Barehand Mode Switching in Touch and Mid-Air Interfaces

by

Hemant Surale

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2020

© Hemant Surale 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Nicolai Marquardt
Associate Professor, Department of Computer Science,
University College London

Supervisor: Daniel Vogel
Associate Professor, Cheriton School of Computer Science,
University of Waterloo

Internal-External Member: James Wallace
Associate Professor, School of Public Health and Health Systems,
University of Waterloo

Internal Members: Edward Lank
Professor, Cheriton School of Computer Science,
University of Waterloo

Edith Law
Associate Professor, Cheriton School of Computer Science,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This dissertation includes first-authored peer-reviewed material that has appeared in conference and journal proceedings published by the Association for Computing Machinery (ACM). The ACM's policy on reuse of published materials in a dissertation is as follows[1]:

> *"Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included."*

The following list serves as a declaration of the Versions of Record for works included in this dissertation:

Portions of Chapters 3:

Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2017. Experimental Analysis of Mode Switching Techniques in Touch-based User Interfaces. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). Association for Computing Machinery, New York, NY, USA, 3267–3280.

`DOI:10.1145/3025453.3025865`
https://doi.org/10.1145/3025453.3025865

Portions of Chapter 4:

Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2019. Experimental Analysis of Barehand Mid-air Mode-Switching Techniques in Virtual Reality. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 196, 1–14.

`DOI:10.1145/3290605.3300426`
https://doi.org/10.1145/3290605.3300426

Portions of Chapter 5:

Hemant Bhaskar Surale, Aakar Gupta, Mark Hancock, and Daniel Vogel. 2019. TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 13, 1–13.

`DOI:10.1145/3290605.3300243`
https://doi.org/10.1145/3290605.3300243

[1] https://authors.acm.org/main.html. Accessed on January 30, 2020.

Abstract

Raskin defines a *mode* as a distinct setting within an interface where the same user input will produce results different to those it would produce in other settings. Most interfaces have multiple modes in which input is mapped to different actions, and, *mode-switching* is simply the transition from one mode to another. In touch interfaces, the current mode can change how a single touch is interpreted: for example, it could draw a line, pan the canvas, select a shape, or enter a command. In Virtual Reality (VR), a hand gesture-based 3D modelling application may have different modes for object creation, selection, and transformation. Depending on the mode, the movement of the hand is interpreted differently. However, one of the crucial factors determining the effectiveness of an interface is user productivity. Mode-switching time of different input techniques, either in a touch interface or in a mid-air interface, affects user productivity. Moreover, when touch and mid-air interfaces like VR are combined, making informed decisions pertaining to the mode assignment gets even more complicated. This thesis provides an empirical investigation to characterize the mode switching phenomenon in barehand touch-based and mid-air interfaces. It explores the potential of using these input spaces together for a productivity application in VR. And, it concludes with a step towards defining and evaluating the multi-faceted mode concept, its characteristics and its utility, when designing user interfaces more generally.

Acknowledgements

First, I would like to thank my supervisor, Daniel Vogel, for his excellent guidance and encouragement throughout my doctoral journey. Working on one of the fundamental issues in the Human-Computer Interaction (HCI) field was an incredible experience for me. My research journey began with a bachelor degree thesis in the field of Real-Time Operating System. With this thesis my computer science journey has made a full circle starting from a systems research to a user focused research. I am in great debt to my supervisors.

Moreover, I am thankful to the post-docs who worked alongside me on various projects: Fabrice Matulic, Aakar Gupta, and Quentin Roy.

I also like to thank my dissertation committee members: Edward Lank, Edith Law, James Wallace, and external examiner Nicolai Marquardt. Beyond the dissertation committee, I'd like to thank Jesse Hoey and Mark Hancock for motivating me and advising at difficult times.

I want to thank my mentors from the industrial research opportunities who introduced me to different research styles and taught me to apply skills to tackle real-world challenges. I felt fortunate to work with Gabriel Rayes, Antonio Gomes, Melodie Vidal, Daniel Perry, and Evan at North Inc. and with Brian Smith, Rajan Vaish, and Andrés Monroy-Hernández at Snap Inc. Moreover, I'd like to extend my gratitude to Ken Hinckley, Jonathan Goldstein, Eyal Ofek, Nathalie Riche, Christian Holz, and Michel Pahud. I owe special thanks to Bill Buxton who would spend hours with me discussing some of the core research directions in HCI and inspiring me to think beyond the obvious. Overall, I enjoyed these industrial experiences thoroughly and learned a ton of new skills.

I want to thank my peers at the University of Waterloo – Nalin Chhibber, Deepak Rishi, Jingjie Zheng, Jeremy Hartmann, Rahul Iyer, Johann Wentzel, Sangho Suh, Greg d'Eon, Edmund Liu, Yunjia Sun, Qifan Li, Libo Gao, Jeff Avery, Keiko Katsuragawa, Lisa Elkin, Blaine Lewis, Rina Wehbe, Mike Schaekermann, Bahareh Sarrafzadeh (Baharjoon), Jay Henderson, Matthew Lakier, Hong Zhou, Mustafa Abualsaud, Hadi Azzouni, Hemant Saxena, Joshua Jung, Nolan Peter Shaw, Ariana Peng, Kamal Laminache, Bismaya Sahoo, Robin James. Moreover, my friends from the industrial experiences – Brian Cohn, Kamalakkannan Ravi, Parisa Kghzgrn, Anand Waghmare, Arjun Srinivasan, Imanol Schlag, Pedro Holanda, Meera Rachamallu, Tanay Singhal, and Manish Kumar. I am sure I am forgetting many other friends who kept me motivated and were supportive at difficult times, so I sincerely apologize in advance.

Dedication

This is dedicated to my grandparents, “Draupadi–Vishwa”, and my parents, “Sharada–Bhaskar”. None of this would have been possible without the foundation and the support you have provided me.

Also, I’d like to thank the little angel in our family, Rajvika, for bringing endless joy and happiness to my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Have you ever tried to enter a password, only to realize CAPS LOCK was on? If the answer is “Yes”, then you have experienced a *mode error*. A mode can be considered as the state of a user interface. For example, a drawing program has different modes like *a paint brush*, *an eraser* tool, or *a selection lasso*. This is an inherent part of almost all interfaces. A mode error occurs when your perception about the state is incorrect [194]. As a result, inadvertent operations are executed, like erasing instead of painting, but the consequences of mode errors could be far more severe. Mode errors have resulted in pilots accidentally shutting down a commercial jetliner engine and killing 47 people [44], or a data centre operator putting Amazon S3 servers to sleep disrupting major services and websites like Netflix, Scribd, and Trello [262].

Modes and mode related errors have received significant attention beyond the Human Computer Interaction (HCI) community, for example in the domain of aviation psychology [60, 117, 147, 246]. Degani et al. [57] trace the origins of the word ‘mode’ to the latin word ‘modus’ meaning manner of acting or doing, and suggested three broader categories of modes. Namely, interface modes that specify the behaviour of the interface, functional modes that specify the behaviour of the machine, and supervisory modes that specify the level of user and machine involvement in a process. However, the definition of mode, or even its categorization, is not standardized in the HCI community.

The definition of *mode* in HCI is surprisingly controversial and researchers have even questioned its existence. In an attempt to put this issue to rest, Johnson and Engelbeck [122] conducted a survey in 1989 to determine the extent at which experts in the domain of HCI agree or disagree with the definition of ‘mode’ and the interpretation of it. The survey results highlighted the disagreement among user-interface designers, and among researchers regarding what modes are.

In the context of this thesis, we align ourselves with the mode definitions stated by MacKenzie and by Raskin. MacKenzie defined *modes* as a functioning agreement or condition [166] and Raskin defines a *mode* as a distinct setting within an interface where the same user input will produce results different to those it would produce in other settings [214]. Assuming a definition of a mode, *mode-switching* is simply the process of switching from one mode to another. Switching between modes can be frequent while interacting with a system, so finding optimum mode-switching methods is important.

For 2D interfaces, there have been numerous experimental investigations comparing mode-switching techniques for pucks, mice, and pens [128, 71, 153, 268, 107], but there has been no comprehensive analysis of mode-switching techniques for touch input, or mid-air input.

This is surprising for touch input considering that a number of touch mode-switching techniques have been developed. Some are unique to touch since they rely on features such as multiple contacts [52, 218], using knuckles or other parts of the hand [96, 169], or characteristics of finger contact [30, 220]. Some touch mode-switching techniques are similar to those evaluated with pens, such as using pressure [182, 100] or using the non-dominant hand [74, 284]. However, generalizing pen-based empirical results to touch is highly speculative considering distinct touch characteristics like reduced precision from "fat fingers" [19, 73, 232] and greater friction [50]. This lack of formal comparisons of touch mode-switching techniques may be one reason why current mode-switching methods for touch seem limited compared to other input methods.

Although breahand mid-air input is a more novel method, its use is increasing. Bare-hand mid-air input performed entirely by a hand posture or movement and without any device is an alternative to device controllers in VR environments. Techniques have been proposed for VR, Augmented Reality (AR), and related contexts using hands only (e.g. [205, 274, 200]) and hands combined with body postures (e.g. [286, 32]). Many of these techniques are suitable for mode-switching, but have only been compared in an ad-hoc manner. Evaluations have focused on tasks like pointing (e.g. [281]), object manipulation (e.g. [215]), selection (e.g. [191]), and annotation [46], but no extensive comparisons of mode-switching time across techniques.

Another challenge is to investigate mode-switching and related interaction techniques when touch and mid-air input are used together. However, the lack of prominent interaction techniques that involve these two input modalities makes selecting techniques for a comparative evaluation difficult. As a first step, we focus on devising an example interaction vocabulary when two input modalities coexist in the same application. Specifically, when a multi-touch tablet is used in VR for 3D solid modelling. Researchers have investigated the use of passive props as a tablet in VR [158], but there is no comprehensive investigation exploring a complete design space for using a modern multi-touch tablet in VR.

Our work is based on the mode definitions provided by MacKenzie and by Raskin, however, the definition of a user interface mode is not yet standardized. This is surprising considering the ubiquity of modes as well as mode-switching techniques, and growing literature on mode related issues [148, 154, 223]. More surprising, is how the definition of interface modes is implicitly defined by the commonly accepted occurrence of mode-errors, yet there is no agreement on what constitutes a user interface mode.

So, in thesis we investigate mode-switching in touch interfaces and in mid-air interfaces, we explore the design space when touch and mid-air interfaces are used in combination in VR, and we present initial work to characterize the concept of mode that is more comprehensive and empirical, hopefully leading to a unified mode theory.

1.1 Research Goals

Before we describe the main research objective and the individual projects with specific research questions, we first define the important terms used in this thesis.

1.1.1 Term Definitions

We use the following terms throughout the thesis, and their definitions are given below:

1. *Mode* – We use the term "mode" to refer to any activity undertaken to change the system state internally or externally by the user [214]. Such activity include, but not limited to changing hand posture(s) while interacting with a tablet or bringing the hands in the range of sensors to change the system state in VR. In the past literature, researchers have discussed multiple definitions of mode [214, 166]. Yet, there is a lack of standardized definition for the term 'mode' [122].

2. *Mode-switching* – We use the term "mode-switching" to refer to switching between modes. Examples of mode-switching include, but are not limited to drawing a line on a tablet with one and two figures interchangeably. This change of drawing using one finger to drawing using two fingers, on a the tablet, is referred as mode-switching in touch-based interface.

3. *Barehand* – We use the term “barehand” to refer to an input performed entirely by a hand posture or movement, without any device. Our work focuses on input with touch-based and mid-air based interfaces.

4. *Device* – We use the term “device” to refer to any type of computing machinery that a user can interact with. Examples of devices include, but are not limited to a tablet, a mobile smartphone, and a head mounted glasses.

5. *Interaction* – we use the term “interaction“ to refer to an activity concerning two entities that determine each other’s behaviour over time as described in the work of Hornbæk and Oulasvirta [115].

6. *Mid-air* – We use the term “mid-air” to refer to an input conducted without contacting a non-body surface. Our work focuses on mid-air interactions in touch-based and mid-air based interfaces.

7. *Mixed-reality* – We use the term “mixed-reality” to refer to the systems that lets user interact in a partially or fully immersive computer mediated environment. This definition is inline with the work by Milgram and Kishino [178] and is frequently utilized in the literature on augmented and virtual reality in human-computer interaction field.

1.1.2 Main Research Objective

The research objective of this thesis is to leverage mode-switching analysis to help application design. We do this in four main projects. The first two projects characterize different barehand mode-switching in touch and mid-air interfaces. In the third project, we apply these results to build an interaction vocabulary for a realistic application. Finally, in the fourth project, we describe a candidate mode theory with a methodology to explore mode characteristics.

1.1.3 Project Description and Research Questions

We compared representative techniques for touch-based input and for barehand mid-air gestural input with respect to mode-switching time, error rate, and subjective ratings. Followed by a design space exploration of using a multi-touch tablet in VR. We conclude with mode definition, characterization, and an experimental methodology to test the interfaces. The broader picture of this thesis is depicted in Figure 1.1, summarizing the research questions, research methodology applied, and the main contribution for each chapter.

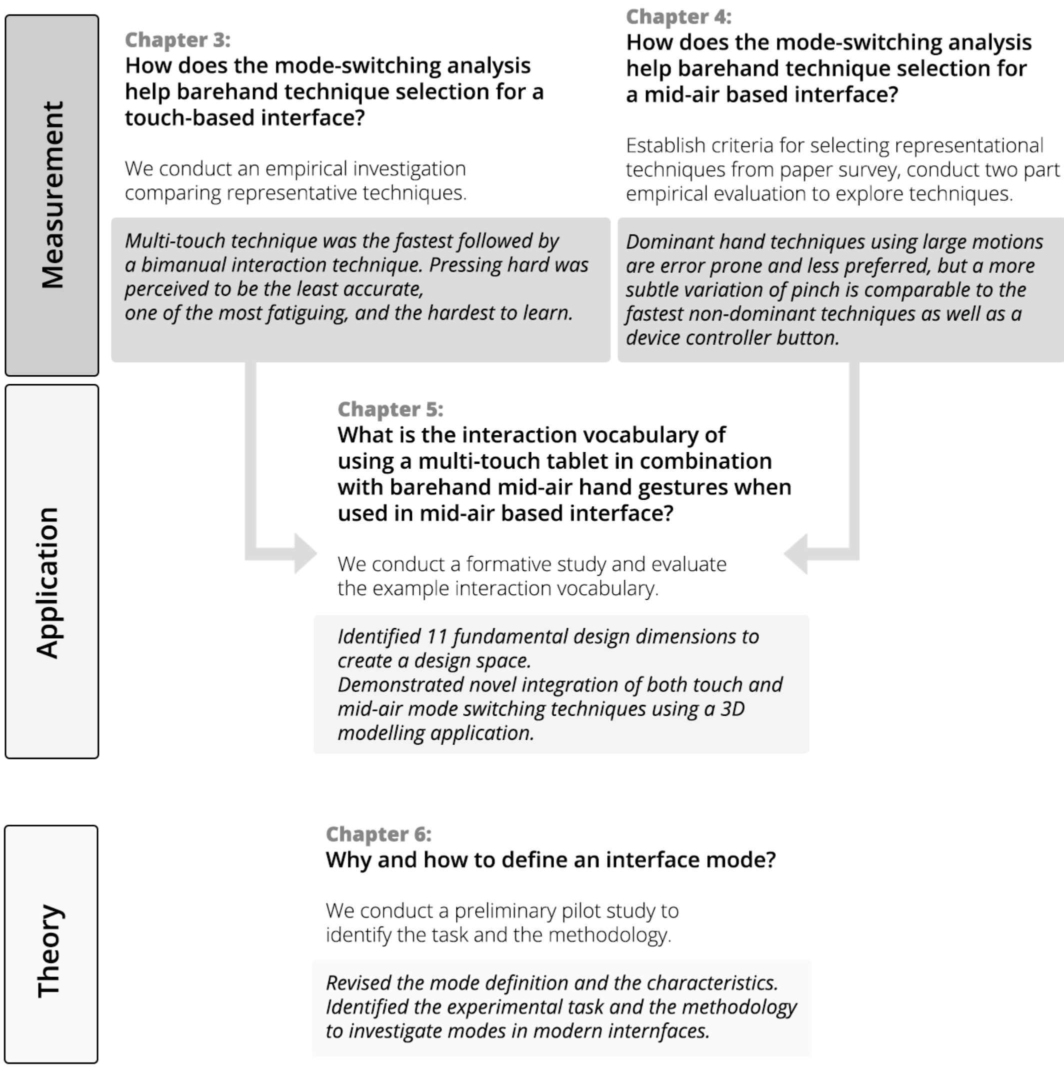


Figure 1.1: Research path showing research problems, methodology, and main results. Bold text below the chapter number is the research problem statement, the middle box of text is methodology applied, and the final italic block of text is the primary contribution. Chapter numbers represent the order of the problems explored.

Touch-based mode-switching:

In this project, we survey touch-based mode-switching techniques and select representative techniques for a comparison. Then, we compare the mode-switching times and error rates of these selected techniques, and measure the user experience using subjective ratings. Finally, we conclude with design guidelines for researchers and practitioners.

This project addresses the following research questions:

- How do future mode-switching techniques compare to established techniques?
- Does the body posture influence the mode-switching performance?
- How do users perceive different touch-based mode-switching techniques?

Barehand mid-air mode-switching for VR:

In this project, we survey barehand mid-air mode-switching techniques and select representative techniques for evaluation. Then, we compare the mode-switching times and error rates of the selected techniques, and measure the user experience through subjective ratings. Finally, we conclude with design guidelines for researchers and practitioners.

This project addresses the following research questions:

- What barehand mid-air hand gestures are suitable for mode-switching action?
- How do dominant hand mode-switching techniques compare against non-dominant hand mode-switching techniques?
- How do barehand mid-air mode-switching techniques compare against controller based mode-switching techniques?
- How well do users perceive mode-switching techniques?

Exploring the design space for using a multi-touch tablet in VR:

In this project, we determine the utility of the tablet's precise touch input, physical shape, metaphorical associations, and natural compatibility with barehand, mid-air input when used in VR. A survey of past investigations utilizing tablet like devices in VR and related MR interfaces was conducted. Then, we conduct a formative study to identify the effective interactions, and, identify design dimensions for building an interaction vocabulary. Followed by, building an interaction vocabulary for using a multi-touch tablet along with barehand mid-air gestures in VR. Finally, we built a prototype system and conducted the user evaluation.

This project addresses the following research questions:

- What is the interaction vocabulary of using a multi-touch tablet in combination with barehand mid-air hand gestures when used in VR?
- Which combinations of touch and barehand mid-air mode-switching can be used in a realistic application?

Exploring mode definition and mode characterization:

In this project, we provide an in-depth discussion of nebulous mode concept using the past literature. We identify abstract mode-switching interface types for empirical investigation, an, we establish an experimental design, task, and methodology to measure and demonstrate mode characteristics.

This project addresses the following research questions:

- What is the definition of a user interface mode? How can it be used to describe an interface effectively?
- What is an experimental task and methodology to identify and test mode characteristics in an user interface?

1.2 Research Contributions

We summarize our contributions by project. For each, we outline the methodology used and the key results that form our contributions.

1.2.1 Touch-Based Mode-Switching

In chapter 3, we describe a project to compare mode-switching time, error rates, and subjective ratings for representative touch input techniques. Given the mobile nature of tablets, both seated and standing poses are tested. The experiment task and design is a near-replication of Li et al.'s [153] pen mode-switching study. The project was published at CHI 2017 [254].

Method

We chose six mode switching techniques among those in current use, described in previous research, or soon plausible given emerging sensor capabilities. Techniques we investigated were two-fingers, non-preferred hand button, finger-on-thumb, knuckle, hard press, and long-press. In the baseline technique, participants used standard touch input. Some mode-switching techniques we tested are analogous to the pen mode switching techniques tested by Li et al. [153]. Our experimental protocol closely follows the work conducted by Li et al. [153] and Dillon et al. [62]. We used their "subtraction method" to measure mode-switching time of each technique. All techniques were designed for a tablet when placed on a desk, or when supported with the non-dominant arm for use while standing.

We recruited 36 participants (8 women, 28 men), 24 participants had experience using multi-touch tablets. The experiment is a repeated measures mixed design, with the participant POSE while using the tablet as a between-subjects factor (SIT or STAND). Half of the participants completed all tasks while seated with the tablet placed flat on a table and the other half completed all tasks while standing with the tablet held on their non-dominant forearm. After completing trials for all techniques, participants provided subjective ratings of the techniques with respect to six aspects: ease-of-learning, ease-of-use, accuracy, speed, eye fatigue, and hand fatigue.

Results

Our results contribute the following insights:

- Techniques ordered from fastest to slowest are: two-fingers, non-preferred hand button, finger-on-thumb, knuckle or hard press, and much slower long-press.
- A sitting or standing pose has no effect on speed and little or no effect on errors (only hard press and non-preferred hand error rates showed some interaction with pose).
- Pressing hard was perceived to be the least accurate, one of the most fatiguing, and the hardest to learn technique.
- Compared to Li et al.'s results for pen, our touch mode-switching timings and error rates are higher (except for knuckle compared to eraser).
- Our results can inform interaction design, returning to the opening example: one finger could draw, two fingers to pan, thumb-on-finger to select, and using a knuckle to open a command menu.

1.2.2 Barehand Mid-Air Mode-Switching for VR

In chapter 4, we describe a project to provide missing empirical evidence for the performance of barehand mid-air mode-switching techniques in VR. Our focus is absolute, single-point input, suitable for the kind of direct object manipulations common in VR. To select techniques to evaluate, we examined barehand mid-air interactions described by over 100 research papers and system descriptions in different settings such as AR, VR, and large displays. We then used three criteria to identify six classes of techniques suitable for mode-switching in VR. In two related experiments, we compare common input actions selected from each class using an adapted "subtraction method" protocol [62], used previously for 2D input [255]. The project was published at CHI 2019 [255].

Method

Both experiments are a within subjects design. The first experiment involved 16 participants (5 women, 11 men), and the second involved 12 participants (5 women, 7 men). Mode-switching technique is the primary factor, with levels corresponding to the seven techniques (non-dominant hand fist, non-dominant hand palm, bringing non-dominant hand in field of view, touching head using the non-dominant hand, dominant hand fist, dominant hand palm, and pointing using the dominant hand) in experiment 1 and an eighth technique using a standard VR device controller with a button held in the non-dominant hand (holding a controller in non-dominant hand). This functions as a non-barehand comparison baseline since the mode is switched by holding the button. The dominant hand draws a line using a pinch.

The second experiment has two further goals. First, test more subtle dominant hand mode-switching techniques to see if actions more similar to a pinch trigger might perform better. Second, test the effect of using a device controller as a mode trigger. Here, mode-switching technique is the primary factor, with levels corresponding to the four techniques (dominant hand orientation, dominant hand middle finger, holding a controller in the non-dominant hand, and non-dominant hand palm).

Results

Our empirically-derived insights can inform the design of VR applications using barehand hand mode switching:

- Dominant hand techniques using large motions are error prone and less preferred, but a more subtle variation of pinch is comparable to the fastest non-dominant techniques.

- With the exception of a few dominant hand techniques, mode-switching times are comparable to most touch methods.

- Using a dominant pinch as a manipulation trigger is comparable to using a device controller button.

- All techniques from fastest to slowest: non-dominant hand holding the device and dominant hand middle finger pinch; non-dominant palm orientation and non-dominant fist; non-dominant head touch, dominant pinch orientation and D palm; non-dominant field-of-view and dominant hand fist; dominant hand point.

1.2.3 Exploring the Design Space for using a Tablet in VR

In chapter 5, we describe a project to use a multi-touch tablet in VR. To develop an interaction vocabulary that tests the combinations of touch and hand mid-air mode-switching. In this project, the interaction vocabulary is built based on the results obtained in the previous two mode-switching investigations. Further, in our investigation, we make observations of behavioural patterns and basic features for a 3D CAD modeling application, then mapped out a design space with twelve dimensions (e.g., 'physical vs. non-physical', 'direct vs. indirect', and 'discrete vs. continuous') and developed a vocabulary of interactions (e.g., 'two-fingers to translate an object', 'five-fingers to navigate', 'mid-air tap to select an object', and 'knuckle to select multiple objects'). This project identified the main criteria to inform the interaction design when a multi-touch tablet is used in VR [253]. The project was published at CHI 2019 [253].

Method

We conducted a formative study to gain insights into how people envision using a physical tablet in a VR environment within the context of a 3D modelling application. Observations were used to build a design space for combining a tablet and barehand gestures. Ten people (7 male, 3 female) participated, three were architecture students, two were mechanical engineering students, and two were amateur users with some experience using 3D modeling applications. Expanding the formative study results, we identified 8 observations and 12 design dimensions, and used these to build an interaction vocabulary. We then prototyped a 3D solid modeling application to test the interaction vocabulary.

Our qualitative user evaluation and goals are similar to Arora et al.'s [7] work. We focus on overall usability of our system by asking the participants to replicate a predefined target model. This allowed us to observe user workflow and analyze user feedback to find the limitations of our system. We also ask participants to use our system to create a 3D model purely out of their imagination. At the end of the study, participants filled out a post-experiment questionnaire rating individual features of the system.

Results

Our design space exploration led to the following results:

- Identified 7 main observations spanning three main categories: delegation of tasks, tablet properties, and posture influenced decisions (termed general observations).
- Formed the design space for using a mutli-touch tablet in VR. The design space has 11 design dimensions. For example, unimanual versus bimanual and interleaved versus simultaneous. Participants used direct tap to select nearby objects, but used two hands while selecting a far object using a tablet. Transforming an object required participants to use both hands simultaneously, while for slicing, order of hand usage was important.
- Described an example interaction vocabulary for 3D solid modelling application.
- Usability evaluation of the example interaction vocabulary that demonstrated novel integration of both touch and mid-air mode switching techniques to facilitate 3D solid modelling in VR.

1.2.4 Mode Theory

In chapter 6, we describe a project to help solidify our understanding of one of the fundamental concepts in HCI, the 'mode'. We hope to uncover the practical significance of this concept amidst varying mode definitions found in the literature. We primarily focus on mode-switching, a process of switching from a command mode and back. This is just one of the characteristics of modes. We speculate that the concept of modes has other properties pertaining to the time spent in a particular mode, types of mode-switching patterns, the way modes can be combined together or separated at times, and so forth. These characteristics impact user interface design differently, so, it is important to characterize them.

Method

We conducted an initial pilot investigation comparing various mode characteristics across four abstract interface types, which fall under moded and un-moded categories. These four interfaces are a simple un-moded interface, and three moded interfaces such as an interface like a document editing application (e.g. MS Word), an interface like web browsers (e.g. Chrome), and an interface like code-editing tools (e.g. Visual Studio). There are five modes in the moded interfaces. The experimental task was to cross two horizontal lines in the direction from top to down in each the interface. The experimental task is balanced across all the interfaces in terms of the cognitive load and the motor movement. We used the un-moded interface to reduce the carry-over effects when switching between the moded interfaces during the experiment. The dependent measures for comparison across four interfaces were mean response times, mode errors, and other types of errors. The pilot experiment was conducted with 12 participants on the online Amazon Mechanical Turk platform.

Results

We obtained preliminary yet promising results, which provide future directions for more thorough investigation. Our specific contributions are:

- A revised mode definition and mode characteristics. Mode characteristics identified based on the experimental investigation are: mode switching, mode occupancy, mode frequency, mode pattern, mode errors, mode scaling, and lastly, mode chunking and mode phrasing.
- An experimental task and the methodology to investigate mode characteristics in the abstract interface layouts.

At a broader level, our research contributions are useful in three different ways. First, our empirically driven results can be directly utilized to inform input technique selection. Secondly, our experimental methodology can be used to analyze future mode-switching techniques. And lastly, we highlight the limits of current definitions used to describe the mode phenomenon.

1.2.5 Dissertation Outline

The remainder of this thesis is organized as follows (see also Figure 1.1).

In chapter 2, we summarize the relevant background literature pertaining to the past mode-switching investigations for mouse, pen, touch, and barehand mid-air input. Then, we describe the experimental protocol to investigate the mode-switching phenomenon in touch and mid-air interfaces.

In chapter 3, we describe the methodology and results for our comparative evaluation of barehand mode-switching techniques in touch-based interfaces. We also discuss the qualitative findings and the interaction design guidelines.

In chapter 4, we describe the methodology and results for our comparative evaluation of barehand mid-air mode-switching techniques in mid-air interfaces. Moreover, we describe the process of selecting these techniques based on a thorough literature survey and carefully selected filtering criteria. We conclude with the qualitative findings and the interaction design guidelines.

In chapter 5, we apply the results obtained from the previous studies (chapter 3 and chapter 4) to formulate the design space of using a multi-touch tablet in combination with barehand mid-air hand gestures in VR. We conclude with the usability evaluation of the prototype system built to exercise an example interaction vocabulary.

In chapter 6, we introduce a revised mode definition, mode characterization, and the experimental task as well as the methodology to investigate modes in modern interfaces.

In chapter 7, we draw conclusions, summarize limitations, and suggest possible future work.

Chapter 2

Background and Related Work

In the past, mode-switching evaluations focused on pucks, mice, and pens (styli). We provide a brief overview of these mode-switching investigations first since they have been arguably the most thoroughly studied and are foundational to our work. We subsequently focus on multitouch input and barehand mid-air input, the most relevant to our work.

2.0.1 Mode-Switching Evaluations

To begin, we review mode-switching investigations for mouse and pen input. Then, we summarize research in mode-switching techniques for touch, barehand mid-air input, and when these modalities are used together.

Mouse Input

Dillon et al. [62] conducted the first investigation to characterize mode-switching with a mouse. Dillon argued that target selection using different selection techniques is not the only performance metric, but, smooth integration of the selection method with the task is equally important. They compared touch, voice, two mice, and single mouse as experimental conditions. They concluded that voice and touch are the fastest to invoke commands compared to any of the mouse-based techniques. They also introduced the *subtraction method*, an accurate way to measure mode-switching time as described in section 2.0.1 and the experimental task is shown in Figure 2.1.

