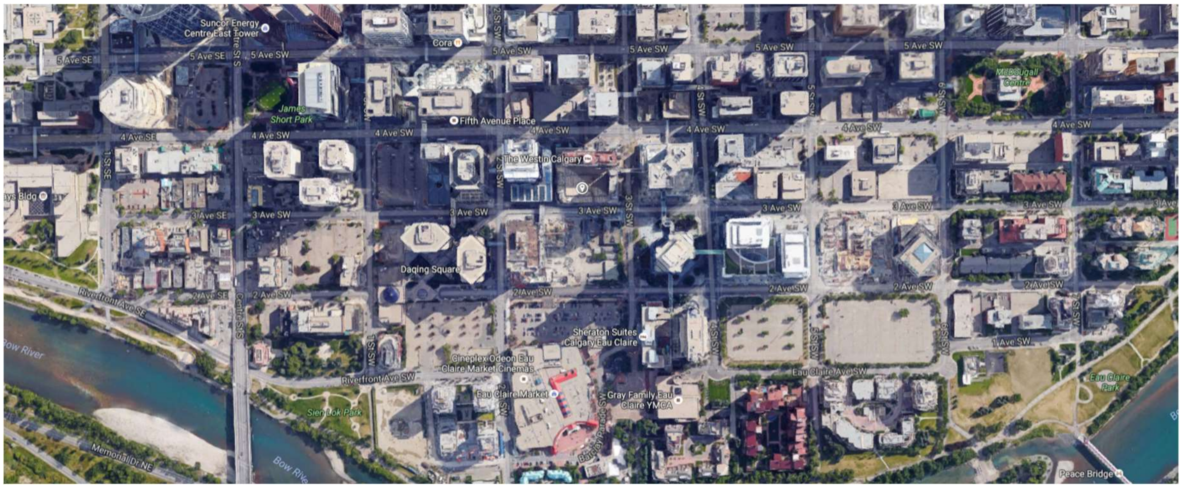


Figure 3.1. A top-down view of a cityscape

3.1.1 Tasks

As a prompt to make participants interact with a digital map system in depth, we tasked participants to **recreate an oblique viewpoint starting from a top-down one**. we showed participants a screenshot of an oblique view of a scene created within Google Earth (Figure 3.2, Figure 3.4) and a link to a north-aligned top-down view of that same region (Figure 3.1, Figure 3.3). From that top-down view, we asked them to recreate the oblique view as best they could.

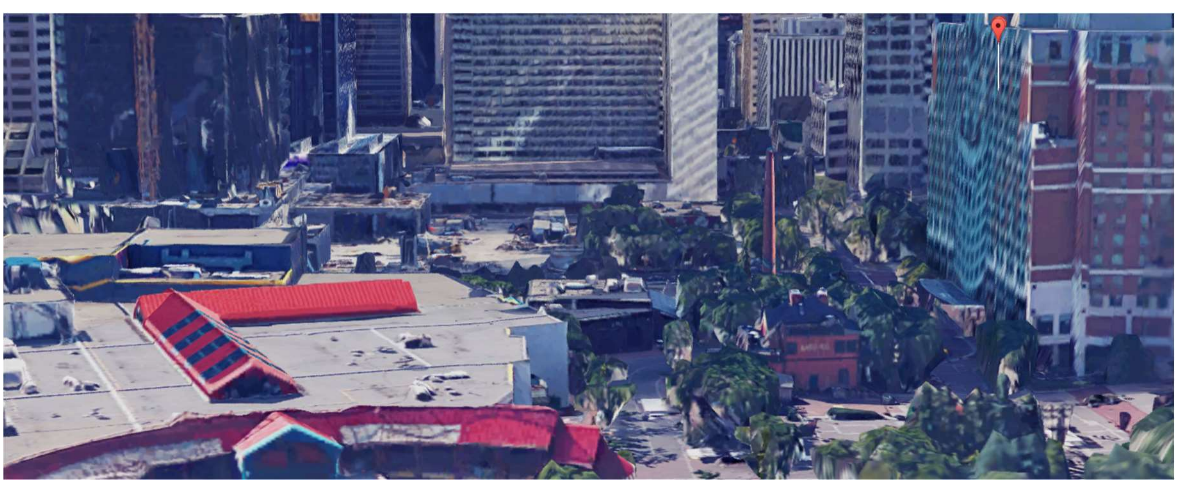


Figure 3.2. An oblique view of the cityscape in Figure 3.1

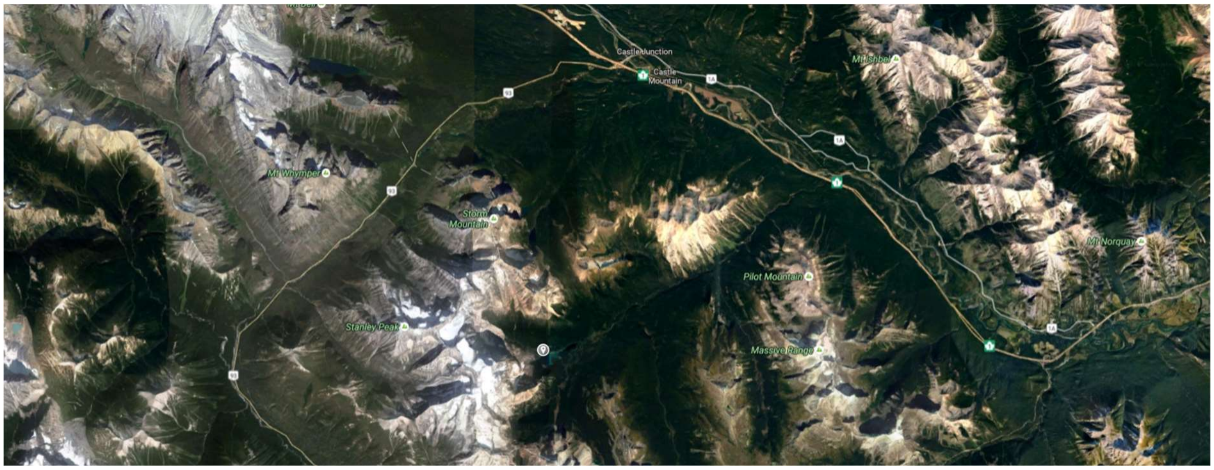


Figure 3.3. A top-down view of a mountainous region

3.1.2 Measures

We timed how long it took for the participant to complete each trial. We timed this from the moment the participant clicked on the link to load the top-down view, until the moment that the participant expressed satisfaction with the recreation of the view. Throughout the trials we recorded participant actions and codified them as pan, zoom, pitch, or rotate actions.

In addition, prior to the trial's we asked participants about their experience with both physical and digital maps.

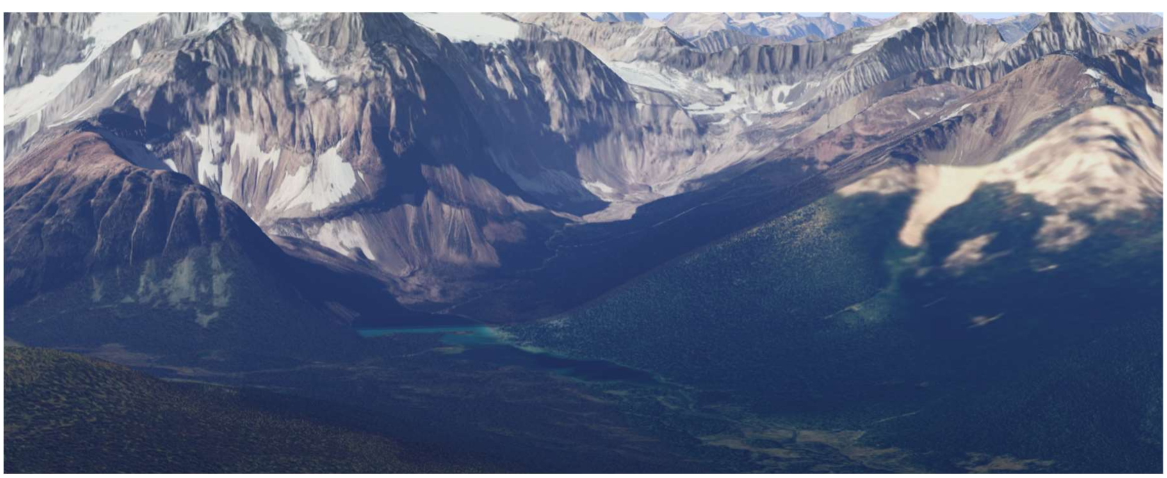


Figure 3.4. An oblique view of the mountainous region in Figure 3.3

3.1.3 Test Environment

We ran this pilot study on a Microsoft Surface 4 with an attached secondary monitor and mouse. On the primary screen we had an instance of Google Earth open and ready, on the secondary screen we showed a slideshow containing target oblique views.

Within Google Earth we presented 10 pairs of scenes with varying geography:

Scene Type	Defining Feature	Location
Urban	University Campus	University of Calgary Campus
	Skyscrapers	Downtown Calgary
	Flat City	Venice
Wilderness	Mountain Range	Rocky Mountains
	Mountain Valley	Rocky Mountains
	River Delta	Auyuittuq National Park
	Tall Mountain	Mount Everest
	Forest	Reserva Tariquia
	Flat Coastline	Puerto Patillos
	Steep Coastline	Cliffs of Dover

3.1.4 Participants

The pilot study had three participants. They ranged in age from 21 to 23 and had varying degrees of experience with maps. One had extensive map reading experience, one had used maps infrequently, and one had never used a physical map before. All three had some prior experience with Google Maps and Google Earth.

3.1.5 Procedure

Participants completed a total of 20 trials, all of which were presented in the same order. Each trial consisted of one task. We gave participants time to take a break between each trial and the entire process took roughly an hour.

3.2 Identifying Challenges for Traditional Camera Controls

When recreating views, participants routinely struggled during the transition between the top-down, oblique, and ground-level perspectives. We observed several recurring problems that make these kinds of transitions challenging:

Chaining interactions. Moving between top-down and ground-level views typically requires a viewer to chain together multiple pan, zoom, rotate, and pitch operations. This requires a degree of foresight as interactions made early in the sequence – such as moving to ground level too early – influence the effect of subsequent interactions. This problem was most frequent in cityscapes and severe mountain ranges where maneuvering around buildings and tall peaks was much more difficult at ground level.

Disorientation. Sequences of pan, rotate, and pitch interactions compound one another and can make it difficult to locate and recreate desired viewpoints. In our pilot, participants often became disoriented, requiring a brief pause between steps to reorient themselves. These brief pauses slowed down the overall process. This occurred most frequently when participants where very far from ground level and landmarks were more difficult to visually resolve, or in scenes that did not contain many obvious landmarks.

Reversibility. Undoing or reversing sequences containing multiple direct manipulations interactions can become very difficult, making it difficult to backtrack to previous viewpoints. In our pilot, participants who had made a mistake often chose to completely reset the scene and start over rather than attempt to reverse their prior interactions. This occurred within all scene types and was not directly connected to the regions themselves.

Figure 3.5. A picture of the testbed running on a Microsoft Surface 4 Tablet.

3.3 Developing a Testbed

Our pilot study helped to identify several challenges, however, our observations were greatly limited in scope. To follow up on these initial observations we developed a testbed for camera controls in map-like environments. We ran the pilot study within Google Earth, which allowed us to make observations about *direct manipulation* camera controls within a popular and contemporary application. However, running the study entirely within Google Earth left us severely limited. We could not make comparisons to other potential camera controls, nor could we modify the existing controls. We were also limited to tasks that could be completed within Google Earth without modification. We could only record what was on screen, or what could be timed by hand. When developing our testbed, We took care to address each of these issues. When designing a new testbed environment, beyond being capable of rendering representations of real-world terrain, it had to be easy to add, or modify, new camera controls, it needed to be possible to support new tasks, and we needed to be able to record the underlying participant actions, not just what was on screen.

We originally created our test environment for Unity 5.6 using C# and later updated it to work in Unity 2017/8 (Figure 3.5). By importing elevation data from Amazon Web Services[2] and satellite imagery from Bing Maps[3] We were able to create large virtual worlds anywhere on earth at ground resolutions between 1:78271 and 1:4.7 pixels per meter (Figure 3.6). Development and iteration of this testbed took place over a focused block of 8-12 weeks.

---

[2] https://aws.amazon.com/public-datasets/terrain/

[3] https://msdn.microsoft.com/en-us/library/bb259689.aspx

By using our own system, we were able to freely implement our own camera controls, as well as tune and modify them as we saw fit. We implemented each camera control as a standalone Unity script that could be easily transferred between Unity projects. Using our testbed, we could develop and test new camera control schemes and experiment with new control schemes and implementations of existing schemes reasonably quickly. Often, we could prototype most camera controls in an afternoon, although many of the more complex controls we implemented took several days to refine and complete. The ability to rapidly prototype and experiment with camera control schemes influenced the final implementation of many camera controls that we did study. We were also free to design our own tasks. The task we gave in the pilot study was very good at making participants use the full range of features of the camera controls. However, task completion was difficult to determine, relied on participant choice and told us more about the tasks then the camera controls. When designing our testbed and our follow-up studies, We took care to choose tasks that had definite completion conditions and would be representative of the kinds of tasks we would expect users to perform in a map-like system. Within our testbed we also had access to the times of every event that occurred within the system. We used that timing information to record camera positions and user interaction events to later visualize or recreate an entire trial.

In Chapter 4, We use our testbed to attempt to alleviate the challenges identified in section 3.2 with the introduction of various look-from camera controls. All the look-from controls address the issue with chaining interactions and disorientation, while some of the introduced controls are designed to be fully reversible as part of the interaction. Others are designed to be simpler.

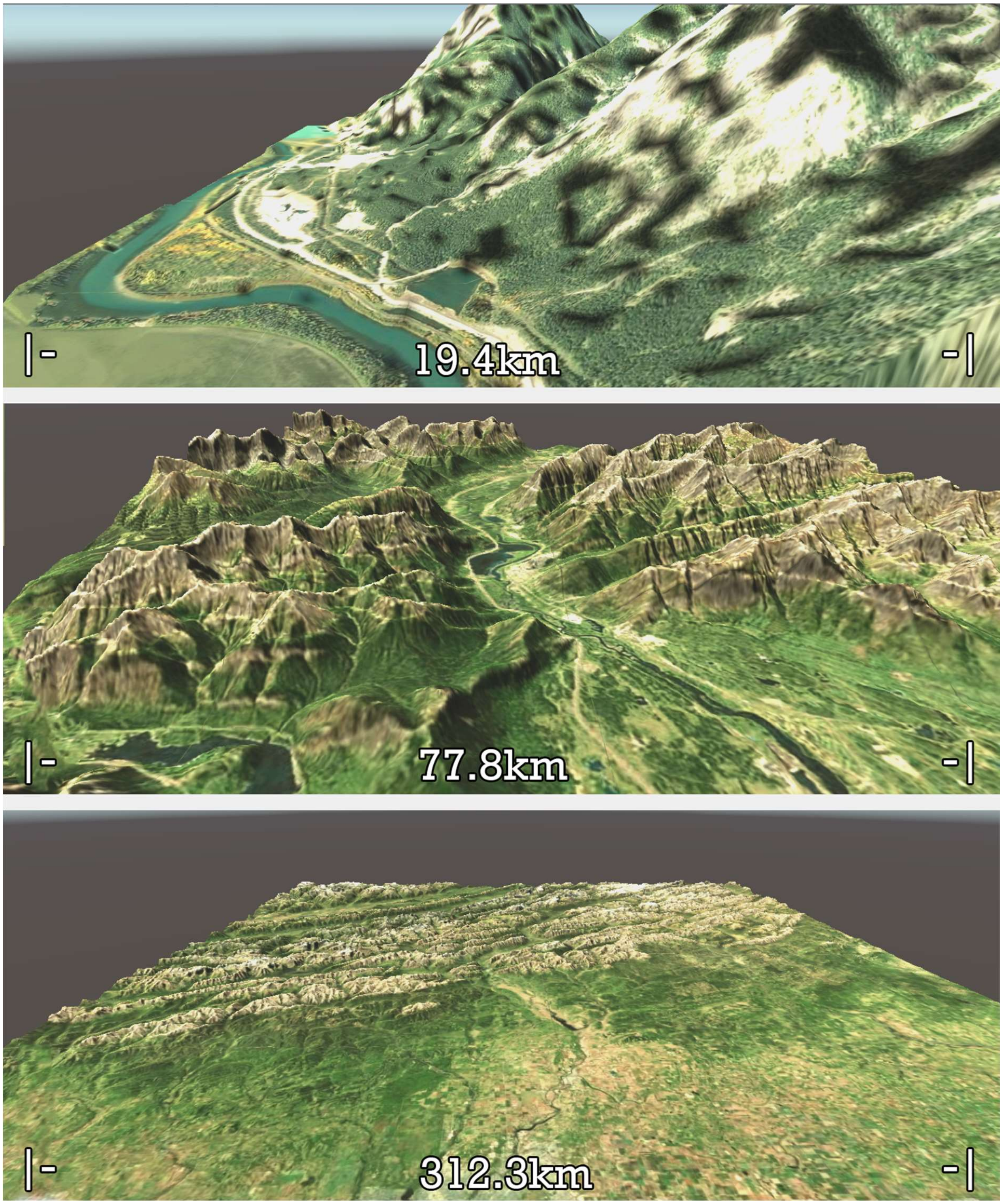


Figure 3.6. Images of the testbed virtual environment at various ground resolutions.

In Chapter 5, we use our testbed to perform a comparative study of navigation techniques designed with VR in mind. Using our testbed, we were able to identify challenges, strengths and weaknesses for a few VR navigation techniques.

Chapter 4 : Look-From Camera Control

In this Chapter, we discuss the design of *look-from* camera controls and a study we performed on those camera controls, the results of that study, and the discussion generated from it. These results help answer RQ1: *Many* a*pplications that rely on touch controls currently use direct manipulation camera control. How can we add to and improve, these touch based systems so that viewers can recreate views and better understand terrain?*

When designing *look-from* camera controls the goal was to create a form of camera control that could address the three issues we identified in Chapter 3: chaining interactions, disorientation, and reversibility. Additionally, we wanted to do so without introducing a new set of issues. *Look-from* camera controls seek to address the three issues identified within Chapter 3, while still retaining compatibility with direct manipulation camera controls.

We begin by describing what *look-from* camera controls are and the design philosophy behind the specific controls we implement. Following that we discuss the technical implementation of each control.

Once we establish the specific *look-from* camera controls used in the study, we discuss the design of the study itself. we describe the test environment, the tasks we asked participants to perform, the measures we used to evaluate the tasks, and the study procedure itself.