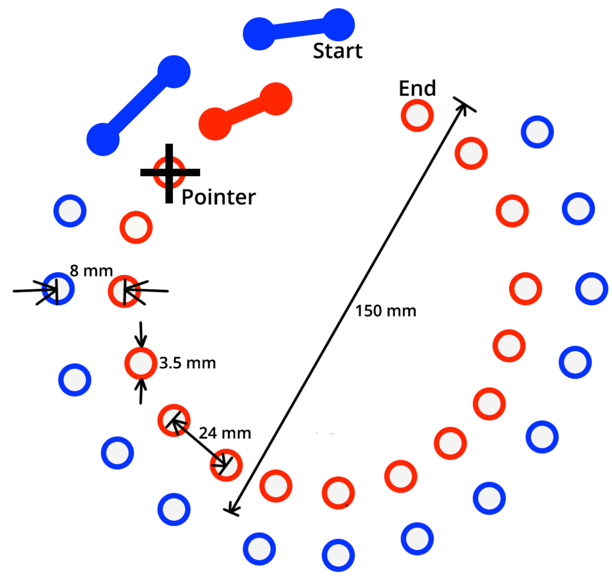


Figure 2.1: Screen display for drawing plus line-colour selection task. User alternated between red and blue lines by command selection. The task began at the 'Start' circle, continued in the anti-clockwise direction until the 'End' circle. (recreated from Dillon et al. [62] Figure 2)

Pen Input

Two most relevant pen-based mode-switching investigations are conducted by Li et al. [153] and Tu et al. [268]. Li et al. compared five techniques: a barrel button, long press, non-preferred hand, pressure, and using the eraser end of the pen. Results suggested that a physical button activated by the non-dominant hand was both faster and more accurate, also confirmed by Ruiz et al. [224]. The experimental task was crossing a pie slice as shown in Figure 2.2. The long press technique was slower and more error prone. The pressure technique, along with the remaining ones, demonstrated poor performance. These results align with past studies where performance of bimanual interaction is found to be better than unimanual interaction [149, 41]. Further, Tu et al. [268] compared five techniques in two form factors of pen-based mobile devices, PDA and Tablet PC. The five techniques were compared: pressure, long press, pressing a barrel button, pressing a button on the handheld device, tapping on the back of the handheld device, and shaking the handheld device. For Tablet PC, pressure performed the fastest and earned most errors. For PDA, tapping on the back offered the fastest performance. Although long press was slower than the other techniques, it resulted in the fewest errors for both form factors. Pressing a button on the handheld device was faster and accurate with both mobile devices.

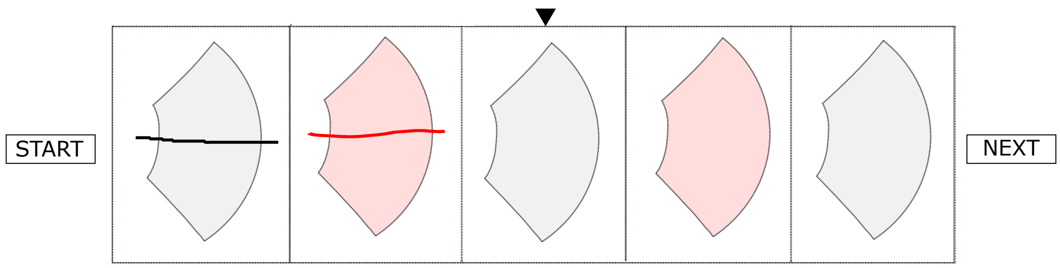
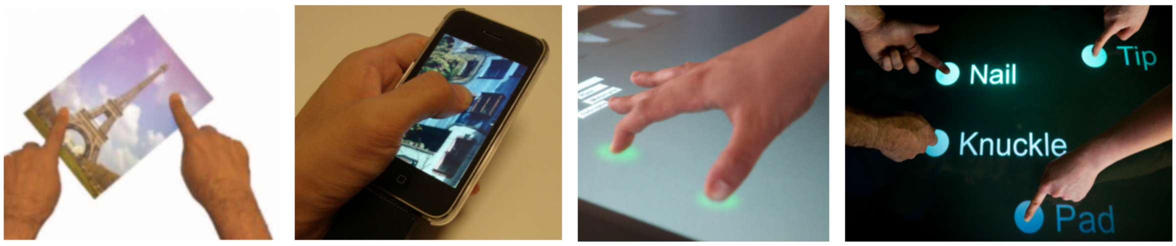


Figure 2.2: Five slices are presented in black and red alternatively, which requires a participant to cross a slice with the slice's colour. The participant needs to switch modes to draw lines with different colours. (recreated from Li et al. [153] Figure 4)

Touch Input

Researchers have introduced several mode-switching methods for touch-based interfaces. For instance, the shape of the non-dominant hand [290, 295, 97] or the number of fingers used [294] can trigger a mode change. Most naturally, using a touch surface of a tablet to activate different modes [284, 74]. Pressure [182, 100, 218] and grip-based [80] controls have also received much attention. More recently, pressure-based technology is integrated into mobile devices, popularly known as 3D Touch or Force Touch [267]. Expressivity has also widened using the number, shape, and mobility of fingers. Multiple fingers performing similar path movement can be used to trigger different actions [151, 294] and if individual fingers can be identified, interactions can be made finger-specific [52]. Further properties of finger input such as contact size [30], slight rolling movements [220], and which part of a finger touches the display [96] can also be recognized to support mode-switching. These techniques are shown in Figure 2.3.

While these various techniques offer multiple ways to switch modes, an empirical investigation comparing their mode-switching times, error rate, and user experience for touch devices has not been investigated yet.



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

Figure 2.3: Touch mode-switching techniques: (a) bimanual interactions (from Wigdor et al. [290]), (b) pressure based technique (from Miyaki et al. [182]), (c) multi-touch technique (from Lepinski et al. [151]), (d) different parts of the finger touching the surface (from Harrison et al. [95]).

Barehand Mid-Air Input

We define *mid-air* as input provided without contacting a non-body surface. In most cases, this means input performed in the space around the body. For on-body contacts, the sensing method is unimportant as long as the technique is conceptually a body contact (e.g. touch the head), and not using a device attached to the body (e.g. tap on a smartwatch).

For mid-air barehand input, mode-switching has only been indirectly evaluated as part of larger interaction technique studies. For large displays, Vogel and Balakrishnan [281] compared a relative pointing technique, which uses a fist mode-switch to “clutch”, but the mode-switch itself is not compared. Similar examples in large display research include Haque et al. [94], Polacek et al. [206], Jota et al. [124], and Katsuragawa et al. [132]. Some of these techniques are shown in Figure 2.4.

Hauqe et al. [94] investigated barehand pointing and clicking interactions with the MYO, which has elecromyograph (EMG) and inertial measurement unit (IMU) sensor. The unimanual gesture set included a clenched fist, spreading all fingers, and a relaxed hand. Similar to Vogel and Balkrishnan, they used a fist mode-switch to “clutch”, but the mode-switch was not investigated. Results showed that using MYO is only 430 to 790 ms slower than using Vicon motion tracking and has acceptable error rates for targets greater than 48 mm. Polacek et al. [206] presented a comparative study of barehand mid-air pointing in unimanual and bimanual settings. They used hand position in mid-air to control pointer position on the large display and the distance of palm to the sensor as a clutch. Results indicated that users primarily use their dominant hand for pointing. Jota et al. [124] compared four variants of raycast pointing using a handheld pointer and barehand mid-air posture. Results indicated that Fitts's law analysis based on angles better approximates the ray pointing performance. Katsuragawa et al. [132] evaluated smartwatch-based barehand mid-air pointing and clicking interactions. A unimanual gesture set used in the study included gestures like raising an arm, changing the orientation of the wrist, and lowering the arm. They demonstrated the use of smartwatch based interactions for pointing in ubiquitous environments. Their Watchpoint technique performed comparable to the the Myopoint technique [94] and a camera based motion tracking system. However, none of these studies focused on comparing the mode-switching performance.

Moreover, in the context of VR and related 3D contexts, Poupyrev et al. [209, 210] evaluated object pointing, manipulation, and selection techniques, Teather and Stuerzlinger compared pointing techniques [261], and Vanacken et al. [272], Grossman et al. [87], and Looser et al. [161] all examine barehand selection techniques. In most cases, these techniques have some explicit activation and deactivation of a mode, but mode-switching performance is not evaluated in isolation.

We are unaware of work comparing mode-switching techniques for barehand mid-air input in VR using the formal subtraction method as used for the mouse and pen.

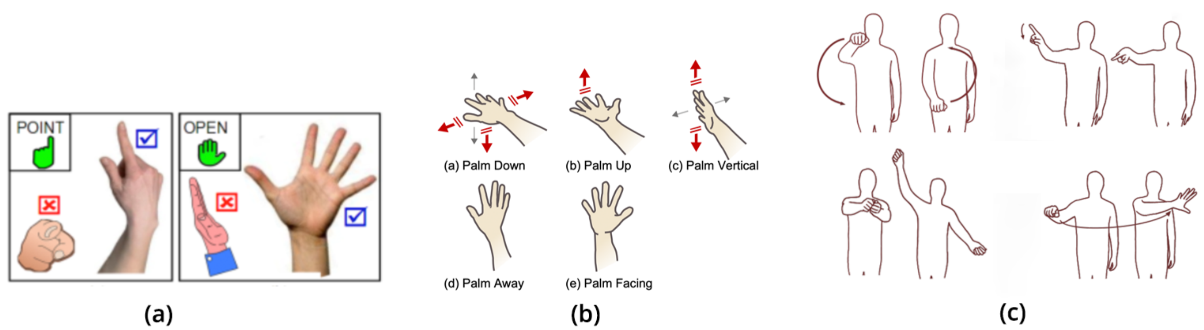


Figure 2.4: Barehand mid-air mode-switching techniques: (a) pointing posture (from Song et al. [243]), (b) palm orientation postures (from Vogel et al. [281]), (c) fist, point, and bimanual hand postures (from Ruiz et al. [226]).

Subtraction Method

Donders [63] introduced the idea that the time between the stimulus and response is occupied by a train of successive processes. If the user completes two tasks, the time difference will reveal the overhead taken by one of the tasks. This general method is known as the *subtraction method*. As described earlier, Dillon et al. [62] were the first to investigate mode-switching time with a mouse. They used the subtraction method as a tool to accurately measure mode-switching time. This method can be used to evaluate mode activation time of a wide variety of command selection techniques and their combinations. It serves as a tool that captures empirical data to make decisions about the alternative mode-switching techniques. Li et al. [153] and Song et al. [242] refined the subtraction method to evaluate mode-switching performance for pen, and their protocol is most relevant to our work.

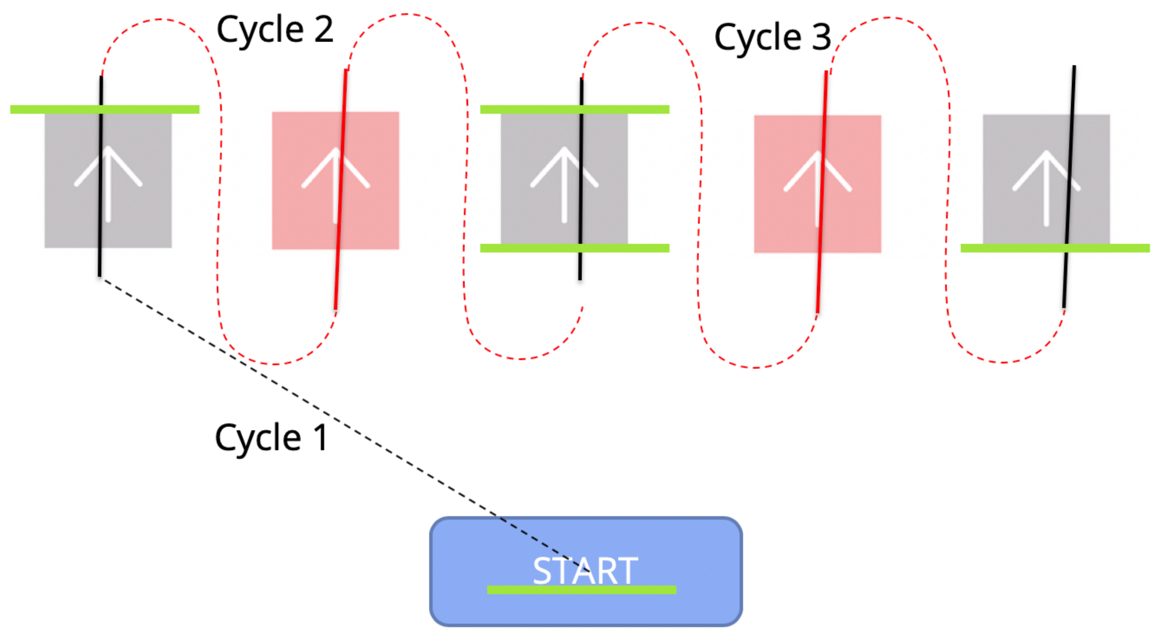


Figure 2.5: Rectangle crossing task. Mode must be switched to draw lines with different colours. Green bars show the cycle separations. The dotted lines represent the in-air hand motion and the solid lines represent the stroke drawn on the tablet. Direction arrows in the rectangle represent the expected stroke direction.

In Li et al.'s experiment, participants were asked to complete a pie crossing task (see Figure 2.2). Experiment involved five mode-switching techniques, nine blocks of trials, and eight screens corresponding to eight compass directions, and five pie-crossing tasks. In total, each participant performed 1800 pie-crossing tasks. The timing for each screen is started when the start button is clicked and automatically ended when the last pie is crossed and the pen is lifted. This duration is divided into three cycles. The first cycle starts when the start button is clicked and ends when the first pie is crossed. The second cycle starts right after the first cycle and ends after the third pie is crossed. And in the third cycle participant crossed last two pie slices. Therefore, one target needs to be crossed in the first cycle and two targets need to be crossed in each of the second and the third cycle. Last two cycles are referred as full cycles and first cycle as the start cycle. In a compound task, a full cycle contains a complete mode switch process.

The mode switching time for each compound block was computed by subtracting the mean of the two adjacent baseline task's average cycle duration from the compound block's average cycle duration. Average cycle duration was the mean duration of all correct full cycles in a block. Note how mode-switching time is the time spent in-between a successive touch-up and a touch-down event, essentially, when the user's hand(s) are in mid-air. Cycle separations are shown in Figure 2.5 (cycle 2 and cycle 3 are the full cycles).

Considering the mode-switching action, touch-based mode-switching techniques are analogues to barehand mid-air mode-switching techniques. We employ subtraction method to evaluate mode-switching times in our work.

2.0.2 Mode-Switching when Combining Touch and Mid-Air Input

Selecting mode-switching techniques becomes an interesting challenge when the multiple modalities are coexisting in a same interface. Specifically, we are interested in a combination of touch-based input and barehand mid-air gestural input. Researchers have investigated the use of a multi-touch tablet in VR and proposed several new mode-switching techniques as described below. While it is obvious to seek an empirical investigation of mode-switching techniques when barehand touch and mid-air input techniques are used together, but without having a well established interaction vocabulary identifying the candidate techniques is a difficult task. Moreover, state transition models for mouse, pen, and touch-based devices has been around for several years. In contrast, for mid-air input it is relatively unknown. As a result, we seek to derive an interaction vocabulary for this hybrid interaction modality. To begin with, we review the past literature on using a 2D surface or a multi-touch tablet in a mixed reality environment.

Tablet and Pen in Mixed Reality

Keefe et al. [134] explored precise mid-air strokes using a haptic-aided input technique for 3D sketching, and Arora et al. [8, 7] investigated the impact of the lack of a physical surface on drawing inaccuracies. Their work explored both 3D sketching in augmented reality (AR) using a mid-air pen-based drawing and 2D surface sketching. More recently, Aslan et al. [9] conducted a series of studies to gauge the potential of pen and mid-air input and noted that mid-air input should complement pen and touch-enabled tablets. However, compared to modern high-fidelity multi-touch tablets, pen input is essentially limited. It does not take the advantages of natural direct interactions with multiple fingers. So, the mode-switching techniques is also limited.

Although sketching using a pen is not our focus, we look at these results through the lens of interaction design. Specifically, the different kinds of mode-switching techniques used in the past work. For instance, the tablet's surface is better suit for continuous input, a tablet could help while drawing in mid-air, and orientating a tablet in arbitrary planes could provide haptic feedback in mid-air.

Tabletop and Hand Gestures in Mixed Reality

One of the most popular form of input in MR is direct hand interactions using hand tracking systems like LEAP. Hand tracking enables quick access to menu [35] as well as mid-air text entry [247, 167]. So, we review literature relevant to multi-touch tabletop surface combined with hand gestures in MR.

Benko et al. [16, 17] have explored the interaction space of using a tabletop surface along with hand gestures in a partially immersive environment. Specifically, they presented a collaborative archaeological analysis wherein user can combine speech, touch, and 3D hand gestures to interact within an immersive environment. They explored 2.5D gestures, wherein user can start interacting with digital contents on the tabletop surface and continue interacting with it in mid-air. As noted in the results, these form of interactions are effective and improved the overall user experience. Similarly, MockupBuilder [55] also demonstrated 2.5D interactions, which start on a planar surface and continues in mid-air, are highly promising for 3D modeling applications. They explored bimanual and continuous interaction on and above multi-touch surfaces to bring direct modeling techniques to semi-immersive virtual environments.

While these studies use tabletop surface for precise input, it does not have the same flexibility as a hand-held tablet, such as orientation tracking, mobility, a mid-sized display, and so forth. Being able to carry the tablet around facilitates interactions without being physically stuck in a certain position. Moreover, tablet enables mode-switching techniques which are suitable for different body posture. For instance, holding a tablet on forearm and interacting with dominant hand or bimanual interaction when placed on the planar surface. However, mode-switching performance might be influenced by body posture.

Tablet as a Prop in Mixed Reality

Moreover, many studies have investigated the use of physical props as a tablet to provide passive haptic feedback in mid-air. For instance, Linderman et al. [158] demonstrated the use of a passive-haptic paddle as a 2D input device for switching between different modes in VR. Results suggested that users prefer interfaces that provide a physical surface, and that allow them to work with UI widgets in the same visual field of view as the objects they are modifying. This result highlights the benefit of using handheld tablet in mid-air.

Poupyrev et al. [211] presented Virtual Notepad, a spatially-tracked, pressure sensitive tablet with pen and handwriting recognition software. Virtual Notepad explored hand writing as a new modality in immersive environment and it was used for text-based applications (note-taking, text input, and annotation using physical pen as a prop). Results highlighted the trade-off between the notepad size and the usability. While bigger notepad obscured the virtual environment and forced users to stretch their hands, a smaller notepad size made it hard to write on. Furthermore, Szalavari et al. [257] demonstrated the use of a passive tablet-like prop for 3D modeling applications. Interactions involved piercing 3D objects for selection, direct manipulation of 3D widgest on the tablet, changing the camera position using the pen, and so forth.

However, none of the past efforts have explored the simultaneous use of multi-touch tablets with barehand mid-air gestures.

Tablet Touch in Mixed Reality

Wang and Lindeman [288] presented the use of position tracked wand and a multi-touch tablet for 3D interaction tasks in AR environment. The wand was used to navigation in a virtual environment and the tablet was used to switch between editing modes. To navigate, the user pointed the wand in the desired direction and pressed a button on a wand to initiate the movement. Switching between different editing modes was facilitated on the tablet. For instance, pressing buttons on the tablet would select a 3D object or edit them based on the mode. Results indicated that using the wand and tablet simultaneously could have improved the user experience. Kim et al. [136] explored a scaled-down locomotion that allows a user to travel in a virtual world as their fingers slide on a multi-touch surface. Their system supported two modes. One finger touch would put the user in walking mode and two hands were used to switch to rotation mode.

While these studies focus on tablet's multi-touch input in the context of MR environment, they still did not consider input interactions beyond two fingers. Many recent studies highlight the utility of touch interactions beyond just two fingers. For instance, Wobbrock et al. [292] studied multi-touch 1080 gestures on a tabletop surface and proposed a user defined gesture set containing more than 20 gestures. Beyond surface gestures, Ruiz et al. [225] presented a user defined motion gesture set with a handheld smartphone. Further, in the context of barehand mid-air gestures, Piumsomboon et al. [204] presented a user defined gesture set of barehand mid-air hand gestures in AR environment. They studied 800 gestures suitable for 40 different tasks and identified six crucial design dimensions to build the taxonomy of gestures. Along these lines, Chen et al. [47] investigated 40 barehand mid-air gestures for manipulating 3D digital contents.

While the combination of 2D input and 3D input has received significant attention in the past, a comprehensive design space exploration of combining 2D and 3D input modalities is still missing. Especially, with modern multi-touch tablets and with 3D interfaces where barehand mid-air gestures are the most intuitive form of input.

Chapter 3

Touch-Based Mode-Switching

This chapter presents the results of a 36 participant empirical comparison of touch mode-switching. Six techniques are evaluated, spanning current and future techniques: long press, non-dominant hand, two-fingers, hard press, knuckle, and thumb-on-finger. Two poses are controlled for: seated with the tablet on a desk and standing with the tablet held on the forearm. Findings indicate pose has no effect on mode switching time and little effect on error rate; using two-fingers is fastest while long press is much slower; non-preferred hand and thumb-on-finger also rate highly in subjective scores. The experiment protocol is based on Li et al.'s pen mode-switching study, enabling a comparison of touch and pen mode switching. Among the common techniques, the non-dominant hand is faster than pressure with touch, whereas no significant difference had been found for pen. Our work addresses the lack of empirical evidence comparing touch mode-switching techniques and provides guidance to practitioners when choosing techniques and to researchers when designing new mode-switching methods.

3.1 Motivation

Most interfaces have multiple modes in which input is mapped to different actions. In a touch interface, the current mode can change how a single touch is interpreted: for example, it could draw a line, pan the canvas, select a shape, or enter a command. Switching between modes can be frequent, so finding optimum *mode-switching* methods is important. There have been numerous experimental investigations comparing mode-switching techniques for pucks, mice, and pens [128, 71, 153, 268, 107], but there has been no comprehensive analysis of mode-switching techniques for touch input.

This is surprising considering that a number of touch mode-switching techniques have been developed. Some are unique to touch since they rely on features such as multiple contacts [52, 218], using knuckles or other parts of the hand [96, 169], or characteristics of finger contact [30, 220]. Some touch mode-switching techniques are similar to those evaluated with pens, such as using pressure [182, 100] or using the non-dominant hand [74, 284]. However, generalizing pen-based empirical results to touch is highly speculative, considering distinct touch characteristics like reduced precision from “fat fingers” [19, 73, 232] and greater friction [50]. This lack of formal comparisons of touch mode-switching techniques may be one reason why current mode-switching methods for touch seem limited compared to other input methods.

In this chapter, we compare the performance of six mode-switching techniques for touch input on a tablet: the standard long press, pressing a button with the non-dominant hand, two-finger multi-touch, pressing hard, using the knuckle, and touching the thumb to the side of the finger. The investigated techniques include current methods, new methods recently made available in commercial devices, and techniques likely possible in the near future. Given the tablet mobility, we also control for two poses: seated with the tablet on a desk and standing while holding the tablet. Our evaluation protocol is based on Li et al.'s widely cited comparison of pen mode-switching [153]. This increases the replicability and validity of our work, and enables a discussion of touch versus pen mode-switching. Direct comparisons are possible for pressure, long press, and non-preferred hand, and to some extent thumb-on-finger and knuckle if considered analogues to Li et al.'s pen barrel button and eraser.

We conclude this chapter with the following results and insights:

- Techniques ordered from fastest to slowest are: two-fingers, non-preferred hand button, finger-on-thumb, knuckle or hard press, and much slower long-press.
- A sitting or standing pose has no effect on speed and little or no effect on errors (only hard press and non-preferred hand error rates showed some interaction with pose).
- Pressing hard was perceived to be the least accurate, one of the most fatiguing, and hardest to learn technique.
- Compared to Li et al.'s results for pen, our touch mode-switching timings and error rates are higher (except for knuckle compared to eraser).
- Our results can inform design, returning to the opening example: one finger could draw, two fingers pan, thumb-on-finger to select, and knuckle for a command menu.

3.2 Background and Related Work

We build on, and extend research developing new mode-switching techniques and formal experiments to analyse them.

3.2.1 Prior Mode Switching Techniques

Early mode-switching techniques focused on pucks, mice, and especially pens (styli). We provide a brief overview of pen techniques first since they have been arguably the most thoroughly studied and are the topic of Li et al.. We subsequently focus on multi-touch input, the most relevant to our work.

Pen Input

With pens, there is a common need to switch between an inking mode and a command input mode, but many techniques can be combined to support multiple modes.

Perhaps the most straightforward method to switch the pen mode, other than the classic "long press" with the nib, is to press a button. This can be single-handed, using the barrel button on the pen [153], a touch sensor below the palm of the writing hand [240], or more commonly with the other (non-dominant) hand [172, 3, 143, 107]. Using two hands exploits the benefits of bimanual interaction [128, 105]. Li et al. [153] found a physical button activated by the non-dominant hand was both faster and more accurate. A later study by Tu et al. led to similar results [268].

Having a well-positioned button on a device is not always a practical solution due to the following reasons. First, there is often a need to trigger multiple mode switches; having multiple buttons on a tablet would be hinder the usability. Second, fixing the position of such buttons might prove to be a difficult task as user's body posture would affect the way they are holding the tablet. Additional techniques have been proposed to overcome those limitations. On pen and touch tabletop systems, there is a large body of work examining different postures performed with the non-dominant hand on the surface to activate command modes for the pen held by the dominant hand [36, 112, 173]. Other techniques include short stroke gestures [104, 150, 88], pressing firmly or lightly [153, 268, 212], stylus rolling [22], contacting with different parts of a multi-faceted crayon [282], and pen-holding postures [242, 109].

Touch Input

Many mode-switching techniques designed for pens or other devices have been applied to direct touch input. For instance, typing capital letters by holding the shift key with the other hand is a simple form of non-preferred hand mode activation. Even the shape of the non-dominant hand [290, 295, 97] or the number of fingers used [294] can trigger a mode change. BiPad [284] and SPad [74] explore the possibility of using the hand holding a tablet to activate different modes by pressing soft buttons. Using soft buttons is a scalable solution compared to using hard buttons on a tablet, as the position of the soft buttons can be easily changed to accommodate user's body posture. Pressure [182, 100, 218] and grip-based [80] controls have also received much attention. Some of the latest mobile devices integrate pressure-based technology and functionality (3D Touch, Force Touch, Press Touch etc.) [267]. The number, shape and mobility of fingers afford further interaction possibilities. Multiple fingers performing similar path movement (two- three-finger swipes etc.) can be used to trigger different actions [151, 294] and if individual fingers can be identified, interactions can be made finger-specific [52]. Further properties of finger input such as contact size [30], slight rolling movements [220] and which part of a finger touches the display [96] can also be recognized to support mode-switching or general interactions.

Mode Switching Analysis

With many possible mode-switching techniques, it is no wonder researchers have attempted to develop models and evaluation protocols to rigorously assess their performance under different settings. Using the non-dominant hand for pen mode switching has been studied in detail by Ruiz et al. [223] who developed a temporal model. They use the Hick-Hyman Law to show the asymptotic cost of adding additional non-dominant hand modes to an interface is a logarithmic function of the number of modes. Experiments indicated that the model is an accurate predictor of the time taken to perform a non-preferred hand mode switch for interfaces containing between two and eight modes when modes are equiprobable. Lank et al. [143] show concurrent mode-switching is the fastest. They conducted an experiment comparing three variants of controlling the mode— pre-gesture mediation, post-gesture mediation, and concurrent mediation. In pre-mediated mode, to draw a moded gesture subjects must depress the mode-switch button prior to the beginning the gesture and hold it until beginning to draw; button state at pen-down indicates mode. In post-mediated mode setting, subjects can press the mode-switch button any time before or during the gesture; button state at pen-up event indicates mode. In the concurrent mode switching, mode can be altered during the beginning of a gesture. Results contradict serial assembly of the human motor control described in the popular 'Kinematic Chain model', demonstrating the non-preferred hand mode initiation is an instance of motor control level interaction where bimanual interference is not serial. Ruiz and Lank [224] who explore aspects of overlapping mode-switching and performance footprints with multiple modes.

To compare performance of mode-switching techniques, a common methodology is Dillon et al's "subtraction technique" [62]. It determines the precise cost of mode-switching by subtracting the time to perform the same series of tasks using a single mode and when alternating between two modes. It is the approach for comparing pen mode switching techniques in Hinckley et al. [107], Song et al. [242], and Li et al.'s[153] highly cited comparison on which we model our work.

We are unaware of a comprehensive study systematically examining and comparing mode-changing techniques for direct touch input. The touch techniques explicitly or implicitly used to trigger mode changes that have been proposed have mostly been superficially or individually evaluated for non-frequent mode-switches. Therefore, it is not clear how well they fare compared to each other and in a context, where state changes are very frequent. Furthermore, the results for pen-based mode-switching may not transfer to touch input, not to mention that touch input enables other techniques such as multiple touches not applicable to pens.

3.3 Mode Switching Techniques

We chose six mode switching techniques among those in current use, described in previous research, or soon plausible given emerging sensor capabilities. Some are analogous to the pen mode switching techniques tested by Li et al. [153]. All techniques were designed for a tablet when placed on a desk, or when supported with the non-dominant arm.

3.3.1 Long Press

Performing a long press (also called "press-and-hold" or "dwelling") is a common method to trigger command modes in current touch interfaces. For example, Android and IOS use a long press to organize app launch icons. We use a long press duration of 500ms , the default Android setting. Li et al. also included a pen long press, but used a 1000ms duration.

Long press detection begins after touch down with a "hold detection phase": as long as finger movement remains within a 3mm radius bounding circle centred on the initial touch point, the finger is considered held still. A circle 25mm in diameter is displayed around the touch point showing the progression towards the 500ms duration. If the finger remains in the box for 500ms, the mode is activated. If the finger exits the box before that time, the hold detection phase restarts with a new bounding circle centred on the new finger position. Our 3mm radius bound is twice that used by Li et al. to account for touch sensor noise. Once detected, the mode remains engaged until touch up regardless of subsequent finger movement. We did not implement a second "hold through" phase to cancel the mode switch like Li et al. because, to our knowledge, this is not used on touch input devices or needed for the experiment.