We continue by presenting and discussing the results of the study. We first discuss the quantitative results of each task before discussing participant feedback. Finally, we give our observations from the study, discussing what tasks participants found difficult and why.

We close this chapter with a presentation of design considerations for *look-from* camera controls, as well as a discussion of alternative implementations of *look-from* camera controls that we did not explore in this study.

This chapter draws heavily from a collaborative work and is reproduced here with permission.

Danyluk, Kurtis Thorvald, Bernhard Jenny, and Wesley Willett. 2019. "Look-From Camera Control for 3D Terrain Maps." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 10. https://doi.org/10.1145/3290605.3300594. (In Press)

4.1 Look-From Camera Control

Look-from camera controls are an alternative way for a viewer to control their camera. User-defined ***look-from*** and ***look-at*** locations constrain the space of possible camera paths that are relevant to the viewer. Given these constraints, a map application can animate the camera directly to the desired view, seamlessly panning, zooming, rotating, and pitching the camera.

Reframing camera control as a *look-from-at* interaction also has the potential to considerably simplify viewer interaction, reducing the sequence of 4+ independent pan, zoom, rotate, and pitch interactions necessary to create a ground-level view down to just two selection operations. Moreover, selecting these *look-from* and *look-at* points can be accomplished easily in a single gesture by mapping them to *touch-down* and *touch-up* interactions respectively. This kind of straightforward *look-from-at* interaction also remains compatible with direct manipulation operations, since viewers can easily use additional pan, zoom, rotate, and pitch interactions to adjust the resulting view.

Unlike direct manipulation operations, which allow viewers to continuously refine and reverse an action, this rudimentary *look-from-at* interaction provides little opportunity for adjustment during the interaction itself. However, by relaxing the requirement for the *look-at* point, we can create a variety of alternative *look-from* interactions which increase the interactivity of the camera control possible within a single touch.

Card (1991), Jacob (1994) and Mackinlay (1990) describe many different ways to map user input controls, however we limit our examination to *look-from* camera control techniques that can be executed using single-touch gestures. This constraint makes it possible for any of the techniques to be executed as a *quasi-mode* (Raskin 2000) initiated via a long press, double-tap, or other explicit interaction. As a result, these *look-from* techniques can be included in environments that also support traditional direct-manipulation camera controls---allowing the two approaches to complement one another.

Additionally, we follow the work of Christianson et al. (1996), which examines declarative camera controls from the perspective of film, to help formulate how *look-from* camera controls should act to best engage a user, motivating some of the details of our implementation.

We explore three distinct look-from variants (*discrete look-from-at*, *continuous look-from-forward*, and *continuous look-from-toward*) which showcase the diversity of camera manipulations possible in a single gesture.

*Discrete Look-From-**At*** (Table 4.1—top) is a straightforward and simple execution of the *look-from-at* metaphor. A viewer selects *look-from* and *look-at* locations on the surface of the terrain. From those points, the system generates an automated camera transition which pans, zooms, rotates, and pitches the camera to position it just behind the *look-from* location, pointed towards the *look-at* location.

Continuous Look From ***Forward*** (Table 4.1—center) generalizes the *look-from-at* pattern by assuming that the *look-at* location is always located in the direction the map is facing, in front of the *look-from* location. This frees the viewer to specify the *look-from* location on touch-down, and then use the rest of the gesture to dynamically move the camera along a path between its initial position and the *look-from* point. Sliding upward advances the camera further along the path---adjusting its position, zoom, and pitch as it progresses. Sliding back reverses that action. This allows the viewer to dynamically refine the camera view, making it possible to smoothly navigate to oblique views above the selected location as well as to locations at ground level.

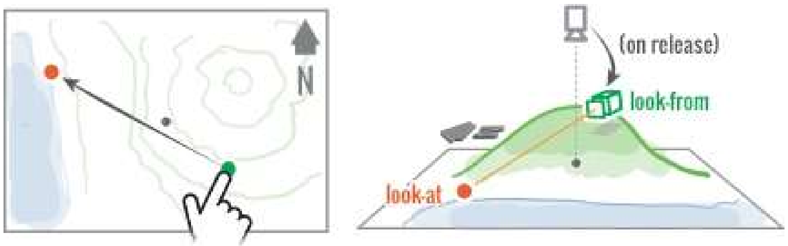
Because the technique is compatible with existing direct manipulation gestures, such as two-finger rotation, viewers can use these to rotate the map either before or after a *look-from-forwards* interaction.

*Continuous Look-From-**Towards*** (Table 4.1—bottom) increases the expressiveness (but also the complexity) of *look-from-forward* by allowing viewers to interactively rotate the map and advance the camera simultaneously. As with the previous techniques, the viewer specifies the *look-from* location on touch-down. Dragging away from the initial touch point in any direction advances the camera towards the *look-from* location, changing its position, zoom, and pitch. Meanwhile, the camera path is rotated based on the on-screen angle between the vector formed by the viewer's initial touch point and the current touch location and a vector pointing straight up from the initial touch point. This greatly increases the number of possible ground-level and oblique views that can be reached in a single interaction. However, it also increases the potential for disorientation as viewers control their camera relative to the starting camera position, not the current camera position, and they are free to break the line of interest (Christianson et al. 1996) — a line parallel to the viewer that passes through the focus of their view. By default, we automatically transition the camera back to its initial view at the end of an interaction to allow for quicker exploration.

<u>Discrete Look-From-At</u>

Touch **down** → specify *look-from* location

Touch **up** → specify *look-at* location
initiate animated transition

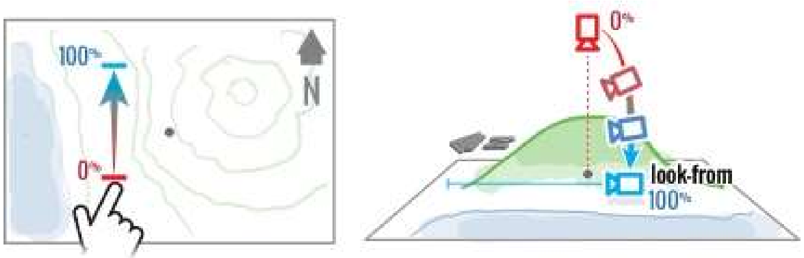


<u>Continuous Look-From-Forwards</u>

Touch **down** → specify *look-from* location

Touch **move** → advance/reverse interactive transition
[0%↔100%] = distance from touch down

Touch **up** → stop transition



<u>Continuous Look-From-Towards</u>

Touch **down** → specify *look-from* location

Touch **move** → advance/reverse interactive transition
[0%↔100%] = distance from touch down
→ rotate camera path
[0º↔360º] = angle relative to touch down

Touch **up** → animated transition back to original view

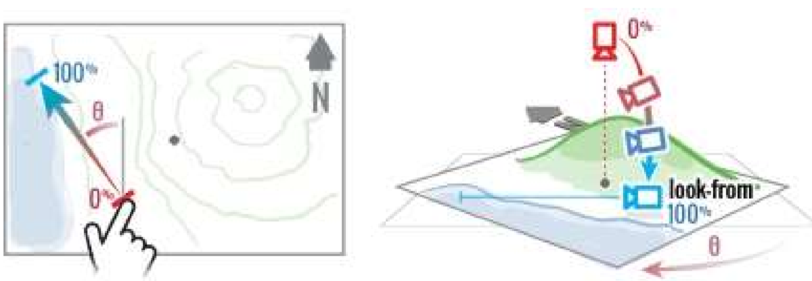


Table 4.1. Input-response mappings for the three *look-from* techniques we consider.

4.1.1 Implementation

To explore the space of possible single-touch *look-from* interactions, we iteratively implemented and examined instances of each of these techniques as well as numerous hybrids and variants. The prototype supports multitouch *direct manipulation* interactions modeled on those used in the mobile versions of Google Earth, including two-finger gestures for zoom, rotate, and pitch. A viewer performs pan with a one finger sliding gesture, zoom with a two-finger pinch to zoom gesture, rotate with a two-finger twisting gesture and pitch with a two-finger gesture where viewers slide both fingers up to pitch the camera up, and both fingers down to pitch down.

Discrete Look-From-At (Table 4.1–top) is implemented as a single discrete transition that begins with a touch-down event. Using that touch point we cast a ray from the camera to find the look-from point of the interaction. The interaction ends with a touch-up event. As with the touch-down, we cast a ray from the camera to find where that touch-up occurred in world space and mark it as the look-at point. Once those points are identified we begin the camera transition. If we cannot cast an unobstructed line between the look-from and look-at points, we raise the look-from and look-at positions at a rate of 2 to 1 until an unobstructed line can be cast between them. This improves the visibility of the camera in the direction of the look-at point while keeping the camera position relatively close to the selected look-from point. The camera is then smoothly animated using a linear interpolation function such that (a) its position is equal to the adjusted look-from point and (b) it is facing the adjusted look-at point.

Continuous Look-From-Forward (Table 4.1–center) is a continuous interaction technique that begins with a touch-down event. Using that touch point we cast a ray from the camera to find the look-from point of the interaction. To find the look-at point in front of the look-from point we then cast another ray from the camera at an angle of 20° above the first ray. At the end of the transition, the camera is located at the look-from position and looking at the look-at location. If the second ray fails to intersect the terrain (for example if 20° overshoots the edge of the map or the horizon) we reduce the angle in steps of 1° until an intersection is found.

The interaction continues with a touch move event. The vertical distance in screen space between the starting point and the current touch point is measured. That distance is then mapped to a range between 0 and 1. If the current point is more than 50% of the screen away from the starting point then it is mapped to 1, if it is 0% away then it is mapped to 0. Between 0% and 50% we map the distance to a value between 0 and 1 using a smooth-step function. We then linearly interpolate the camera position between the starting position and end position using the mapped distance. We also spherically linearly interpolate the camera rotation between the start and end rotation by the square of the mapped distance. Finally, the camera position is moved back slightly along the vector between the look-from and look-at point, keeping the look-from point in view to reduce viewer confusion. While the finger remains on the screen this interaction can be advanced or reversed by moving further or closer to the starting point, respectively. The interaction ends with a touch-up event and the camera remains in its last position.

Continuous Look-From-Towards (Table 4.1–bottom) is a continuous interaction technique that works very similarly to Continuous Look-From-Forward. To do this we try to first rotate the camera’s starting position to replicate the orientation that we assume in a Look-From-Forward interaction scheme. The viewer interaction begins with a touch-down event. Using that touch point we cast a ray from the camera to where that touch was in world space, this defines the look-from location.

The interaction continues with a touch move event. We cast a ray from the near plane of the starting camera frustum from the moved touch point to find our look-at position. This sets look-from and look-at points such that the projection of the vector between them on the near plane of the starting camera is the same as the vector between the starting and moved touch points. Using that vector, we immediately rotate the camera's starting position such that that vector is up. This avoids camera rolls and odd camera transitions where the horizon appears to be skewed, which can confuse viewers. We then measure the Euclidean distance in screen space between the starting point and the current touch point and use it to interpolate the camera position and rotation using exactly the same method as for Continuous Look-From-Forward.

While the finger remains on the screen this interaction can be advanced or reversed by moving further from or closer to the starting point, respectively. At any point the viewer can lock the camera position by using another finger to tap a lock icon located at the starting point. The viewer interaction ends with a touch-up event. Unless the camera position was locked, the camera animates back to the starting location and rotation. If locked, the camera remains in place and the viewer can continue to adjust the view using direct manipulation or additional Continuous Look-From-Torward interactions.

4.2 Study

To explore the use of these three *look-from* variations we conducted a within-subjects lab study in which we asked participants to perform two common and difficult map reading tasks, elevation comparison and line-of-sight assessment, across four different types of terrain. In addition, we asked participants to provide feedback about the difficulty of using the techniques and about which techniques they enjoyed using. Our goal was to understand how the addition of *look-from* controls changed participants' navigation strategies and examine any impact these interaction techniques had on participants' speed, accuracy, and overall experience.

4.2.1 Test Environment

We conducted the experiment using a Microsoft Surface 4 tablet using the testbed described in Chapter 3.3 Developing a Testbed. Participants interacted using the touch screen with no keyboard or other peripherals attached. Using our interactive prototype, we rendered four large virtual environments (Figure 4.1) at ground resolutions between 1:121 and 1:32 pixels per meter. We chose these locations to represent a variety of environments and scales including: mountainous terrain (a 124km^2 section of the Canadian Rocky Mountains centered on Canmore, Alberta), rolling flat-topped peaks (a 32km^2 region around Mount Wutai in China's Shanxi Province), a volcanic plateau (a 64km^2 region surrounding Mount Waialeale on the Hawaiian island of Kaua'i), and deep canyons (a 32km^2 region along the Grand Canyon in northern Arizona).

4.2.2 Tasks

We asked participants to perform two types of tasks: elevation comparison and line-of-sight. In both tasks we showed participants two points on the map, one marked in red and one marked in blue. These tasks have been used to compare the effectiveness of maps and map tools in both early (Phillips, Lucia, and Skelton 1975; Potash, Farrell, and Jeffrey 1978) and recent studies (Willett et al. 2015; Li et al. 2017; Eynard and Jenny 2016). We generated these tasks randomly prior to the study.

Elevation comparison tasks required the participant to answer whether the red point or blue point was at a higher elevation. The randomly generated points were always at least 8% of the map height apart and at most 14%. In addition, we removed any trivially easy trials where one of the points fell in an open flat area such as a plateau, lake, or ocean. Participants responded to the prompt “Which Point is Higher?” by pressing buttons at the upper left-hand corner of the screen labeled “Blue” or “Red”.

Line-of-sight tasks required the participant to answer if there was a clear line of sight between one point to the other. We generated line-of-sight tasks by placing a point to the left and right of the red and blue point and performing line casts between all red points to blue points. If at least two of the line casts, and at most 6, were obstructed, we considered the trial to be sufficiently challenging. When choosing trials, we selected exactly half where there was clear line of sight and half where line of sight was obstructed. Participants responded to the prompt “Can Blue see Red?” by pressing “Yes” or “No” buttons at the upper left-hand corner of the screen.

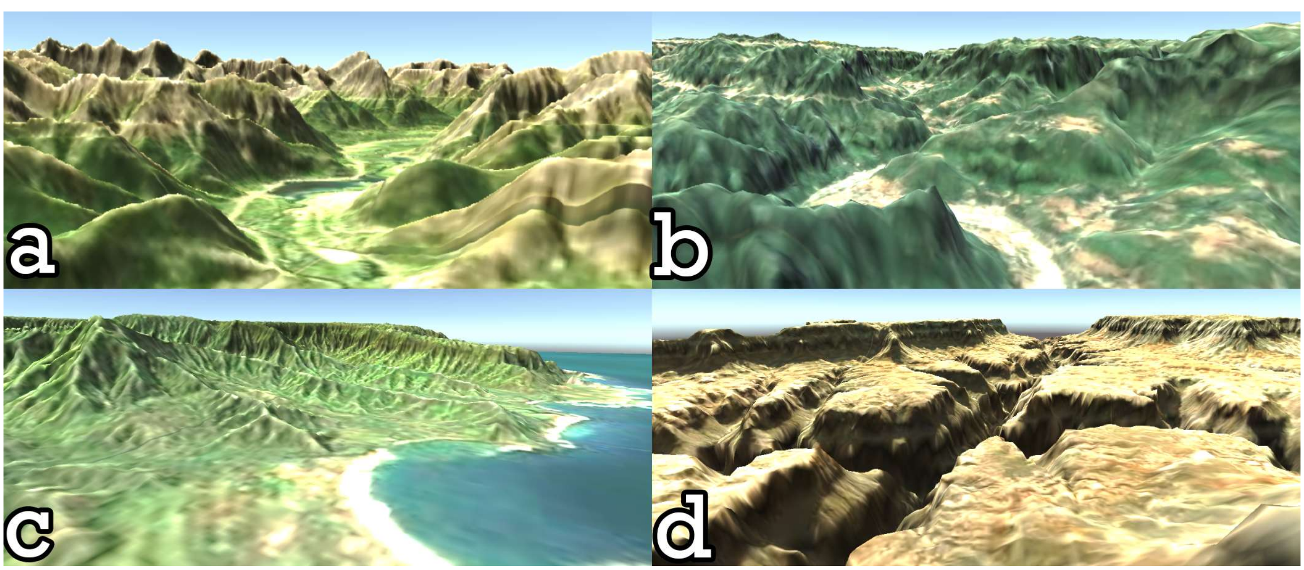


Figure 4.1. Maps used in user study: (a) Rocky Mountains, (b) Mount Wutai, (c) Mount Waialeale, (d) Grand Canyon

4.2.3 Measures

During the trials we recorded participant accuracy in completing the tasks and the time it took to complete each task. After the trials we asked participants about which technique they found easiest, how difficult they found each technique, which technique they enjoyed the most, and how enjoyable each technique was.

We measured **Accuracy** as either correct or incorrect based on the task. We asked participants to focus on accuracy when completing trials and did not expect there to be any notable difference between technique accuracy. This is because given enough time it is possible for every technique to be equally accurate.

We recorded **Time** from the first touch event in a trial until the participant answered the question. Because we asked participants to focus on accuracy, we expected that any noticeable differences between techniques would be reflected in differences of time.

We measured **Perceived difficulty** in two ways. First, we asked participants to select which technique they found the easiest. Second, we asked participants to rate on a 1–5 Likert scale how difficult they found each technique.

We measured **Perceived enjoyability** similarly. First, we asked participants to select which technique they found the most enjoyable. Second, we asked participants to rate on a 1–5 Likert scale how enjoyable they found each technique.

4.2.3 Participants

We recruited 16 participants aged between 21 and 36 (6 male, 10 female). Thirteen reported experience using digital map software such as Google Maps. Five reported experience with paper maps, such as hiking trail maps. Two reported no experience with maps or mapping software. Participants were recruited with a combination of university email lists, word of mouth, and snowball sampling.

4.2.4 Procedure

Prior to the study we asked participants for simple demographic information and their familiarity with maps and cartography. During the study, we compared four camera control techniques against each other: *direct manipulation, discrete look-from-at, continuous look-from-forward,* and *continuous look-from-towards*. In the case of each *look-from* technique, participants also had access to *direct manipulation* controls.

We asked each participant to complete 64 total trials, including 32 elevation comparison and 32 line-of-sight. We administered trials in alternating order (elevation comparison, then line-of-sight) and changed the scene in sequence after each line-of-sight (Canmore, Mount Wutai, Mount Waialeale, Grand Canyon). At the beginning of each trial we reset the participant's camera to the initial top-down position. Each block contained 8 trials and consisted of one full sequence of locations: in total there were 8 blocks. We gave participants identical blocks of trials in the same order. Within each block we gave participants access to one technique, the order of which was permuted using a Latin square.