3.3.2 Two-Finger Multi-touch

One of the simplest distinctions for touch input is whether one or two fingers contact the display at the same time. This is common in Android and IOS, and researchers have used two fingers to activate marking menus [151] and distinguish between dragging and hovering in the DTMouse technique [68]. There is no equivalent technique with pen input.

A two-finger touch is detected when two correlated touches occur soon after initial touch down. For our experiment, two touches must be detected before crossing into the first rectangle (typically less than 80ms). To remain comparable with other single touch techniques, a single input position is defined using the midpoint between touch points. We selected the midpoint based on pilot tests examining the perceived input point for two touches. This positioning is also used for DTMouse in hover mode. Once detected, the mode remains engaged until touch up regardless of the number of touches.

3.3.3 Non-Preferred Hand

Touch interfaces can support using the non-preferred hand to activate a mode with soft buttons, a simple example is holding the SHIFT key while typing. Li et al. found that pressing a physical button was one of the fastest ways to activate a mode with pen input. Our equivalent technique uses a rendered touch button since it is more practical with current tablets.

To engage the mode, the non-dominant hand presses and holds a 45 × 25mm button before the dominant hand touches down. Once the touch down event occurs, the mode remains engaged until both the dominant-hand touch up event occurs *and* the mode switch button is released. We require the button to be pressed before the dominant hand touch down to be consistent with Li et al.. The mode button location is dependent on the

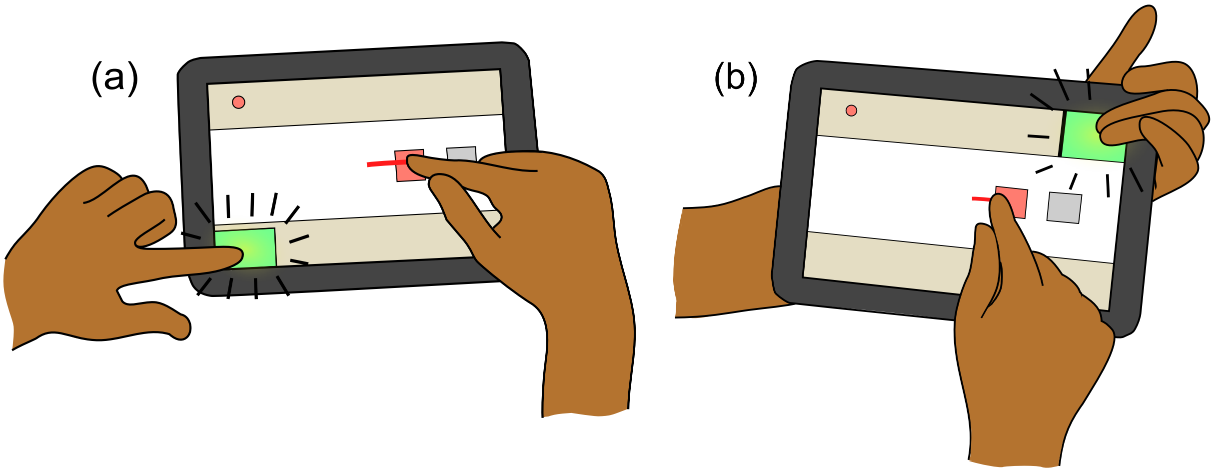


Figure 3.1: Position of the mode-switch button (in green) activated by the non-preferred hand when (a) sitting; and (b) standing.

participant's handedness and whether the tablet is supported by a surface (e.g. sitting at a desk) or supported by the non-dominant forearm (e.g. when standing). When supported by a surface, the button is displayed at the bottom-left (or bottom-right) corner (Fig. 3.1-left). When supported by the non-dominant arm, the button is displayed at the top-right (or top-left) corner (Fig. 3.1-right). This enables the user to reach and tap the mode switch button comfortably with the fingers of the hand holding the device, a common posture reported by Wagner et al. [284] We fixed button locations and sizes for our experiment, but techniques exist to automatically detect how a mobile device is held so such mode-switch button could be positioned accordingly [80].

3.3.4 Hard Press

Pressure-based touch interaction has been described in previous work [18, 21, 100, 218] and recent technology developments suggest pressure sensing will be supported on commercial touch devices in the near future [267]. Using pressure-based mode-selection for pens has been well studied (e.g. [212]) and it was a method evaluated in Li et al.'s experiments.

Most current touch devices report a simulated pressure reading based on the size of the touch contact. On vision-based tabletops the actual contact size is captured by a camera [21], but capacitive devices estimate it from the signal strength. We found simulated pressure with capacitive tablets is unreliable due to factors like skin moisture, relative humidity, and body hydration. To get a true measure of pressure, we initially experimented with placing multiple external force-sensitive resistors under the tablet and training a classifier to recognize touch events with normal and hard pressure. This worked reasonably well on a desk, but designing a housing for accurate sensing when standing proved difficult. Instead we detect hard presses indirectly, based on muscle tension sensed using a MYO electromyographic (EMG) armband. Benko et al. used the same technique with a similar EMG sensor [18]. Note that our objective is to simulate a future pressure sensing technique in our experiment; we are not proposing that people wear a MYO armband when using a tablet.

A simple threshold-based classifier is trained for each participant (Li et al. used a global threshold for pressure across all participants, but found it unreliable). To train, the participant crosses through five rectangles, alternating between a normal touch and a hard press touch according to rectangle colour. This is repeated 4 times. The data from the 8 armband EMG sensors are smoothed using the one-euro filter [43] and synchronized with the touch events and expected type of touch. The median and standard deviation of each sensor signal for normal touches and hard presses is calculated using events logged from touch down until the rectangle is entered. All sensors where the hard press median minus two standard deviations is greater than the normal touch median plus two standard deviations are considered *differentiating* sensors. If less than two differentiating sensors are found, the armband is adjusted and the training repeated. Otherwise, each sensor is assigned a threshold equal to the hard press median minus two standard deviations (EMG signals for hard press are always greater than normal press). Once trained, a touch is considered a hard press if two or more of the differentiating electrodes exceed the thresholds determined in training. A 5-person pilot found the method was almost 99% accurate.

To use hard press for mode-switching, sufficient pressure must be applied to the tablet screen to cross the threshold. During the experiment, a hard press must be identified before crossing into the rectangle. The mode is disengaged upon touch up.

3.3.5 Thumb-on-Finger

Pressing a "barrel button" is a classic way to change the mode using a pen, and this technique was included in Li et al's study. We approximate barrel button mode-engagement for touch input with a thumb press on the index finger, similar to techniques used for mid-air clicking [89, 281] and NanoStylus [298]. We anticipate that with technologies such as Project Soli [83], these types of gestures will be able to be sensed.

In our experiment, we use a wearable device with a force-sensitive resistor (FSR) taped

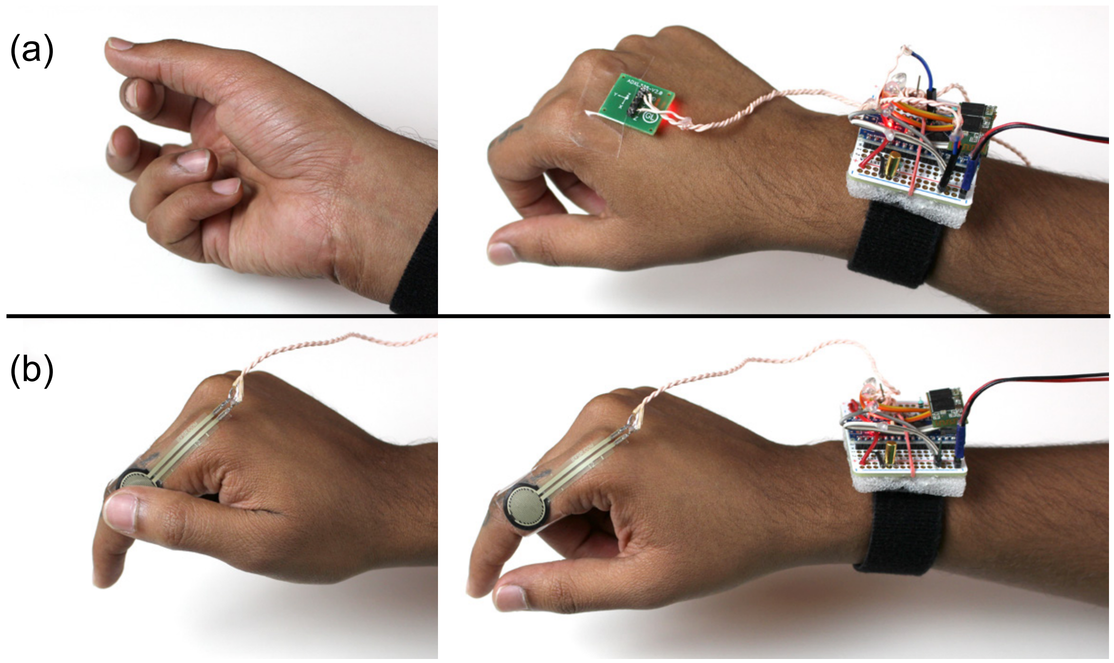


Figure 3.2: Hardware to robustly detect future input actions for the purpose of the experiment: (a) knuckle touches using accelerometer; and (b) thumb-on-finger using pressure sensor attached to index finger.

to the proximal phalanges of the index finger (Fig. 3.2 b). We ensured this apparatus did not impede natural touch interaction with the tablet. The FSR is 12.7 mm in diameter and 0.47 mm thick. The sensing range is 0 to 175 psi and we used a global threshold of 51 psi to detect when the thumb lightly contacts the finger. The FSR is connected to an ATmega328 Arduino strapped to the wrist. The Arduino sends pressure readings to the tablet over Bluetooth. To reduce weight, an external battery is connected to the wearable device with a lightweight wire.

The mode is activated by pressing the thumb to the side of the index finger before touching down. Once the touch down event occurs, the mode remains active until both the touch up event occurs *and* the thumb is released from the finger.

3.3.6 Knuckle

Using the knuckle for touch input has been described in Marquardt et al. [169] and Tapsense [96]. Knuckle-sensing is already offered on some smartphones [77]. Turning the hand over to engage the knuckle also bears some similarity to using the eraser end of a pen, a mode-switching technique included in Li et al.'s study.

Our tablet does not sense knuckles natively, so we simulate a future knuckle sensor. An ADXL335 3-axis accelerometer mounted on the back of the hand with tape detects wrist rotation. Specifically, the mode is switched when the z axis of the accelerometer exceeds a 90 degree angle (it is 0 degree when the sensor is horizontal). This simple threshold is sufficient to differentiate between knuckle and normal finger pad touches. The accelerometer is connected to the same wrist-mounted apparatus used for thumb-on-finger sensing. In our experiment, all but one participant used their middle finger knuckle to perform this technique. Given the mechanics of the movement, the mode must be engaged before the first touch. The mode remains active until both the touch up event occurs *and* the wrist rotates back to the finger pad touch orientation.

3.4 Experimental Setup and Apparatus

The goal of this experiment is to compare mode-switching time, error rates, and subjective ratings for the six techniques described above. Given the mobile nature of tablets, both seated and standing poses are tested. The experiment task and design is a near-replication of Li et al.'s [153] pen mode-switching study.

3.4.1 Participants

We recruited 36 participants (mean age 24.1 SD = 2.4, 8 women, all right-handed). 24 participants had experience using multi-touch tablets. A $10 remuneration was provided. Each participant completed a questionnaire before the experiment began and after the experiment was conducted. A copy of these pre and post questionnaires are included in Appendix A.

3.4.2 Apparatus

The experiment was performed on a Google Nexus 10 tablet (1.7 GHz Cortex A15 CPU with 1 GB RAM) running Android OS 5.1.1. The tablet's 264 × 178 mm display has a resolution of 2560 × 1600 px, a density of 11.8 px/mm (300 PPI). The device weighs approximately 603 grams. The experiment task code was written in Processing using the

Android export library. Using Ng. et al.'s method [190] and a 240 fps camera, end-to-end latency was 100 ms, comparable to current apps.

3.4.3 Tasks

Our experimental tasks are closely based on Li et al. [153]. Five 20 × 22 mm rectangles are crossed in succession where the 20 mm ends form two parallel crossing targets (Fig. 4.3). Note that 20 mm crossing targets are 63% larger than the minimum size recommend by Luo and Vogel to achieve a 4% error rate [164]. The rectangle is a simplification of the pie section used by Li et al.. All five rectangles are displayed in a horizontal row, all oriented in the same direction with the required crossing direction indicated by a white arrow.

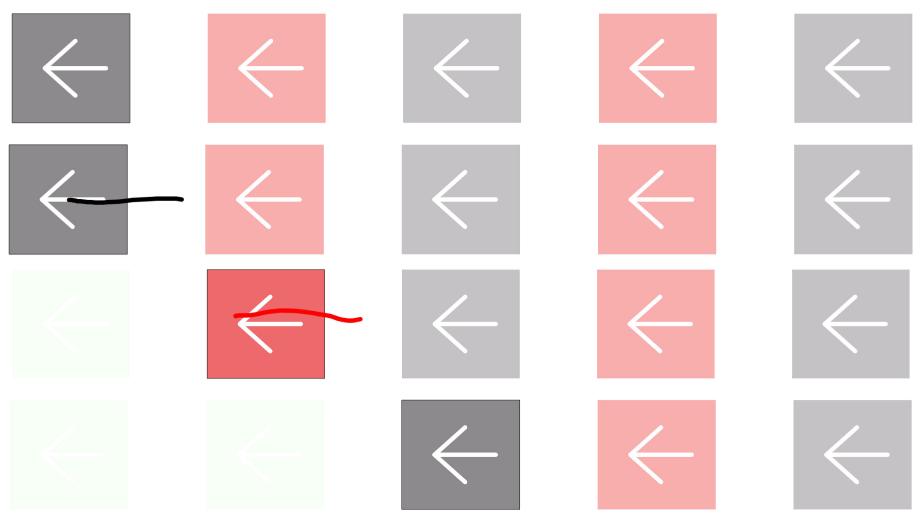


Figure 3.3: Compound task: five oriented rectangles (180° shown) are crossed while alternating between default and command modes. Each row above is a screen capture of the task (top to bottom): about to cross first baseline target; crossing first target; crossing second "moded" target; about to cross third baseline target. The baseline task looks identical except all rectangles are grey and only the default mode is used.

There are two task variations. In the *baseline* task, five grey rectangles are shown and the participant crosses them using standard touch input. In the *compound* task, the five rectangles alternate between grey and red, with the first rectangle grey. The participant must cross each grey rectangle using standard touch input and each red rectangle using the specified mode-switching technique. All touches leave a trail for feedback, black for standard touches and red when the mode is engaged. Note that the red or black trails function only as an abstract representation of two different modes. A small circle in the top-left corner of the display also turns red when the mode is engaged. If there is any crossing or mode error, a beep sounds and the rectangle must be crossed again to continue. Error detection and classification are described below.

3.4.4 Design and Procedure

The experiment is a repeated measures mixed design. The participant POSE while using the tablet is a between-subjects factor (SIT or STAND). Half of the participants completed all tasks while seated with the tablet placed flat on a table and the other half completed all tasks while standing with the tablet held on their non-dominant forearm. The mode-switching TECHNIQUE is a within-subjects factor with levels corresponding to the six mode-switching techniques (LONGPRESS, TWOFINGER, NONPREF, HARDPRESS, THUMB, KNUCKLE).

Participants were randomly assigned to a POSE condition and TECHNIQUE order was counter-balanced using a 6 × 6 Latin square. For each TECHNIQUE, there was a 1 to 3 min training period (after wearable hardware was attached and calibrated for HARDPRESS, THUMB, and KNUCKLE). Once training was over, the participant completed 9 BLOCKS of tasks. Odd numbered blocks were entirely baseline tasks and even numbered blocks entirely compound tasks. Before each block, the participant had to press a start button. Each block presented the task using 4 crossing directions (N, E, S, W) in random order. Note that Li et al.'s design had 8 directions, but they report no significant differences. Our four cardinal directions are representative of common actions like swiping, and a reduced number of directions enabled all six techniques to be tested in less than 1 hour with minimal fatigue. Participants were allowed to take breaks between blocks.

In sum there were: 6 TECHNIQUES × 9 BLOCKS (5 baseline, 4 compound) × 4 directions × 5 rectangle crossing = 1,080 rectangle crossings per participant.

3.4.5 Quantitative Measures

We calculated three measures from the experiment event logs, described below.

Errors and Error Rates

Like Li et al. [154], we identify three types of errors. A *crossing error* occurs if the touch stroke did not cross both ends of the rectangle in the correct order and direction. This captures errors related to crossing accuracy. An *out-of-target error* occurs if the touch stroke did not intersect with any part of the rectangle. This most often captures a case when the participant intentionally aborted a rectangle crossing. These errors are only possible on the current rectangle, strokes intersecting with other rectangles are ignored. A *mode error* occurs when the wrong mode is used to cross a rectangle. In other words, stroking a grey rectangle with red or vice versa. Mode errors are only possible in compound tasks.

We further distinguish between mode-in and mode-out errors. A *mode-in error* occurs when the participant fails to transition from standard touch input to the specified mode-switching technique. This is detected during the second or fourth rectangle crossing. A *mode-out error* is when the participant fails to transition from the specified mode-switching technique to standard touch input. This is detected during the third or fifth rectangle crossing. Finally, a *combined error* occurs if any of the errors above happen. Each of these error types are recorded as an indicator variable: 1 if the error occurred and 0 otherwise. The mean value of one type of indicator variable across trials produces the corresponding error rate.

Crossing Time

The crossing time is the duration between the touch up event after the previous rectangle was crossed until the touch up event after the current rectangle is crossed. There are four measurable rectangle crossings per task.

Mode-switching Time

Naively, one might directly compare crossing times in the baseline task with crossing times in the compound task (where a mode switch was required). However, both crossings share a common overhead of moving from the end of the previous rectangle to the start of the current rectangle. Therefore, we use the "subtraction method" used by Li et al. (adopted from Dillon et al. [62]) to isolate mode-switching time.

The method defines three cycles during a task. The first cycle is from the moment the start button is pressed until the touch up event after crossing the first rectangle. The second cycle begins immediately after, ending when all fingers are lifted after crossing the third rectangle. The third cycle begins immediately after, ending when all fingers are lifted after crossing the fifth rectangle. The second and third cycles are *full cycles*. During the compound task, each full cycle captures a complete mode-switch operation: the participant switches into a mode using the specified technique, crosses a rectangle, switches out of the mode, crosses another rectangle, and lifts their finger(s). The purpose of the first cycle is to ensure standard touch input is used before the second cycle. In each block, there are 8 full cycles (two cycles per direction). For each technique, each participant completes 32 full cycles with mode switching (in the 4 compound task blocks) and 40 full cycles with standard touch input only (in the 5 baseline task blocks).

The subtraction method isolates mode switching time using mean times from second and third cycles. For each block, the mean time for second and third cycle is calculated using error-free cycles (recall there are four task directions per block, so each full cycle is repeated four times). The mode-switch time is calculated by subtracting the mean full cycle time of two adjacent baseline blocks from the mean full cycle time from a compound block. In total, this provides 8 mode-switch time measurements per-participant, per-technique (2 per block).

3.5 Results

All results, including subjective ratings, are continuous, so the same analysis procedure was used for all data. Specifically, we performed repeated measures ANOVA and pairwise t-tests with Bonferonni corrections when main or interaction effects were found. For interaction effects, we restricted pairwise tests to comparing means across factor dimensions independently. When the assumption of sphericity was violated, degrees of freedom were adjusted using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$). Non-normal skewed distributions, were corrected using a log transform or Aligned Rank Transform [291] depending on the severity and direction of skewness.

3.5.1 Data Pre-Processing

We examined error-free *crossing times* to identify outliers more than 3 standard deviations from the mean for each TASK divided by POSE. This removed 4.6% of the rectangle crossing trials (3.8% to 6.7% per technique), comparable to similar touch experiments [164].

Using the remaining error-free full cycles, we used the subtraction method described above to calculate *mode-switch times*. Visual inspection of the mode-switch times distribution suggested non-normality, confirmed by a Shapiro-Wilk and Anderson–Darling tests. To compensate, we log-transformed all data points for mode-switch time. There were 33 data points with slightly negative mode-switch times. This only appeared for TWOFINGER, the fastest technique. To compensate, we first added 306ms to all times to guarantee positive values required by the log function. Note that this log transformed data is used for statistical tests involving mode-switch time. All times presented in the paper are actual measured values. Error rate distributions did not suggest non-normality.

3.5.2 Learning Effects

To determine if performance changed during the four compound blocks, we tested for effects of POSE ×TECHNIQUE ×BLOCK on *mode-switch time* and *combined error rate*. We found no statistically significant interaction involving block indicating no learning effect across blocks. This matches the performance stability noted by Li et al. All blocks are used in subsequent analyses.

3.5.3 Mode-Switching Time

We expected POSE would alter how the techniques were performed, but there was no significant main effect of POSE, or POSE ×TECHNIQUE interaction effect on MODE-SWITCH TIME. There was a significant main effect of TECHNIQUE ($F_{5,170} = 109.52$, $p < .0001$, $\eta_p^2 = .76$). Post hoc tests found all TECHNIQUES significantly different $p < .0001$, except HARD PRESS and KNUCKLE. Ranking TECHNIQUES from fastest to slowest mode-switch time: TWOFINGER (222ms); NONPREF (311ms); THUMB (408ms); KNUCKLE and HARDPRESS (500ms and 568ms respectively, not significantly different); and LONGPRESS (1244ms). LONGPRESS is more than twice as slow as the next fastest techniques and more than five times slower than the fastest technique.

Comparing the switching times of techniques used in Li et al., we notice that our values are systematically higher. For NONPREF and HARDPRESS, the authors report mean times of 139ms and 284ms respectively, which are roughly half of our values. Tu et al. [268] also evaluated those techniques and while the pressure-based technique is reportedly even faster (mean time 228ms), the timing for their version of NONPREF, 304ms, is very similar to ours.

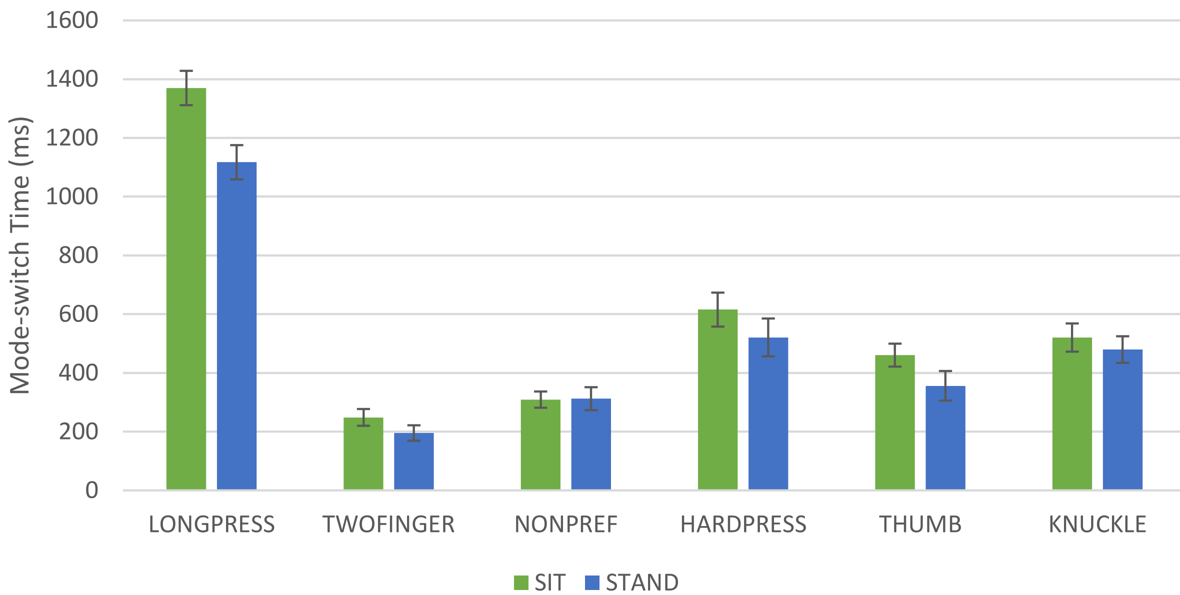


Figure 3.4: Mean mode-switch times by POSE and TECHNIQUE with 95% CI.

Although there was no effect of POSE, we examine TECHNIQUE by POSE given our a priori control. For each POSE, we ran a one-way ANOVA for TECHNIQUE on *mode-switch time*. Main effects were found for SIT ($F_{5,85}$ = 69.60, p < .0001, η_p^2 = .80) and STAND ($F_{5,85}$ = 44.42, p < .0001, η_p^2 = .72). The pattern of post hoc differences was very similar to the TECHNIQUE main effect (all $p < .0001$). The only difference is for STAND: there was one other non-significant difference between THUMB and NONPREF. Without that one exception, the order of TECHNIQUES from fastest to slowest is consistent between POSES and with combined poses (see Fig. 3.4).

3.5.4 Error Analysis

Before examining specific error rates for techniques and poses, we note that the *overall error rate* for baseline crossing cycles is a 4.1% with no detectable differences between POSE. This rate suggests participants were balancing speed and accuracy [302] and is low enough to suggest using the subtraction method is valid. Li et al. do not report baseline rates.

Unlike mode-switch time, we did find a significant interaction for POSE ×TECHNIQUE on *combined error rate* ($F_{3.40,115.66}$ = 3.02, p = .012, η_p^2 = .081). There was a significant

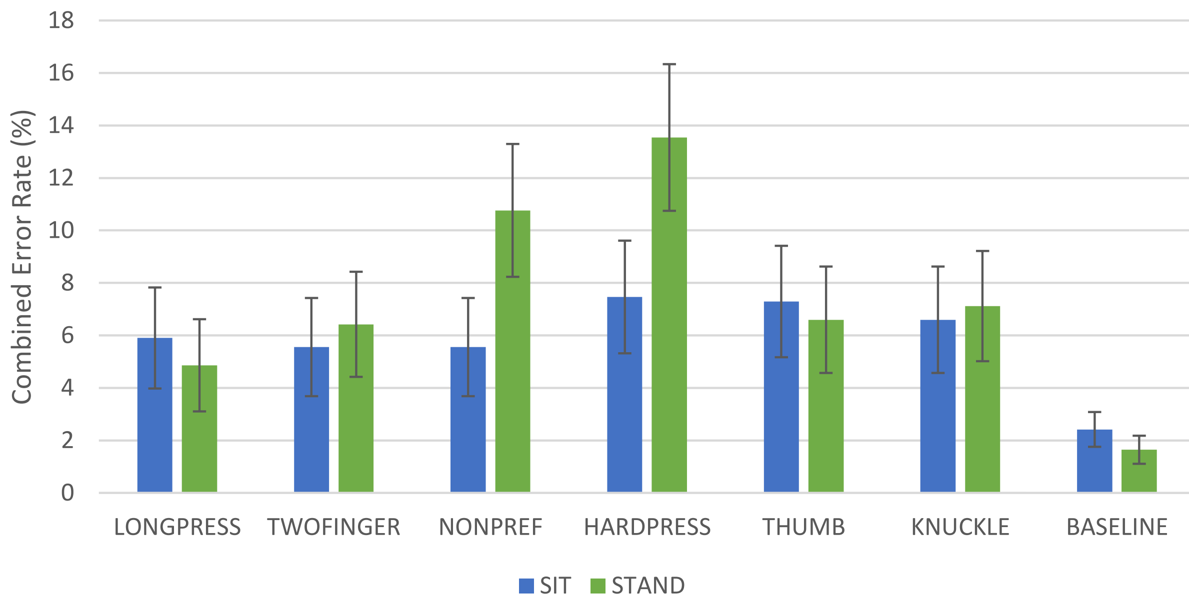


Figure 3.5: *Combined error rate* for TECHNIQUES by POSE

main effect for TECHNIQUE as well, but the interaction with pose is most relevant. To determine if the technique combined error rate was significantly different between poses, we performed six pairwise tests. With Bonferonni correction, only a borderline difference in HARDPRESS $p = .06$ exists. Like mode-switch time, it appears that POSE has little or no effect on overall error rate.

Continuing to explore the significant interaction, we examine if the technique *combined error rates* were significantly different using pairwise tests between techniques when considering SIT and STAND separately. For SIT, we found no significantly different techniques. The measured rates ranged between 4.8% and 7.5%. Again, we observe a contrast with Li et al.'s reported error rates, which are systematically lower. Their hold technique led to the most errors (due to a slippery screen) and NonPrefHand to the lowest rates. Tu et al. once more show different results (Holding being the least error-causing technique in their experiment) and error rate ranges that are partially closer to ours.

For STAND, we found significant differences between TWOFINGER and HARDPRESS, LONGPRESS and HARDPRESS (both $p < .05$), and a borderline difference between NONPREF and LONGPRESS ($p = .05$). The measured rate for HARDPRESS is 13.5% and NONPREF is 10.8%, with the remaining techniques ranging between 3.3% and 7.1%. This provides more evidence that, relative to the other techniques, HARDPRESS and NONPREF are harder to perform when standing.

We also investigate the effect of POSE and TECHNIQUE on specific types of errors (illustrated by POSE in Fig. 3.6 & 3.7).

For *crossing error rate*, we found a main effect for TECHNIQUE ($F_{5,170} = 3.97, p < .001, \eta_p^2 = .104$). Post hoc tests showed NONPREF had a higher rate (3.9%) than HARDPRESS (1.2%) and THUMB (1.5%). Measured rates were between 1.2% and 3.9%.

For *out-of-target error rate*, we also found a main effect for TECHNIQUE ($F_{5,170} = 5.41$, $p < .0001$, $\eta_p^2 = .14$). Post hoc tests showed the rate for THUMB (1.0%) was lower than KNUCKLE (2.6%), NONPREF (5.0%), and TWOFINGER (5.9%). Also HARDPRESS (2.5%) was lower than TWOFINGER. Measured rates were between 1.0% and 6.3%. Note that Li et al. had almost no out-of-target errors.

For *mode-in error rate*, TECHNIQUE had a main effect but we focus on the more relevant significant POSE ×TECHNIQUE interaction ($F_{5,170} = 2.44$, $p = .036$, $\eta_p^2 = .066$). For SIT, post hoc tests showed TWOFINGER (1.0%) was lower than LONGPRESS (3.1%). For STAND, KNUCKLE (0.7%) was lower than HARDPRESS (6.9%) (all $p < .05$). Measured rates were between 1.0% and 5.0% for SIT and between .07% and 6.9% for STAND.

For *mode-out error rate*, TECHNIQUE had a main effect ($F_{3.56,121.07} = 6.28$, $p < .001$, $\eta_p^2 = .16$). Post hoc tests showed TWOFINGER (0.3%) was lower than NONPREF (2.4%) and HARDPRESS (3.1%) HARDPRESS was higher than THUMB (0.9%) and LONGPRESS (0.6%) (all $p < .05$). Measured rates were between 0.3% and 3.1%. There is some evidence that NONPREF and HARDPRESS both are more difficult to disengage.

3.5.5 Subjective Ratings

After completing trials for all techniques, participants provided subjective ratings of the techniques with respect to six aspects: ease-of-learning, ease-of-use, accuracy, speed, eye fatigue, and hand fatigue. All ratings were on a continuous numeric scale from 1 to 5, with 1 being the worst score (e.g. low accuracy, hard to learn, very fatiguing) and 5 the best (e.g. high accuracy, easy to learn, not fatiguing).

Table 3.1 summarizes subjective ratings by POSE. The distributions for ratings was non-normal due to high negative skewness, so the values were transformed using the Aligned Rank Transform method [291]. ANOVAs performed on this transformed data did not reveal any significant main effects or interactions involving POSE on any any of the ratings. However, there are significant main effects for TECHNIQUE regardless of POSE. We report the main results of pairwise comparisons between TECHNIQUE for each rating.

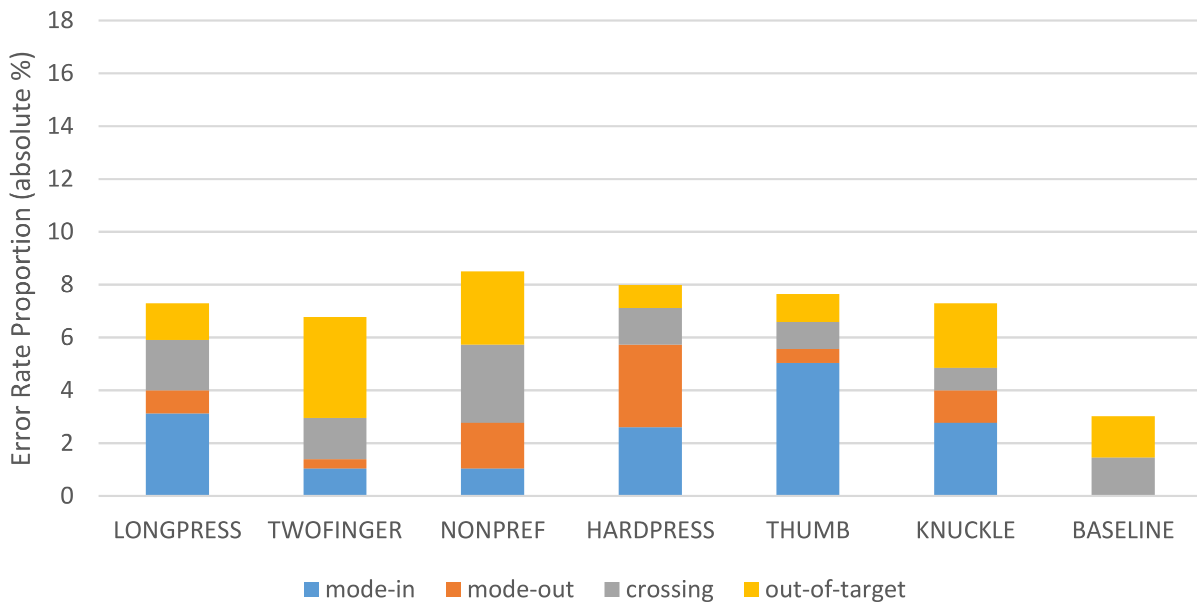


Figure 3.6: Proportion of specific error rates for TECHNIQUES for SIT. Note more than one type of specific error can occur during a cycle.

- For *hand fatigue* and *ease-of-use*, HARDPRESS and KNUCKLE were reported respectively more tiring and less easy to use compared to the other techniques (all $p < .032$). This is understandable, given the pressure and wrist efforts required. Participant feedback confirmed the difficulty and physical demand of HARDPRESS (seven people) with one participant commenting that it almost felt like breaking the tablet. As for KNUCKLE, two participants reported it was tiring and two pointed out that it resulted in increased occlusion; however, two people also said it was "fun". For NONPREF, four participants remarked that the technique required well-timed coordination and thus getting used to to be efficient. For the standing position, we also observed that participants with small hands (two in particular) sometimes had trouble reaching the button with the fingers of the non-preferred hand across the bezel of the tablet.

- In terms of *ease-of-learning*, HARDPRESS was rated significantly harder to master than LONGPRESS and TWOFINGER ($p = .0043$ and $p = .0102$ respectively). We believe this is because participants had to learn to adjust touch pressure levels to be able to activate the two different modes reliably as well as because of the extra training required for the EMG classifier.

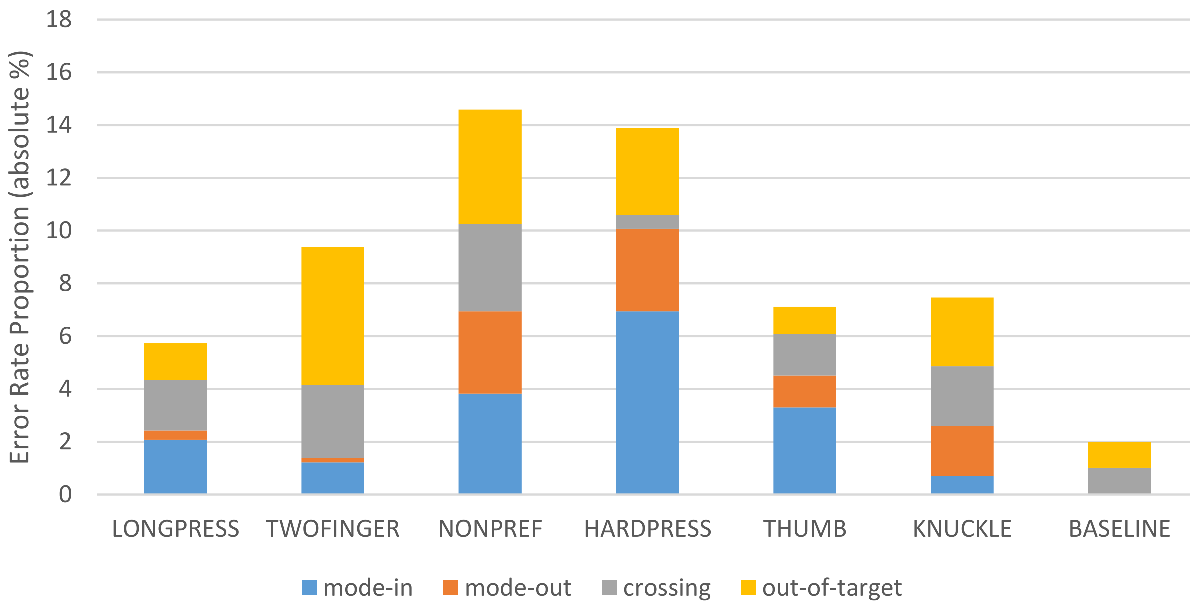


Figure 3.7: Proportion of specific error rates for TECHNIQUES for STAND. Note more than one type of specific error can occur during a cycle.

- With respect to *accuracy*, HARDPRESS was consistently perceived as having the lowest precision ($p < .007$ compared to all other techniques). Note that HARDPRESS and NONPREF have the highest overall error rates, yet HARDPRESS was rated more accurate. Hence, there appears to be an increased perception of poor accuracy with HARDPRESS. For TWOFINGER, we observed that, during training, participants needed a short adaptation time to position their two fingers for the ink to appear at the desired spot.

- Finally, regarding *speed*, TWOFINGER was rated significantly faster than all other techniques except THUMB ($p < .01$), THUMB was judged faster than HARDPRESS, KNUCKLE and LONGPRESS ($p < .013$) and LONGPRESS considered significantly slower than all techniques except HARDPRESS ($p < .03$). All those results are consistent with the time measurements.

3.6 Discussion

Our results provide evidence that regardless of whether a tablet is used flat on a desk, or held by the non-dominant arm while standing, the performance characteristics and

SIT	LONG PRESS	TWO FINGER	NON PREF	HARD PRESS	THUMB	KNUCKLE
Learning	$4.4_{\pm.21}$	$4.4_{\pm.20}$	$4.7_{\pm.13}$	$3.7_{\pm.32}$	$4.6_{\pm.14}$	$4.1_{\pm.23}$
Ease-of-Use	$3.9_{\pm.24}$	$4.2_{\pm.22}$	$4.0_{\pm.28}$	$3.0_{\pm.31}$	$4.3_{\pm.24}$	$2.7_{\pm.26}$
Accuracy	$4.3_{\pm.20}$	$4.1_{\pm.26}$	$4.5_{\pm.16}$	$3.2_{\pm.26}$	$4.3_{\pm.20}$	$4.2_{\pm.18}$
Speed	$2.6_{\pm.23}$	$4.3_{\pm.22}$	$4.2_{\pm.17}$	$3.4_{\pm.30}$	$4.2_{\pm.20}$	$3.4_{\pm.33}$
Eye Fatigue	$4.7_{\pm.15}$	$4.8_{\pm.10}$	$4.8_{\pm.12}$	$4.4_{\pm.25}$	$4.7_{\pm.17}$	$4.5_{\pm.21}$
Hand Fatigue	$3.7_{\pm.27}$	$4.1_{\pm.27}$	$4.4_{\pm.22}$	$3.0_{\pm.32}$	$4.0_{\pm.26}$	$3.0_{\pm.31}$
Combined	$3.9_{\pm.1}$	$4.3_{\pm.1}$	$4.4_{\pm.1}$	$3.5_{\pm.1}$	$4.4_{\pm.1}$	$3.6_{\pm.1}$
STAND	LONG PRESS	TWO FINGER	NON PREF	HARD PRESS	THUMB	KNUCKLE
Learning	$4.6_{\pm.22}$	$4.7_{\pm.16}$	$3.9_{\pm.32}$	$3.8_{\pm.26}$	$4.3_{\pm.24}$	$4.0_{\pm.27}$
Ease-of-Use	$3.9_{\pm.22}$	$4.2_{\pm.24}$	$3.2_{\pm.36}$	$3.0_{\pm.31}$	$3.9_{\pm.29}$	$3.2_{\pm.33}$
Accuracy	$4.5_{\pm.18}$	$4.2_{\pm.19}$	$3.9_{\pm.33}$	$3.1_{\pm.28}$	$4.1_{\pm.23}$	$3.8_{\pm.26}$
Speed	$2.8_{\pm.30}$	$4.6_{\pm.20}$	$3.4_{\pm.30}$	$3.3_{\pm.27}$	$4.1_{\pm.26}$	$3.4_{\pm.25}$
Eye Fatigue	$5.0_{\pm.0}$	$5.0_{\pm.0}$	$4.9_{\pm.05}$	$4.9_{\pm.05}$	$5.0_{\pm.0}$	$4.7_{\pm.17}$
Hand Fatigue	$4.3_{\pm.30}$	$4.5_{\pm.18}$	$3.6_{\pm.32}$	$2.9_{\pm.34}$	$4.3_{\pm.26}$	$3.6_{\pm.30}$
Combined	$4.2_{\pm.1}$	$4.5_{\pm.1}$	$3.8_{\pm.1}$	$3.5_{\pm.1}$	$4.3_{\pm.1}$	$3.8_{\pm.1}$

Table 3.1: Mean subjective ratings: SIT (top) and STAND (bottom).

subjective impressions of these techniques are comparable. This may bolster the validity of other non-mode-switching touch input studies evaluated only when a tablet is laid flat on a desk (e.g. [165, 82, 187]). However, more styles of touch input need to be tested in sitting and standing poses to verify any general claims.

Regarding mode-switching techniques, our results show long press is the worst and two-finger multi-touch the best, unless target accuracy is critical. When accuracy is needed, thumb-on-finger is the best option with lower out-of-target errors than two-finger (and others) and lower crossing errors than non-preferred hand. Subjective ratings for thumb-on-finger do not indicate a pattern of any strong preference or dislike. One caveat is that, although there were no significant differences for thumb-on-finger in mode-in errors, the measured values are high (Figs. 3.6 and 3.7). Given the high variance preventing statistical difference, individual mastery of thumb-on-finger mode-switching varies.

When considering mode-switching time only, non-preferred hand, knuckle, and hard press are all within 100ms from the mean time for thumb-on-finger. At first this may suggest they are all comparable, but high rates of mode-switching errors and a pattern of lower subjective ratings cast doubt on using pressure for touch mode-switching. Although there is subjective support for the non-preferred hand, it has a higher mode-out error rate than other techniques and a surprising pattern of frequent crossing and out-of-target errors.

Given the novelty of using a knuckle for touch input, we were surprised to see it perform as well as it did. It has an overall error rate comparable to the best techniques with one of the lowest mode-in error rates. Although out-of-target errors are higher than some, there is no statistical evidence of a higher crossing error rate. This is encouraging considering we expected the increased occlusion to make knuckle crossing wildly inaccurate. However, subjective ratings and comments show a pattern of mild dislike due to higher fatigue and perceived slower speed. Note that only two-finger and knuckle *instantly* combine the mode-switch with the initial touch position. This "merging of command selection and direct manipulation" has been shown to be beneficial [93].

3.6.1 Subtraction Method Validity

Like Li et al., we confirm that we can apply the subtraction method in our experimental protocol since drawing and positioning movements are not strictly fixed as in Dillon et al.'s point-connecting task [62]. We accomplish this by comparing the *total movement distance* in a full cycle between baseline and compound tasks. A sufficiently small difference indicates that movements required for the two types of tasks were similar. Due to how we logged multiple simultaneous touches, automatic calculation of movement time for non-preferred hand and multi-touch proved error-prone. We chose to not include them in this analysis. Among techniques, non-preferred hand and multi-touch are arguably the most similar to non-mode touching crossing movements.

For the remaining four techniques, we found a mean movement distance for a full cycle to be 55.2mm (690px). This is 1.4mm greater than the baseline condition, in which mean movement distance was 53.8mm (672px). This difference of only 2.7%, compares favourably with Li et al.'s difference of 4.7mm (20px)[1], or 3.4%. Li et al. also report a similar single full cycle mean movement distance of 66 mm (290px). This demonstrates using the subtraction method was valid.

3.6.2 Temporal Pattern Analysis

We analyse the temporal submovements pattern to understand different techniques, and use this to classify them into different temporal models. This is directly based on the

[1]Li et al. used a 12.1" diagonal, 1024 × 768px TabletPC.

	BASELINE	LONG PRESS	TWO FINGER	NON PREF	HARD PRESS	THUMB	KNUCKLE
T_{P1}	323	392	375	486	415	502	546
T_{C1}	196	1239	210	189	420	276	267
T_{P2}	317	415	435	419	514	433	510
T_{C2}	195	220	200	208	234	218	218
T_{ENG}	N/A	1130	401	318	459	536	544
T_{DIS}	N/A	51	100	421	91	61	531
T_{GES}	N/A	552	284	779	466	303	799

Table 3.2: Cycle decomposition times (ms).

"keystroke level analysis" performed by Li et al.. We use the same models and compare our results with their findings.

A full cycle can be decomposed into four submovements with corresponding times:

$$T_{cycle} = T_{P1} + T_{C1} + T_{P2} + T_{C2}$$

where T_{Pi} is the time taken to position the finger in the air before crossing the i^{th} rectangle and T_{Ci} is the time taken to drag the finger on the display and cross the i^{th} rectangle. T_{Pi} begins on touch up of the previous rectangle and ends on touch down of the i^{th} rectangle. T_{Ci} begins on touch down and ends on touch up after crossing through the i^{th} rectangle.

Each two-rectangle full cycle during a compound task requires crossing rectangle 1 (the 'red' one) with the mode engaged and crossing rectangle 2 (the 'grey' one) with the mode disengaged. Therefore, the mode is engaged either during T_{P1}, or near the beginning of T_{C1}. The time to engage a mode T_{ENG} is the duration from the start of a cycle until the mode is engaged. Note that T_{ENG} equals T_{P1} if the mode is engaged precisely at touch down (e.g. knuckle); T_{ENG} may be less than T_{P1} if the mode can be engaged before touch down (e.g. non-preferred hand); and T_{ENG} may be greater than T_{P1} if the mode is engaged after touch down (e.g. hard press). T_{GES} is the time spent gesturing, defined from the later of mode engagement and touch down until touch up. Beyond these durations defined by Li et al., we define a mode disengage time T_{DIS} as the duration from disengagement until touch up.

Table 3.2 provides mean times for these submovements for all techniques. A further verification that the task and subtraction method worked as it should is that the T_{C2} times are the same for the baseline task and all techniques. As in Li et al., our absolute timings for T_{C2} all appear close to the baseline. To verify this, we conducted a one-way ANOVA for the effect of technique (including baseline) on T_{C2}. There was a main effect

($F_{4.07,138.27} = 9.5754$, $p < .001$) and post hoc tests reveal that all but two-finger multi-touch are significantly different. However, the differences between all techniques and the baseline is less than 40 ms.

3.6.3 Temporal models

All techniques follow the temporal models described in Li et al.'s keystroke level analysis. Although the authors did not perform statistical tests on timing decompositions, we provide this extra level of validation when applicable.

Using Non-Preferred Hand and Thumb-on-Finger. The two techniques obey the same temporal model, as they require the mode to be engaged before touch down, with release possible during, or after completion of the gesture. As in Li et al., we calculate an estimate of the gesture engagement time by subtracting T_{P1} of the baseline task from T_{P1} of the compound task. We obtain 164ms for NONPREF and 179ms for THUMB, with no significant difference between them. This contrasts with Li et al., where the mode engagement time for NonPrefHand and BarrelButton were 65ms and 144ms respectively, a greater difference.

Disregarding the absolute timings which are systematically lower for Li et al., we believe there may be two reasons for this greater difference. First, it may be easier and faster to hit the index finger with one's thumb than to reach and press a barrel button on a stylus. Second, although our participants held their finger poised above the touchscreen button before each non-dominant hand trial, Li et al.'s NonPrefHand participants could exploit the physical button by resting their finger on it to minimize activation and eliminate targeting. The physical button may have been the primary reason for the very strong performance of NonPrefHand in Li et al.. Perhaps the full potential of non-preferred hand mode-switching cannot be realized on a touchscreen.

Using Long Press (Hold). Li et al. note that there is a difference between the T_{P1} timings of Hold and the baseline even though gesture mode engagement is started upon touch down and thus, at first glance, the two T_{P1} should be close. They attribute this difference to an additional preparation time needed when slowing down the pen movement to hold it in a steady position. We also observe this phenomenon with a statistically significant difference of 69.5ms between the T_{P1} timings ($p < .0001$), albeit a much lower one.

Li et al. also calculate the time participants took to respond to the feedback showing that the mode had been engaged (a full circle) using the formula:

$$T_{response} = T_{ENG} + T_{P1} - Holdtime$$

Their response time is 137ms. Ours is: 1130 – 392 – 500 = 238ms.

Using Hard Press (Pressure) and Two-Finger Multi-touch. In Li et al., Pressure is the technique with the lowest T_{P1}, but again, without statistical analyses we do not know if timings are significantly different from other techniques. Even though HARDPRESS appears to be third best only from the mean values, ANOVAs and post hoc tests do not reveal any significant differences between the T_{P1} values for HARDPRESS, TWOFINGER and LONGPRESS and hence we are not able to conclude which of the three techniques has the shortest positioning time.

Similar to Li et et al., we calculate the time to increase touch pressure to the required level in order to activate the gesture mode: $T_{ENG} - T_{P1}$ = 45ms. This value is much lower than their 176ms. We attribute that to the possibility that it might be easier to sense and apply the required pressure level using direct input than with an instrument such as a stylus. The fact that we used an EMG armband with a high data rate and adapted pressure thresholds for each participant might also have been factors.

Like Li et al, we observe that drawing with HARDPRESS takes more time than without (the difference between the two T_{C1} values is 223ms, which is significant). The difference is likely even more pronounced with a finger than with a stylus due to the increased friction of dragging (Li et al. report a drawing time of 176ms, but, once more, we cannot determine if the differences are significant).

The model for TWOFINGER is similar, as the mode is engaged after touch down and has to be maintained throughout the moded action. However, we expect the time to engage the mode after touch down to be very short, as the two fingers are usually put down almost simultaneously. Similar to hard press, we calculate the engagement time after touch down, which is $T_{ENG} - T_{P1}$ = 26ms. A t-test confirms that the difference with HARDPRESS is significant ($p < .0001$).

Using Knuckle. Our knuckle technique follows the temporal model of the Eraser in Li et al.. The authors notice very similar T_{C1} timings for the Eraser and the baseline task, meaning that drawing with the eraser end of the stylus requires no extra effort. This was not the case with KNUCKLE. Drawing with the knuckle took an extra 71ms, statistically significant compared to the baseline ($p < .0001$). Li et al. further calculate times to turn and revert the pen in order to use the eraser by subtracting the T_{P1} and T_{P2} timings. They report values of 661ms and 555ms. A stylus being typically longer and the rotation required to use its eraser larger, our wrist-rotation times for KNUCKLE are predictably lower: 223ms and 193ms respectively.

3.6.4 Improvements to Mode-Switching Techniques

None of the techniques we tested were perfect in all aspects. Even two-finger multi-touch suffers from accuracy problems when the demands for targeting and tracing precision are high. To improve that, a cursor could appear between fingers when approaching the screen (assuming pre-touch sensing is available [108]). For hard press, it would be interesting to see if a device with built-in touch pressure sensors and a smart adaptive thresholding algorithm could improve the measured and perceived accuracy as well as reduce fatigue.

For KNUCKLE, our realisation of the technique is based on the rotation of the wrist, which causes both strain and occlusion problems. Those issues might be alleviated if knuckle interaction could be performed without turning the hand (i.e. by bending the finger and using the distal interphalangeal joint).

Regarding techniques which require the mode switch to be engaged before touch down, mode-in errors would presumably be reduced if that condition is relaxed so the mode change could occur after touching the device. This has been shown in pen mode-switching for direct manipulation contexts that permit late mode activation [143, 223]. The extended period allowed for the switch could be chosen depending on the application context and which functions the moded and non-moded actions are mapped to: a long period if late mode engagement has minimal disturbance (e.g. changing stroke style) and a short period when late mode engagement would be disruptive (e.g. switching between panning and inking).

Hybrid techniques are an interesting avenue for exploration. For example, pressure-based activation could be combined with non-preferred-hand to simulate behaviour of a physical button: the fingers of the non-preferred hand could rest on the screen to minimize activation time and eliminate targeting. This might be especially helpful when supporting the tablet with that non-preferred hand, as the holding posture is maintained at all times and therefore stability is likely increased. It would be particularly interesting to see how techniques can be combined to support several mode changes, such as combining two-finger or knuckle with non-preferred hand.

3.7 Limitations

While our experiments provide insightful results, we acknowledge their scope of validity and the limitations within which they can be interpreted. First, since our study design followed Li et al., we operate with the same constraints regarding a single mode switch applied with a relatively high and regular frequency in a synthetic linear task. This design allowed us to perform fine-grained analyses, but further studies could validate our results in more realistic and less controlled tasks. For instance, as opposed to a continuous engagement task like drawing a line in our experiment, testing the mode-switching performance with a tapping task might reveal interesting performance differences. Testing varying body postures like running or walking could expand the applicability of the results. Furthermore, applications often include more than two modes, so techniques could also be evaluated with multiple modes to assess scalability.

Although we strive to design optimized and representative techniques, there are possible limitations in the way they were implemented. Knuckle, hard press, and pressure required participants to wear extra sensors. Although none reported any particular impediment or discomfort, results for those techniques might change if alternate sensing was used. For non-preferred hand, the size and position of the trigger button was fixed for all participants. Prior work and several pilot tests informed our final design, but size and position could be personalized to individual people, especially when used in a standing position. Finally, the dwell time used to activate a long-press naturally influences the performance and accuracy of the switch. Our choice of 500ms is used in popular operating systems, but this could be further optimized.

3.8 Conclusion

We presented a detailed analysis of touch input mode-switching techniques. Our results can be used as guidelines for selecting mode-switching techniques. When restricted to current device capabilities, two-finger multi-touch should be selected if accuracy is not critical and a non-preferred hand button otherwise. As more advanced sensors are available, touching the thumb to the side of the finger will also be a good choice. Using the knuckle works surprisingly well, though many people perceive it as being inaccurate and uncomfortable. In most cases, long press should be avoided. In contrast to reasonable performance for pen pressure mode-switching reported by Li et al. [153], using pressure for touch mode-switching appears problematic. It is possible that hard press performance and perceived inaccuracy may improve on a device with built-in pressure sensors, but we suspect this has more to do with touch friction and fatigue.

Though numerous experimental investigations have compared mode-switching techniques for pucks, mice, and pens, we believe we are the first to do so for touch input. This work fills an important knowledge gap, especially when considering how fundamental mode-switching is to touch interaction.

Chapter 4

Barehand Mid-Air Mode-Switching in VR

This chapter present an empirical comparison of eleven bare hand mid-air mode-switching techniques suitable for virtual reality in two experiments. The first evaluates seven techniques spanning dominant and non-dominant hand actions. Techniques represent common classes of actions selected by a methodical examination of 56 examples of prior art. The standard "subtraction method" protocol is adapted for 3D interfaces, with two baseline selection methods, bare hand pinch and device controller button. A second experiment with four techniques explores more subtle dominant-hand techniques and the effect of using a dominant hand device for selection. The results provide guidance to practitioners when choosing bare hand, mid-air mode-switching techniques, and for researchers when designing new mode-switching methods in VR.

4.1 Motivation

Raskin defines a *mode* as a distinct setting within an interface where the same user input produces results different from those it would produce in other settings [214]. *Mode-switching* is simply the transition from one mode to another. Modes are common in all interfaces, including interfaces for Virtual Reality (VR) and Augmented Reality (AR). For example, a hand gesture-based 3D modelling application may have different modes for object creation, selection, and transformation. Depending on the mode, the movement of the hand is interpreted differently. Completing a task typically requires frequent mode-switching, so understanding the performance of different methods is important.

In VR, bare hand mid-air input is an alternative to device controllers. Techniques have been proposed for VR, AR, and related contexts using hands only (e.g. [205, 274, 200]) and hands combined with body postures (e.g. [286, 32]). Evaluations have focused on tasks like pointing (e.g. [281]), object manipulation (e.g. [215]), selection (e.g. [191]), and annotation [46], but no extensive comparisons of mode-switching techniques have been performed yet. Mode-switching techniques for mice [62], styli [153, 268], and touch [254] have been evaluated, but generalizing those results to 3D environments like VR is not straightforward.

We provide missing empirical evidence for the performance of bare hand mid-air mode-switching techniques in VR. Our focus is absolute, single-point input, suitable for the kind of direct object manipulations common in VR such as pointing at, grabbing and moving 3D elements in the virtual environment. To select techniques to evaluate, we examined bare hand mid-air interaction in different settings, then used three criteria to identify six classes of techniques suitable for mode-switching in VR. In two related experiments, we compare common input actions selected from each class using an adapted "subtraction method" protocol [62], used previously for 2D input.

The first experiment compares seven techniques, with a dominant-hand pinch as the fundamental manipulation trigger. The mode-switching techniques include three dominant hand postures: a fist, an open palm and pointing the index finger; and four non-dominant hand postures: a fist, an open palm, bringing the hand into the field-of-view and touching the head. As a comparison baseline, the button of a device controller held in the non-dominant hand was also included in the tests. A second experiment explores questions emerging from the results. The effect of more subtle dominant hand techniques is examined by testing pinching with wrist rotation, and pinching with different fingers; the effect of pinching as a manipulation trigger is compared with a controller held in the dominant hand.

Our empirically-derived insights can inform the design of VR applications using bare hand mode switching:

1. Dominant techniques using large motions are error prone and less preferred, but a more subtle variation of pinch is comparable to the fastest non-dominant techniques.

2. With the exception of a few dominant techniques, mode-switching times are comparable to most touch methods.

3. Using a dominant pinch as a manipulation trigger is comparable to using a device controller button.

4. All techniques from fastest to slowest: Non-Dominant (ND) device and Dominant (D) middle finger pinch; ND palm orientation and ND fist; ND head touch, D pinch orientation and D palm; ND field-of-view and D fist; D point.

4.2 Background and Related Work

We define *bare hand* as input performed entirely by a hand posture or movement, without any device. Note that this definition does not specify the sensing method, so early work using instrumented gloves for hand tracking are considered bare hand for our purposes. We define *mid-air* as input conducted without contacting a non-body surface. In most cases, this means input performed in the space around the body. For on-body contacts, the sensing method is unimportant as long as the technique is conceptually a body contact (e.g. touching the head), and not using a device attached to the body (e.g. tapping on a smartwatch).

4.2.1 Formal Mode-Switching Evaluations

Mode-switching has been more commonly studied in 2D interfaces. Dillon et al. [62] introduced a formal "subtraction method" for comparing mouse mode-switching, which was adopted by Li et al. [153] and Tu et al. [268], who each compared five mode-switching techniques, and Surale et al. [254], who compared six techniques. Using the non-dominant hand for stylus mode-switching has been studied in detail by Ruiz et al. [223] who developed a temporal model, Lank et al. [143] who showed concurrent mode-switching is fastest, and Ruiz and Lank [224] who explored related aspects with multiple modes.

4.2.2 Related Evaluations of Interaction Techniques

For mid-air bare hand input, mode-switching has only been indirectly evaluated as part of larger interaction technique studies. In the context of large displays, Vogel and Balakrishnan [281] compare a relative pointing technique, which uses a fist mode-switch to "clutch", to a ray-cast technique without any mode-switch, but the mode-switch itself is not compared. Similar examples in large display research include Haque et al. [94], Polacek et al. [206], Jota et al. [124], and Katsuragawa et al. [132]. In the context of VR and related 3D contexts, Poupyrev et al. [209, 210] evaluated object pointing, manipulation, and selection techniques, Teather and Stuerzlinger compared pointing techniques [261], and Vanacken et al. [272], Grossman et al. [87], and Looser et al. [161] all examine barehand selection techniques. In most cases, these techniques have some explicit activation and deactivation of a mode, but mode-switching performance is not evaluated in isolation.