Additionally, prior to starting any study blocks we administered 3 training trials. In the first training trial we demonstrated how to use each camera control technique and asked them to practice using each technique. In the second training trial we demonstrated to participants a sample elevation comparison tasks and taught them how to use the study interface. In the third training trial we demonstrated to participants a sample line-of-sight task and again taught them how to use the study interface.

After completing all blocks, we debriefed the participant and asked them to complete a follow up questionnaire in which we asked about their perceived difficulty and enjoyability of each technique. We also asked for a short response detailing what they found easy/difficult and enjoyable/least enjoyable about each technique. On average the whole procedure took one hour to complete.

4.3 Results

Due to concern in a variety of fields about the use of null hypothesis significance testing (Cumming 2014; Dragicevic, Chevalier, and Huot 2014), we report our results using estimation techniques and report effect sizes using confidence intervals (CI) as is consistent with recent APA recommendations (*Publication Manual of the American Psychological Association* 2017). To do so we compute average scores for each participant, then compute 95% confidence intervals using the aggregate scores, applying a Bonferroni correction, when appropriate, to control for multiple comparisons.

4.3.1 Accuracy

Mean accuracy values (reported as accuracy over random) across all conditions were in the range of 30-40% — 50% is perfect accuracy. The most accurate was *discrete look-from-at,* with a mean accuracy of 38% (CI= [32%,45%]) in elevation comparison trials and 37% (CI= [26%,46%]) in line-of-sight trials. The least accurate was *continuous look-from-towards,* with a mean accuracy of 30% (CI= [21%,38%]) in elevation comparisons and *direct*

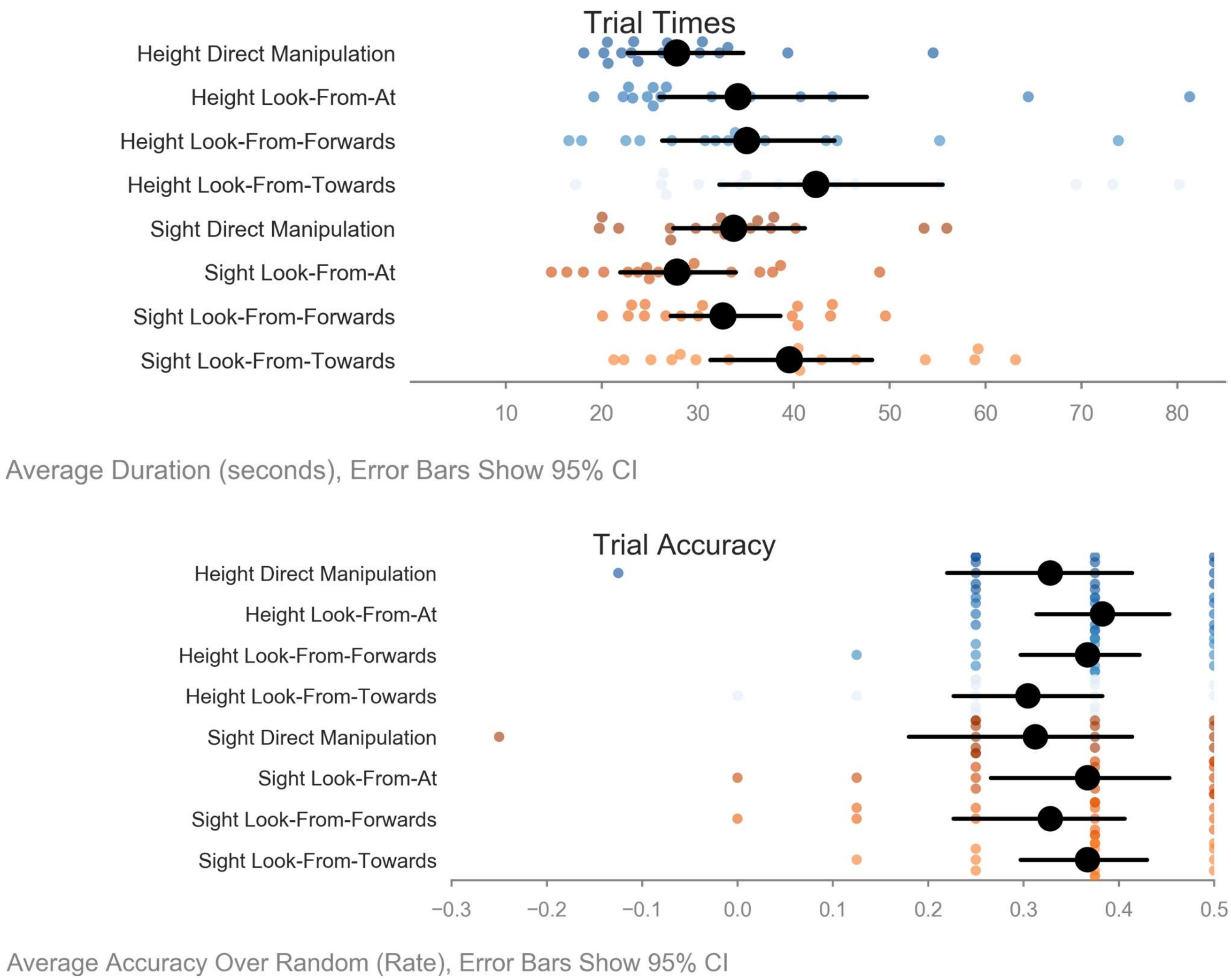


Figure 4.2. Average trial time and accuracy per participant. error bars show 95% CI

manipulation with a mean accuracy of 31% (CI = [16%,41%]). However, variation across participants was high and overall results showed no conclusive difference in accuracy between conditions (Figure 4.2–top).

4.3.2 Time

Mean times were in the range of 27s to 40s. The mean fastest technique was *direct manipulation,* with a mean time of 27.6s (CI = [22.7,34.8]) for elevation comparison trials and *discrete look-*

from-at, with a mean time of 27.8s (CI = [22.3,34.2]) for line-of-sight trials. The slowest technique was *continuous look-from-towards,* with a mean time of 42.2s (CI = [31.3,55.3]) for elevation comparisons and 39.5s (CI = [30.9,49.0]) for line-of-sight trials. However, much like in accuracy there was a large degree of variation across participants and overall results showed no conclusive difference in speed between conditions (Figure 4.2–bottom).

4.3.3 Difficulty and Enjoyability

While no one of the three look-from techniques we tested was unambiguously superior to the others in terms of speed or error rate, our results point to differences related to perceived difficulty and enjoyability. Participants showed a strong and conclusive preference for the simpler look-from technique, and also showed a readiness to integrate all three techniques with direct manipulation interactions.

Directly reflecting the Likert responses (Figure 4.3) 10 of 16 participants responded that *discrete look-from-at* was the easiest technique to use, 3 responded *direct manipulation*, 2 *continuous look-from-forward,* and just 1 *continuous look-from-towards*. Closely reflecting difficulty, 9 of 16 participants ranked *discrete look-from-at* as the most enjoyable control scheme, 3 ranked *continuous look-from-forward* and *continuous look-from-towards* as their favorite, and only 1 ranked *direct manipulation* as their favorite technique to use.

4.3.3.1 Direct Manipulation

The most common positive comment of *direct manipulation* was that participants found the controls familiar (16 participants). While other comments noted specific functions (pan, zoom, or rotate) felt great to use (3 participants). Additionally, P4 noted that it gave them very “Fine

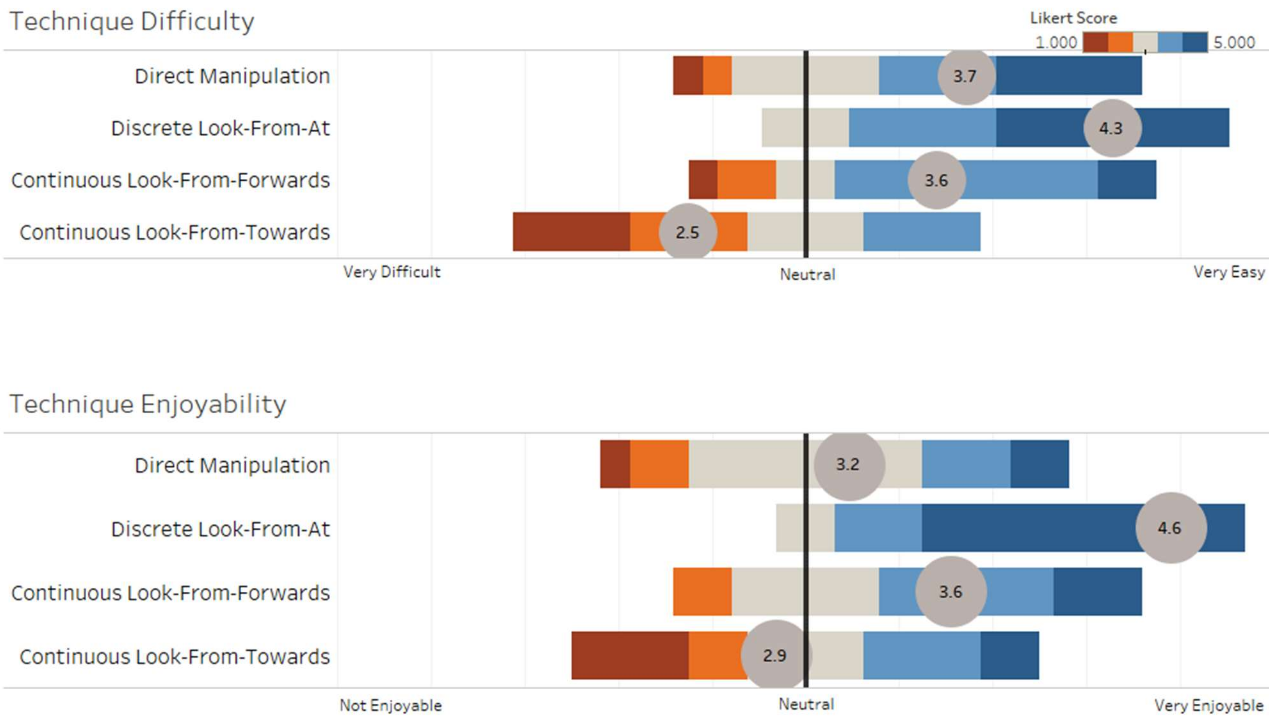


Figure 4.3. Likert survey results. Circled values indicate average Likert score. Staggered bars show stacked responses offset with positive responses to the right of the reference line and negative to the left

granularity to control things”. Negative comments noted that *direct manipulation* was “reliable but slow”, required multiple steps (4 participants) was “tedious” (2 participants), or wasn’t precise enough (2 participants).

4.3.3.2 Discrete Look-From-At

Participants positively noted that *discrete look-from-at* was effective at very quickly zooming in to target destinations (P6) while rotating them to where they wanted to look all in one interaction (3 participants). This caused the interaction to feel easy and fast to use (8 participants). They also liked that it felt “perfectly suited” for line-of-sight tasks (2 participants) and that they could quickly flip their view (2 participants). They also noted that the results of their actions were always as expected (5 participants). Negative comments noted that it could take a few interactions to get used to (P7) and that they initially found it disorienting (P2).

4.3.3.3 Continuous Look-From-Forward

Participants positively noted that *continuous look-from-forward* was simple to use (7 participants) and that it worked as a straightforward alternative to pitch (3 participants) and zoom (3 participants). But negative comments noted that it required a lot of actions like *direct manipulation* and wasn't much different from it (8 participants).

4.3.3.4 Continuous Look-From-Towards

Participants positively noted that *continuous look-from-towards* had great control of the six different axis of motion (P11), made easier by a fixed point of rotation (P16). Participants also found it felt "very powerful" (6 participants) and "responsive" (3 participants). But negative comments overwhelmingly noted that the rotational controls could be very confusing or disorienting (14 participants).

4.4 Observations

When completing trials, the most important factor in determining success at the tasks was how well the participant could find and frame both points into their view and maintain that framing. Especially during the process of changing their viewing perspective to one in which they could discern line-of-sight or elevation. A participant successfully framed both points when both could be seen clearly from their current camera position.

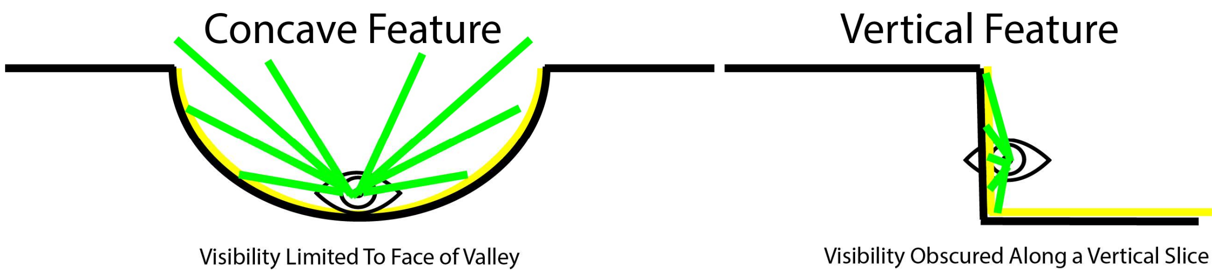


Figure 4.4. Diagram of a concave and vertical feature. Yellow Shows viewer viewshed. Green Lines show obstructed sight lines.

4.4.1 Navigating Concave Features

Concave terrain such as bowls or valleys can pose navigation problems when the camera comes close to the surface of the terrain. Once the camera dips into a concave region, the sides of the feature tend to obscure the surrounding terrain, hiding landmarks and making it difficult for viewers to maintain a clear frame of reference.

In our study, participants most often became trapped by concave features in the Direct Manipulation condition, either by zooming in too quickly or by dramatically pitching the camera down into a valley. With Discrete Look-From-At controls, participants sometimes encountered similar issues when choosing a look-from location deep in a valley. Once they had entered a concave region, the terrain often obscured one or both of the target points and participants generally found it easier to reset the camera back to a top-down view than to attempt to backtrack and locate them. This occurred most frequently in the Rocky Mountains and Mount Waialeale environments where the terrain had tight concave features that were not readily apparent from above.

Some participants learned to avoid entering concave features in the Direct Manipulation condition by pitching the camera slightly before zooming in, mimicking the behavior of

Continuous Look-From-Towards/Forward. This initial camera tilt made it easier for those participants to identify regions that could potentially trap the camera before dropping into them. In the Continuous Look-From-Towards/Forward conditions, meanwhile, participants were typically able to escape concave regions by interactively reversing the transition once they noticed an occlusion.

4.4.2 Navigating Vertical Features

Participants also experienced problems when the target points were placed on the sides of vertical terrain features like cliff faces or canyon walls. In these cases, the vertical features often obscured the target points from most directions, greatly reducing the number camera positions from which a viewer could see both points. This occurred most frequently in the Grand Canyon, where points along the canyon walls were often only visible from above or from a small number of locations within the canyon itself. It also occurred in a few places on Mount Wutai where points along the edge of a plateau could only be seen if the camera was positioned on that side of the feature.

Navigating along these vertical features (especially those in the canyon) was difficult in all conditions. However, it was particularly challenging with Direct Manipulation controls, where positioning and angling the camera to see both points simultaneously required considerable manual dexterity and planning. The three look-from techniques somewhat mitigated this challenge, since participants could typically ensure that at least one of the points remained continuously in view by choosing a look-from location immediately adjacent to it. The Continuous Look-From-Towards/Forward techniques also allowed participants to refine and reverse the camera movement and make corrections whenever the vertical feature obscured the second point.

4.5 Design Considerations

Based on participants' feedback, as well as our own experience iteratively designing and testing numerous different *look-from* techniques (including the three presented in detail here), we provide several considerations for the design of future look-from techniques for touch maps on mobile devices.

D1. Keep *look-from* points in frame during transitions. In general, we find that maintaining the visibility of the original selection throughout transitions preserves a common reference point and helps reduce viewers' sense of disorientation. In our implementations, we achieve this by ensuring that the final camera position falls slightly behind the *look-from* point, allowing that point to stay in view even at the end of the transition.

When a transition involves a specific *look-at* point, keeping that location visible is also beneficial. In cases where terrain blocks the line of sight between the *look-from* and *look-at* points, we find that shifting the final camera position both *upward* and *backward* can help maintain the visibility of both. Not including this sort of adjustment often results in views where the *look-at* position is entirely occluded.

D2. Sequence camera rotation and pitch to minimize roll. Simultaneously rotating and pitching the camera can result in visible roll where the camera tilts and the horizon appears diagonal, and few points from the initial view remain in frame. This was apparent in our initial explorations of *look-from-towards*, where pilot testers complained that they had a hard time maintaining a clear frame of reference, especially when performing rotations greater than ±90º.

Staging rotations at the beginning of a transition minimizes this issue by aligning the horizon and look-at direction before pitching so that the camera never appears tilted. This makes it possible for pitch to be smoothly interpolated throughout the animation or adjusted to ensure that the *look-from* location always remains in frame.

D3. Make *look-from* interactions reversible if possible. The inability to reverse after completing a camera move was a major shortcoming of both our *look-from-at* and *look-from-towards* implementations, both of which require viewers to use *direct manipulation* interactions or a hard reset to return to a top-down view. Reversible *look-from* interactions, meanwhile, can make it easier to recover from mistakes and support more seamless transitions between top-down, oblique, and ground-level views in both directions.

D4. Preserve compatibility with existing direct manipulation techniques where possible. While look-from techniques can streamline common transitions between top-down, oblique, and ground-level views, they are less suited to search and navigation tasks that require repeated zooming and panning. Moreover, *direct manipulation* interactions can be very useful for making small adjustments to views produced using *look-from* interactions, allowing viewers to independently pan, zoom, or pitch the camera after transitioning down to an oblique view or rotating around a ground-level point of interest.

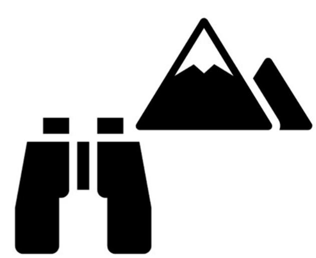
Based on our experience, adding *look-from* navigation as quasi-modes initiated via a distinct interaction such as a long-press or double-tap facilitates these transitions well – allowing viewers to quickly enter and exit a look-from navigation mode while retaining access to the standard vocabulary of map interactions. However, the vocabulary of simple gestures is relatively limited, and many mapping tools and operating systems already assign functions to gestures like double-taps (often mapped to *zoom*) or long presses (often used to trigger context menus). This limitation may make it difficult to add quasi-modal *look-from* gestures to some existing systems without relying on additional hardware inputs or interface elements. Moreover, it may make it difficult to support multiple different kinds of *look-from* gestures simultaneously in an interface.