We are unaware of work comparing mode-switching techniques for bare hand mid-air input in VR using the formal subtraction method used for mouse, pen, and touch.

4.3 Mode-switching Techniques

To identify bare hand mid-air interaction techniques suitable for VR mode-switching, we examined research and commercial systems to create a list of candidate techniques.

Our examination included general surveys of 3D interaction from Argelaguet and Andujar [5], Jung et al. [126], Poupyrev et al. [209], Bowman et al. [33], Aigner et al. [2] and Groenewald et al. [86]. We also examined the results of elicitation studies for bare hand mid-air interaction with large displays [226, 273, 293, 160], general ubiquitous computing [45] and AR [205]. Finally, we found examples of bare hand mid-air techniques in many papers on interaction techniques (e.g. [281, 243]), new sensors (e.g. [135]), and commercial devices (e.g. Leap Motion [144], Kinect [138], Myo [186]). For this work we did not consider other non-hand input that may also be suitable for mode-switching, such as feet (e.g. [276]), voice (e.g. [29]), gaze (e.g. [201]), or exocentric interaction (e.g. [251]).

From our initial list of bare hand mid-air interaction techniques gathered from the literature we extracted a subset of candidates that we considered s suitable for mode-switching based on three filtering criteria.

4.3.1 Actions Suitable for Mode-Switching

We identified 40 different mid-air bare hand actions in 56 publications or device manuals. We initially considered actions suitable for dominant or non-dominant hands, even if the source considers a specific hand. To filter those actions to a subset suitable for mode-switching techniques, we created three criteria based on observable mechanics of hand or finger actions. These criteria are not explicitly about performance, because no previous work specifically evaluated mode-switching, nor are they about learnability, comfort, and social acceptability, because these aspects are often not reported. The filtering criteria are:

Class	**N**	**Actions** (with sources)
Pinch Finger(s)	29	**thumb touches index** [135, 286, 273, 20, 200, 64, 45, 304, 237, 205, 114, 145, 176, 127, 202]; thumb touches all fingers [45, 200, 275, 300, 274, 205, 125, 12]; thumb touches side of hand [159, 304, 271, 281]; thumb touches middle [45, 237, 176]; thumb touches index and middle [237, 125]; thumb touches ring [45, 176]; thumb touches pinky [45, 176]; thumb touches index, middle, and ring [45]; thumb touches ring and pinky [45]; thumb touches three fingers [205]
Extend Finger(s)	31	**extend index** [281, 135, 287, 243, 271, 200, 64, 76, 159, 275, 160, 304, 205, 222, 144]; extend thumb [304, 140, 160, 273, 116, 200, 300, 274, 205]; extend thumb-index-middle [76, 159]; clench index [271]; two hand point [243, 222, 205]; point with dwell [271]
Close Hand	18	**make fist** [135, 287, 243, 271, 270, 101, 116, 45, 300, 205, 215, 200, 53, 238, 76, 281, 176, 125]; make partial fist [64, 281]
Open Hand	62	**open hand with oriented palm** in/out [287, 188, 13, 280, 191, 301, 278, 101, 137, 79, 99, 78, 131, 200, 53, 64, 238, 250, 1, 76, 146, 293, 279, 275, 160, 300, 274, 304, 205, 114, 138], up/down [222, 191, 140, 301, 278, 101, 116, 137, 200, 64, 275, 304], right/left [116, 137, 79, 99, 53, 76, 293, 279, 275, 304]; open hand [281, 135, 243, 1, 274, 280]; open hand with finger(s) bent [271, 281]
Raise Hand (ND)	6	**hand raised into field-of-view** [259, 278, 137, 274]; raised above shoulder [250, 304]
Touch Body (ND)	6	**finger(s) touch head** [286, 99, 304], behind ear [160]; mouth [64]; hand touches waist [286]

Table 4.1: Bare-hand mid-air interaction techniques suitable for mode-switching in VR based on examination of research papers and commercial systems. Similar *Actions* are grouped into *Classes*. The number of sources (N) is an approximate indication of popularity. Boldface actions are those tested in our experiments.

Independent

A mode-switch action should be fast to recognize and independent of previous tracking states, meaning it should not rely on time-based actions such as a specific movements. Conceptually, this means the technique can be recognized in a single sensor time frame (in implementations, it may actually be a few frames to compensate for noise). Examples of independent actions include pinching the thumb and finger together, making a fist, or raising an arm. Examples of non-independent actions include dwelling, gestures like drawing an 'X', or mimicking knocking.

Kinesthetic

The action should enable the mode to be maintained by the user, not the system. This means the posture, position, or gesture action can be "held" as long as the mode is needed, and the mode ends when no longer held. This creates a *kinesthetic quasimode* [214], known to reduce mode errors [236]. Examples of kinesthetic actions include pinching, making a fist, or a repeated gesture, where changing the posture or stopping the gesture releases the mode.

Unconstrained

The action can be executed with the dominant hand at any position and, in the case of a kinesthetic action, the dominant hand can easily reach any position for subsequent operations. Unconstrained actions include any non-dominant hand action that does not impede the dominant hand, and dominant hand actions such as pinching or making a fist. Examples of constrained actions are placing the dominant hand on the head, or pointing the dominant index finger towards the body, since those actions physically constrain the motion range of the arm.

We considered candidate actions suitable for mode-switching techniques when they satisfied all three criteria. This narrowed the list down to 29 actions found in 53 papers, listed in Table 4.1. Analogous actions are grouped into 6 classes. For example, all actions that involve a pinch of some kind are grouped into the general "Pinch Finger(s)" class. Most actions can be used for mode-switching with either hand, except those in the "Raise Hand" and "Touch Body" classes that are only suitable for non-dominant usage.

4.3.2 Selected Techniques for Evaluation

We selected the most popular action from each class to evaluate (boldface in Table 4.1). Five of these form eight mode-switching techniques because some actions are performed with both dominant and non-dominant hand.

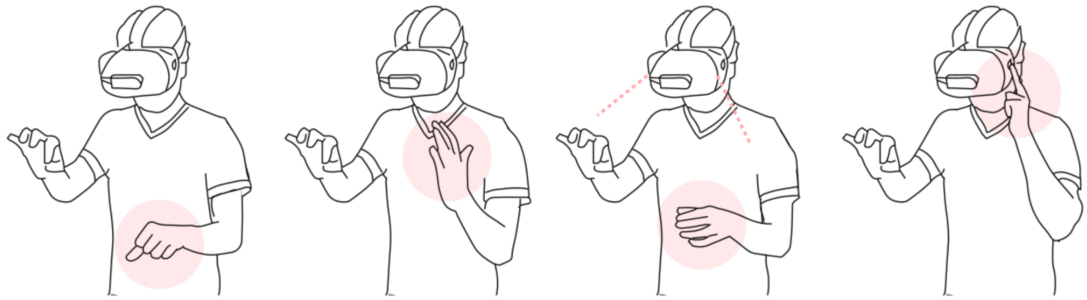


Figure 4.1: Selected non-dominant hand mode-switching techniques: (a) non dominant fist (ND-FIST); (b) non dominant palm (ND-PALM); (c) hand in field of view (ND-FOV); (d) touch head (ND-HEAD).

With bare hand mid-air input, a mechanism is required for the user to indicate when their hand is just moving, or if it is performing an operation such as drawing a line. Buxton's *Simple 2-State Transaction Model* [39] calls these states "tracking" and "dragging". With a VR device controller or a mouse, this is achieved by pressing a button. For our bare hand mid-air system, we use a thumb-index pinch since it is a popular action, and it uses a subtle movement that can be easily sensed. We believe using a pinch is the most obvious choice, but other possibilities include tapping the palm with the thumb [45] and a partial finger bend or "Airtap" [281].

Non-Dominant Techniques. The non-dominant hand controls the mode and the dominant hand manipulates using the thumb-index pinch (see Figure 4.1). The techniques are:

- *Non-Dominant Fist* (ND-FIST) — The mode is active when the non-dominant hand is clenched, and released when the hand relaxes so one or more fingers begin to open.
- *Non-Dominant Palm* (ND-PALM) — The mode is active when all fingers of the non-dominant hand are extended, roughly pointing up, with the palm facing to the right (assuming the left hand is non-dominant). The mode is released when the palm orientation changes significantly or one or more fingers are no longer extended.
- *Hand Moved into Field of View* (ND-FOV) — The mode is active when the non-dominant hand is moved into the user's field of view, and released when it moves outside.

Figure 4.2: Selected dominant hand mode-switching techniques: (a) dominant fist (D-FIST); (b) dominant palm (D-PALM); (c) point (D-POINT).

- *Touch Head* (ND-HEAD) — The mode is active when the non-dominant hand touches the side of the head (HMD).

Dominant Hand (D) Techniques. The dominant hand controls the mode by forming a posture, and a thumb pinch against the side of the index finger engages and disengages the “dragging” state for manipulation (see Figure 4.2). The techniques are:

- *Dominant Fist* (D-FIST) — The mode is active when the dominant hand is clenched, and the mode is released when the hand relaxes, i.e. when one or more fingers begin to open.
- *Dominant Palm* (D-PALM) — The mode is active when all the fingers are extended, roughly pointing up, and the palm is facing away.
- *Point* (D-POINT) — The mode is active when the index finger is extended and all remaining fingers are closed.

4.3.3 Technique Sensing

To track the user’s hands in the VR environment, we use a LEAP motion mounted on the front of the HMD [6, 235, 156]. We found the LEAP reliable for rendering the hand and tracking hand position, but the built-in posture recognizer often misclassified dominant fist, palm, and pinch postures during rapid mode switching, and detection of pinch actions to trigger manipulation was imprecise. To address these issues, we developed a user-calibrated classifier to discriminate between problematic postures, and added a force-sensitive resistor (FSR) to detect pinches and head touches.

User-calibrated Posture Classifier

In the experiment, we only need to independently discriminate between palm and pinch, or between fist and pinch. We train a simple classifier for each case, calibrated to each user. We describe the method using fist and pinch, but both are similar. A user draws 15 lines each with the pinch, palm, and fist postures. At each frame, a ten-dimensional feature vector consisting of fingertip positions and distances relative to the centre of the palm is recorded. The median and standard deviation (SD) in each dimension is calculated for the 20 frames before and after the start of each line. The classifier selects differentiating dimensions in which the fist median plus two SD is less than the pinch median minus two SD. In the rare case that two differentiating dimensions are not found, training is repeated with the user instructed to form the postures more clearly. Otherwise, each selected dimension is assigned a threshold half-way between the two medians. Once trained, a posture was considered a pinch when two or more selected dimensions exceeded the threshold. A 6-person pilot test found this method almost 99% accurate. During the experiment, this calibration process was conducted immediately before the first block testing dominant fist or palm.

Pinch and Touch Detection

An FSR taped to the distal phalanx of the dominant thumb is used to detect thumb-index pinches. The FSR is 7 mm in diameter and 0.3 mm thick with a sensing range of 0 to 175 psi. To detect a touch to the head, a larger FSR (25.4 mm diameter, 0.21 mm thick) is taped to the non-dominant side of the HMD. We used a threshold of 17 psi for both FSRs to detect a light pinch or touch. Thin wires from each FSR were connected to an Arduino Nano (Atmega328) mounted off the body. We verified that the sensors and wires did not impede the user's movements.

System Latency and Performance

Our code and 3D scene were simple and we used a high-end computer, so the Unity application ran at an optimal 90 FPS to supply the 90Hz HMD. The Leap motion provided a stream of hand postures at 110Hz, the Arduino updated pressure values approximately every 2ms, and the controllers were tracked at 250Hz to 1kHz. Our posture classifier did not use any temporal filtering. Since cycle duration is measured between two input times, the effective latency is 11ms at 90 FPS.

4.4 Experiment 1

The goal is to empirically compare mode-switching performance of the seven techniques listed above. We adapt the standard "subtraction method" [62, 153, 268, 254] protocol to VR. This determines the precise cost of mode-switching by subtracting the time to perform tasks using a single mode, from those when alternating between two modes.

4.4.1 Participants and Apparatus

16 participants (mean age 28.5 SD = 6.3, 5 women, all right-handed) were recruited. 7 had experience using a VR device. Remuneration was $10. Each participant filled out a questionnaire before the experiment began and after the experiment was completed. These questionnaires are included in Appendix B. An HTC Vive VR head-mounted display (HMD), with a resolution of 1080 × 1200 px per eye, 90Hz refresh rate, and 110° FOV was used. Focal length was initially set to 63 mm, but participants were given the possibility to adjust it. A high-end Windows 10 machine (3.6GHz Intel i7 CPU, GeForce GTX 1080 GPU) ran the experiment application written in Unity 5.5.3f1. A LEAP motion and two FSRs were attached as described above.

4.4.2 Task

We adapt the 2D task used in previous mode-switching investigations [62, 254, 153] to a 3D task for VR. Considering that a common class of VR consumer applications are for sketching, painting, and 3D modeling (e.g. [26, 199, 14, 266, 85, 241, 199, 171]), with much prior research in these areas [119, 8, 234, 133, 260], we use 3D line drawing as our fundamental task. Note that this is an abstraction of many 3D tasks, such as creating objects other than lines (e.g. cubes, spheres), transforming an object (e.g. moving, scaling), or panning a world scene. Regardless, for the purposes of our experiment, the explicit mode change is more important than the gross movement of the hand during the task. All tasks were performed in a standing position.

Line Drawing

A series of five aligned pairs of spheres is presented to the user in the VR world (Figure 4.3). The spheres in a pair need to be connected by drawing a 3D line between them in the proper mode. Each sphere has diameter 100 mm, the distance between spheres in a pair is 50 mm, and pairs are stacked with 82 mm overlap. The position is calibrated so the topmost pair is at the participant's shoulder level. Sizes and distances among the pairs are chosen so that participants can easily reach each sphere without stepping, regardless of pair orientation or movement direction. We chose the centre of the palm as the line "ink" anchor as it is more stable than the fingers, which are used for mode-switching.

For each pair of spheres, the line drawing task can be seen as a four-step process. First, place the dominant hand inside the starting sphere (a change of opacity indicates the centre of the palm is inside). Second, engage line drawing using the pinch trigger. Third, move the hand to draw the line until it is inside the second sphere. Lastly, release the pinch to disengage line drawing. Subtle audio feedback ('tick') indicates line drawing engagement and disengagement. If engagement or disengagement occurs outside the sphere, an error is logged, a buzz sounds, and the participant has to redraw the line. This is repeated for all five pairs of spheres. The current pair is indicated using colour saturation, and the required direction of the line drawing is indicated through transparency (most opaque sphere to semitransparent sphere).

Baseline and Compound Task Variations

There are two task variations. In the *baseline* task, five pairs of blue spheres are shown and the participant draws lines using only the dominant hand pinch. In the *compound* task, the five pairs alternate between blue and red colours, with the first pair being blue. The participant must draw lines to connect a pair of blue coloured spheres using a pinch, and red paired spheres with the specified mode switching technique. To switch modes, the participant must have formed the current mode-switching technique posture at the moment line drawing is engaged. As visual feedback, "moded" lines drawn with the mode-switching technique are red, and "unmoded" lines are blue. These red or blue trails function only as an abstract representation of two different modes. The colour of a small sphere rendered in the middle of the palm also indicates the mode. If there is a mode-switch error, a buzz sounds and the participant has to redraw the current line.

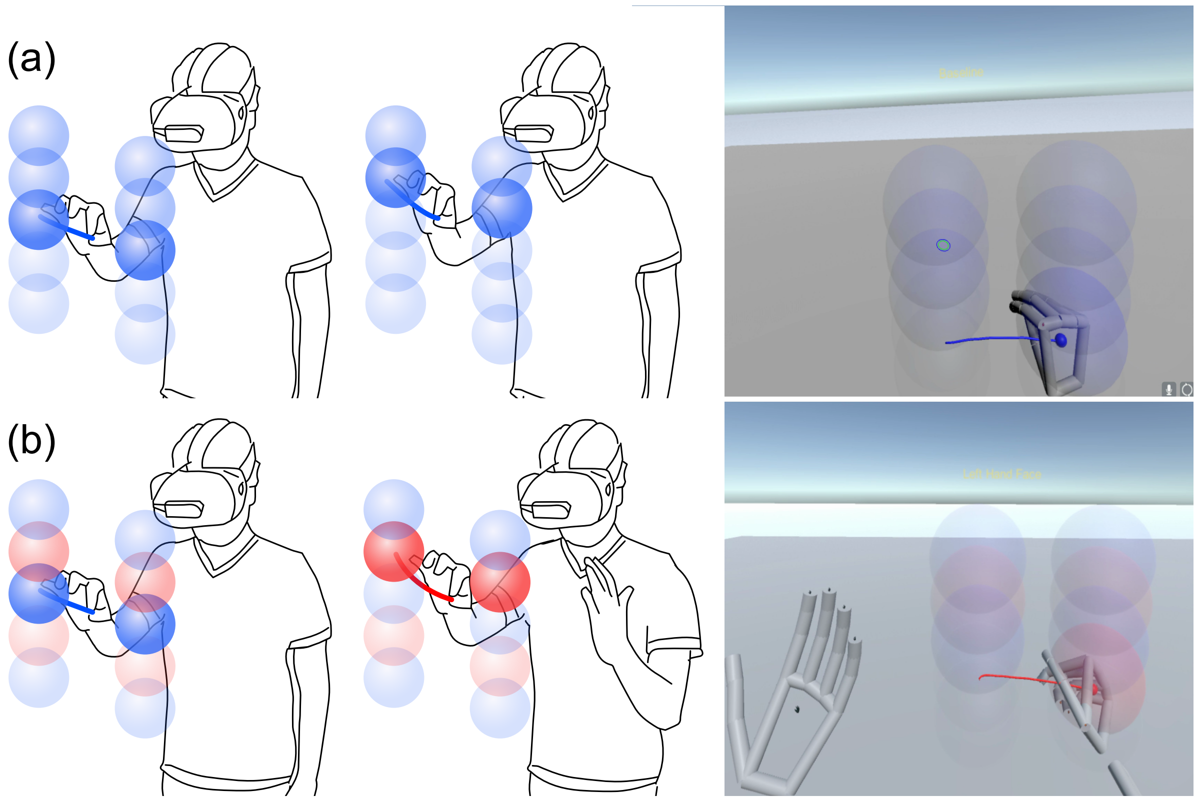


Figure 4.3: Line drawing task: (a) baseline; (b) compound.

4.4.3 Design and Procedure

The experiment is a within subjects design. Mode-switching TECHNIQUE is the primary factor, with levels corresponding to the seven techniques described above (ND-FIST, ND-PALM, ND-FOV, ND-HEAD, N-FIST, D-PALM, D-POINT) and an eighth technique using a standard HTC controller with a button held in the non-dominant hand (ND-DEVICE). This functions as a non-bare hand comparison baseline since the mode is switched by holding the button. Like all other non-dominant hand techniques, the dominant hand draws the line using a pinch. Our block design deviates slightly from previous mode-switching studies [153, 254] in the following ways.

Pilot Study to Refine Design

Previous mode-switching studies used 9 blocks: 5 baseline (B) task blocks separated by 4 compound (C) task blocks (i.e. BCBCBCBCB). Considering that we wished to test 8 techniques (previous mode-switching studies compared 5 or 6), and mid-air gestures are more fatiguing than pen or touch interactions, we conducted a 4-person pilot test to see if some blocks could possibly be removed without impacting the results too much. Each session took more than 90 minutes. Examining a graph of task times by block for each technique gave no indication of pronounced systematic differences between any of the baseline blocks, and we did not find any significant interaction for TECHNIQUE × BLOCK. As a result, we reduce the number of baseline blocks from 5 to 2 by keeping only the first and the last baseline blocks (i.e. BCCCCB). The pilot study also tested 6 directions for the drawing task: left and right horizontal, up and down vertical, and in and out along the depth direction. A graph of time by task direction did not reveal any major differences, and we found no significant effect, so we reduced the directions to 3 for the main study: horizontal left, vertical down, and depth inward. With the optimized experimental design, all eight techniques can be tested in less than an hour with less fatigue.

Final Design

To minimize order effects, TECHNIQUE was counter-balanced using a 8 × 8 Latin Square. The session began with 5 baseline blocks for training. Then, for each TECHNIQUE, there was a 3 min practice period followed by 6 BLOCKS of tasks. The starting and ending block were *baseline* tasks, remaining blocks were *compound* tasks (i.e. BCCCCB). In each block, the participant had to complete 5 line drawing tasks in 3 directions (left-to-right, up-to-down, out-to-in) in random order. Breaks were encouraged between blocks. After the experiment, participants provided subjective ratings for each technique.

In sum there were: 8 TECHNIQUES (7 mid-air bare hand, 1 device) × 6 BLOCKS (2 *baseline*, 4 *compound*) × 3 directions × 5 line drawings = 720 line drawings per participant.

4.4.4 Dependent Measures

Mode Switching Time

The line drawing time is the duration starting from line engagement in the starting sphere until line disengagement in the ending sphere. Naively, one might directly compare line drawing times in a baseline block to a compound block (where a mode switch was required). However, all line drawings share a common overhead of moving from the ending sphere of the previous line to the starting sphere of the current line. Therefore, we use the "subtraction method" [62, 254] to precisely isolate mode-switching time.

In a set of line drawing tasks, there are three "cycles". The first is from the moment spheres are visible until the pinch trigger is released after drawing the line between the first pair of spheres. The second cycle begins immediately after, and ends when the pinch trigger is released after drawing the line between the third pair of spheres. The third cycle begins immediately after, ending when the pinch trigger is released after drawing the line between the last pair of spheres. During the compound task, the second and third cycles are *full cycles*. Each captures a complete mode-switch operation: the participant switches into a mode using the specified technique, draws a line connecting spheres, switches out of the mode, draws a line connecting another set of spheres, and disengages line drawing mode. The first cycle only guarantees the baseline mode is active before the second cycle. In each block, there are 6 full cycles (2 per direction). For each technique, each participant completes 24 full cycles with mode switching (in 4 compound task blocks) and 12 full cycles with no mode switching (in 2 baseline task blocks).

The subtraction method isolates mode switching time using mean times from cycles. For each block, the mean time for the second and third cycle is calculated using error-free cycles. The mode-switching time is calculated by subtracting the mean full cycle time of the first and the last baseline blocks from the mean full cycle time of a compound block. In total, this provides 8 *mode-switch time* measurements per participant, per technique (2 per block).

Errors and Error Rates

Like Li et al. and Surale et al., we identify three error types. A *start error* occurs if the line drawing is initiated outside the active starting sphere. An *end error* occurs if the line did not connect the two active spheres in the correct direction. A *mode error* occurs when the wrong mode is used to connect the spheres in compound tasks (e.g. connecting blue spheres with a red line). We further distinguish between mode-in and mode-out errors. A *mode-in error* occurs when the participant fails to transition from baseline to the specified mode-switching technique. A *mode-out error* is when the participant fails to transition from the specified mode-switching technique to baseline. Each of these error types are recorded per trial as 1 if the error occurred, and 0 otherwise. The mean value across trials produces an error rate.

4.4.5 Results

We examined error-free *line drawing times* to identify outliers more than 3 standard deviations from the mean for each TASK. This removed 1% of the line drawing trials. Using the remaining error-free full cycles, we used the subtraction method explained above to calculate *mode-switch times*. Visual inspection of the mode-switch time distribution suggested non-normality, confirmed by a Shapiro-Wilk test. To compensate, all mode-switching time data points were log transformed. There were 29 data points with slightly negative mode-switching times, primarily for the fastest techniques ND-PALM and ND-DEVICE. To compensate, we added 440 ms to all times to guarantee positive values required by the log transform. Note that log transformed data is only used for statistical tests, all reported times are actual measured values.

Analysis Method

For mode-switching time and error rates, we performed a TECHNIQUE × BLOCK repeated measures ANOVA. Pairwise t-tests with Bonferroni corrections are used. For interaction effects, we restrict pairwise tests to comparing means across factor dimensions independently. When the sphericity condition is violated, degrees of freedom were adjusted using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$) corrections.

Learning Effects

To determine if performance changed during the four compound blocks, we tested for effects of TECHNIQUE × BLOCK on *mode-switch time* and *combined error rate* (one measure capturing whether any error occurred). Also, to determine if performance changed between the start and end baseline blocks, we tested effects of TECHNIQUE ×BLOCK on *cycle time* of baseline blocks only. We found no statistically significant interactions involving block in any of these test, indicating no or minimal learning effects. This matches the performance stability noted by Li et al. [153] and Surale et al. [254] All blocks are used in subsequent analyses, with BLOCK used as a repeated measure.

Mode Switch Time

Most non-dominant techniques were faster than dominant ones, with the device controller being the fastest (Figure 4.4). There was a significant main effect of TECHNIQUE

($F_{7,105}$ = 4.93, $p < .0001$, η^2_G = .23). Post hoc tests found ND-DEVICE faster than all dominant techniques (D-FIST, D-POINT, and D-PALM) (all $p < .001$), and two non-dominant techniques (ND-HEAD, ND-FOV) (both $p < 0.05$). ND-PALM was faster than D-PALM ($p < .05$), D-FIST ($p < .0001$), D-POINT ($p < .001$), and ND-FOV ($p < .01$). Furthermore, ND-FIST was faster than D-FIST and D-POINT (both $p < .05$). No significant differences were among the dominant hand techniques. Ranking TECHNIQUES from fastest to slowest measured mode-switch time: ND-DEVICE (331 ms), ND-PALM (459 ms), ND-FIST (513 ms), ND-HEAD (528 ms), D-PALM (729 ms), ND-FOV (793 ms), D-FIST (846 ms), and D-POINT (955 ms).

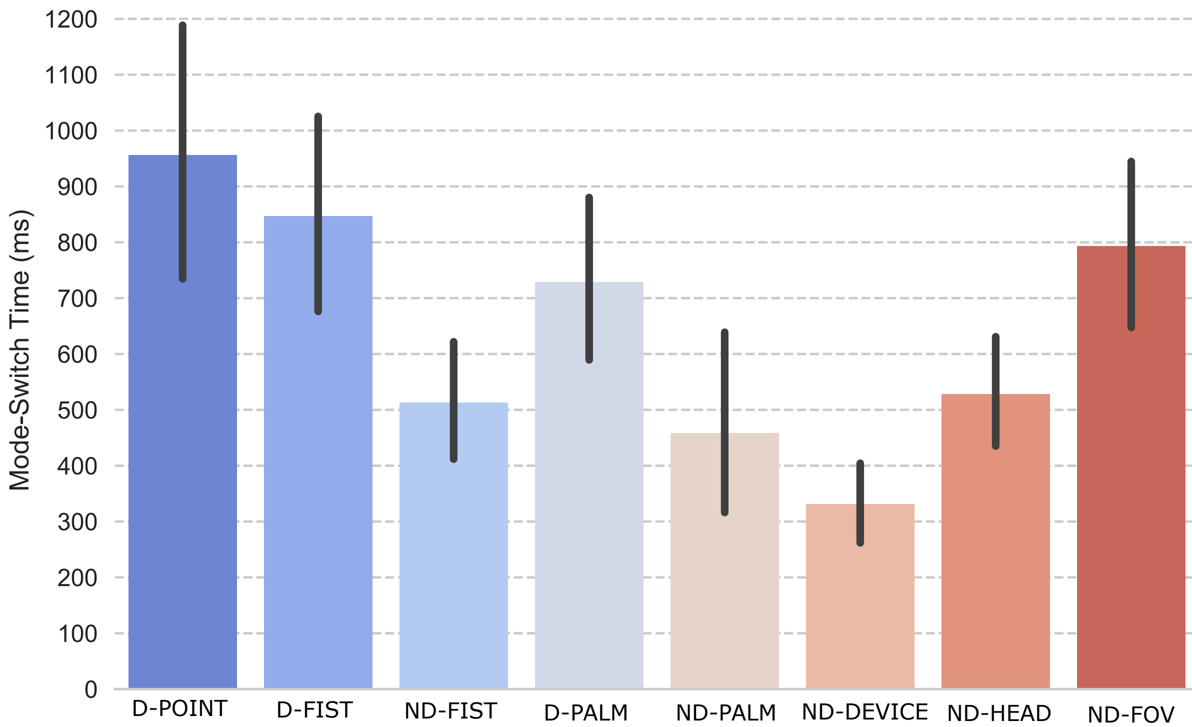


Figure 4.4: *Mode-switch time* by TECHNIQUE (error bars in all graphs are 95% CI).

Overall Error Rate

Overall error rates were between 5.3% and 18.3%, with dominant techniques more error-prone than non-dominant techniques. Non-dominant hand mid-air techniques were not significantly different from using a device controller. There is a significant main effect of TECHNIQUE on *overall error rate* ($F_{7,105}$ = 5.95, $p < .0001$, η^2_G = .28). Post-hoc tests revealed that the rate for ND-DEVICE (5.3%) was lower than that of D-POINT (18.3%), D-PALM (13.6%), and D-FIST (15.3%) (all $p < .001$). The D-POINT error rate was higher than that of ND-HEAD (7.8%), ND-PALM (8.0%), ND-FOV (9.2%) (all $p < .001$), and ND-FIST (9.5%) ($p < .05$). Finally, D-FIST also had a higher error rate than ND-FOV, ND-HEAD, and ND-PALM (all $p < .05$). Note that the overall error rate for the baseline task is 8%, so rates between 5.3% to 18.3% with mode-switching are quite similar. Error rates in this range are in line with reported error rates for mid-air interactions [163, 162].

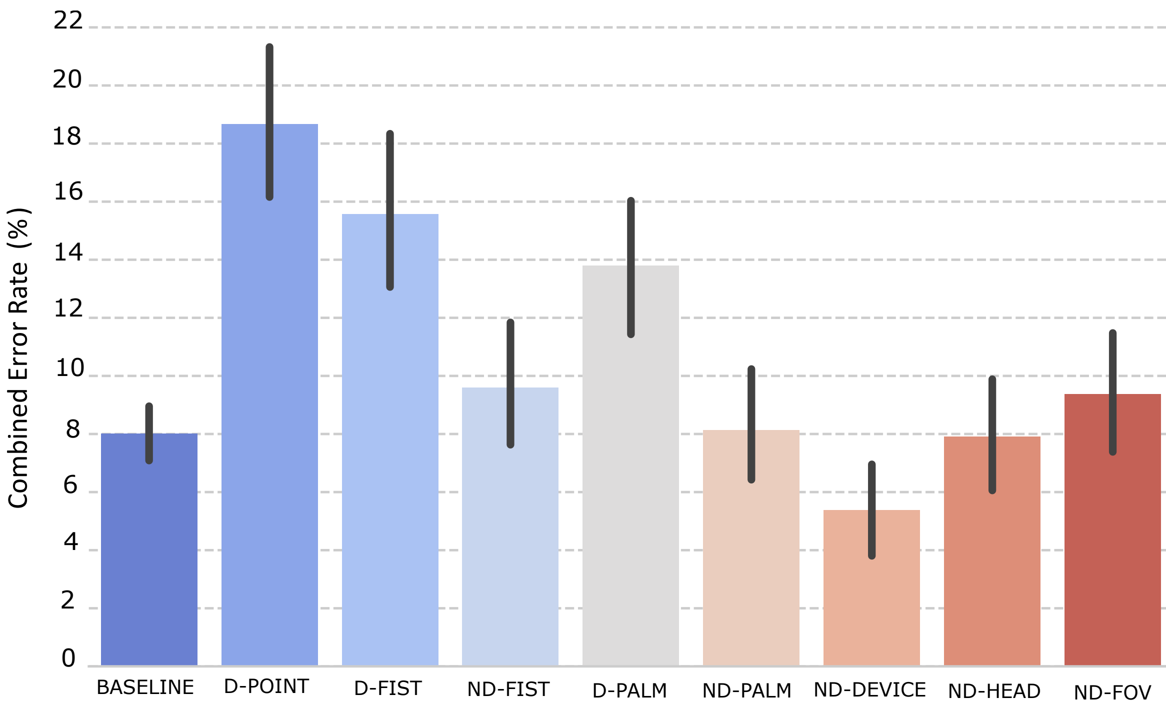


Figure 4.5: *Overall error rate* by TECHNIQUE.

Start and End Error Rates

start error rate was the largest contributor to *overall error rate* (Figure 4.6). Measured rates were between 3.7% and 14.7%, with dominant techniques more error prone. A main effect of TECHNIQUE on *start error rate* ($F_{7,105} = 5.26$, $p < .0001$, $\eta^2_G = .26$) with post hoc tests showed ND-DEVICE (3.7%) was more robust than D-FIST (12%), D-POINT (14.7%) (both $p < .001$), and D-PALM (9%) ($p < .05$). Error rates of ND-FIST (7.6%) ($p < .05$), ND-FOV

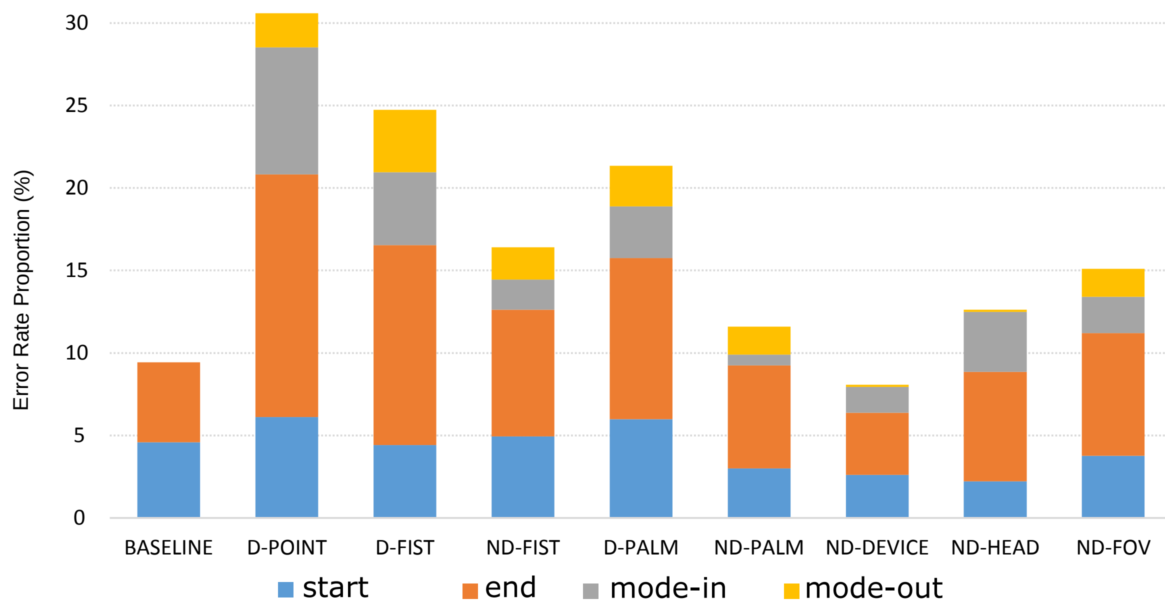


Figure 4.6: Proportion of specific error rates by TECHNIQUE. Note multiple error types can occur in a cycle.

(7.4%), ND-HEAD (6.6%), and ND-PALM (6.2%) ($p < .001$) were significantly lower than D-POINT (14.7%), and the ND-PALM rate was lower than D-FIST (all $p < .001$). For *end error rate*, we did not find significant effect of TECHNIQUE, BLOCK, or their interaction. Measured rates were between 2.2% and 6.1%.

Mode-In and Mode-Out Error Rates

For *mode-in error rate*, Non-dominant hand techniques were comparable to using a controller, with dominant fist and palm slightly more error-prone. Dominant pointing had high mode-in errors. All rates were between 0.6% and 7.6%. There is a main effect of TECHNIQUE on *mode-in error rate* ($F_{3.79,56.92} = 6.10$, $p < .0001$, $\eta^2_G = .29$). Post hoc tests showed D-POINT had a higher rate (7.6%) than all other techniques: D-PALM (3.1%), ND-DEVICE (1.5%), ND-FIST (1.8%), ND-FOV (2.2%), ND-HEAD (3.6%), and ND-PALM (0.6%) (all $p < .05$). Moreover, the ND-PALM rate was lower than D-FIST (4.4%), D-PALM, and ND-HEAD (all $p < .05$).

Except D-FIST, most of the techniques had *mode-out error rates* below 2.4%, suggesting that mode disengagement was a minor contributor toward *overall error rate*. Measured

	D-POINT	D-FIST	ND-FIST	D-PALM	ND-PALM	ND-DEVICE	ND-HEAD	ND-FOV
Accuracy	3.1 ±.3	3.7 ±.3	3.9 ±.2	3.2 ±.2	4.4 ±.2	**4.9 ±.1**	4.4 ±.2	3.9 ±.3
Learning	3.1 ±.4	3.7 ±.3	4.3 ±.3	3.2 ±.3	4.7 ±.1	**5.0 ±.0**	4.4 ±.2	4.1 ±.3
Ease of Use	3.2 ±.3	3.5 ±.3	3.9 ±.3	2.9 ±.3	4.7 ±.2	**4.8 ±.2**	4.1 ±.3	4.0 ±.3
Eye Fatigue	**4.6 ±.3**	4.8 ±.2	4.9 ±.1	4.9 ±.1	4.9 ±.1	4.9 ±.1	4.9 ±.1	**4.6 ±.3**
Hand Fatigue	**3.9 ±.4**	4.2 ±.3	4.2 ±.3	**3.9 ±.4**	4.4 ±.2	4.9 ±.1	4.4 ±.3	**3.9 ±.4**
Speed	3.4 ±.3	4.1 ±.2	4.2 ±.2	3.6 ±.3	4.5 ±.2	**4.8 ±.2**	4.3 ±.2	3.8 ±.3
Combined	3.6 ±.1	4.0 ±.1	4.2 ±.1	3.6 ±.1	4.6 ±.1	**4.9 ±.0**	4.4 ±.1	4.1 ±.1

Table 4.2: Subjective ratings for Experiment 1 (mean ± SEM).

rates were between 0.1% and 3.7%. There is a main effect of TECHNIQUE on *mode-out error rate* ($F_{4.13,62} = 3.28$, $p < .01$, $\eta^2_G = .18$). Post hoc tests showed ND-DEVICE (0.01%) and ND-HEAD (0.01%) had a lower error rate than ND-FIST (1.9%), D-FIST (3.7%), and D-PALM (2.4%) (all $p < .05$).

Subjective Ratings

After the experiment, we asked participants to rate each technique with respect to six aspects: ease-of-learning, ease-of-use, accuracy, speed, eye fatigue and hand fatigue. A 5-point continuous scale was used, with 1 being the worst score (e.g. low accuracy, hard to learn, very fatiguing) and 5 being the best (e.g. high accuracy, easy to learn, not fatiguing).

Table 4.2 summarizes the ratings. The distribution of ratings was non-normal due to high negative skewness, so values were transformed using Aligned Rank Transform [291]. ANOVAs performed on transformed data revealed significant main effects of TECHNIQUE on all the aspects except eye fatigue. The main results of pairwise comparisons between TECHNIQUE for each rating are:

- For *hand fatigue*, D-PALM ($p = .049$) and ND-FOV ($p < .03$) were perceived as more fatiguing than ND-DEVICE ($p < .05$). However, for the remaining techniques, reported fatigue levels were not significantly different. This indicates that fatigue levels for the overall experiment were low.

- For *speed* and *accuracy*, D-POINT and D-PALM were rated significantly slower and less accurate than ND-DEVICE ($p < .01$). For the remaining techniques, there was no significant difference in terms of speed. ND-DEVICE was perceived as being more accurate than all the dominant techniques ($p < .001$) and some non-dominant techniques, ND-FIST and ND-FOV ($p < .05$). The remaining techniques, ND-HEAD and ND-PALM were rated as being more accurate than D-POINT and D-PALM ($p < .05$).

- For *ease-of-use* and *ease-of-learning*, D-PALM was rated worse than all non-dominant techniques ($p < .05$), followed by D-POINT, which was rated worse than all non-dominant techniques ($p < .05$) except ND-FOV. D-FIST was perceived as being harder to learn than ND-DEVICE and ND-PALM. ND-DEVICE was easier to use compared to all the dominant hand techniques ($p < .01$).

4.4.6 Summary

Two non-dominant techniques, forming a fist or palm, were not significantly different from using a controller device, and had error rates comparable to the un-moded baseline task. Yet, almost half of the participants picked the controller as most preferred. To understand how the pinch engagement trigger affects mode-switching, the next experiment evaluates a dominant device controller as a trigger.

Overall, non-dominant techniques are generally faster and less error prone than dominant ones. However, using only a dominant hand would free the other hand, and in theory, this should reduce fatigue. So why did dominant techniques perform relatively poorly? Participant comments suggest some confusion: *"Difficult to change the modes using the same hand. It is frustrating when it recognizes hand inaccurately"* [P3], *"I feel Index, Fist, and Pinch are all same"* [P7]. When choosing fist, palm, and point for dominant techniques, we felt these would reduce confusion since they are very different from the un-moded pinch trigger. Instead, using very different actions seem to have increased confusion, and perhaps introduced a time penalty for larger finger movements required to switch between pinching and the mode-switch action. We explore this in the next experiment by testing subtle variations of a pinch for mode-switching actions.

4.5 Experiment 2

This experiment has two goals. First, test more subtle dominant hand mode-switching techniques to see if actions more similar to a pinch trigger might perform better. Second, test the effect of using a device controller as the manipulation trigger. The apparatus, quantitative measures and experimental protocol are the same as in Experiment 1.

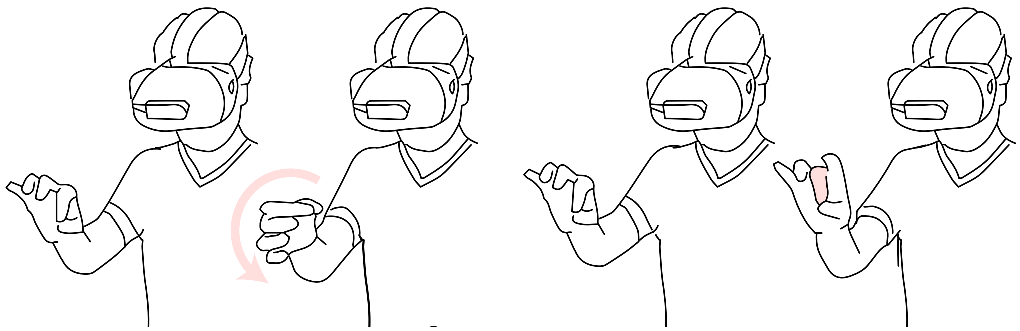


Figure 4.7: Mode switching techniques evaluated in experiment 2. (a) orientated pinch (D-ORIENT); (b) middle finger pinch (D-MIDDLE)

4.5.1 Participants

We recruited 12 participants (mean age 29.6 SD = 3.37, 5 women, 1 left-handed). 6 participants had experience using VR device. Remuneration was $10. Each participant filled out a questionnaire before the experiment began and after the experiment was conducted. A copy of these questionnaires are included in Appendix B.

4.5.2 Techniques

We tested two sets of techniques. The first set consists of two subtle variations of a dominant hand pinch as a mode-switching action. We use a *pinch baseline* in this set, meaning a pinch is used to engage line drawing as in Experiment 1. The techniques are:

- *Oriented Pinch* (D-ORIENT) — The mode is active when the wrist is rotated clockwise (from the user's perspective) more than 45°. Manipulation is engaged and disengaged using the thumb-to-index pinch.
- *Middle Finger Pinch* (D-MIDDLE) — The mode is active when the middle finger and thumb are pinching. This also acts as simultaneous engagement of the manipulation trigger. Two 7 mm diameter FSRs were taped to the tips of the index and middle fingers for precise detection.

For the second set, we re-use two non-dominant techniques from Experiment 1, but this time with *controller baseline*, where selection is triggered by the button of a device

	D-ORIENT	D-MIDDLE	ND-DEVICE*	ND-PALM*
Accuracy	4.2 ±.2	4.3 ±.3	**4.5 ±.2**	3.8 ±.3
Learning	4.4 ±.3	**4.3 ±.4**	**4.3 ±.3**	3.8 ±.2
Ease of Use	4.1 ±.2	**4.3 ±.4**	4.2 ±.3	3.7 ±.2
Eye Fatigue	**4.8 ±.1**	4.9 ±.1	4.9 ±.1	4.9 ±.1
Hand Fatigue	4.4 ±.3	4.8 ±.1	4.6 ±.3	**4.3 ±.3**
Speed	4.2 ±.3	**4.6 ±.3**	4.4 ±.2	3.8 ±.3
Combined	4.4 ±.1	**4.6 ±.1**	4.5 ±.1	4.1 ±.1

Table 4.3: Subjective ratings for Experiment 2 (mean ± SEM).

controller held in the dominant hand. This enables us to compare drawing a line with and without holding a physical controller. An asterisk post-fix denotes these are versions of the same techniques, but with a controller for selection.

- *Non-Dominant Palm* (ND-PALM*) — The mode is active when all fingers of the non-dominant hand are extended, roughly pointing up, with the palm facing right.
- *Non-Dominant Controller* (ND-DEVICE*) — The mode is active when the button of the non-dominant hand controller is pressed. Note that both hands hold controllers.

4.5.3 Design and Procedure

The design and procedure are similar to the first experiment. The mode switching TECHNIQUE is a within-subjects factor with levels corresponding to the four mode-switching techniques. The TECHNIQUE order was counter-balanced using a 4 × 4 Latin Square. In sum: 4 TECHNIQUES × 6 BLOCKS (2 *baseline*, *4 compound*) × 3 directions × 5 line drawings = 360 line drawings per participant.

4.5.4 Results

Data Pre-Processing

The same methods were used as in Experiment 1. Less than 1% of the line drawing trials were removed as outliers, and mode-switch times were log transformed to correct non-normality. This resulted in 29 negative values so 275 ms was added to all times to obtain positive values. Again, log transformed data is used only for statistical analyses. Reported times are measured values.

Learning Effects

To determine if performance changed during the four compound blocks, we tested for effects of TECHNIQUE ×BLOCK on *mode-switch time* and *combined error rate*. We found no statistically significant interaction indicating no learning effect across blocks, so all blocks are used in the subsequent analysis.

Mode Switch Time

Middle finger pinch, a more subtle dominant technique, was among the fastest techniques. There was a significant main effect of TECHNIQUE ($F_{3,33} = 3.96$, $p < .05$, $\eta^2_G = .26$). Post hoc tests found that ND-DEVICE∗ was faster than ND-PALM∗ and D-ORIENT ($p < .05$). Furthermore, D-MIDDLE is faster than D-ORIENT ($p < .001$). However, we did not find significant differences between ND-DEVICE∗ and D-MIDDLE. Ranking TECHNIQUES from fastest to slowest mode-switch time: ND-DEVICE∗ (226 ms) or D-MIDDLE (233 ms), ND-PALM∗ (467 ms), and D-ORIENT (669 ms).

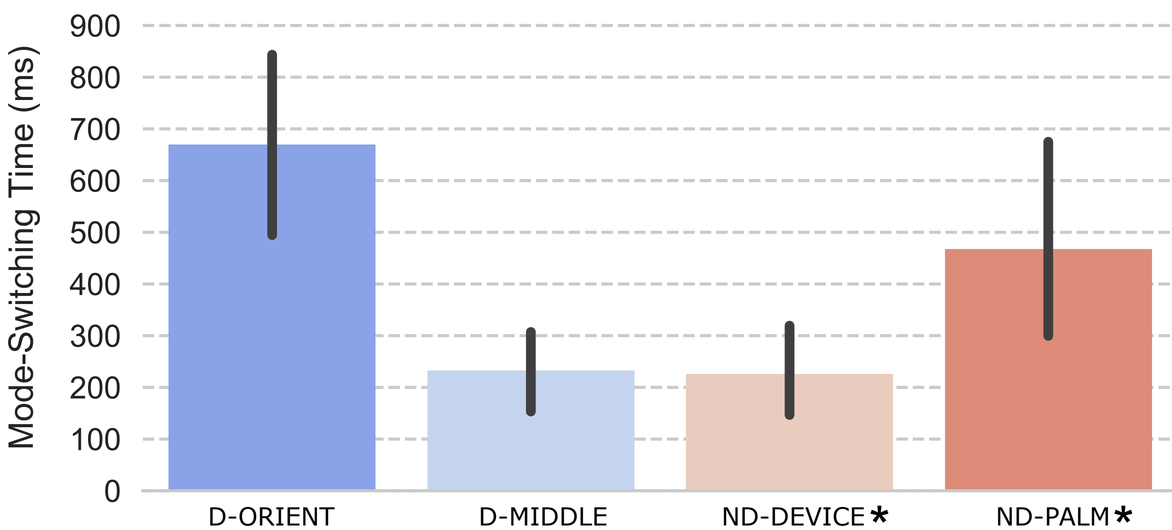


Figure 4.8: Mode-switch times by TECHNIQUE.

Error Rates

Overall error rate for the pinch and controller baseline tasks are below 5% (3.9% to 4.5%), and between 5% to 9.5% for mode-switching techniques (Figure 4.9). All techniques have comparable error rates, with the exception of middle finger pinch, which had a higher end error rate than non-dominant palm (Figure 4.10). There is no significant effect of TECHNIQUE on any type of error, except for *end error rate*. We found a main effect of TECHNIQUE on *end error rate* ($F_{3,33} = 3.52$, $p < .05$, $\eta^2_G = .26$). Post hoc tests show that D-MIDDLE (5.0%) is higher than ND-PALM∗ (0.6%) ($p < .05$).

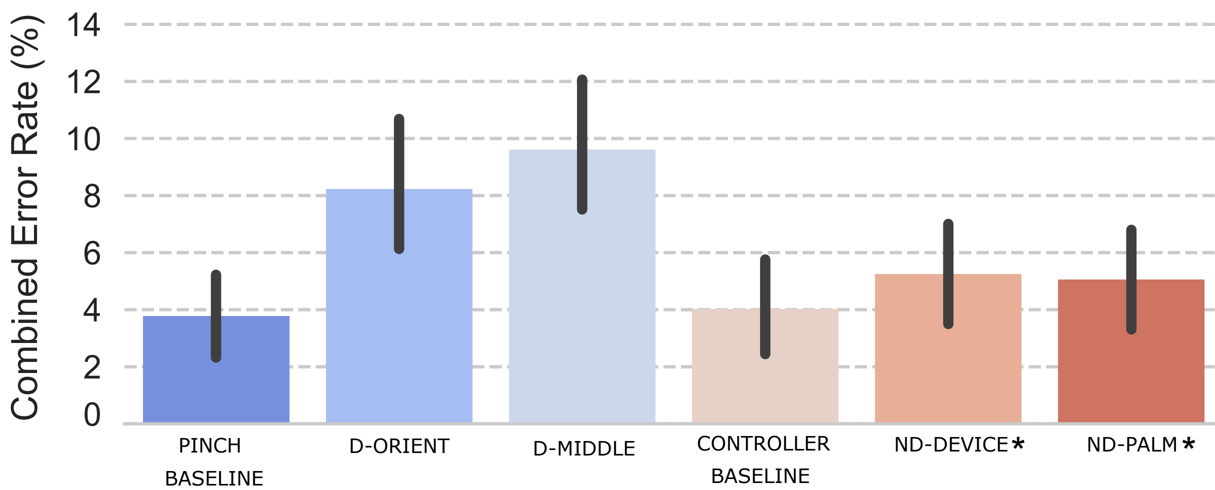


Figure 4.9: *Overall error rate* by TECHNIQUE. Baseline techniques have *start* and *end error rates*.

Subjective Ratings

Similar procedures were followed to collect and process subjecting ratings. However, no statistically significant differences were found for any aspect, suggesting participants may have perceived all four techniques equally. Table 4.3 summarizes those ratings. For overall preference, 42% of the participants liked D-MIDDLE and ND-DEVICE∗ and 42% disliked ND-PALM∗.

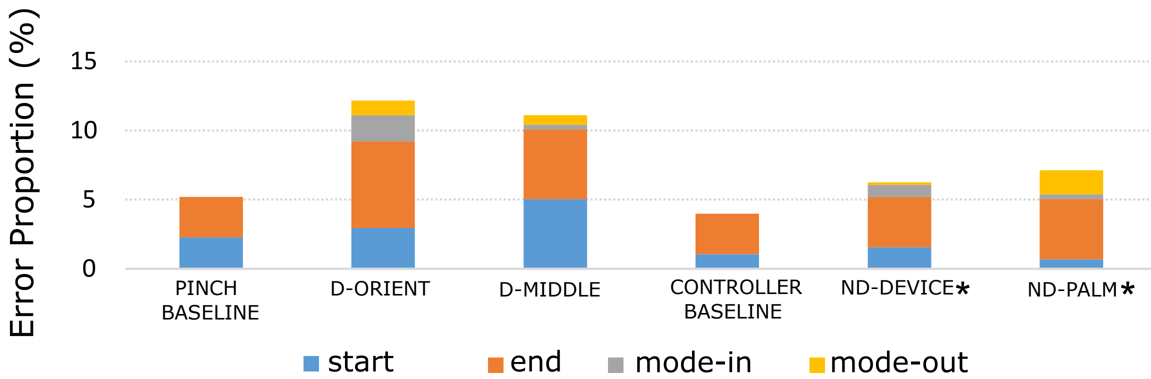


Figure 4.10: Proportion of specific error rates by TECHNIQUE.

4.6 Discussion

We discuss findings considering both experiments, and their potential impact on the design of VR interfaces.

Use non-dominant actions for accuracy-focused tasks and dominant actions for longer tasks. When considering accuracy only, non-dominant techniques such as raising a palm or touching the head are superior. If dominant hand mode switching is required, a subtle variation on the selection trigger, such as the middle finger pinch, is recommended. Slower and more error prone dominant postures, such as fist or palm, may be suitable for infrequent mode switching and tasks with less critical accuracy demands, such as panning. Also consider that while the non-dominant techniques are precise and fast, fatigue may be an issue for longer periods of use [121]. A mix of dominant and non-dominant mode-switching might prove to be most effective.

Avoid the pointing posture for frequent mode-switching. Dominant pointing was the slowest technique, and led to the most errors. This is surprising, since it is a commonly used mid-air bare hand action. Using it for raycast pointing is likely suitable [281], but other pointing techniques should be considered if frequent mode switching is expected.

Subtle dominant hand techniques are a promising alternative to non-dominant controller technique. Experiment 1 showed that the device controller technique was least error prone and fastest, but Experiment 2 shows that dominant-hand techniques can reach the performance levels of non-dominant ones when made subtle, as demonstrated by using the middle finger pinch as a simultaneous mode-switch and manipulation trigger. While we found significant differences between using a device and all dominant techniques in Experiment 1, mode-switching time for middle finger pinch (233 ms) was not significantly different from the device controller (331 ms) in Experiment 2. Considering the device controller was used for both mode-switching and manipulation trigger, this is quite remarkable. For overall errors, middle-finger pinch was comparable for all error rates, except the end error rate. However, the overall error rate for middle finger (9.5%) may be evidence of some potential increase compared to the very low rates when using a device controller. Nevertheless, subtle dominant hand techniques were perceived to be less fatiguing. So, in the future, more variations of subtle gestures need to be tested, for instance, single handed microgestures [45], which could be useful for long hours of use.

Switching between very different dominant postures may be confusing. For dominant hand techniques, we noted that higher error rates may have had more to do with motor-control confusion caused by rapidly switching between the baseline pinch and a qualitatively different posture such as fist, point, or hand. We believe the similarity to the basic pinch for the oriented pinch and middle finger pinch in Experiment 2 alleviated this confusion.

A pinch is a practical alternative to a controller button We saw little practical difference between using a pinch trigger compared to a controller button. When the non-dominant mode-switching action is held constant, any difference between a pinch or a controller button for line drawing trigger seems negligible. Comparing across experiments, t-tests found no significant differences in mode-switch time between ND-DEVICE and ND-DEVICE$\star$ ($t(97.2) = 1.7941$, $p = .07$), or between ND-PALM and ND-PALM$\star$ ($t(99.2) = -0.0656$, $p = .94$). Comparing only line drawing times during baseline tasks for these same two pairs further demonstrates a similarity. The times are actually significantly faster when using a pinch trigger compared to a device trigger ($t(1775.5) = -4.20$, $p < .001$), but the effect size is as little as 41ms between ND-DEVICE and ND-DEVICE$\star$. Finally, we observed no differences between overall baseline task time in Experiment 2 when pinching for a trigger, or when using a device button ($t(102.3) = 1.39$, $p = 0.16$). This is an encouraging result showing that a dominant pinch action for selection may be as effective as a device controller button.

4.7 Conclusion

We presented an analysis of bare hand mid-air mode-switching techniques for VR. Techniques in Experiment 1 were selected using a principled review of related work, and techniques in Experiment 2 were selected to investigate specific questions raised by the first experiment. We found non-dominant techniques to be fast and accurate compared to most dominant techniques, but a dominant middle finger pinch shows comparable performance. Most dominant hand techniques, including popular actions like fist and point, also incurred high error rates likely due to confusion with the unmoded pinch manipulation trigger. Our findings can assist designers in making informed decisions when mapping techniques to mode-switching actions for VR applications. Our results may generalize to other 3D interfaces using absolute positioning and direct manipulation, like AR. We hope these results prove as useful to the VR community as previous mode-switching studies have been for pen and touch.

Chapter 5

Combining Touch and Mid-Air Input

Complex virtual reality (VR) tasks, like 3D solid modelling, are challenging with standard input controllers. In this chapter, we propose exploiting the affordances and input capabilities when using a 3D-tracked multi-touch tablet in an immersive VR environment. Observations gained during semi-structured interviews with general users, and those experienced with 3D software, are used to define a set of design dimensions and guidelines. These are used to develop a vocabulary of interaction techniques to demonstrate how a tablet's precise touch input capability, physical shape, metaphorical associations, and natural compatibility with barehand mid-air input can be used in VR. For example, transforming objects with touch input, "cutting" objects by using the tablet as a physical "knife", navigating in 3D by using the tablet as a viewport, and triggering commands by interleaving bare-hand input around the tablet. Key aspects of the vocabulary are evaluated with users, with results validating the approach.

5.1 Motivation

While virtual reality (VR) has been around in various forms since at least the 1960s (e.g., [256]), advances in display technology have sparked a new interest from both researchers and the public. There are clear advantages to virtual reality, like the ability to look and move around in an immersive 3D environment. Yet, VR interaction is challenging due to limited tactile feedback, poor input precision when drawing [8], and lack of a consistent interaction vocabulary. Past research has introduced methods for haptic feedback [174, 289, 49, 252, 15], techniques to increase precision [208], and more standardized control schemes [244]. In our work, we leverage the familiarity and ubiquity of multi-touch tablets as a means of interacting with 3D content in a VR world.

We introduce a "TabletInVR" design space combining a 3D-tracked tablet with mid-air barehand gestures, which we demonstrate in an example interaction vocabulary for 3D modelling.

Exploring VR interaction in the context of 3D modelling is particularly compelling because the task *should* be a good fit for VR, but in practice, supporting the many required operations is challenging (e.g. object creation, selection, transformation; world navigation; copy, paste, undo, etc.). Although past research has considered the use of 2D surfaces in VR, this has focused on 3D-tracked props without real multi-touch input [158, 211, 157], or using multi-touch tablets for transforming 3D objects without exploiting 3D tablet tracking [48, 221].

Our work combines the affordances of a 3D-tracked tablet with the input capabilities of its multi-touch surface. We advocate that the tablet's precise touch input capability, physical shape, metaphorical associations, and natural compatibility with barehand, mid-air input can be effectively used in VR. Interactions involving precise mutli-touch input could begin on the tablet followed by coarse hand gestures in VR, or tablet input could be used to transform objects or navigate the world in a familiar mutli-touch way. This suggests interesting aspects when combining these two modalities. Interactions can leverage physical qualities like the 2D tablet input providing a continuous tactile sensation and a mid-air gesture enabling free movement in space. Interactions spanning the tablet and mid-air gestures could more effectively exploit bimanual input, since bimanual multi-touch input is known to work well [285, 75], but pure bimanual mid-air gestures are less reliable in VR [111, 245].

Our primary contribution is the definition and exploration of a design space of using a multi-touch tablet in VR.

In a formative study, we asked participants familiar with 3D software to envision how they would perform standard 3D modelling tasks using a tablet in VR.