4.6 Improving Discoverability of Look-From Camera Controls

L*ook-from* camera controls implemented as quasi modes on top of existing camera controls may have limited discoverability. We propose two methods to improve their discoverability.

Suggested Views. By pre-placing *look-from* and *look-at* points at interesting locations in the environment viewers can perform a *look-from* interaction through these pre-placed points. To perform this interaction the viewer would simply have to select a *look-from* point that is featured on the map, this will cause *look-at* points to be highlighted nearby, selecting that will cause a *look-from* camera transition. This will suggest to viewers how to effectively use *look-from* camera controls and provide an opportunity to explain how the viewer can perform these interactions at arbitrary points within the scene.

Interactive UI Elements. By providing interactive UI elements, *look-from* interactions can be suggested to a viewer through the interface. For example, a pair of binoculars and a mountain on the viewers' toolbar that are placed next to each other. The viewer could then drag the binoculars from the UI to choose a *look-from* location within the scene. The system would then prompt the viewer to to drag the mountain to choose a *look-at* location.



4.7 Alternative Control Schemes

While we considered just three examples of *look-from* techniques for digital terrain maps, variants of these approaches may also merit further examination – especially approaches that relax the requirement of using a single touch gesture. For example, using multiple touches (either sequentially or bimanually) could permit continuous versions of *look-from-at*, where one touch sets the *look-from* point, and a second touch gesture sets the *look-at* location and interactively advances the camera.

There also exists a rich space of alternative input schemes for more complex *look-from* techniques, including alternatives to our version of *continuous look-from-towards.* In our implementation, using the angle between the cursor and the initial touch-down to control rotation often resulted in confusion, since rotations greater than 90º inverted viewers' frame of reference. Once at ground level, viewers often expected to be able to rotate right or left by dragging in the corresponding direction, rather than circling the original point clockwise or counter-clockwise. A cartesian input scheme that used forward/backward movement to move the camera along the path and left/right motion to control rotation might align more closely with common mental models of camera movement.

Finally, although we considered input schemes that rely only on finger position, commodity touch-enabled devices generally also capture touch pressure and information that can be used to reliably estimate finger pitch and pose (Schwarz et al. 2015). These additional input factors could support even more nuanced transitions – for example, by using finger pitch or pressure to dynamically vary the vertical height of the camera path.

4.8 Chapter Conclusion

Look-from camera control techniques can serve as lightweight additions to traditional *direct manipulation* controls, allowing viewers to simplify common and challenging map navigation tasks using simple gestures. By integrating camera zoom, pan, and pitch and building camera paths based on one or two viewer-specified points on the map, these techniques can support smooth transitions between top-down, oblique, and ground-level views.

Chapter 5 : Navigating Maps in VR

In the previous chapter we were interested in designing camera controls for applications that use touch controls and that could address the issues we identified in Chapter 3. However, the space of interactive camera controls is a broad one that includes a much wider diversity of devices and ecosystems then just 2D touch devices. In this chapter we examine another digital map ecosystem, immersive virtual reality. We present the design and results of two studies that compared the effectiveness of four common navigation techniques for VR environments in a large scale open virtual environment that represented real-world terrain. This evaluation focused on existing navigation techniques, as opposed to generating new ones. Much like the cooperative relationship between *direct manipulation* and *look-from* camera controls shown in Chapter 4, in this Chapter we demonstrate how navigation techniques in VR can be combined to improve the viewer's experience.

We begin with a description of the four navigation techniques — flight, teleportation, world-in-miniature, and 3D cone drag. we first describe how each technique works, we then follow by describing how we implemented our own versions of the techniques.

Afterwards, we discuss the overall design of the study. we provide study motivation, and describe the test environment, study tasks, and measures we used to evaluate the results.

We continue by recounting the specific participants and procedure for the first study, and we present the results of the study. We then do likewise for the second stage of the study.

Once we have presented the quantitative results of each stage of the study, we provide a discussion of the entire study in aggregate. we discuss participant feedback on the individual navigation techniques before providing detail on feedback centered around cybersickness. Based on this feedback and our observations we discuss the strengths and weaknesses of each navigation technique, detailing the when each technique is most appropriately employed.

Continuing with these strengths and weaknesses in mind, we present how users combined the navigation techniques together to overcome the weaknesses of each and more effectively perform the study tasks.

We finish with a discussion of what makes for good combinations of navigation techniques and a statement motivating combining navigation techniques in large scale open virtual environments.

We use the results of the studies and the following discussion to answer RQ2: *Immersive virtual reality systems have several options for camera control. Given many options, which option(s) are most appropriate for representations of real-world terrain?*

This chapter draws heavily from a collaborative and content is reproduced here with permission.

Danyluk, Kurtis and Wesley Willett. 2018. “Evaluating the Performance of Navigation Techniques in Large Scale Open Virtual Environments.” University of Calgary. (In Submission)

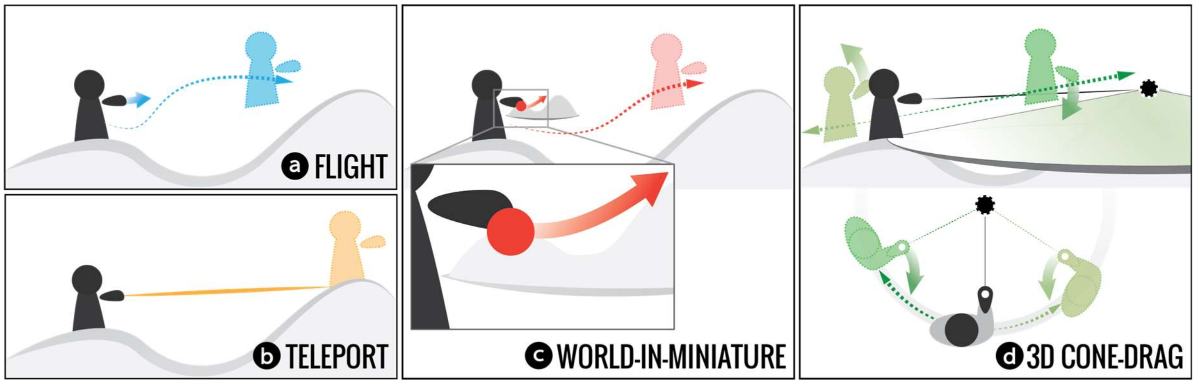


Figure 5.1. The four different VR navigation techniques:(a) flight, (b) teleport, (c) world-in-miniature, (d) 3D cone drag.

5.1 Navigation Techniques for Large VR Spaces

The design space of navigation techniques for virtual reality includes both *continuous* navigation techniques in which users move smoothly through their environment, and *discrete* techniques in which they move instantaneously between two discrete locations. Large VR spaces are a challenge because their sheer size rules out familiar navigation techniques like locomotion (physically walking), which most closely resemble navigation in the physical world. In large spaces, origins and destinations may be far apart, and viewers need mechanisms for transitioning these distances quickly yet precisely. While a variety of navigation techniques for navigating large spaces exist in the research literature and in commercial applications, the trade-offs between them remain poorly understood. Here, we introduce four common yet contrasting techniques we chose to compare in our studies and describe how we implemented our own versions of each technique.

5.1.1 Continuous navigation techniques

Flight is one of the earliest and most prevalent navigation techniques for VR (Ware and Osborne 1990). When using this technique, a viewer moves through virtual environments as if they were controlling a flying vehicle, like a helicopter (Figure 5.1a). *Flight* involves a smooth, predictable, and reversible translation to a position immediately adjacent to the starting position.

In our implementation, the viewer controls their motion by pointing a handheld controller in the desired travel direction and then pressing forward/back on a directional pad to move in that direction. Motion stops immediately as soon as the button is released. We added a small acceleration (acceleration=0.6m/s^2, Velocity$_{initial}$=0.1m/s, Velocity$_{final}$=120m/s) to flight to help travel more quickly across the scene. Because the motion is continuous, viewers can reorient the controller during *flight* to fluidly adjust their direction of travel.

World-in-miniature allows the viewer to change their position by manipulating a virtual version of themselves in a miniature version of the environment (Figure 5.1c). For example, using *world-in-miniature*, a viewer can navigate to a nearby mountain by grabbing a virtual token representing their position in a miniature version of the virtual world, and moving that token to the top of the miniature version of the peak. This miniature copy of the larger environment provides an overview which complements the viewer's local perspective, allowing them to quickly move to regions outside their immediate field of view. Moreover, because of the scale differences between the two representations, viewers can use small motions on the miniature world to cover large distances very quickly.

Our implementation (Figure 5.2) uses purely virtual representations of the *world-in-miniature* components. The miniature world is centered on a 1m diameter mini-map, located at the center of the viewer's physical play area. While early implementations of *world-in-miniature* (Stoakley, Conway, and Pausch 1995; Pausch et al. 1995) used a virtual token to control both the position and orientation of the viewer's camera, the token in our implementation only translates the viewer's play area within the larger virtual world. In our system, moving the token shifts the play area in real time and allows viewers to continue to control the camera orientation using their head position and walk freely around the play area as it moves, this is similar to the system presented by Pausch et al. (1995). We perform velocity filtering using a One Euro Filter (Casiez, Roussel, and Vogel 2012) to smooth user inputs and help mitigate the risk of cybersickness, especially when performing detailed movements.

3D Cone Drag allows the viewer to directly manipulate the scene as though they were directly grabbing and dragging the landscape (Figure 5.1d). *3D cone drag* translates the direct manipulation metaphor commonly used on touch screen devices to work in VR. This inverts the traditional navigation metaphor, allowing viewers to move the environment around them rather than changing their position relative to it. This is similar to *look-from* camera control in that the viewer chooses a grab point, or *look-from* point, and manipulates their position relative to that point. *While* not widely used in past research, we chose to examine *3D Cone Drag* because of its prominent use in Google Earth VR, a popular commercial application which involves exactly the type of large scene navigation tasks that we are interested in exploring.

We based our implementation of 3D Cone Drag on the approach used in Google Earth VR (Käser et al. 2017) which is itself an extension of the approach described by Mapes and Moshell (1995). To navigate using *3D cone drag*, a viewer points towards a location in the scene and 'grabs' that point using the controller. Using that grab point we create a virtual cone with the grab point serving as the vertex of the cone and the point at the viewer's feet serving as a point on the cone's edge. Pointing the controller up or down translates the viewer forward and backward along the surface of the cone. Pitching the controller straight down moves the viewer forward all the way to the vertex, while pitching straight up moves the viewer backward to twice the original cone radius. When the cone has less than a 10m radius we treat the radius as 10m to make navigation up close easier. Moving the controller left or right orbits the viewer laterally around the cone so that the cone's vertex always remains in the direction the controller is pointing. These manipulations give the illusion that the viewer is grabbing the world and dragging it by a rigid arm.

5.1.2 Discrete navigation techniques

Teleportation allows the viewer to specify a point in space and then directly and instantly move to it (Figure 5.1b). Viewers can move to any location that is within their line of sight.

We chose to evaluate *teleportation* because it is one of the earliest and most prevalent navigation techniques for VR. The most obvious advantage of *teleportation* is that transit time is instant. Further, because there are no intermediate transitions, users don't experience any velocity in navigation, and avoid one of the most common causes of cybersickness.

Our implementation uses a linear ray-cast pointer projected from the viewer's controller. By holding down a controller button the viewer triggers a straight laser pointer, which they can use to point towards any visible location in the environment. Once the viewer releases the button, they are teleported immediately to that location.

5.1.3 Mini-maps

Of these techniques, *world-in-miniature* requires an additional navigation aid – in this case, a mini-map showing a small-scale, abstracted version of the surrounding environment. However, because mini-maps provide an alternative perspective on the environment that can help viewers evaluate distances and see terrain outside of their immediate field of view, they may also benefit other navigation techniques. In particular, mini-maps may be a useful navigation aid when using *teleportation*, where they make it possible for viewers to teleport to parts of the virtual world that are not directly visible from their current location.

In our implementation, we used a circular 1m diameter mini-map, which hovered in the center of the play area at waist height (Figure 5.2, Figure 5.4). The orientation of the mini-map mirrored the orientation of the world and both its position and orientation were fixed. This mini map was a 1:1024 scale recreation of the virtual world, using the same geometry and textures, but clipped using a cylinder to ensure that participants could reach its center from all sides. The mini-map always included a prominent red token showing the viewer's current position.

5.2 Studies

While prior research has discussed and evaluated many of these techniques independently, the trade-offs and complementarities between have not been deeply examined.

Each VR navigation technique has a unique set of strengths and weaknesses. As a result, different approaches may be more effective for particular classes of tasks or for different kinds of viewers. Contemporary guidelines for VR developers (*Oculus Best Practices* 2017; "Movement in VR" 2018) often advocate for minimizing continuous movement, citing the risk of nausea.

This poses a challenge in large virtual environments, where navigation (often over large distances) may be essential. *Teleportation* offers one possible solution, allowing viewers to quickly cover large distances while reducing the potential for cybersickness. However, *teleportation* can often be disorienting (Bowman, Koller, and Hodges 1997) and can become increasingly difficult to use in larger worlds where destinations are no longer easily visible.

Meanwhile, continuous techniques (including *flight*, *3D cone drag*, and *world-in-miniature*) produce continuous movements through the environment, likely reducing viewers' chances of becoming lost or losing track of their orientation but increasing the risk of cybersickness. Even within the space of continuous techniques, trade-offs exist. Straightforward approaches like *flight* make it easy to cover short distances using smooth, predictable movements, but can be prohibitively slow in large environments. Meanwhile, *3D cone drag* embraces the familiar direct manipulation metaphors common on 2D displays, making it easier for viewers to move and rotate the terrain simultaneously. *World-in-miniature* also leverages direct manipulation, allowing viewers to quickly traverse large distances and providing overviews that can reveal unseen parts of the world and help viewers track their own position.

We conducted a study in two stages that examine the trade-offs between these techniques. In both stages, we compared (1) which techniques allowed viewers to navigate and search more quickly in large virtual environments and (2) how each technique impacted viewers' sense of orientation. We also characterized (3) viewers qualitative preferences for and experiences with these four techniques, and (4) examined how viewers combined the approaches to compensate for their relative strengths and weaknesses. We discuss the results of both stages in aggregate in section 5.5 Discussion.

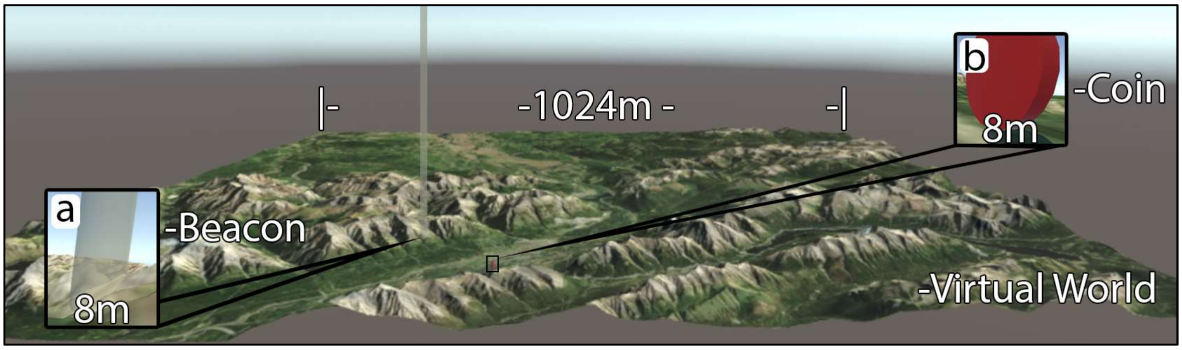


Figure 5.3. Overview of the entire virtual world used during the study tasks. Insets show beacons (a) used for navigation tasks and coins (b) used for search tasks

5.2.1 Test Environment

We developed our study environment for the HTC Vive virtual reality headset and Vive controllers, see section 3.1.3 Test Environment. Our test map included a 124km×124km region centered around the hamlet of Exshaw, Alberta. This area is situated at the edge of the Canadian Rockies, near Banff National Park, and features a diverse mix of flat, rolling, and mountainous terrain. We rendered this region at a scale of roughly 1:120, resulting in a 1024m×1024m virtual terrain (Figure 5.3). Within this virtual world, the viewer occupied a 2.5m×2.5m play area (mirroring the size of the physical test area) within which they could move and interact freely (Figure 5.2).

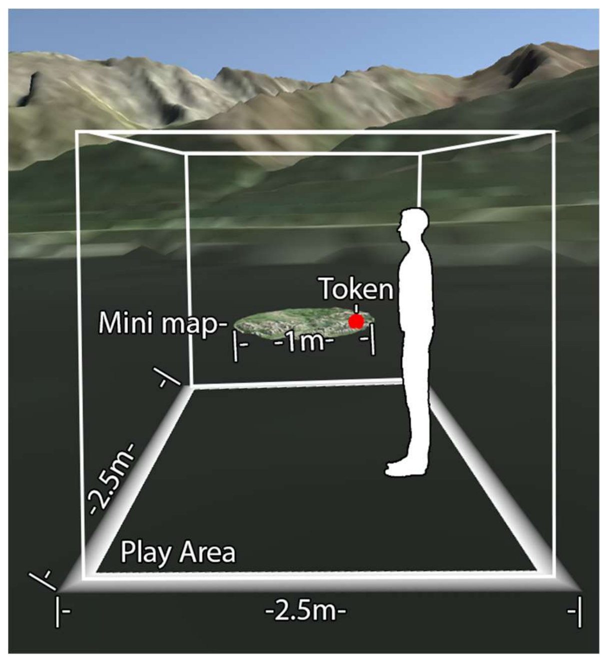


Figure 5.2. The 2.5m^2 play area and 1m diameter mini-map configuration used in our test environment.

5.2.2 Tasks

We asked participants to perform two different kinds of common spatial wayfinding and orientation tasks (Darken and Peterson 2001). These included *navigation tasks* which required viewers to move as quickly as they could to a known location, and *search tasks* which required them to explore the virtual world in order to locate a hidden object. These are similar to the tasks used by Darken and Peterson (Darken and Peterson 2001). Each trial consisted of one task.

Navigation tasks involved moving from a starting position to a large, clearly visible beacon placed at a random location in the environment (Figure 5.3a). The beacons also appeared as bright yellow pins on the mini-map.

Search tasks involved finding and collecting a large red coin (Figure 5.3b) placed at a random location in the environment. Coins were not marked on the mini-map.