Based on observations of behavioural patterns and proposed features, we mapped out a design space with eleven dimensions (e.g., 'physical vs. non-physical', 'direct vs. indirect', and 'discrete vs. continuous') and developed a vocabulary of interactions (e.g., 'two-finger drag to translate an object', 'five-finger drag to navigate', and 'swipe-in to delete the object'). Lastly, we validated our system similar to Arora et al. [7], where participants created 3D models to test the design space.

5.2 Background and Related Work

In this section, we summarize prior investigations in the domain of mixed reality environments, using props as tablet, touch input, tablet and pen, and hand gesture input, while focusing on 3D solid modelling.

5.2.1 Tablet and Pen in Mixed Reality

One common approach found in past work is to use a tablet with a pen as an input device in VR. Aspin et al. [10] explored the use of a 3D tracked tablet and stylus in a CAVE-like system for navigation and exploration of small, complex 3D structures. Bowman et al. [34] explored using a tablet and stylus in VR for the assessment of building structures. Billinghurst et al. [24] used a pen operated pressure sensitive pad to support content creation in a virtual environment. Similarly, Bornik et al. [31] used a 6-DOF tracked pen with a tablet to view and manipulate medical data, Reitmayr and Schmalstieg [217] explored the use of a pen and a tablet-like pad (both are props tracked using markers) for a collaboration task, and Sareika et al. [228] investigated bimanual interactions for urban planning using a pen and tablet. Keefe et al. [134] explored precise mid-air strokes using a haptic-aided input technique for 3D sketching, and Arora et al. [8, 7] investigated the impact of the lack of a physical surface on drawing inaccuracies. Their work explored both 3D sketching in augmented reality (AR) using a mid-air pen-based drawing and 2D surface sketching . More recently, Aslan et al. [9] conducted a series of studies to gauge the potential of pen and mid-air input and noted that mid-air input should complement pen and touch-enabled tablets. However, compared to modern high-fidelity multi-touch tablets, pen input is essentially limited. It does not take the full advantages of direct multi-touch finger input.

Albeit sketching using a pen is not our focus, we look at these results through the lens of interaction design. A few lessons to learn before we address 3D modelling. For instance, tablet's continuous input on tablet surface; drawing in mid-air with the help of the tablet, orientating tablet surface in arbitrary plane, and so forth.

5.2.2 Tabletop and Hand Gestures in Mixed Reality

Benko et al. [16, 17] explored the interaction space combining a tabletop and hand gestures in a partially immersive environment. Marquardt et al.'s *continuous interaction space* [168] and MockupBuilder [55] both demonstrated these sorts of 2.5D interactions. They start on a planar surface and continue in mid-air, something highly promising for 3D modeling applications. However, a tabletop surface does not have the same flexibility as a hand-held tablet, such as orientation tracking, mobility, a mid-sized display, and so forth. Being able to carry the tablet around facilitates interactions without being physically constrained to a certain position.

5.2.3 Tablet as a Prop in Mixed Reality

Another way to provide 2D input in VR is to use a prop like a tablet, without a touch sensor. For instance, Linderman et al. [158] demonstrated the use of a passive-haptic paddle as a 2D input device for widget selection in VR, Poupyrev et al. [211] used it for text-based applications (note-taking, text input, and annotation using physical pen as a prop), and Szalavari et al. [257] used it for 3D modelling application. However, none of the past efforts have explored the simultaneous use of multi-touch tablets with hand gestures like in our work.

5.2.4 Tablet touch in Mixed Reality

Wang and Lindeman [288] presented an AR environment consisting of a semi-transparent HMD (Head Mounted Display), a wand in the right hand, and a multi-touch tablet mounted on the left forearm. The interface enabled looking at the virtual environment, as well as seeing the tablet mounted on the non-dominant hand of the user. However, tablet touch interaction was cumbersome since the wand had to be held somewhere other than the right hand temporarily (in the left hand, or between the legs). Kim et al. [136] explored a scaled-down locomotion that allows a user to travel in a virtual world as their fingers slide on a multi-touch surface. However, finger motions were not precisely detected and only two finger touch was investigated . In contrast, our system adds more expressivity by tracking two or more touch points [292], utilizing device orientation [225] to navigate in arbitrary plane and using mid-air hand gestures [47, 204].

5.3 Formative Study

The goal of this formative study is to gain insights into how people envision using a physical tablet in a VR environment, using the context of a 3D modelling application. Our approach

Task	Subtask	Suggested Actions
Create	Primitive	Draw on tablet and extrude or push away, menu buttons
	Clone	Grab longer and drag
Select	Object	Grab, tap on tablet
	Face	Tap on object
	Group	Non dominant grab, two finger, lasso on tablet
Transform	Rotate	Rotate hand
	Translate	Move hand
	Scale	Pinch to zoom, distance between two hands
Modify shape	Slice	Menu buttons, slice with a hand, tablet to slice
	Extrude	Draw 3D path, pinch face
Modify texture	Colouring	Menu buttons

Table 5.1: List of the tasks and corresponding actions suggested by formative study participants.

is similar to work by Hinckley et al., who observed how people used physical paper and notebooks to inform new design spaces for combining touch and stylus input [113] and stylus grip sensing [110]. Like those works, the observations from this formative study are later used to build a design space and an example interaction vocabulary, and we also conduct a preliminary user study to validate our design space through the example interaction vocabulary.

Participants

Ten people (7 male, 3 female, ages 22–26) participated. Three were architecture students experienced with Revit, Fusion360, Sketchup, and SolidWorks software; two were mechanical engineering students experienced with SolidWorks and Fusion360; and two were amateur users with some experience using software like 123D, Blender, and Sketchup. Five participants had experience with VR.

Procedure

Participants were told to imagine how they would use a multi-touch tablet together with mid-air barehand gestures in a fully immersive 3D environment containing 3D objects. Like past work [113, 110], participants were asked to act out specific tasks. In our case, these were basic 3D modelling operations (see Table 5.1). Each participant used a multi-touch tablet (which was not turned on). A chair was provided, but the decision to sit or stand was left to the participant. Two small and one medium sized cardboard cubes were placed around the user, some within arm's reach, and some beyond. These cubes helped them visualize an object to create or manipulate without wearing an HMD.

We asked each participant to imagine and act out 3D object creation, selection, manipulation, and annotation. While performing the tasks, they simultaneously explained their envisioned system using a think-aloud protocol. This included the steps they took, and their opinions about important considerations and choices they made. Observations were recorded by the experimenter as written notes.

5.3.1 Observations

We observed participants' behaviour and analyzed notes using affinity diagramming to reason about the role of a tablet in VR. Our design space is a manifestation of the following seven core observations.

Delegation of Tasks:

O1. *Granular and coarse actions*: participants preferred using mid-air hand gestures for coarse actions, followed by input on the tablet for finer control, "I'd grab an object and then use the tablet to rotate it." [P3]

O2. *Near and far actions*: instead of navigating to a distant object, beyond arm's reach, participants preferred indirect object selection using the tablet. For instance, the tablet's screen could depict a birds-eye view [288, 251], where a tap on the tablet selects an object. Participants also suggested treating the tablet as a remote control, so they could raycast to select. However, to select objects within arm's reach, they preferred to reach out and grab with their hands.

Tablet Properties:

O3. *Tablet as interface*: participants suggested using menu buttons (2D) on the tablet to create objects, invoke commands, and select modes. Although they utilized a mixture of mid-air hand and touch-based gestures, most tasks were initiated on the tablet with a tap of a button. Tracking the tablet orientation and position creates novel precision-focused interactions. For example, to translate an object, a user can tap on the tablet to select it, and then drag with their fingers to translate while adjusting the translation axis through the tablet's orientation.

O4. *Tablet as a tool*: Despite not being common, a few participants used the tablet to define a slicing plane, and some used their dominant hand for slicing an object (like a knife). this behaviour was from a fruit ninja game, where players use a sword to cut through fruits. Other participants used bare hands to slice an object, but were skeptical about accuracy and unsure it was a suitable operation.

The physical form of the tablet affords a variety of operations when tracked and rendered virtually in VR. It can be made to resemble a knife, a tray, a rectangular block, a ruler, a storage unit, among other physical forms. The plane of a tablet can be aligned with the face of an arbitrary 3D object to extrude, color, or even delete it. A corner of a tablet can be used as a pointer, which can be used to select objects.

O5. *Haptic feedback*: mid-air hand gestures seemed suitable for discrete interaction and touch-based interactions were favoured for continuous manipulation. This behaviour appeared to be linked with the demand for haptic feedback and the perceived precision requirements of the task. For instance, participants suggested hand gestures to grab an object and a pinch-to-zoom gesture on the tablet to scale it. Moreover, prior research has shown that haptic sensations in VR can greatly improve the user experience [289, 49, 118].

Symbolic input and UI interactions (e.g., buttons, menus, etc.) can also benefit from having a physical, tactile surface. The tablet can act as an arbitrary UI (e.g., to annotate objects or select modelling operations), and the tactile feedback can improve typing speed when compared to mid-air typing without haptic feedback [90].

General Observations:

O6. *Occlusion avoidance*: participants felt that using a tablet for continuous manipulation tasks made more sense than using hand gestures, as it avoids occluding the object of interest and requires minimal efforts, "[...] and my hands will not even occlude the object." [P1].

O7. *Sit vs. Stand*: all participants preferred sitting, except one who demonstrated a willingness to stand, "I can stand if I have to look at the cube from the top side." [P2], but still opted to sit throughout the exercise.

5.4 Design Space

Following a systematic approach, we consider these observations ($O1$ - $O7$), depicting user behaviour, to build the design space, followed by designing the interaction vocabulary. For each candidate dimension, in design space, we pose intriguing questions that instigate design considerations. Such considerations would help interaction designers assign different roles to the tablet. Note that our focus is on building a design space and interactions for using a tablet in VR. We do not contest to investigate 2D input or hand gesture input in isolation [120, 25, 292] or passive 2D input in VR, as it has been studied elsewhere [158, 67].

Informed by our set of high-level observations from the formative study along with past research, we shape a design space. Recall that we asked participants to envision interactions for three settings, so, we assemble these interactions together with corresponding dimensions in each setting to bring out novel and rational interactions. These dimensions are essentially the lenses thorough which we can envision the possibilities of the TabletInVR concept.

Design Dimensions

Design dimensions are the core components of interaction space, where each interaction we envision is composed of one or more of the following design dimensions.

D1. *Tablet vs. mid-air properties*: Table 5.2 describes different properties of tablet and mid-air interactions in VR. Tablet in VR could be mutually beneficial given high precision input space on 2D surface. However, when does high precision input is essential? Is 3DOF input adequate? In VR, which interactions need tactile feedback? or UI?

D2. *Non-dominant (ND) vs. dominant (D) hand assignment*: Participants used their dominant hand while using the tablet as a tool (O4). How the ND and D hand roles are defined based on the use of the tablet in VR?

D3. *Sit vs. stand*: Body posture can have an impact on fatigue and on the interaction experience in general. As pointed out in O7, only one participant was willing to stand, and this depended on the task. What tasks are suited to sitting vs. standing? Can VR provide the flexibility to either stand or sit irrespective of the task?

Properties \ In VR	Tablet touchscreen	Mid-air tablet	Mid-air hand
Precision	High	Low	Low
Input space	2D (3DOF)	3D (6DOF)	3D (high DOF)
Tactile feedback	Yes	No	No
UI	Familiar (WIMP)	No (tilt)	No (gesture)
Midas touch	No	Yes	Yes

Table 5.2: Tablet vs. Mid-air properties.

D4. *Attention to device vs. scene*: Recently, Yan et al. [299] found that, compared to eyes-engaged, the eyes-free approach is significantly faster, provides satisfying accuracy, and introduces less fatigue and sickness. Can interaction spaces be divided into different regions, either on the tablet or around the user, to guide attention and leverage the benefits of eyes-free interaction?

D5. *Unimodal vs. multimodal*: There are many modes of interaction for a tablet in VR. For instance, combined mid-air hand and touch gestures, touch-only, unimanual or bimanual hand gestures, tablet orientation and touch, and so forth. How and when can these modes be applied to reduce fatigue or to improve accuracy, and in general, reduce user frustration?

D6. *Unimanual vs. bimanual*: Past research has explored the benefits of bimanual interaction [91, 41, 258]; however, bimanual interaction may not be suitable for every task, for example, grabbing a virtual object using only the dominant hand. Should a task be performed using either one or both hands? Does it improve the task completion time? Modern VR devices are equipped with reasonably accurate hand-gesture recognizers. Mid-air barehand gestures along with touch gestures enable unique workflows. For instance, a pinch gesture could be used to select an object, followed by a pinch-to-zoom on the tablet. A long pinch could be used to create a ghost copy of an object, followed by a two-finger rotation gesture on the tablet.

D7. *Environment reality vs. virtuality*: There exist multiple ways to provide different levels of visual feedback. For instance, with a standard tablet, the 3D world can exist only in the confined window of a tablet screen. Similarly, when the tablet is tracked to create a viewport, the world can be seen through a tablet screen; however, the virtual objects are physically stuck to the real environment, like in augmented reality. The tablet acts as a portal to the virtual world around the user [92]. Furthermore, when a tablet is used in VR, a portal could let a user view the real world while being in VR [175], creating a 'portal to reality'. While wearing an HMD, the ability to be aware of one's surroundings is essential. With the tablet in hand, a user can peek into reality whenever desired. Prior research has explored using a flat surface to create a viewport [257], however, interacting through a viewport is an unexplored area of research. Further, When transitioning between modes of operation, how and when can an awareness of the real world be provided in the virtual world and vice versa? Can the tablet' s screen used as a portal to and from reality? Does it break the immersion?

D8. *Interleaved vs. simultaneous*: Does an interaction require simultaneous use of both input techniques, such as touch and mid-air hand gestures, or given a task would only one of them would suffice? Is it preferable to use both input techniques in an interleaved fashion, or to solely rely on one of them?

D9. *Discrete vs. continuous input*: While tapping on a virtual object to select it is an example of discrete input, changing the scale of an object is an example of continuous input. However, the mapping from task to input type is not always clear. For instance, consider a relatively complex task of selecting an object from a stack of objects, which is placed beyond arms reach, would we still resort to discrete input or would mixed input be more efficient? What scenarios drive such a mapping? When should a designer opt for discrete input and when should they opt for continuous input?

Furthermore, tablets provide a high-fidelity input space in a low-fidelity virtual environment. We can go beyond taps and clicks to recognize hand-drawn gestures in VR. This would allow users to provide a more advanced form of input. For instance, gesture-based menu invocation [303] and hand-drawn shape recognition could be used to invoke commands, or more traditional forms of input such as pinch-to-zoom and two-finger swipe could be used.

D10. *Direct vs. indirect*: As pointed out in O6, to avoid occlusion, participants used the tablet screen instead of mid-air hand gestures. Similarly, in O2, participants used the tablet screen to select distant objects. These observations hint toward the need for an indirect manipulation technique. Does the interface leverage the full potential of available input methods for direct and indirect tasks?

D11. *Physical vs. non-physical*: O4 highlighted the use of the tablet as an entity which does not necessarily follow physical laws from the real world. We identify this being a crucial factor while assigning roles to the tablet and the user's hands in VR. We try to reason about the possibility of assigning direct interaction with the hands to abide by physical laws of the real world and non-physical interactions using the tablet. For instance, direct tap using a finger might displace a virtual object, while the tablet could pierce through a virtual object to select it. Moreover, could such physical and non-physical interactions lead to a better experience of using tools in VR? Would switching such roles make interactions difficult and unusable? In what cases should direct interactions using hands not follow physical laws?

5.4.1 TabletInVR Prototyping System

How the interaction techniques were implemented was partially influenced by the capabilities of our prototyping system. The application runs in Unity 5.6.1, on a high-end Windows 10 machine (3.6GHz Intel i7 CPU, 1.6GHz GeForce GTX 1080 GPU). The VR HMD is an HTC Vive (1080 × 1200 px per eye, 90Hz refresh, 110° fov). Hand tracking uses a LEAP motion device mounted on the front of the HMD with the interaction engine v1.1.1. The mounting angle and 135° field-of-view of the LEAP camera enables the hand using the tablet to be tracked when the user looks at the tablet. The interaction engine Unity LEAP plug-in displays 3D models of the users hands in VR. It should be noted that LEAP hand tracking is not robust to IR reflection, especially when the tablet is near and when the finger positions are pointing away from the LEAP. As a result, our implementation avoids these kinds of in-air gestures with touch interaction available on the tablet.

The tablet is a 9.7" Samsung Galaxy Tablet S3 (1536 × 2048 px display, 264 ppi), weighing 429g. The 3D position and orientation of the tablet is tracked using an HTC Lighthouse tracker. Tracking of this tracker is glitchy when the docking area is facing the Lighthouse [277], however, it did not hinder the usability. The 9.9cm × 4.2cm tracker is screwed into a lightweight aluminum bar, which is attached to the back of the tablet using high-strength hook-and-loop fasteners. The bar is attached such that the tracker extends approximately 4.5cm out beyond one corner. This mounting position enables portrait and landscape orientation when held in one or two hands, and the tablet can be flipped to use the back as a haptic surface. We perform calibration of the virtual tablet model and the physical tablet manually, by adjusting the rotation and translation offsets until they align. Since the tracker is securely fixed to the tablet, this one-time manual calibration is acceptable. Multitouch events registered by the tablet (x and y coordinates of each touch point) are sent in real-time to the server over a high speed WiFi network.

5.4.2 Example Interaction Vocabulary

We describe an example interaction vocabulary for using a tablet in VR for the purpose of 3D solid modelling. The interaction techniques are informed by the formative study observations and are constructed to span and illustrate the design space dimensions. Table 5.3 shows how the design dimensions informed the interactions. For instance, the *Create* interaction is a result of flipping the tablet using the non-dominant hand (D2) and discrete taps (D1, D6, D9) on the back of the tablet using the dominant hand. Each family of interaction techniques are described generally, with specific implementation details from our application provided to make the ideas more concrete.

Dimensions \ Interactions	Create	Select, Deselect	Delete	Transform	Modify	Annotate	Navigation	Help	System Menu
D1. Tablet vs. mid-air properties	Tap on tablet	-	Mid-air gesture	Drag on tablet	-	-	-	-	-
D2. Hand assignment	Dominant	-	-	-	Dominant	-	-	-	Non-dominant
D3. Sit vs. stand	-	-	-	-	-	-	Both	-	-
D4. Attention to device vs. scene	-	-	-	-	-	On the device	On the device	On the scene	-
D5. Unimodal vs. multimodal	-	-	-	-	Tablet orientation and touch	Touch	Tablet orientation and touch	-	-
D6. Unimanual vs. bimanual	Bimanual	Both	Unimanual	Bimanual	-	-	Bimanual	Bimanual	-
D7. Environmental reality vs. virtuality	-	-	-	-	-	-	Tablet viewport	-	-
D8. Interleaved vs. simultaneous	-	Interleaved	Interleaved	-	-	-	-	-	-
D9. Discrete vs. continuous	Discrete	-	-	Continuous	-	Discrete	Continuous	-	-
D10. Direct vs. indirect	-	Both	-	-	-	-	-	-	-
D11. Physical vs. non-physical	-	Both	-	-	Non-physical	-	-	-	-

Table 5.3: Interactions used in vocabulary with mappings from TabletInVR design dimensions.

3D Modelling Tasks

Foley et al. [72] provide a fundamental interaction task set, independent of application and hardware, for 3D environments—select, position, orient, path, quantify, and text. Our interaction vocabulary includes these tasks and builds upon them with more advanced interactions: selection, deselection, manipulation (rotate, scale, translate), modify (slice and extrude), creation, deletion, annotation, and so on. In our interaction vocabulary that follows, we demonstrate how a tablet's precise touch input capability, physical shape, metaphorical associations, and compatibility with barehand mid-air input can be used in VR to perform these 3D modelling tasks.

Tablet Viewport

We design several interactions to use the affordance and physical properties enabled by a view of the 3D scene rendered in the HMD's view of the tablet display. In addition, we add a 3D ray emanating out from the centre of the back of the tablet. The combination of the viewport rendering, the ray, and available multi-touch input of the tablet creates a useful direct and indirect interface.

Note that the viewport rendering is only on the front display side of the tablet, and it is hidden when the tablet is being used for other purposes, like rotating, translating or scaling an object, or when using the tablet to slice objects. This helps the user maintain focus on the object(s) being transformed (D4). During navigation, only this viewport rendering is visible with the virtual world made uniformly black. This enables the viewport rendering to function as a small view of the 3D scene during navigation, which is less likely to induce motion sickness [197, 70].

Creation

To select an object for creation, a user flips the tablet over with their non-dominant hand, browses a list of objects using a scrolling list, and taps on one to select it (O3, D6, D9). The selected object appears as a 3D icon near the top of the tablet, indicating creation mode is active and what object has been selected for creation (see Figure 5.1).

When an object is selected, there are two ways to create it in the scene. First, the user can *remotely* create objects by pointing the ray from the back of the tablet at the grid on the ground plane. A tap on the tablet screen creates an object at that point on the grid (O5). Second, the user can create objects in *mid-air* (50cm in front of them) by

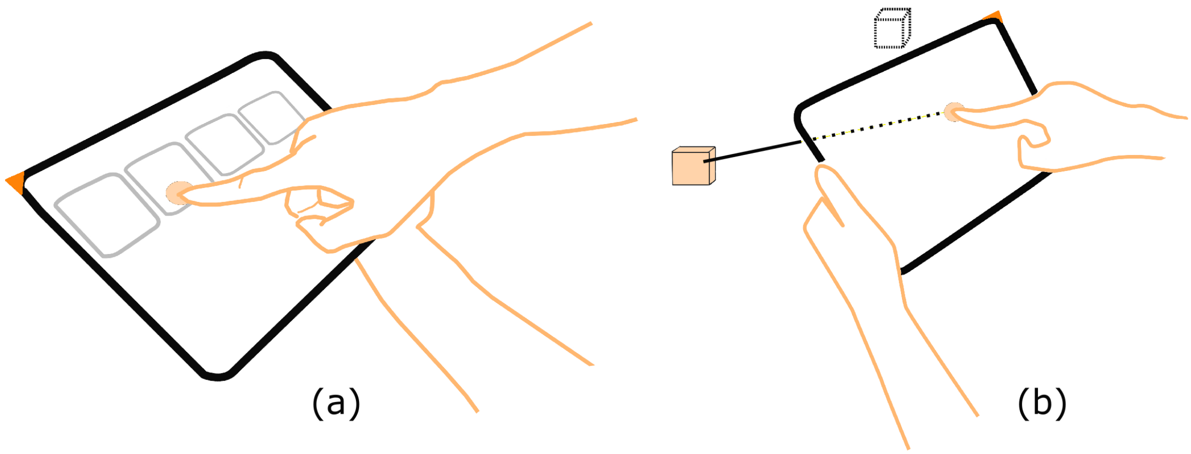


Figure 5.1: Creation. (a) Flip the tablet, select the object for creation, (b) Tap on tablet viewport to create.

pinching the thumb and index finger of their dominant hand while holding the tablet in their non-dominant hand (D1, D2).

Multiple objects can be created by repeating either a remote tap or mid-air pinch, and a different object can be created by flipping the menu and selecting a different object. To exit creation mode, a "swipe-in" movement is performed using the dominant hand just over the surface of the tablet. This follows the affordance of brushing off the icon of the creation object (O1, D9). We use the same gesture for deleting a selected object, explained later.

Our application supports primitive-shape creation (cube, cylinder, sphere, capsule) and Minecraft-style [181] blocks.

Selection (and Deselection)

In VR, selection methods differ based on how far away the object of interest is (O2, O4). So, we employ three different selection methods that take advantage of the tablet's form in conjunction with hand-tracking (Figure 5.2).

First, we use the tablet viewport for selecting a distant object, usually beyond arm's reach and within sight (O2). The user points the tablet's ray at a distant object and taps on the tablet screen to select it (O3, D10). Second, a corner pointer with a bright

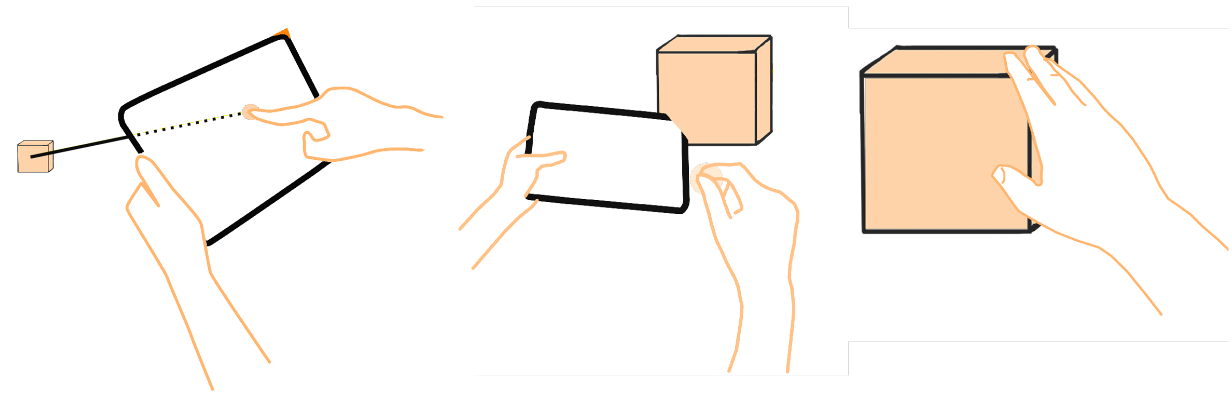


Figure 5.2: Object selection. (a) Tap on tablet viewport, (b) Pierce tablet corner in the object and pinch, (c) Tap on the face of the object.

yellow highlight at the top right corner of the tablet can be used to select an object by first piercing through the object with the corner, and then using a dominant-hand 'pinch' gesture (O2, O4, D6, D8, D11). Third, a user can also select an object by tapping with their dominant hand on the face of the object (O2, D6, D10).

In order to select multiple objects, a knuckle hand posture (see Figure 5.3, described in TapSense [95] and in the mode-switching study by Surale et al. [254]) is used along with one of the selection methods. Selected objects are highlighted using custom shaders (orange color). When the tablet is piercing two adjacently placed objects, only the object enclosing the corner pointer will be highlighted yellow (O6). Highlighting is used to indicate the hover state (O2, O4, D10, D11). This makes the corner pointer a precise object selection method, especially in case of a cluttered scene. Objects can be individually deselected by selecting them again using any of the techniques, and all objects are deselected when selecting "nowhere" with the corner pointer or tablet viewport, or selecting another object (Figure 5.3 (b-c)).

Deletion

Deletion follows selection. To delete an existing object, select it and perform a "swipe-in" movement using the dominant hand just over the surface of the tablet (see Figure 5.4 (a)).

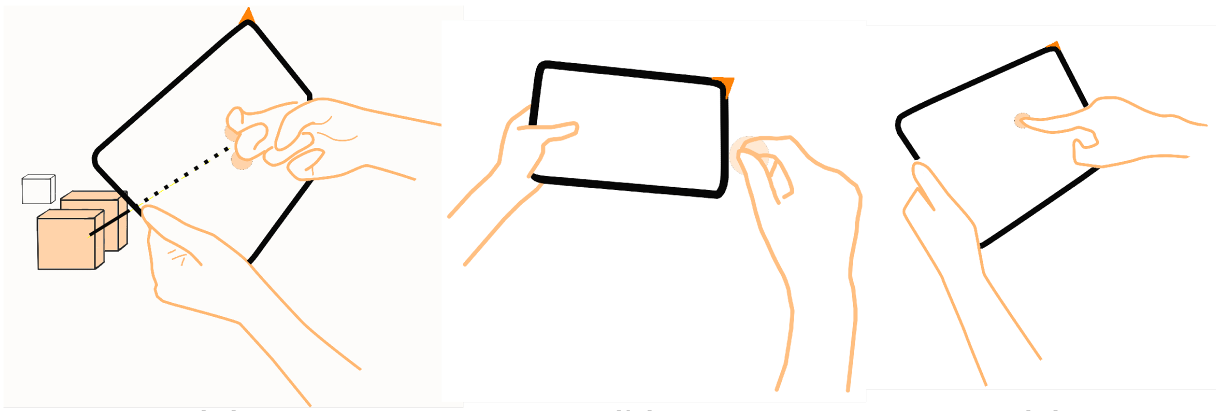


Figure 5.3: (a) Knuckle for multiple object selection, (b) Pinch to deselect, (c) Tap on viewport to deselect.

Rotate, Scale, and Translate

Rotate, scale, and translate transformations (Figure 5.4(b-c)) follow selection, and can be performed simultaneously. Once an object is selected, orienting the tablet fixes an axis and plane of transformation (Figure 5.5) (O1, O3, O5, O6, D9). The tablet orientation has to be maintained until the end of the transformation. Two-finger touch on the tablet starts transformation, releasing the contact disengages. Users can perform transformations with 9DOF.

Modify

The shape of an object can be modified by either slicing the object in an arbitrary plane or selecting a face for extrusion. Both operations follow selection. A user can slice an object by placing the tablet through the object with their non-dominant hand to determine a slicing plane and then using a dominant-hand pinch gesture to trigger the slice (D2, D5, D11). The half of the object above the tablet's screen will be removed, and the remaining portion of the object will be kept (Figure 5.6 (a)). To extrude, choose a face of an object by orienting the tablet in one of the three orientations (similar to Figure 5.5). Once the desired face is selected, two finger horizontal drag on the tablet extrudes the face (inward or outward). Our application supports extrusion with cube(s).