5.2.3 Measures

During both study stages we tracked participants' position in the virtual environment, as well as the physical positions of their head and hands at one second intervals. We also collected timing data for each task, along with detailed notes on participants' behaviors and vocalizations.

To gauge participants' spatial awareness and orientation, we asked them to complete a *point-back test* immediately after they finished each search or navigation task. Upon reaching a beacon or coin, the interface prompted participants to use the controller in their non-dominant hand to point back towards the location of the previous coin or beacon and confirm their selection by pressing the controller's trigger for three seconds. We measured point-back error by calculating the acute horizontal angle between where the viewers pointed and the actual position of the previous beacon/coin.

After the study, participants also completed a 5-point Likert survey in which they rated the difficulty and enjoyability of the navigation techniques and conditions. Finally, we asked participants to discuss their favorite and least favorite navigation techniques.

5.3 Study Stage 1

The first study stage compared *flight*, *teleportation*, and *world-in-miniature*. We also evaluated the effect of including a *mini-map* in the *flight* and *teleportation* conditions and examined participants' behavior when they were free to combine all three techniques.

5.3.1 Participants

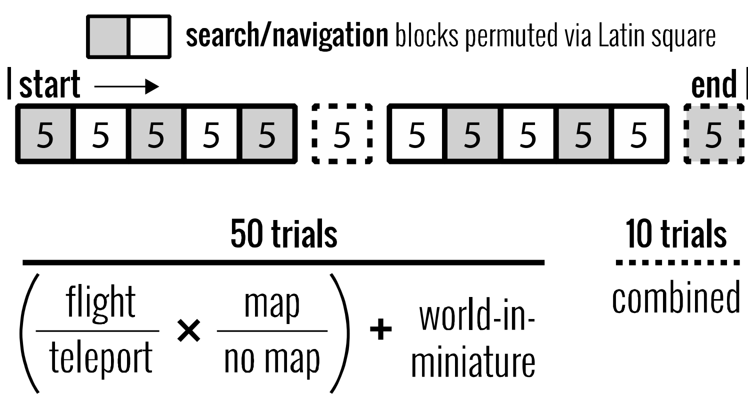
We recruited 10 Participants (2 female / 8 male, ages 22-47). Three participants had no prior experience in VR, three had very little prior experience, one had casual gaming experience in VR, one had extensive gaming experience in VR, and two had development experience in VR. We recruited via word of mouth, snowball sampling, and a university study recruitment portal and compensated participants with a $15 gift card. Eight participants showed low to negligible risk of simulation sickness on the Simulation Sickness Questionnaire (Kennedy et al. 1993; Kolasinski 1995), while 2 users showed a risk of minor symptoms. We refer to these participants using codes P1-P10.

5.3.2 Procedure

Participants completed 60 total trials, alternating between 5-trial blocks of *search* and *navigation* tasks. In blocks 1-5 and 7-11 participants alternated between 5 different experimental conditions – *flight* and *teleportation (*each *with/without mini-map)* plus *world-in-miniature.* Blocks 6 and 12 used a *combined* interface where participants had access to *flight, teleportation*, *world-in-miniature,* and the *mini-map* simultaneously. We permuted both task and condition using a Latin square design. Participants had the opportunity to rest between each block, and we asked them to remove the VR headset and take an enforced break after every three blocks. Participants also performed a 10-minute training block where they were familiarized with all of the navigation techniques and how to perform each task.

After the study was complete, participants debriefed with the experimenter and completed a follow-up questionnaire.

5.3.3 Quantitative Results



Due to growing concern in a variety of fields about the use of null hypothesis significance testing (Dragicevic, Chevalier, and Huot 2014; Cumming 2014) we analyze our results using estimation techniques and report effect sizes with confidence intervals (CI) rather than p-value

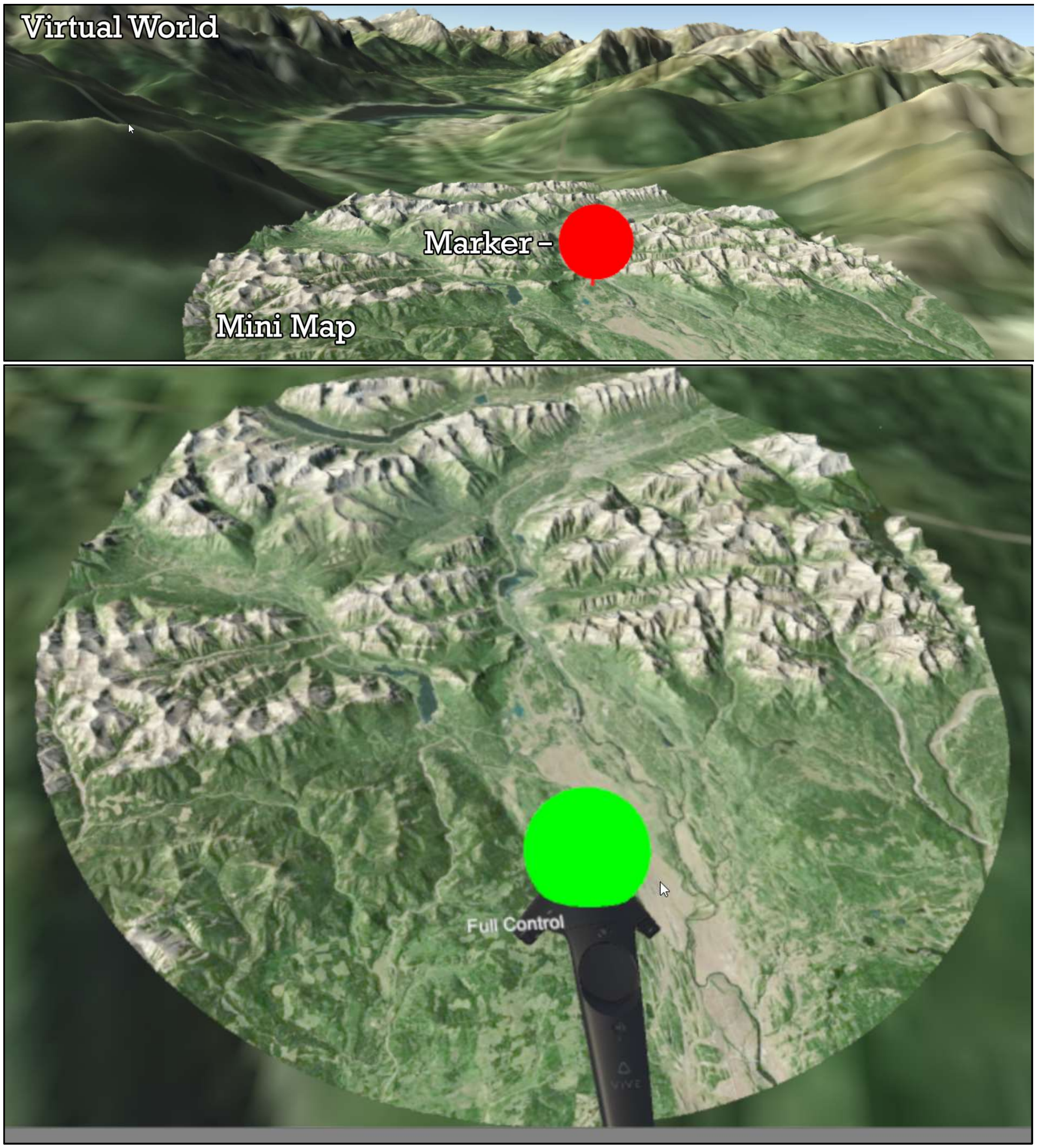


Figure 5.4. A side view of the mini-map (top) and top down oblique view of the mini map (bottom) presented to participants

statistics. The reporting methodology is consistent with recent APA recommendations (*Publication Manual of the American Psychological Association* 2017). We first computed average scores for each participant, then computed averages and 95% confidence intervals using the aggregate scores, applying a Bonferroni correction to control for multiple comparisons. Where appropriate we also computed pairwise differences between conditions, again using 95% confidence intervals with a Bonferroni correction.

5.3.3.1 Mini-map (Figure 5.4)

Although we included variations of flight and teleport both with and without a mini-map, the inclusion of a mini-map appeared to have very little impact on task performance. For navigation tasks, pairwise comparisons (Figure 5.5) showed no differences in task duration or point-back error between the *map* and *no-map* variations for either *flight* or *teleportation.* For search tasks, we only observed one case in which the *map* appeared to impact performance. Participants took an average of 16.5s (CI = [14.3,18.9]) to complete search tasks while using flight without a mini-map and 21.8s (CI = [18.5,25.8]) with one. This difference was less pronounced in navigation trials where participants took an average of 15.2s (CI = [12.2,19.9]) without a mini-map and 16.7s (CI = [13.9,20.4]) with a mini-map. Participants took an average of 51.6s (CI = [36.9,70.4]) to complete search tasks using teleportation without a mini-map and, taking 62.3s (CI = [49.0,81.9]) with one. In navigation tasks participants took an average of 19.1s (CI = [16.3,22.4]) with a mini-map and 23.1s (CI = [18.7,28.5]) without. Pairwise comparison (Figure 5.5) shows that the addition of a mini-map made no meaningful difference in participant performance with a 95% CI, except during navigation trials with flight where it was slower.

Participant point-back error during search tasks using flight averaged 19.4º (CI = [13.9,29.2) error without mini-map and 15.8° error (CI = [12.2,20.3]) with. Error with teleportation is 41.9° error (CI = [29.7,56.9]) without a mini-map and 34.0° error (CI = [25.0,45.0]) with. During navigation tasks flight had 15.9º (CI = 12.0,20.4) error without a mini-map and 19.4º (12.0,27.9) with. Teleportation had an average of 29.0º (CI = [18.8,43.4]) error without a mini map and 21.6º (CI = [15.9,31.0]) with. Pairwise comparison (Figure 5.5) shows no meaningful difference between conditions with and without a mini-map with a 95% CI.

While it had no significant effect on performance participants, used the mini map as a guide for point-back tests, trying to remember their previous position on the mini map to compare with their current position. This was especially true while using teleportation. In addition, they used the mini map to find the highest mountain peaks both as targets for teleportation and to confirm their current location. P8 used the Vive controllers to serve as markers that spanned the path they believed they had taken during the trials. P1 and P5 wanted to turn off the mini-map because it

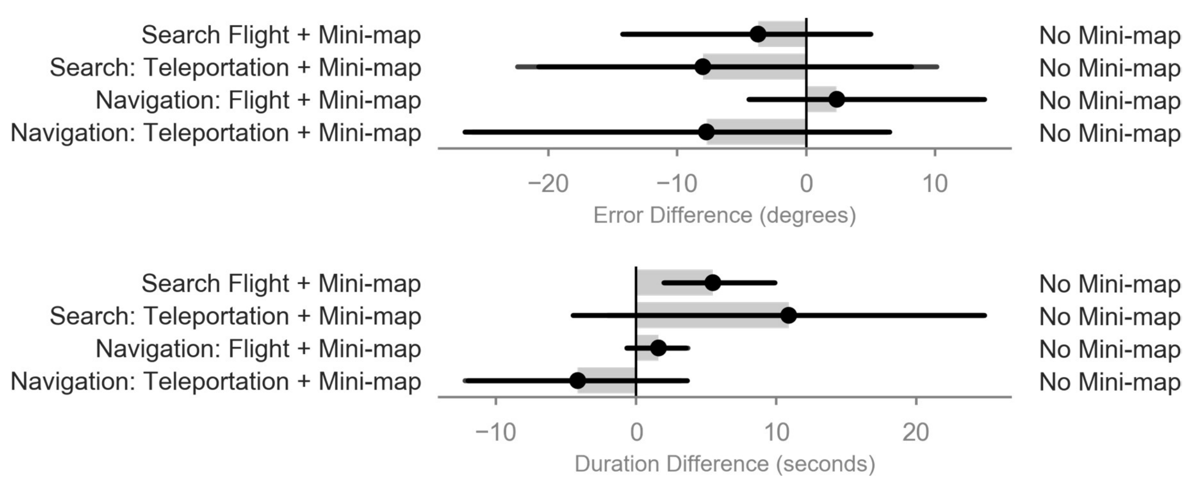


Figure 5.5. Pairwise comparison of task duration (top) and point-back error (bottom) for flight and teleportation with and without a mini-map. Error Bars show 95% CIs.

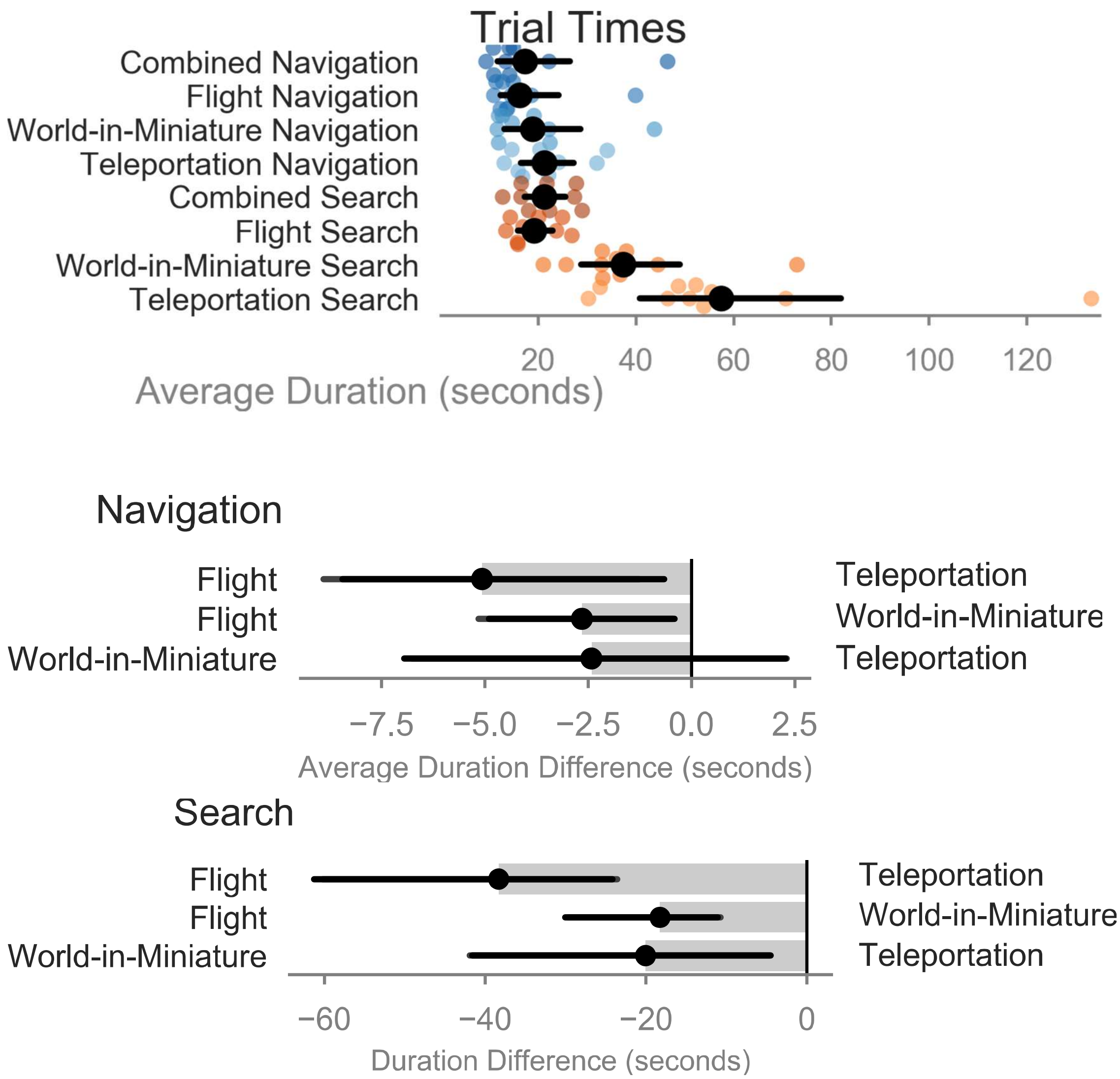


Figure 5.6. Average task duration (shorter is better) by condition for Study stage 1 (top). Swarm plots show individual participant averages. Pairwise comparisons between conditions for navigation tasks (middle) and search tasks (bottom). All error bars show 95% CIs.

obstructed their field of view. While P2, P4, P6, and P8 expressed that they liked the mini-map and wished they could turn it on in trials where it was disabled.

Due to the very small contribution of the mini map in both trial duration and point back test error we provide all subsequent analysis of stage 1 with an aggregation of trials containing the mini-map and trials without.

5.3.3.2 Task Duration (Figure 5.6)

While using *flight* participants took an average of 15.9s (CI = [12.1,24.6]) to complete navigation trials and 19.1s (CI = [15.5,23.1]) to complete search trials. *World-in-miniature* was slower, averaging 18.5s (CI = [13.0,28.1]) to complete navigation trials and 37.0s (CI = [28.4,50.8]) for search trials. *Teleportation* was also slower, averaging 21.1s (CI = [16.2,27.3]) for navigation tasks and 56.9s (CI = [40.6,84.3]) for search tasks.

Pairwise comparisons show that *flight* was faster than either *teleportation* or *world-in-miniature* for navigation tasks but show no clear difference between *teleportation* and *world-in-miniature*. For search tasks *flight* outperformed both other approaches by a considerable margin, and *world-in-miniature* clearly outperformed *teleportation*. Although *flight* performed best, 3 of 10 participants noted that they felt *world-in-miniature* was as fast or faster than *flight*, even though they performed more slowly with it.

5.3.3.3 Orientation (Figure 5.7).

Point-back error was comparable between *flight* and *world-in-miniature* for both task types. In navigation trials the average point-back error for *flight* was 16.9° (CI = [10.0,30.8]) and for *world-in-miniature* was 21.1° (CI = [14.7,27.9]). For search trials *flight* averaged 17.7° (CI = [13.2,23.5]) and *world-in-miniature* 21.7° (CI = [12.7,32.5]). *Teleportation* introduced much more error, averaging 25.5° (CI = [17.4,38.0]) in navigation trials and 38.2° (CI = [27.5,48.7]) in search trials. Pairwise comparisons show the similarity between *flight* and *world-in-miniature* as well as their divergence from *teleportation.*

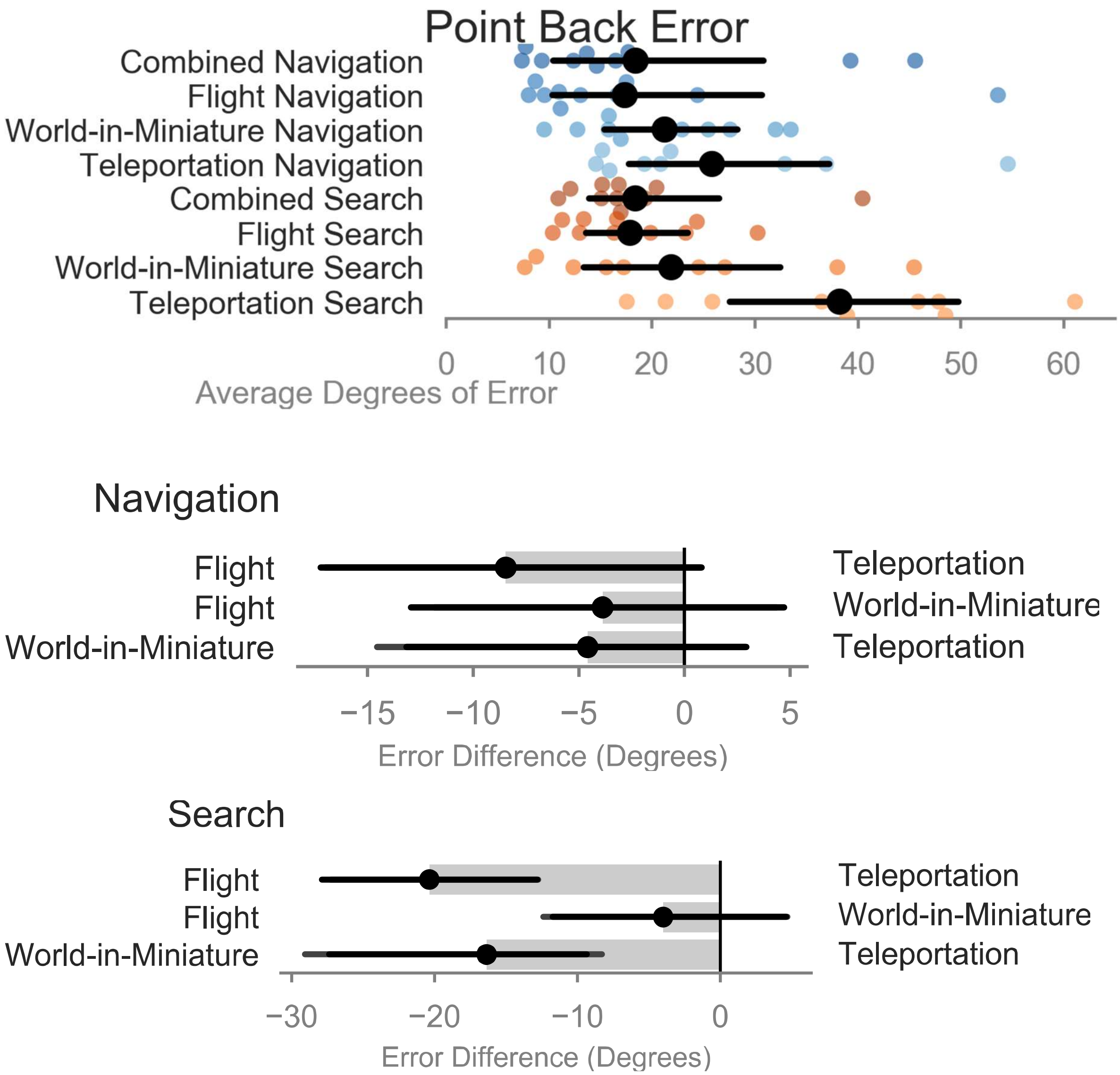


Figure 5.7. Average point-back error (less is better) by condition for Study stage 1 (top). Swarm plots show individual participant averages. Pairwise comparisons between conditions for navigation tasks (middle) and search tasks (bottom). All error bars show 95% CIs.

5.4 Study Stage 2

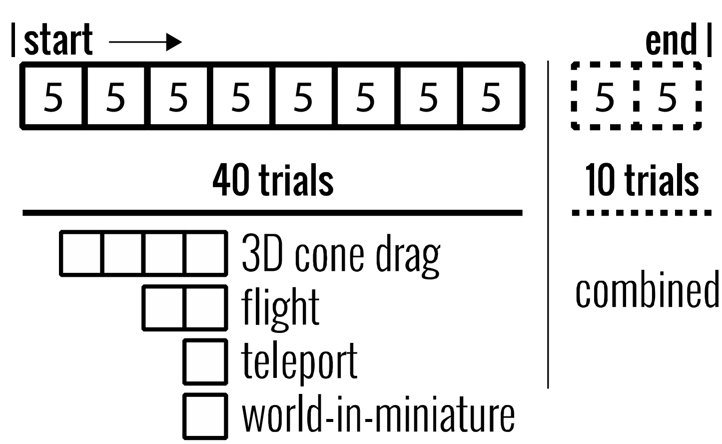
In the second stage of the study we compared the three techniques from the first study stage (*flight*, *teleportation*, and *world-in-miniature)* with *3D cone drag*, the navigation technique now popularized by Google Earth VR (Käser et al. 2017).

5.4.1 Participants

We recruited 8 participants (2 female / 6 male, ages 22-43). Four participants (P6-P10) were repeat participants from stage 1, while the other four (P11-14) were new. As in study 1, we compensated participants with a $15 gift card.

5.4.2 Procedure

Because differences between the navigation techniques were more pronounced for search tasks, we did not include navigation tasks in study 2. Instead, participants completed 50 *search* trials split across 10 blocks. To capture more data about the new condition while reducing participant fatigue, we biased the number of blocks allocated to *3D cone drag* (4 blocks) and *flight* (2 blocks) and included only one block each of *teleport* and *world-in-miniature*. We permuted block order using a Latin Square. To explore how participants combined all four techniques, we included two *combined* blocks at the end of the study. As in study 1, participants had the opportunity to rest between each block, and we asked them to remove the VR headset and take an enforced break after 4 blocks. We also used the same pre- and post-study procedures as in stage 1.



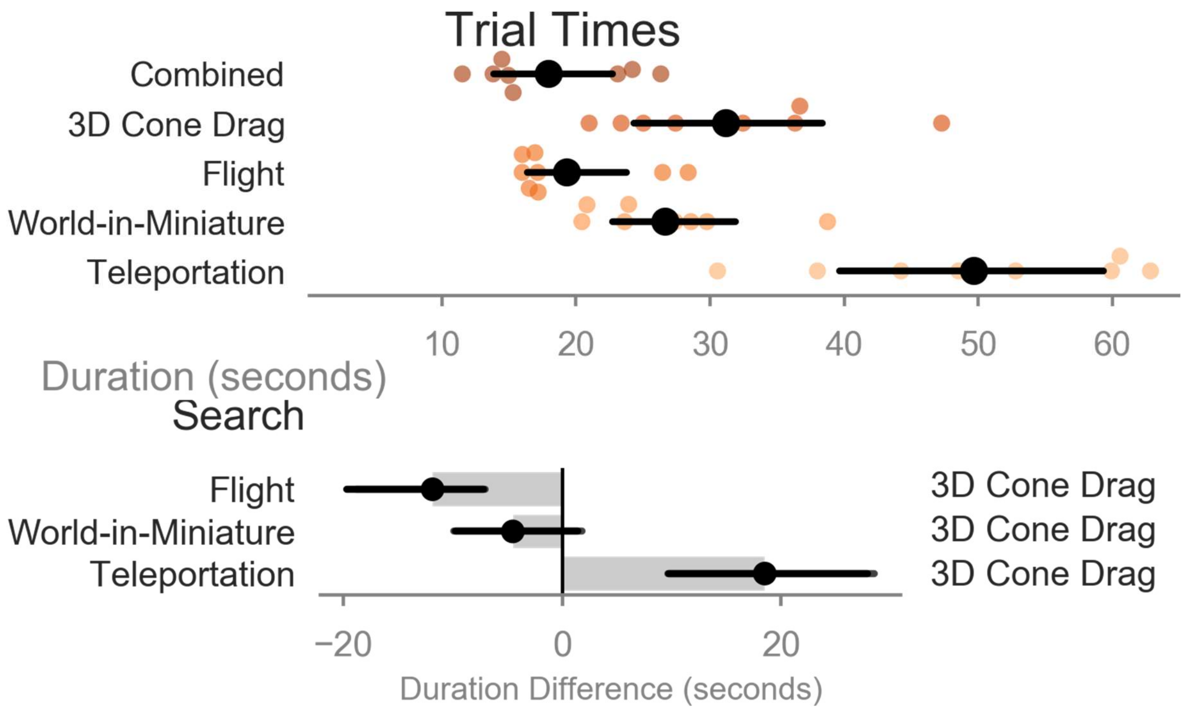


Figure 5.8 Average duration (shorter is better) by condition for Study 2 (top). Swarm plots show individual participant averages. Pairwise comparisons between conditions (bottom). Error bars show 95% CIs.

5.4.3 Results

Task Duration (Figure 5.8).

Participants performed the best with *flight* taking an average of 19.2s (CI = [16.3,23.5]) to complete tasks. This was faster than *world-in-miniature* which took on average 26.5s (CI = [22.6,31.9]), *3D cone drag* which take on average 31.0s (CI = [24.7,38.4]) and *teleportation* which took on average 49.7s (CI = [40.3,58.3]).

Pairwise comparisons show that *flight* clearly outperformed *3D cone drag, world-in-miniature* had comparable performance to *3D cone drag* and *3D cone drag* outperformed *teleportation.*

Orientation (Figure 5.9). Participants again performed best with *flight* with an average of 16.3° (CI = [11.5,20.9]) of point-back error. This was close to *world-in-miniature* which had an average of 17.5° (CI = [13.1,21.9]) of error. *3D cone drag* had an average of

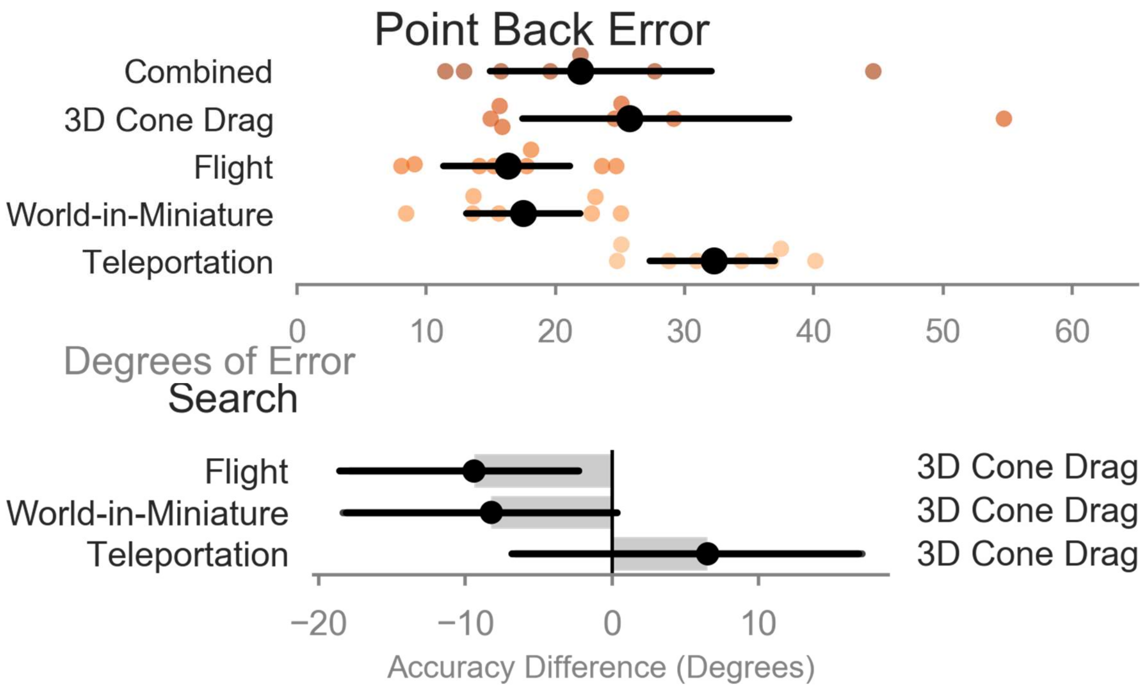


Figure 5.9. Average point-back error (smaller is better) for Study 2 (top). Swarm plots show individual participant averages. Pairwise comparisons between conditions (bottom). Error bars show 95% CIs.

25.3° (CI = [17.8,37.6]) of error which, while worse than *flight* and *world-in-miniature* outperformed *teleportation* which had an average of 32.2° (CI = [27.7,36.8]) of error. Pairwise comparisons show that participants were consistently faster when using *flight* than *3D cone drag*.

5.5 Discussion

In this section we discuss a combination of user feedback and researcher observations.

5.5.1 Participant Feedback

Overall, participants found *flight* much easier to use than any other individual technique and preferred to use it when possible. However, participants also highlighted distinct advantages of each of the other techniques and frequently combined techniques to leverage their strengths.

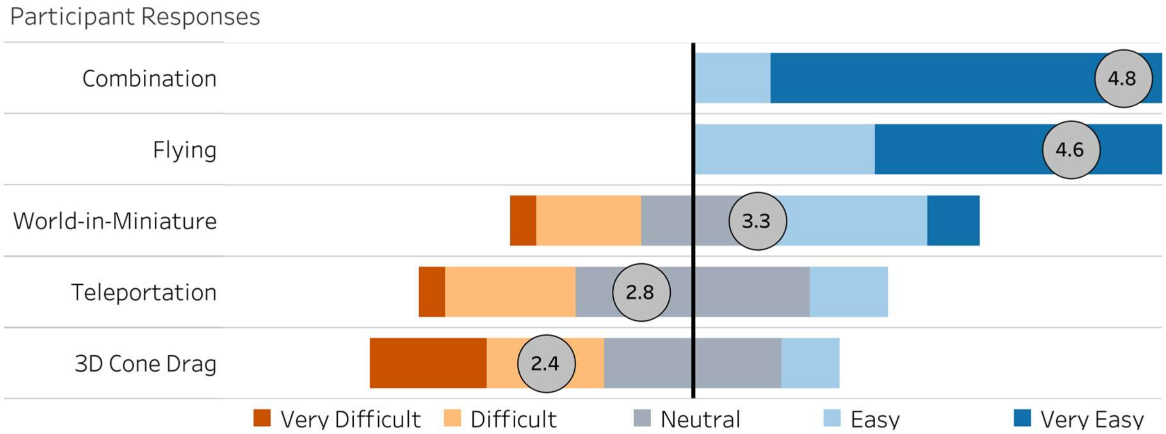


Figure 5.10. Likert responses on participants perceived difficulty of techniques. Circles show mean score. Staggered bars show stacked responses offset with positive responses to the right of the reference line and negative to the left.

Flight. Overall, participants articulated a strong preference for *flight* and found it very easy to use (Figure 5.10). This ease of use is reflected in the results as *flight* consistently performed the best of all techniques. In stage 1, when asked for their favorite and least favorite techniques 7 participants rated *flight* as their favorite while 1 rated it as least favorite. In stage 2, five participants rated *flight* as their favorite and 0 as least favorite. Overall, participants found *flight* intuitive to use and effective at all tasks. The only complaints were that it could be nauseating at higher velocities and that movement could be a bit slow. P5 and P6 noted that they felt immersed in the environment while using *flight.*

P7	*"it was very smooth and seamless"*
P9	*"easy to do everything"*
P6	*"flight was [the] most intuitive"*

	"[it] feels like one is actually moving, physically, through the space"
P8	*"as you […] start going faster and faster that was a little bit nauseating."*
P5	*"I still felt like I was part of the environment"*

World-in-Miniature. Users were divided in their opinion on *world-in-miniature* and opinions varied on whether it was easy or difficult (Figure 5.10). This matches study results which found *world-in-miniature* to fall in the middle of performance. In stage 1, when asked for their favorite and least favorite techniques 3 participants rated *world-in-miniature* as their favorite and 3 as least favorite. In stage 2, 1 participant rated it as their favorite if combined with flight and 2 as least favorite. Interestingly, while five participants noted some cybersickness during the trials none mentioned it when asked for feedback about *world-in-miniature*. Participants liked that they could use their whole body to control the token. However, they also found that attempting precision controls was challenging and would often spend a lot of time close to a coin but unable to collect it. P5 found that *world-in-miniature* disengaged them from the scene.

P2	*"I loved [world-in-miniature], even with other things accessible to me I chose to use [world-in-miniature]. It was a lot funner."*
P4	*"I could feel where I was going to go, and [use my body to] slow down."*
P6	*"the way the user is moving isn't necessarily coupled to the vision"*
P5	*"it disengaged me from the environment"*

Teleportation. Participants disliked when they were forced to use *teleportation* and found it difficult to use (Figure 5.10). This matches our study results that consistently found *teleportation* to perform poorly compared to other navigation techniques. In stage 1, when asked for their favorite and least favorite techniques 6 participants rated *teleportation* as their least favorite and 0 as favorite. In stage 2, 1 participant ranked *teleportation* combined with flight as their favorite and 2 participants rated it as their least favorite. Participants felt that it was ill suited for the tasks they were asked to perform. The biggest reason for this was that it was difficult to find an overview of the scene and most of the map was generally hidden from view. They also found it very difficult to see where they were pointing to, especially at long distances or while using the mini-map. This made mistakes especially punishing as participants would sometimes end up hundreds of meters past where they intended to go. P8 and P12 expressed that by forcing them to directly use the mountains to gain a view of the scene to search for coins they felt that *teleportation* felt more like exploring a game world than the other two techniques.

P7	*"you couldn't get any view of the scene."*
	"It was hard to aim precisely"
P5	*"I didn't have freedom of movement."*
P9	*"it's easy to lose which direction you were facing"*
P2	*"It was the most difficult to wrap my brain around."*

3D Cone Drag. Participants were mixed on *3D cone drag* but generally preferred other navigation techniques when given the choice and found it difficult to use (Figure 5.10). This rated difficulty diverges from expectation as *3D cone drag* performed comparably to *world-in-miniature,* but participants felt it was much more difficult. When asked about their favorite and least favorite techniques 1 participant ranked it as their favorite while 4 ranked it as their least favorite. Participants liked that it was very precise, allowing for fine movements. Several participants found that using the technique was very fun and made them feel like "*Tarzan or Spiderman*" (P13). However, when grabbing from far away participants found that the velocity was too fast and could be nauseating. Two participants also found that it was tedious and tiring to use for longer periods.

P9	*"It's easy to be really precise"*
P13	*"I felt like I was Tarzan or Spiderman"*
P8	*"It made me really nauseous"*
	"it had an exploration kind of feel"
P11	*"It was tedious"*

5.5.2 Cybersickness

Continuous navigation techniques are known to increase participants' risk of cybersickness, and while we did not focus on evaluating cybersickness in our studies our results reflect this. Over the course of the two studies 7 of 14 participants reported some degree of cybersickness, and one additional participant dropped out of stage 1 early as a result of it. While using *world-in-miniature* three reported mild cybersickness symptoms while two reported moderate. This is consistent with the original findings in “Virtual reality on a WIM” (Stoakley, Conway, and Pausch 1995), where the authors went so far as to decouple the moving of the token from updating the view in order to reduce the effect. Two participants noted mild cybersickness while using *flight*. One of those participants noted that cybersickness only occurred at greater speeds, an observation also consistent with past studies (Bowman, Koller, and Hodges 1997; So, Lo, and Ho 2001). Two participants noted minor cybersickness symptoms while using *3D cone drag* and one, P7, noted severe symptoms causing them to only complete 15 out of 20 trials while using *3D cone drag.* Cybersickness caused by *3D cone drag* was most severe when navigation velocity was high. As we expected, no participants noted any symptoms while using *teleportation*.

A number of strategies exist for reducing the risk of cybersickness when using *flight*, including cube map techniques (McCrae et al. 2009; Trindade and Raposo 2011) and scaling strategies like GiAnt (Argelaguet and Maignant 2016). Meanwhile many VR games, as well as systems like Google Earth VR, employ a variety visual tweaks to reduce false feelings of motion – including trimming the viewport and adding a persistent ground plane during motion (Käser et al. 2017). While these strategies have been demonstrated for both *flight* and *3D cone drag*, there exists less guidance when working with *world-in-miniature.* The original *world-in-miniature* systems (Stoakley, Conway, and Pausch 1995) separated the movement of the token from movement in the world, allowing the viewer to specify their target location and then jump directly to that point. Such approaches minimize the risk of cybersickness but also lose the benefits of continuous motion. Speed-based filters such as the 1 Euro filter (Casiez, Roussel, and Vogel 2012) can be used to limit maximum velocity. However too much filtering can introduce lag and a disconnect between the input and output, potentially exacerbating cybersickness risk.

5.5.3 Strengths and Weaknesses

The four navigation techniques we examined have strengths and weaknesses when used alone.

Flight is the most all-around effective navigation technique. However, its navigation speed is slow and designers must actively work to tune it to reduce the risk of cybersickness. There exists several options to automate the choice of an optimal speed, like cube maps (Trindade and Raposo 2011) or automatic scaling, such as GiAnt (Argelaguet and Maignant 2016).

Teleportation allows for instant movement. However, using it can be disorienting and it is hard to obtain good overviews of the environment. This can be mitigated through mini-maps and related tools like “Bird’s Eye” overview images (Fukatsu et al. 1998). In addition, aiming in VR follows a Fitts law model (C. Wingrave and Bowman 2005) and aiming at far destinations can be challenging. Many modern systems limit the distance of teleportation by using an arc instead of a line to limit maximum selection distance.

World-in-miniature enables fast motion in a scene and makes it very easy to navigate to positions that provide good overviews. However, precise movements can be challenging especially as the scale of the world and the mini-map diverge. Precise object placement is a known challenge even in the physical world (Poupyrev et al. 1998; Kerr and Langolf 1977) and complicates a variety of selection and manipulation tasks in virtual reality (Argelaguet and Andujar 2013). This can be mitigated with approaches like PRiSM (Frees and Kessler 2005), which reduces shakiness for precise motions, or SSWIM (C. A. Wingrave, Haciahmetoglu, and Bowman 2006) which uses scrolling and scaling to keep the scene and mini-map at similar scales.

Finally, *3D cone drag* provides a direct-manipulation alternative to flight. However, we found that using the technique for long, fast movements also lead to high velocities and strong accelerations, increasing the chance of cybersickness. In addition, the control motions themselves can be tedious, requiring a lot of arm movement. High velocities caused by grabbing far away can be mitigated by limiting how far away a user can grab. However, by limiting the distance a user can grab the technique becomes more tedious as the user most perform more intermediate grabs to navigate a scene. Balancing how far a user can grab and how many steps the interaction requires is necessary for a successful implementation of *3D Cone Drag*.

5.6 Combinations

Interestingly, when participants in our studies had access to all the interaction techniques simultaneously, they tended to combine techniques to address the relative weaknesses of

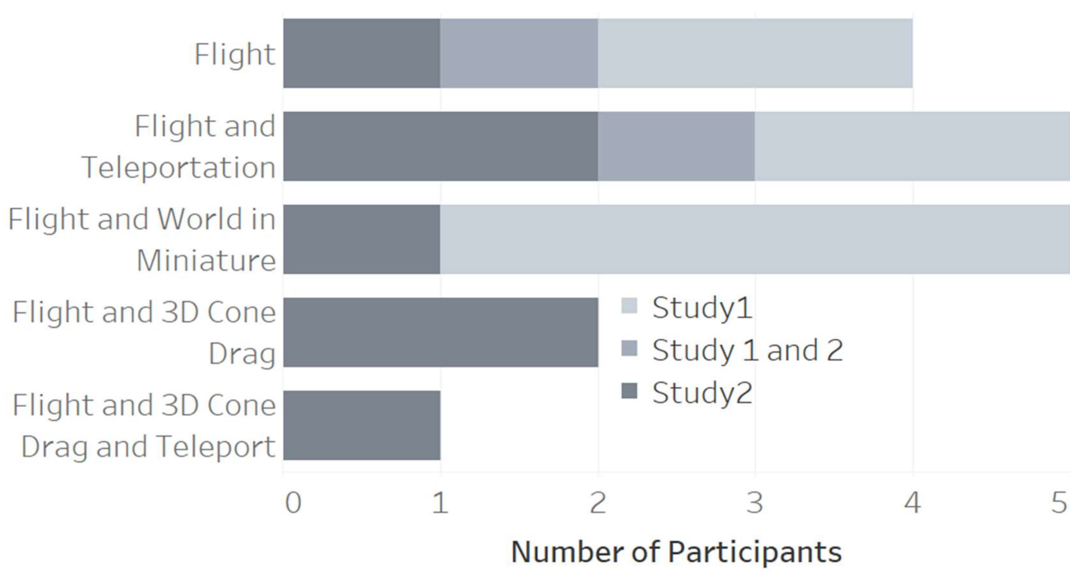
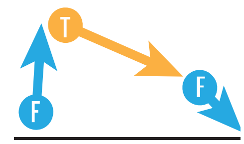


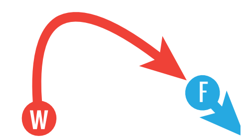
Figure 5.11. Number of participants who used each combination of navigation techniques by study.

individual approaches. We observed five unique combinations of techniques across the two studies (Figure 5.11) and suggest some general combination guidelines.

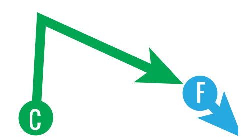
***Teleportation* + *Flight*.** 5 total participants combined these approaches, using *teleportation* for rapid, long-distance travel that might otherwise be tedious. Participants would fly directly up into the air to get a good view of the scene and their target, teleport as close to the destination as possible, then fly the remaining distance.



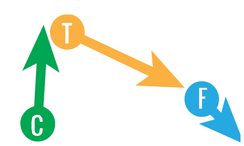
***World-in-Miniature* + *Flight*.** Similarly, 5 participants used *world-in-miniature* to cover large distances, before shifting to *flight* for small precision movements. Participants would often move upward to get a good overview of the scene using *world-in-miniature* then use it to get as close as possible to their target, before shifting to *flight* for the final more difficult movements.



***3D Cone Drag* + *Flight*.** In our second study, two participants used *3D cone drag* in a similar way – first moving upward to gain an overview of the scene and then grabbing and dragging to move most of the way to their target before flying the last portion.

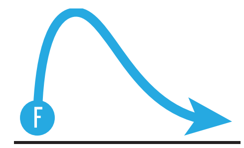


***3D Cone Drag* + *Teleportation* + *Flight*.** One participant in stage 2 combined all three techniques to great effect, using *3D cone drag* to move up to a better vantage point, then combining *teleportation* and *flight* to move close to and reach the target.



In all of these cases, participants used *flight* mostly for precise movements, while relying on *teleportation, world-in-miniature,* and *3D cone drag* for longer transits across the large virtual environment. Participants found *world-in-miniature* and *3D cone drag* especially useful for moving upward quickly to gain an overview of the scene, even when the mini-map was present.

While many participants combined navigation techniques when given the chance, even when given access to all techniques, 5 of the 14 participants still relied exclusively on *flight.* This speaks most strongly to flights capability as a stand-alone navigation technique.



5.6.1 Pairing Navigation Techniques Effectively

There are three features of a navigation technique that allow it to effectively perform in a large VR environment:

1) Capable of Precise Local Movement
2) Capable of Large, Quick Movement
3) A Mechanism to Obtain an Overview of the Scene

The capacity for both precise local movements and large, quick movements through the scene is necessary for navigating throughout the scene and a mechanism for obtaining an overview of the environment is also particularly valuable, especially when environments include complex terrain or obstructions. Many navigation techniques fit several of these three qualifications. For example, *flight* can be used for high precision movement and can easily be used to obtain an overview, making it the most reliable single technique out of those we evaluated. However, when used for long, protracted movements, it (as well as the other continuous techniques) may trigger nausea or discomfort. *Teleportation* is capable of precise local movements and large quick movements but is particularly weak at obtaining an overview of the environment. *World-in-miniature* is well suited for large quick movements and obtaining an overview of the environment, but precision local movement is challenging. *3D cone drag* is technically capable of all three requirements, but precise local movements are very tedious to perform, such that using them is undesirable for users. As a result, pairing flight, the strongest stand-alone navigation technique, with *teleportation* and/or a technique like *world-in-miniature* or *3D cone drag* allows viewers to navigate efficiently and reducing their risk of disorientation. This is exactly what Google Earth VR does, pairing flight and 3D Cone Drag.

5.7 Chapter Conclusion

When working in a large VR environment every navigation technique has its own strengths and weaknesses. The weaknesses of each technique can either be mitigated by technique specific design choices or through combinations of techniques. We have highlighted these weaknesses and how to address them in discussion. In addition, when possible, combinations can be used to greatly increase user power and freedom and can counteract technique specific weaknesses. When considering which techniques to use in a system, designers should likely include some variant of *flight* if appropriate as it is stand-alone the most powerful, as well as at least one other navigation technique to supplement it.

Chapter 6 : Conclusion

In this thesis, we presented two explorations that evaluated camera control techniques designed for representations of real-world terrain. In the first, we presented and evaluated an alternative camera control scheme designed specifically for representations of real-world terrain on 2D touch devices. In the second, we compared and evaluated the effectiveness four common VR navigation techniques at navigating large virtual map-like spaces. In this concluding chapter, we revisit the thesis questions and contributions presented in Chapter One and suggest some potential future work, as well as a closing statement.

6.1 Thesis Questions

The guiding question of this thesis was: **How can we improve camera controls specialized for representations of real world-terrain?** We created two sub questions from the guiding question to focus our research.

> *Research Question 1 (RQ1): Many applications that rely on touch controls currently use direct manipulation camera control. How can we add to and improve, these touch based systems so that viewers can recreate views and better understand terrain?*

We address this question in Chapter Four, where we designed three new camera control schemes based on a look-from metaphor. We validated these new schemes in Chapter 4, where we compared each against *direct manipulation* and each other. Based on our experience designing, implementing, and evaluating these three camera controls we presented a few key design guidelines that can be used to develop new camera control.

Research Question 2 (RQ2): Immersive virtual reality systems have several options for camera control. Given many options, which option(s) are most appropriate for representations of real-world terrain?

We address this question in Chapter Five, where we implemented and directly compared four navigation techniques in a large map-like immersive virtual environment. Based on this evaluation we present the benefits and trade-offs of those four navigation techniques, as well as several ways in which a designer can mitigate weaknesses — one way being combinations of navigation techniques.

In addressing these two research questions we examined a small portion of the overall guiding question. However, answering the guiding question in full beyond the scope of this thesis and will need to be addressed through continued iteration on these researcher questions. This work forms a meaningful start at addressing the guiding question.

6.2 Contributions

This thesis has four main contributions in the domain of designing interactive camera controls for use in representations of real-world terrain:

Contribution 1: A side-by-side comparison of *direct manipulation* camera controls and three variations of *look-from* camera controls at line-of-sight and elevation comparison tasks. We performed this on representations of representations of real-world terrain presented on a tablet sized touch device. This shows how *look-from* camera controls can be integrated smoothly on top of *direct manipulation* controls without impacting performance but improving the user experience.

This comparison is one of the results presented in Chapter Four and highlights the quantitative results of the study.

Contribution 2: Design considerations for future *look-from* navigation tools for digital maps based on my observations of and experience with these initial variants. These considerations highlight many non-obvious features necessary for a clean introduction of *look-from* camera controls to system.

These design considerations are one of the results presented in Chapter Four, and highlight lessons learned in the design process of look-from camera controls, through observations made during the look-from study, and from participant feedback.

Contribution 3: A side-by-side comparison of *flight, teleportation, world-in-miniature,* and *3D cone drag* at navigation and search tasks on a representation of real-world terrain presented in an immersive virtual environment. This shows the effectiveness of *flight* as a general-purpose navigation technique for map-like environments in VR.

This comparison is one of the results presented in Chapter Five, and it highlights the findings made from the quantitative data from the study.

Contribution 4: An examination of the five specific ways in which participants combined *flight, teleportation, world-in-miniature,* and *3D cone drag* to more effectively complete search and navigation tasks by compensating for the weaknesses of some techniques with the strengths of other techniques.

The observed combinations are another result presented in Chapter Five, and highlight the qualitative data derived in the study from observation and user comments.

6.3 Future Opportunities

One of the features that makes maps the most interesting is that they don't just refer to a digital world — but also represent a fragment of the real world. Because of this, an interesting way to expand upon this work is to consider camera control for mixed reality systems that are situated in the map environment that they represent. With many mixed reality systems there is no need to consider camera control too deeply, the viewer physically handles the camera. With a head mounted mixed reality system that means that the viewer rotates their camera by turning their head and moves it by physically locomoting. With a handheld augmented reality system (like a cellphone) the viewer moves and rotates their camera by manipulating their handheld like a lens into the mixed world. However, by disconnecting the camera from the mixed reality device's physical location a viewer can gain one of the greatest benefits of maps — gaining an overview and quickly exploring alternative views. For example, by allowing the viewer to transform their camera straight up into a bird's eye view of their surroundings they could view an overview of their environment while still being able to see the world at ground level through their own eyes. This transition from a personal perspective to a bird's eye view is the exact opposite of how most map systems operate — the viewer transitions from a ground level oblique view to a top down view, as compared to transitioning from a top down view to an oblique one. A view in which the viewer can consider two views at once, such as a ground level view and a top down view, is a split view. By considering this split view links could be formed between the two perspectives. For example, by touching a point in the overview that same point could be highlighted in the camera view (Figure 6.1). Or conversely, image information from the device camera could be used to texture the overview (DiVerdi, Wither, and Hollerer 2008).

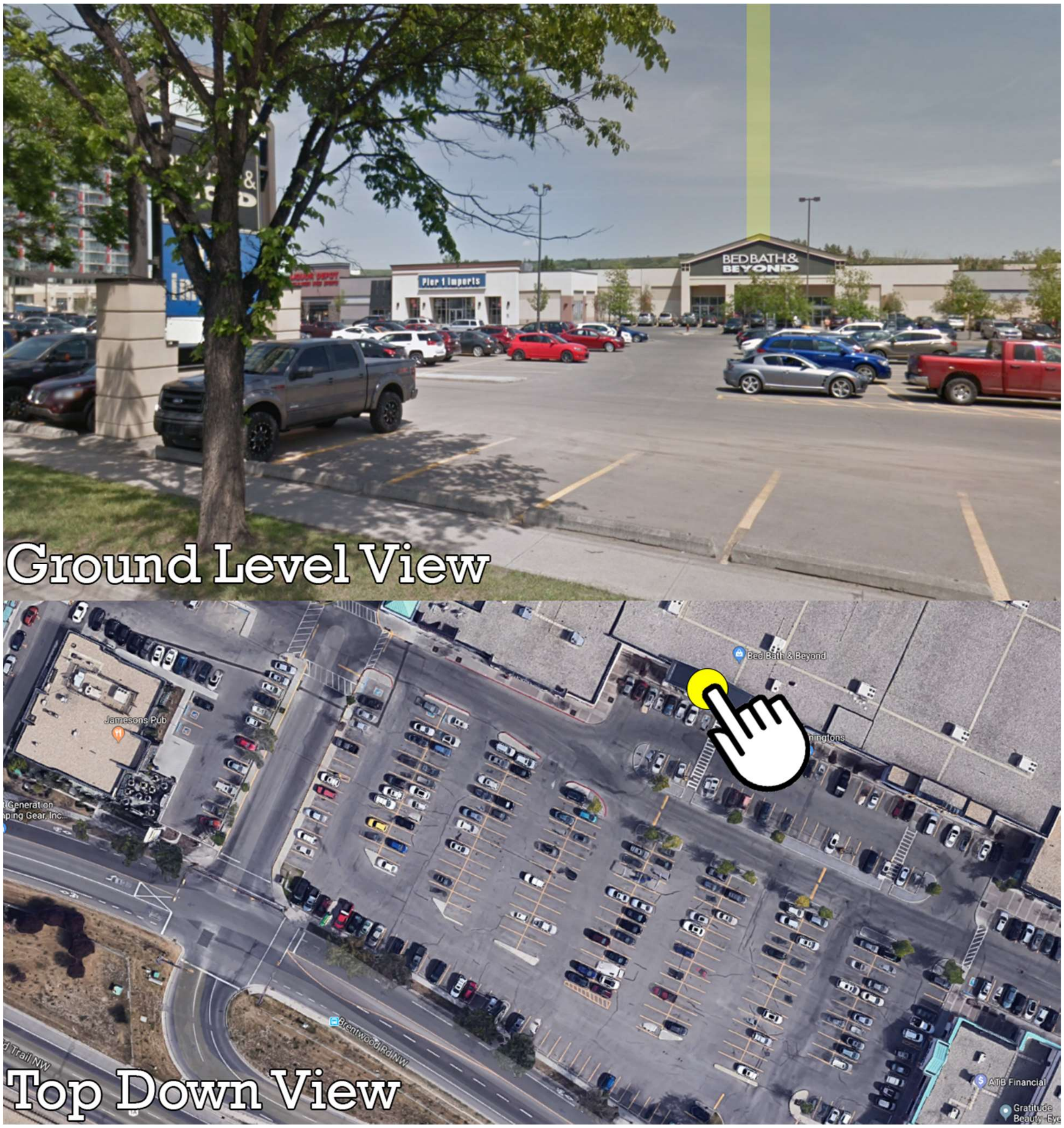


Figure 6.1. A Mock up of a split mixed reality view showing how an interaction can be linked between two views. Augmented reality view (top) and top down map view (bottom). The yellow point in the top-down view is the yellow column in the ground level view, seen from above.

The content of this thesis can help to guide the design of mixed reality camera controls in several ways. The first is from the guidelines outlines in Chapter 4:

- **D1. Keep *look-from* points in frame during transitions**
- **D2. Sequence camera rotation and pitch to minimize roll**
- **D3. Make *look-from* interactions reversible if possible.**

In Chapter 4 viewers using *look*-from camera controls primarily used them to transition from top down views to oblique views, however the guidelines should still applicable when moving from an oblique view to a top down view. Demonstrating their applicability to the inverted circumstance, either positively or negatively, would be a useful starting point for the design of a mixed reality camera control.

The second way the results of this thesis can contribute to the development of mixed reality camera controls is from the observations in Chapter 5. Due to the similarities between headset-based mixed reality devices and virtual reality devices many of the navigation techniques explored in Chapter 5 have the potential to act as effective means for camera control within mixed reality. The observations shown in Chapter 5 should help to narrow down the selection of appropriate techniques. Mixed reality, and split views both present an interesting challenge for interactive camera control, but the techniques detailed in Chapter 5 will likely act as an effective starting point.

6.4 Closing Remarks

It can be challenging to effectively read and interpret maps. However, digital maps can make that experience easier. We can improve how viewer's engage with maps by improving the camera control techniques that viewers have. In this thesis, we presented two laboratory studies each comparing different navigation techniques designed for representations of real-world terrain. The first introduced a new style of camera control designed with representations of real-world terrain in mind, look-from camera controls. The second compared the effectiveness of four, different common camera control techniques in virtual reality within the context of a representations of real-world terrain. Both studies highlight many different camera control schemes designed for representations of real-world terrain. They also show how different camera controls change how viewers interact with their environment. While this work only scratches the surface of different camera controls, it sets a starting block that can be extended to motivate future camera control designs, and to inform the choice of existing camera controls.

Bibliography

Abend, Pablo, Tristan Thielmann, Ralph Ewerth, Dominik Seiler, Markus Mühling, Jörg Döring, Manfred Grauer, and Bernd Freisleben. 2012. "Geobrowsing Behaviour in Google Earth – A Semantic Video Content Analysis of On-Screen Navigation." In *GI_Forum 2012: Geovisualization, Society and Learning*, 2–13.

Argelaguet, Ferran, and Carlos Andujar. 2013. "A Survey of 3D Object Selection Techniques for Virtual Environments." *Computers & Graphics* 37 (3). Elsevier:121–36.

Argelaguet, Ferran, and Morgant Maignant. 2016. "GiAnt: Stereoscopic-Compliant Multi-Scale Navigation in VEs." In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, 269–77. VRST '16. ACM. https://doi.org/10.1145/2993369.2993391.

Bowman, Doug A, David Koller, and Larry F Hodges. 1997. "Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques." In *Virtual Reality Annual International Symposium, 1997., IEEE 1997*, 45–52.

Buchholz, H., J. Bohnet, and J. Dollner. 2005. "Smart and Physically-Based Navigation in 3D Geovirtual Environments." In *Ninth International Conference on Information Visualisation (IV'05)*, 629–35. IEEE. https://doi.org/10.1109/IV.2005.117.

Buddeberg, Jonas, Bernhard Jenny, and Wesley Willett. 2017. "Interactive Shearing for Terrain Visualization: An Expert Study." *GeoInformatica* 21 (3). Springer US:643–65. https://doi.org/10.1007/s10707-016-0283-9.

Card, Stuart K., Jock D. Mackinlay, and George G. Robertson. 1991. "A Morphological Analysis of the Design Space of Input Devices." *ACM Transactions on Information Systems* 9 (2). ACM:99–122. https://doi.org/10.1145/123078.128726.

Casiez, Géry, Nicolas Roussel, and Daniel Vogel. 2012. "1€ Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2527–30.

Chen, Michael, S. Joy Mountford, Abigail Sellen, Michael Chen, S. Joy Mountford, and Abigail Sellen. 1988. "A Study in Interactive 3-D Rotation Using 2-D Control Devices." *ACM SIGGRAPH Computer Graphics* 22 (4). ACM:121–29. https://doi.org/10.1145/378456.378497.

Christianson, D.B. David B, Sean E S.E. Anderson, Li-wei He, David H. Salesin, D.S. Daniel S Weld, and Michael F. Cohen. 1996. "Declarative Camera Control for Automatic Cinematography." In *National Conference on Artificial Intelligence*, 148–155. https://doi.org/10.1016/j.cell.2011.08.008.

Cumming, Geoff. 2014. "The New Statistics: Why and How." *Psychological Science* 25 (1). Sage Publications Sage CA: Los Angeles, CA:7–29.

Darken, Rudolph, and Barry Peterson. 2001. "Spatial Orientation, Wayfinding, and Representation." *Handbook of Virtual Environment Technology* 2002. https://doi.org/10.1201/b17360-24.

DiVerdi, Stephen, Jason Wither, and Tobias Hollerer. 2008. "Envisor: Online Environment Map Construction for Mixed Reality." In *2008 IEEE Virtual Reality Conference*, 19–26. IEEE. https://doi.org/10.1109/VR.2008.4480745.

Dragicevic, Pierre, Fanny Chevalier, and Stephane Huot. 2014. “Running an HCI Experiment in Multiple Parallel Universes.” In *CHI’14 Extended Abstracts on Human Factors in Computing Systems*, 607–18.

Eynard, James D., and Bernhard Jenny. 2016. “Illuminated and Shadowed Contour Lines: Improving Algorithms and Evaluating Effectiveness.” *International Journal of Geographical Information Science*, April. Taylor & Francis, 1–21. https://doi.org/10.1080/13658816.2016.1144885.

Frees, Scott, and G Drew Kessler. 2005. “Precise and Rapid Interaction through Scaled Manipulation in Immersive Virtual Environments.” In *IEEE Virtual Reality Conference, 2005.*, 99–106.

Fukatsu, Shinji, Yoshifumi Kitamura, Toshihiro Masaki, and Fumio Kishino. 1998. “Intuitive Control of Bird’s Eye Overview Images for Navigation in an Enormous Virtual Environment.” In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 67–76. VRST ’98. New York, NY, USA: ACM. https://doi.org/10.1145/293701.293710.

Gleicher, Michael, and Andrew Witkin. 1992. “Through-the-Lens Camera Control.” In *ACM SIGGRAPH Computer Graphics*, 26:331–40. https://doi.org/10.1145/142920.134088.

Hachet, Martin, Fabrice Decle, Sebastian Knodel, and Pascal Guitton. 2008. “Navidget for Easy 3D Camera Positioning from 2D Inputs.” In *2008 IEEE Symposium on 3D User Interfaces*, 83–89. IEEE. https://doi.org/10.1109/3DUI.2008.4476596.

Hagedorn, Benjamin, and Jürgen Döllner. 2008. “Sketch-Based Navigation in 3D Virtual

Environments.” In *Smart Graphics*, 239–46. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-85412-8_23.

Hanson, A.J., and E.A. Wernert. 1997. “Constrained 3D Navigation with 2D Controllers.” In *Proceedings. Visualization ’97 (Cat. No. 97CB36155)*, 175–182,. IEEE. https://doi.org/10.1109/VISUAL.1997.663876.

Hildebrandt, Dieter, and Robert Timm. 2014. “An Assisting, Constrained 3D Navigation Technique for Multiscale Virtual 3D City Models.” *GeoInformatica* 18 (3). Springer US:537–67. https://doi.org/10.1007/s10707-013-0189-8.

Jacob, Robert J. K., Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen. 1994. “Integrality and Separability of Input Devices.” *ACM Transactions on Computer-Human Interaction* 1 (1). ACM:3–26. https://doi.org/10.1145/174630.174631.

Jankowski, Jacek, and Martin Hachet. 2015. “Advances in Interaction with 3D Environments.” In *Computer Graphics Forum*, 34:152–90.

Jia Wang, and R. W. Lindeman. 2012. “Comparing Isometric and Elastic Surfboard Interfaces for Leaning-Based Travel in 3D Virtual Environments.” In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, 31–38. IEEE. https://doi.org/10.1109/3DUI.2012.6184181.

Käser, Dominik P, Evan Parker, Adam Glazier, Mike Podwal, Matt Seegmiller, Chun-Po Wang, Per Karlsson, et al. 2017. “The Making of Google Earth VR.” In *ACM SIGGRAPH 2017 Talks*, 63:1--63:2. ACM. https://doi.org/10.1145/3084363.3085094.

Kennedy, Robert S, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. 1993. “Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator

Sickness." *The International Journal of Aviation Psychology* 3 (3). Taylor & Francis:203–20. https://doi.org/10.1207/s15327108ijap0303_3.

Kerr, Barry A, and Gary D Langolf. 1977. "Speed of Aiming Movements." *The Quarterly Journal of Experimental Psychology* 29 (3). Taylor & Francis:475–81.

Kolasinski, Eugenia M. 1995. "Simulator Sickness in Virtual Environments."

Kopper, R, Tao Ni, D A Bowman, and M Pinho. 2006. "Design and Evaluation of Navigation Techniques for Multiscale Virtual Environments ." *IEEE Virtual Reality Conference (VR 2006)* . IEEE . https://doi.org/10.1109/VR.2006.47.

Li, Nico, Wesley Willett, Ehud Sharlin, and Mario Costa Sousa. 2017. "Visibility Perception and Dynamic Viewsheds for Topographic Maps and Models." In *Proceedings of the 5th Symposium on Spatial User Interaction - SUI '17*, 39–47. ACM Press. https://doi.org/10.1145/3131277.3132178.

Liu, James, Hirav Parekh, Majed Al-Zayer, and Eelke Folmer. 2018. "Increasing Walking in VR Using Redirected Teleportation." In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*, 521–29. ACM Press. https://doi.org/10.1145/3242587.3242601.

Mackinlay, Jock, Stuart K. Card, and George G. Robertson. 1990. "A Semantic Analysis of the Design Space of Input Devices." *Human-Computer Interaction* 5 (2). L. Erlbaum Associates Inc.:145–90. https://doi.org/10.1207/S15327051HCI0502&3_2.

Mapes, Daniel P., and J. Michael Moshell. 1995. "A Two-Handed Interface for Object Manipulation in Virtual Environments." *Presence: Teleoperators and Virtual Environments*

4 (4):403–16. https://doi.org/10.1162/pres.1995.4.4.403.

McCrae, James, Igor Mordatch, Michael Glueck, and Azam Khan. 2009. "Multiscale 3D Navigation ." *Proceedings of the 2009 Symposium on Interactive 3d Graphics and Games* . ACM . https://doi.org/10.1145/1507149.1507151.

"Movement in VR." 2018. Introduction to VR. 2018. https://unity3d.com/learn/tutorials/topics/virtual-reality/movement-vr?playlist=22946.

Oculus Best Practices. 2017. Irvine: Oculus VR, LLC. https://static.oculus.com/documentation/pdfs/intro-vr/latest/bp.pdf.

Pausch, Randy, Tommy Burnette, Dan Brockway, and Michael E Weiblen. 1995. "Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures." In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, 399–400. SIGGRAPH '95. ACM. https://doi.org/10.1145/218380.218495.

Peck, T. C., H. Fuchs, and M. C. Whitton. 2012. "The Design and Evaluation of a Large-Scale Real-Walking Locomotion Interface." *IEEE Transactions on Visualization and Computer Graphics* 18 (7):1053–67. https://doi.org/10.1109/TVCG.2011.289.

Phillips, Richard J., Alande Lucia, and Nicholas Skelton. 1975. "Some Objective Tests of the Legibility of Relief Maps." *The Cartographic Journal* 12 (1). Taylor & Francis:39–46. https://doi.org/10.1179/caj.1975.12.1.39.

Pierce, Jeffrey S., and Randy Pausch. 2004. "Navigation with Place Representations and Visible Landmarks." In *Proceedings - Virtual Reality Annual International Symposium*, 173–80. IEEE. https://doi.org/10.1109/VR.2004.1310071.

Potash, L. M., J. P. Farrell, and T. S. Jeffrey. 1978. “A Technique for Assessing Map Relief Legibility.” *The Cartographic Journal* 15 (1). Taylor & Francis:28–35. https://doi.org/10.1179/caj.1978.15.1.28.

Poupyrev, Ivan, T Ichikawa, S Weghorst, and M Billinghurst. 1998. “Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques.” In *Computer Graphics Forum*, 17:41–52.

Publication Manual of the American Psychological Association. 2017. Sixth Edit. Washington, D.C.: American Psychological Association.

Raskin, Jef. 2000. *The Humane Interface : New Directions for Designing Interactive Systems*. Addison-Wesley.

Schwarz, Julia, Old Middlefield Way, Mountain View, and Robert Xiao. 2015. “Estimating 3D Finger Angle on Commodity Touchscreens.” In *ACM International Conference on Interactive Tabletops and Surfaces*, 47–50. https://doi.org/10.1145/2817721.2817737.

Silva, Bruno Marques Ferreira da, Selan Rodrigues dos Santos, and Jauvane Cavalvante De Oliveira. 2009. “Using a Physically-Based Camera to Control Travel in Virtual Environments.” In *2009 XI Symposium on Virtual and Augmented Reality*, 146–56.

Slater, Mel, Martin Usoh, and Anthony Steed. 1995. “Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality.” *ACM Transactions on Computer-Human Interaction* 2 (3). ACM:201–19. https://doi.org/10.1145/210079.210084.

So, Richard H Y, W T Lo, and Andy T K Ho. 2001. “Effects of Navigation Speed on Motion Sickness Caused by an Immersive Virtual Environment.” *Human Factors* 43 (3):452–61.

https://doi.org/10.1518/001872001775898223.

Stoakley, Richard, Matthew J Conway, and Randy Pausch. 1995. “Virtual Reality on a WIM: Interactive Worlds in Miniature.” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 265–72. CHI ’95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. https://doi.org/10.1145/223904.223938.

Tan, Desney S., George G. Robertson, and Mary Czerwinski. 2001. “Exploring 3D Navigation.” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’01*, 418–25. ACM Press. https://doi.org/10.1145/365024.365307.

Trindade, Daniel R, and Alberto B Raposo. 2011. “Improving 3D Navigation in Multiscale Environments Using Cubemap-Based Techniques.” In *Proceedings of the 2011 ACM Symposium on Applied Computing*, 1215–21. SAC ’11. ACM. https://doi.org/10.1145/1982185.1982454.

Turner, Russell, Francis Balaguer, Enrico Gobbetti, and Daniel Thalmann. 1991. “Physically-Based Interactive Camera Motion Control Using 3D Input Devices.” In *Scientific Visualization of Physical Phenomena*, 135–45. Springer Japan. https://doi.org/10.1007/978-4-431-68159-5_8.

Wan, M, F Dachille, and A Kaufman. 2001. “Distance-Field Based Skeletons for Virtual Navigation.” In *Proceeding of the Conference on Visualization’01*, 239–46. https://doi.org/10.1109/VISUAL.2001.964517.

Ware, Colin, and Steven Osborne. 1990. “Exploration and Virtual Camera Control in Virtual Three Dimensional Environments.” *SIGGRAPH Computer Graphics* 24 (2). New York, NY, USA: ACM:175–83. https://doi.org/10.1145/91394.91442.

Willett, Wesley, Bernhard Jenny, Tobias Isenberg, and Pierre Dragicevic. 2015. "Lightweight Relief Shearing for Enhanced Terrain Perception on Interactive Maps." In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 3563–72. ACM Press. https://doi.org/10.1145/2702123.2702172.

Wingrave, C A, Y Haciahmetoglu, and D A Bowman. 2006. "Overcoming World in Miniature Limitations by a Scaled and Scrolling WIM." In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium On*, 2006:11–16. IEEE.

Wingrave, C, and D Bowman. 2005. "Baseline Factors for Raycasting Selection." In *Proceedings of HCI International*.

Xiao, Dongbo, and Roger Hubbold. 1998. "Navigation Guided by Artificial Force Fields." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '98*, 179–86. ACM Press. https://doi.org/10.1145/274644.274671.

Yan, Zhixin, Robert W. Lindeman, and Arindam Dey. 2016. "Let Your Fingers Do the Walking: A Unified Approach for Efficient Short-, Medium-, and Long-Distance Travel in VR." In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, 27–30. IEEE. https://doi.org/10.1109/3DUI.2016.7460027.

Yannakakis, Georgios N., Héctor P. Martínez, and Arnav Jhala. 2010. "Towards Affective Camera Control in Games." *User Modeling and User-Adapted Interaction* 20 (4). Springer Netherlands:313–40. https://doi.org/10.1007/s11257-010-9078-0.

Zeleznik, Robert, and Andrew Forsberg. 1999. "UniCam---2D Gestural Camera Controls for 3D

Environments." In *Proceedings of the 1999 Symposium on Interactive 3D Graphics - SI3D '99*, 169–73. ACM Press. https://doi.org/10.1145/300523.300546.

UNIVERSITY OF CALGARY

November 12, 2018

University of Calgary
2500 University Drive NW
Calgary, AB T2N 1N4
Canada

I, Wesley Willett, give Kurtis Danyluk permission to use co-authored work from our publication, “Evaluating the Performance of Navigation Techniques in Large Scale Open Virtual Environments” for his MSc thesis.

Sincerely,

[redacted]

Wesley Willett
Assistant Professor
Computer Science
University of Calgary
wesley.willett@ucalgary.ca

UNIVERSITY OF CALGARY

November 12, 2018

University of Calgary
2500 University Drive NW
Calgary, AB T2N 1N4
Canada

I, Wesley Willett, give Kurtis Danyluk permission to use co-authored work from our publication, "Look-From Camera Control for 3D Terrain Maps" for his MSc thesis.

Sincerely,

[redacted]

Wesley Willett
Assistant Professor
Computer Science
University of Calgary
wesley.willett@ucalgary.ca

MONASH University

November 12, 2018

University of Calgary

2500 University Drive NW

Calgary, AB T2N 1N4

Canada

I, Bernhard Jenny, give Kurtis Danyluk permission to use co-authored work from our publication, “Look-From Camera Control for 3D Terrain Maps” for his MSc thesis.

Sincerely,

[redacted]

Bernhard Jenny
Associate Professor
Faculty of Information Technology
Monash University, Melbourne, Australia
bernie.jenny@monash.edu