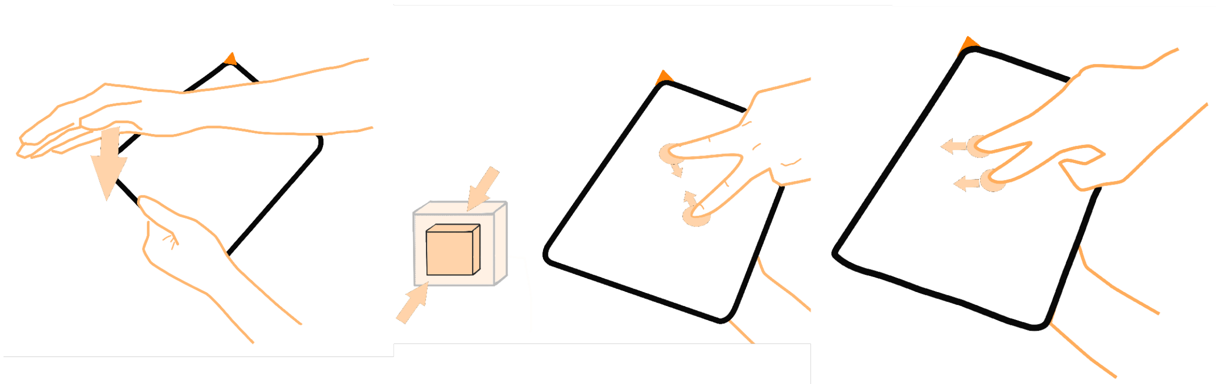


Figure 5.4: (a) Swipe-in to delete the object, (b) Two finger scale, (c) Two finger drag to translate.

Text Annotation

Having a physical tablet has the major benefit of providing a means for text input (O5). To create an annotation, we leverage the creation techniques and add ‘text’ to the scrolling list of available objects. Thus a user can place it in the scene remotely or in mid-air, similar to the Virtual Notepad system [211]. Annotations are interactive objects and can be selected or repositioned. When selected, a keyboard will appear on the tablet for typing and ‘Enter’ is used to commit the changes (O3, O5). While editing, a textbox will appear just above the keyboard showing the current annotation text. This helps maintain the focus on typing without needing to look at the annotation object directly.

Navigation

Simulator sickness or motion sickness is a well-known issue in virtual reality, and navigation without physical movement can exacerbate the problem. One effective way to mitigate this issue is to limit the user view. For instance, Fernandes et al. [70] used varying sized vignettes to limit the visual input to the user resulting in reduced motion sickness. An extreme version of this is recommended by Oculus [197], by fading the scene entirely to black.

We employ a similar approach. Five-finger touch on the tablet initiates navigation (see Figure 5.6 (b)), and while navigating, the scene quickly fades to black, except for the tablet and viewport. As a result, it is possible to see through the tablet screen to view the scene.

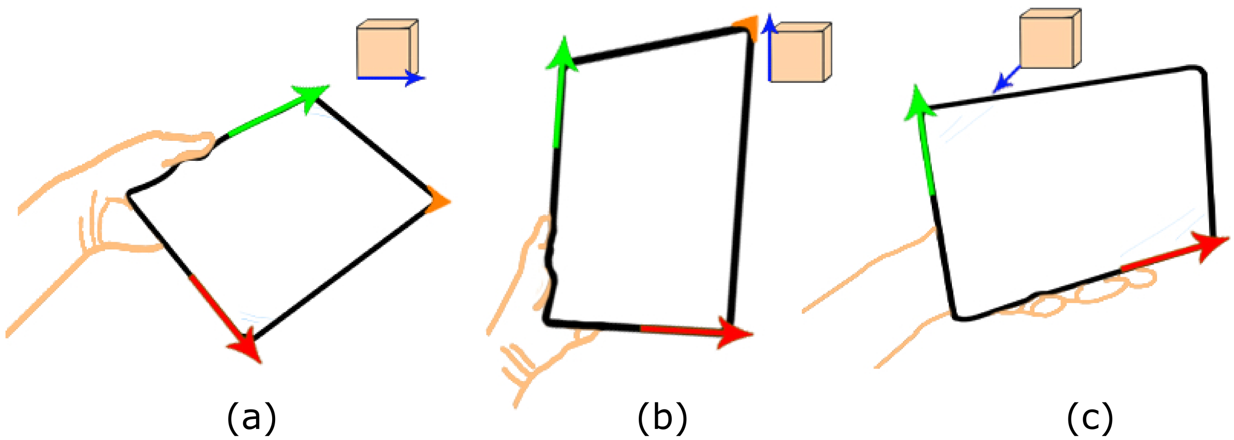


Figure 5.5: Select axis of transformation using the orientation of the tablet. (a) Facing up to select the x-axis, (c) Portrait vertical to select the y-axis, and (c) Landscape left to select the z-axis.

The moment navigation stops, by lifting one's fingers off the tablet, the scene is brought back to full visibility with a 3-second fade (O3, D4).

Zoom-in/out, rotate, and drag gestures are used for navigation. To rotate the view, touch the tablet with five fingers and rotate the wrist either to the left or to the right. Fiver-finger drag will intiate a move along the tablet plane (orientation), which can be adjusted with the non-dominant hand (O5, O6, D6, D7). Like transformation, navigation uses two points, the mid-point of the five-finger touch and the first point of contact to enable navigation. Note that five-finger drag moves the person in the opposite direction of the drag, which has the effect of the view rendered on the tablet moving in the same direction as the fingers. The five-finger zoom in/out gesture navigates from the initial position to the forward/backward direction pointed by the tablet, respectively (D5). In our application, rotation rotates the user view around the up axis pointing toward the sky. Note tablet orientation makes no difference for scene rotation.

Seeking Help

To request help, the user can old the tablet with two hands (D6) up to their chin (see Figure 5.6 (c)) and query into the mic (O4). Voice recognition on the tablet responds to the query. Here, a metaphor of a person thinking while holding a writing pad against the chin is used. A quick help video is played a meter in front when the distance between the

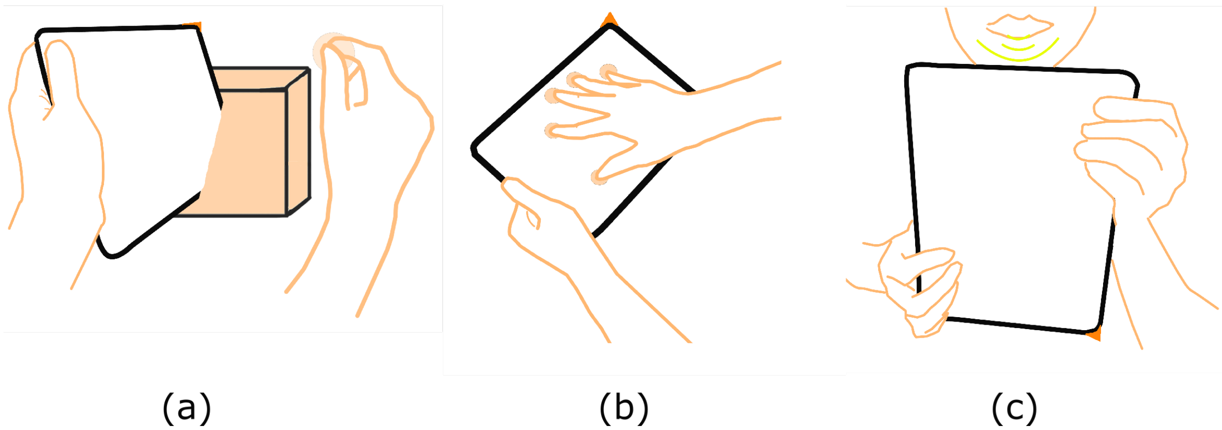


Figure 5.6: (a) Slicing an object, (b) Five-finger touch to navigate, (c) Speak to tablet and ask for help.

HMD (D4) and the tablet is within range ($\approx$ 10cm). The video stays in the view as long as the gesture is maintained.

System Menu

Butterworth et al. [38] demonstrated the use of system menu in early work on VR 3D modelling for operations such as undo, redo, cut, copy, and paste. In our system, hold the tablet in the dominant hand to access the system menu (D2). Multiple options are available; For instance, share, clear, exit.

5.5 User Evaluation

Our evaluation protocol and the goals of the user evaluation are similar to Arora et al.'s investigation [7]. We focus on overall usability of the system to replicate a predefined target model, understand user workflow, and analyze user feedback to spot shortcomings. We also ask participants to use our system to create a model purely out of their imagination. At the end of the study, participants filled out a post-experiment questionnaire indicating their experience with individual features of the system.

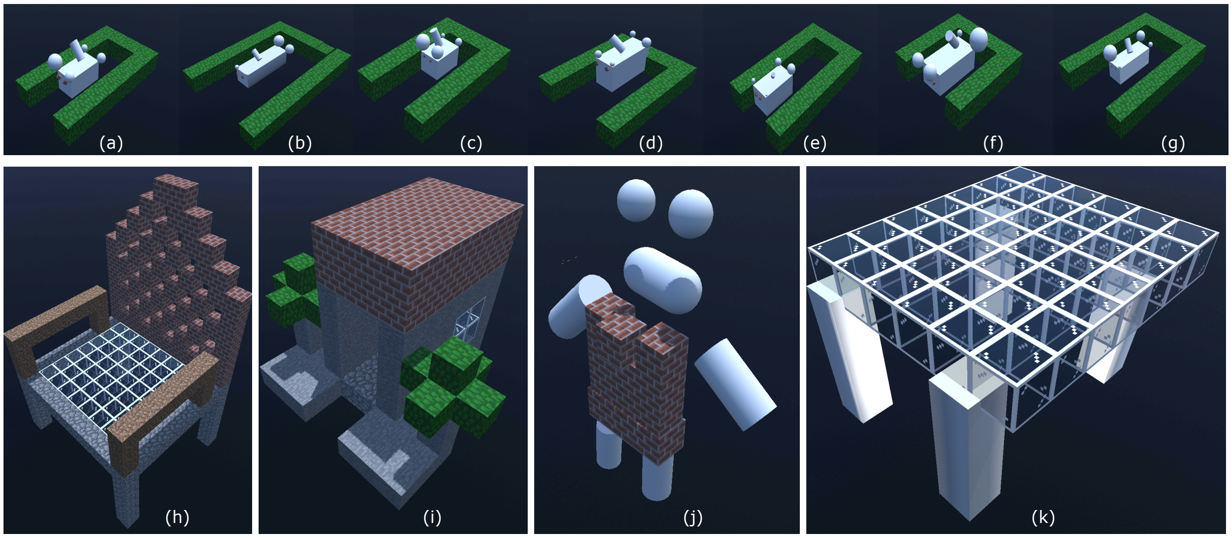


Figure 5.7: Sample results from 'replication' and 'freeform exploration' task. (a) Target Model, (b-g) Participant's replication ($P1$-$P6$), (h-k) Participant's creations in 'freeform exploration' ($P1$-$P4$)

Participants

Six people (all male, ages 19-34) participated in our study. They were experienced with Fusion360 ($P1$, $P2$), SolidWorks ($P2$), and other 3D modelling ($P3$-6) tools. All received $15 for successful completion of the study. Each participant filled out a questionnaire before the experiment began and after it was completed. A copy of these questionnaires are included in Appendix C.

Procedure

The study had three parts which took approximately 90 minutes in total to complete:

Part 1: Training (20-30 minutes). Participants were introduced to the system and how to use it. Then, the experimenter demonstrated and simultaneously explained five main features of the system: create and delete, select/deselect, transform, navigate, and modify. Participants practiced using the main features until they felt confident.

Part 2: Replication Task (20-30 minutes). The purpose was to exercise all the primary features. Essentially, testing the overall usability of the system by making the participants replicate the target model in half an hour. The target model is a predefined spatial arrangement of specific 3D objects placed on the floor as shown in Figure 5.7 (a). Replication includes completing a set of tasks in any order: 1) Create a cube on the floor using the grid; 2) Extrude it; 3) Create four spheres near the top corners of the cube in decreasing order of scale; 4) Create and place a cylinder on the centre of the top face of the cube; 5) Rotate the cylinder by -45°; 6) Create two 'brick' blocks, scale them down by more than 50% and place it on the front face of the cube; 7) Rotate these blocks by 45°; 8) Create a fence around the cube using 'grass' blocks. Before starting to replicate, participants were asked to familiarize themselves with the target model. The target model was always visible at their front-left side (tilted 45° facing the participant) as a reference. Note they were not required to match the exact dimensions of the target model. Instead, we used visual inspection to validate the match between the reference model and the model produced by the participant.

Part 3: Freeform Exploration (20-30 minutes). Participants were told to explore the system on their own to make their own creation, and were allowed to search the internet for inspiration. Participants were told to use their preferred features to create the 3D model they imagined. After completing the session, they filled out a post-experiment questionnaire.

Participants were encouraged to take breaks between each part of the study and to notify the experimenter if they were feeling nauseated. However, none of the participants reported feeling discomfort. Except $P2$, all the participants preferred sitting throughout the study.

5.5.1 Results and Qualitative Feedback

All participants successfully replicated the target model within the specified time limit (Figure 5.7 (b-g)). During the freeform exploration, participants created a chair ($P6$), an android ($P4$), a tree ($P1$), houses ($P1$, $P5$), and a glass table ($P3$) (Figure 5.7 (h-k)).

Overall, the system was perceived to be useful and interesting: $P5$ noted, "Most of the features, gestures were very intuitive and easy to follow", $P6$ noted, "[...] was an amazing experience, it really felt like we are interacting with the real world objects." Additionally, we analyze user feedback to understand the strengths and the limits of our system.

Create, delete, and modify. Observations and comments from participants indicate that creation, deletion, and modify were intuitive interactions. Except $P2$ ("Sometimes it created objects despite not being intended"), most of the participants could hold the tablet without accidentally touching the tablet screen; however, $P2$ had difficulty holding the tablet in a way that it would not cause unintended taps. To mitigate such problems, we could ignore the touch points near the grip [110] or a provide a longer handle on the left side of the tablet. Also, using design dimension D1, we can assign the trigger to a mid-air pinch [102] or fist gesture [244], and using dimension D2, non-dominant hand touch events can directly be discarded when near the tablet.

Select and deselect. The majority of participants found selection and deselection easy to understand and did master it quickly. Also, a few participants preferred raycast over corner selection. $P1$ felt corner selection was "weird". All participants found orienting the tablet to select the face of the cube for extrusion to be useful and reported positively. However, multi-object selection using the knuckle hand posture received mixed reviews. Except $P2$ and $P5$, participants reported it to be hard to use. They felt rotating the dominant-hand wrist to be tiresome. Using dimension D6, a dominant hand touch can trigger selection, while a non-dominant hand touch on the tablet can be used to switch between single-object or multi-object modes. Using the non-dominant hand for mode selection while holding the tablet has been effective in prior work [285, 75].

Transform (Scale, Rotate, and Translate). Recall that all three transformations can be performed simultaneously. This approach is similar to many pre-existing tablet applications like maps and image viewers, but for controlling object transformations, participants expressed mixed reviews. Participants found it easy to transform (rotate, scale, and translate) an object, but maintaining the distance between the fingers while rotating proved challenging, as finger distance corresponded to the scale of the object, indicating that explicit modes may be useful [297]. Also, instead of relying on two-finger touch, we can use D1 and D6 by touching the tablet with the index finger on the dominant hand to fix the axis of rotation, and rotating the tablet with the non-dominant hand to rotate the selected object as if the user is turning an object with a wrench. Overall, participants could understand and transform objects easily ($P1$-5).

Navigate. Navigation was perceived to be a hard task for numerous reasons, except $P6$, "Navigating with 5 fingers is easy and it doesn't conflict with other tasks." $P1$ felt that the movement directions were backwards (i.e., that it should have been world-centric, rather than tablet-centric movement, despite the world fading to black), and found it hard to navigate. Moreover, similarly to the transform operation, we let participants rotate, move, and zoom-in/out simultaneously. However, as noted in prior studies, separating DOF could improve the control during navigation [177, 297]. Or using D4 and D9, instead of continuous navigation, a user can select a fixed point on the map shown on the tablet and a tap would instantly teleport the user.

Menu Navigation. $P1$ felt uncomfortable interacting with the back of the tablet due to our custom tracker mounting bar and hook-and-loop fasteners. On the other hand, P6 reported, "[...], sometimes it is hard to hold the tablet in left hand." Except P4 and P6, participants found menu selection to be difficult. The primary reason for discomfort during the menu selection task was from flipping the tablet and interacting with the the back. Arora et al. [7] speculated about a similar issue in their work. We believe it was cumbersome to hold the tablet with the non-dominant hand, and tracking to interact with the menu was far less reliable than multi-touch on its front. To tackle unreliable tablet tracking and to avoid flipping the tablet, we can use D4 and D6 to select menu items on the front side of the tablet, which allows more precise 2D input and does not rely on 3D position tracking of the tablet with the dominant hand.

5.5.2 Discussion

Overall, the results show participants could use the example TabletInVR interaction vocabulary, as implemented in the proof-of-concept system, to accomplish core 3D modelling operations. This further suggests the associated TabletInVR design space and design dimensions were useful for exploring these new types of interactions.

While our system demonstrated integration of both mid-air hand gestures and tablet input to facilitate 3D solid modelling in VR, user evaluation pointed out some limitations. They can be circumvented using alternative combinations of the design dimensions. For instance, to tackle unreliable tablet 3D position tracking of the tablet, we can use D10 (indirect input) on the tablet screen, which is precise for 2D input (D1) and does not rely on 3D position tracking of the tablet. To tackle issues pertaining to using the dominant hand, we can rearrange roles using D2. To tackle motion sickness, D4 can be used to direct user attention to the device, rather than the surrounding VR environment, and so forth.

As a result, design dimensions would help tackle engineering issues until the technology matures. Moreover, we have presented only a small subset of possible interactions; we believe design dimensions could allow interaction design beyond 3D modelling. While a comparison of more technically mature and robust TabletinVR systems with the controller is warranted, our work demonstrates the feasibility of using the design dimensions to build a usable interaction vocabulary.

5.6 Conclusion

We are the first to investigate the design of an example interaction vocabulary for using a multi-touch tablet in VR for 3D solid modelling. We approach the design methodically and propose design dimensions that inform the design of our vocabulary, but can also inform the design of alternate vocabularies. We validate this interaction vocabulary with a proof of concept system that addresses the core components of 3D modelling and a user study that shows that the interface is useful in replicating and creating original designs.

Our study also identified some limitations which we discuss with possible solutions, but it also hints at future possibilities in this largely unexplored design space. While our focus was on 3D solid modelling, the design dimensions can also inform vocabularies for other applications like gaming, data visualization, and simulation control. Our work can guide future researchers and designers by extending the VR interaction space beyond traditional input devices.

Chapter 6

Mode Theory

"Don't mode me in." – Larry Tesler. [263]

Tesler, the inventor of cut-copy-paste [264], censured the presence of modes in user interfaces. Despite such strong reaction from a prominent figure, it is surprising that the *mode* concept has not been standardized in the HCI community. Regarding Tesler's view, it is hard to avoid something that is not well defined in the first place. In fact, definition attempts have gone through many iterations since 1989 without reaching a consensus [122, 214, 166, 194], and the existence of modes is still disputed by some HCI researchers [122]. Much has been written and discussed around this concept in the literature, however, there is no consensus in the way modes are defined, or the effects of modes on user performance or cognitive load.

The previous chapters focus on mode-switching in depth, but this is only one of many characteristics related to the larger mode concept. While conducting our mode switching work, we increasingly became aware that other mode characteristics were not well defined, and there was a need to examine mode switching within the larger concept of mode and other mode characteristics. So, in this chapter, we investigate the definition of "mode" more deeply including its utility and various characteristics in the context of designing user interfaces. After a thorough literature review, we selected three distinct mode-switching interface types that are commonly used. These are abstract representations of frequently used desktop applications for text editing (e.g. MS Word), web browsing (e.g. Chrome), and code editing (e.g. Visual Studio). We refer to them as *single canvas*, *overlapping canvas*, and *multiple canvas* respectively. We implemented a simple line crossing task that can require switching between different ink colours using the assigned mode-switching interface. We test our methods with 12 participants using an online crowdsourcing platform.

Our preliminary results show that a multiple canvas interface incurred the least amount of mode-errors, while requiring the longest response times. Using an overlapping canvas incurred the highest number of errors. The single canvas interface showed moderate performance in terms of response time and mode-error rate. We conclude with initial results and argue why this topic and the methodology creates promising directions for future research.

6.1 Motivation

Despite a prolonged discussion of the mode concept in the HCI community [122, 214, 166, 155, 154, 193], there are scarce resources to evaluate how exactly it informs user interface design. The utility of the mode concept maybe unknown due to its implicit nature. By utility, we refer to the mode characteristics that potentially have an impact on the user interface design and user performance. We hypothesize that like mode-switching, modes have other characteristics that can impact user performance differently across interfaces. However, the lack of understanding regarding mode characteristics hinders interface design. As a result, the idea of a 'mode' is left with the user interface designers and researchers to interpret in their own way.

In the early 1980s, researchers focused on building modeless text editors [263, 130, 11]. However, modern user interfaces are increasingly complex to build and it is rather tedious to apply those ideas to today's application design. Also, the consequences of mode errors could be far more severe than selecting a wrong colour in an interface or sending 'reply-all' emails [196]. As explained earlier in the introduction to this thesis, mode errors have resulted in pilots accidentally shutting off a commercial jetliner engine and killing 47 people [189, 28], and a data centre operator putting Amazon S3 servers to sleep which were running Netflix, Scribd, and Trello websites [219, 4]. Further, a NASA scientist, Miller et al. [180] investigated the implications of mode-errors in aircraft accidents when a pilot is interacting with automated controls. So, mode related issues must be investigated with great care.

6.2 Background and Related Work

In this section, we summarize past work pertaining to mode definitions, characteristics of modes, and potential confusion that might occur when the same term is used in different contexts.

6.2.1 Mode Definition

Norman laid out the theoretical foundation of human errors [195], classifying modes as a part of "slips during the formation of an intention". He described modes in terms of their manifestation as human errors, an "Erroneous classification of the situation". Norman suggests that mode errors could also be intertwined with description errors, hinting at the intricate nature of the concept. He continued to describe modes as a device state.

Brewster et al. [37] provide a similar mode definition, "A mode is a state within a system in which a certain interpretation is placed on information". For instance, typing characters can form a word in a text editor or execute a command depending on the system state. However, a device state does not necessarily establish the application context. Mackenzie defines modes as "a functioning arrangement or condition" [166] and provides an example of how a small region of the screen, while using a text editing tool, can put a system into various modes. This example relates to modes that are separated in the spatial domain. Each button on the screen occupies a separate position and has a region defined by the shape and size of the button. Interacting with these buttons would change the system state, in turn the system's mode. He defines modes by their ability to put a system into a different state.

Raskin provided an alternative definition of modes, "a distinct setting within a computer program or any physical machine interface, in which the same user input will produce perceived results different to those that it would in other settings" [214]. This definition is similar to the one provided by Poller and Garter [207], "a system has modes if the effect of a given user action is not always the same." Raskin, however, suggests a caveat, stating that an interface is not modal as long as the user is fully aware of its current state. Similar to Norman, he describes another concept intertwined with modes, *locus of attention*: a user's awareness of the system state. Such awareness is limited in practice. In VR, one can change the scene contents without being noticed by the user [170]. It is possible to do so, because if the user's locus of attention changes to a different area in VR, the state of the interface may then represent a mode since the user is no longer aware of it. This adds more complexity to the term 'mode'. It also means that simply presenting feedback and indicating system state is not enough to eliminate mode errors. Beyond striving for a firm definition, we argue that the utility of the concept should be studied.

Johnson et al. [123] have discussed the existence of modes in non-computing systems, for instance, in ovens, blenders, and toasters. In a toaster setting, the control to toast bread either LIGHT or DARK represent modes, and a mode-error could manifest as burned toast or raw bread. Leveson et al. [152] defined modes in general settings as "a mutually exclusive set of system behaviours." They suggested using state machine models to study modes as a transition from one state to the other. State machine models group all the possible system states. For example, ON and OFF are all the possible system states for a switch. Degani [56, 58, 59] cites a Webster dictionary to define modes: "(1) manner of acting or doing; method; way. (2) the natural disposition or the manner of existence or action of anything: form". They focus on flight controls used in aviation industry.

Now, is a mode a specific system state? or the manner of doing a task with a computing system? or the user's awareness of the system state? Whether the mode concept relates purely to the system's internal state, user's mental state, or in-between the system and the user, is unknown. Collectively these definitions provided by all researchers paint a rough picture of the mode concept, however, none of these definitions are the standard.

6.2.2 Mode Types

Researchers have put forth many different classifications of modes. Identifying mode types helps understand the breadth of the mode concept. Sellen et al. [236] recommended using kinesthetic modes to reduce the cognitive load and the mode errors. Kinesthetic modes [296] are also variously referred to as 'user-maintained', 'quasi-mode' [214], and 'spring-loaded' [107, 109]. While these types of modes are inherently tied to the time dimension, we can tie modes to space dimension as well, called 'spatial' modes. Browser tabs are a good example of spatial modes, clicking on them might reveal different result depending on which tabs are closed and which are opening. Note how the location of a click might not change across tabs, but the results certainly can change. Another good example of spatial modes are 'modal dialogues' [192]. These block access to other windows, often an unpleasant experience to the user. Modal dialogues appear when the user wants to save their work. For instance, while working on a photoshop file if the user wants to save the file, the 'save file' pop-up window would block the access to the photoshop application window.

Leveson et al. [152] defined three types of modes using control theory. Supervisory modes, component operating modes, and controlled system operating modes. Supervisory modes determine who or what is controlling the component at any time. Component operating modes control the behaviour of the control component itself. Controlled system operating modes specify sets of related behaviours of the controlled system and are used to indicate its operational state.

Mode classification does not end yet, Gow et al. [84] discussed two types of modes: action modes and indicator modes. Action modes are sets of states in which particular combinations of user actions, or other events, have a consistent effect. The user believes these actions in the absence of a clear visual feedback. Indicator modes are sets of states in which the interface sends or displays consistent feedback to the user. So, a user can actively observe the indicator mode. Interestingly, Gow et al. discuss the possibility of combining modes into "compound modes", a union of two similar types of modes. For instance, combining two action modes like using longpress and hardpress together [81] would result in a compound mode.

Some of these mode classifications might seem redundant. For instance, Leveson's controlled system operating modes resemble to Gow's indicator modes. However, most of the classification is still ambiguous. For instance, Sellen et al.'s [236] kinesthetic modes are not included in the mode classification suggested by Gow and Leveson. We speculate that there exists many such inconsistencies that limit understanding of the mode concept. We attribute such issues primarily to different mode definitions and return attention back to a lack of standardization.

6.2.3 Mode Errors

Mode errors are the only characteristic of the mode concept that the HCI community collectively has agreed upon. Researchers have studied mode errors and focused on reducing or avoiding them, yet surprisingly what constitutes a mode is unknown.

Past work has investigated various mode error prevention strategies. Among which, using kinesthetic feedback as a primary means of mode indication reduces mode errors significantly [236]. This is relevant to the 'locus of attention' idea that Raskin introduced [214]. Maintaining muscle tension is a better feedback of interface state than a visual indicator. The constant muscle tension is an explicit signal to the brain, which brings the action into user's locus of attention. Similarly, using an audio indicator can also reduce mode errors [184] and the associated cognitive load [37]. Thimbleby et al. [265] suggest reducing the number of modes in order to make the interface more predictable. They highlight the necessity of aligning the system image with the user's mental model to avoid errors. However, this raises important questions, like: Does a successful task completion mean the user has learned the right mental model? If so, the user should never make an error following a successful task completion? But the reality is certainly different. Moreover, it would be possible to trade speed for accuracy to ensure the successful task completion. Would that ensure the user's mental model is correct? We cannot be sure that the user has learned the right mental model based on the accuracy of the task completion and vice versa. Another relevant issue here is the way errors are measured and the flexibility of undoing mistakes following Shneiderman's principle for direct manipulation interface design, easy reversal of actions [239]. Mode-errors become less critical, if it is easy to undo an erroneous action. This also provides an opportunity for the user to correct their mental model. But not all mode-errors are reversible. For instance, pressing 'CMD + DEL' keys will permanently delete the file in the macOS file manager.

Levenson et al. [152] identify six main sources of mode-errors and added examples of each: interface interpretation errors, inconsistent behaviour, indirect mode changes, user's authority limits, unintended side effects, and lack of feedback. A common example of an input interface interpretation error occurs with many word processors, where the user may think they are in insert mode but instead are in command mode and their input is interpreted differently than they intended. We can relate these types of errors with the 'locus of attention'. A user's locus of attention is elsewhere, the result might be an error.

Sarter and Woods [229] conducted an experimental simulation study of mode awareness and pilot automation coordination on the Airbus A320. They asked 18 experienced A320 pilots to fly a 90-minute scenario on a full-mission A320 simulator. The goal of the study was to distinguish between the errors of commission, where an operator takes an inappropriate action, and errors of omission, where an operator fails to take a required action. The study identified several different types of errors were later consolidated by Levenson et al. [152].

An inconsistent behaviour error occurred during an A320 accident involving a protection function. This function prevented autothrust system when the flight altitude was 100 feet above the ground, however, the pilots were unaware of this behaviour. They believed the system can be put in autothurst mode in all the conditions [229, 231]. An example of an accident that has been attributed to an indirect mode change occurred while an A320 was landing in Bangalore. In this case, the pilot wished to select a lower altitude, but the automation was in an altitude acquisition mode. This resulted in the activation of an open descent mode which led to the accident [229, 231]. An authority limiting error is a type of lockout or interlock that fails to prevent user actions that could cause the system to enter a hazardous state. An example occurred in the Sarter and Woods A320 simulator study [229, 231] where it was discovered that pilots were not aware that entering a runway change after entering data for the assigned approach results in the deletion of all previously entered altitude and speed constraints, even though they may still apply.

Incomplete feedback is often implicated in accident scenarios. For example, in the A320 Bangalore accident, the pilot-flying had disengaged his flight director during the approach and was assuming that the pilot-not-flying would do the same thing. The result would have been a mode configuration in which air speed is automatically controlled by the auto throttle (the speed mode), which is the recommended procedure for the approach phase. However, the pilot-not-flying never turned of their flight director, and the open descent mode became active when a lower altitude was selected. This indirect mode change (explained above) led to the hazardous state and eventually the accident [229].

These error types essentially describe the flaws in user interface design as well as the limits of human cognition, both of which lead to an error. Moreover, our work on investigating mode-switching in touch-based user interfaces [254] suggests out-of-target errors are caused by an unintended motor action. In practice, such motor actions can invoke inadvertent commands, such as mode errors, in a real application.

In summary, mode errors are associated with user cognition, user's motor control, and the user interface. In this chapter, we use mode errors to understand the differences among mode-switching interface layouts.

6.2.4 Mode Prediction

Mode prediction is a process of inferring interface modes. Such prediction can help a user rely on the system to make an automatic mode-switch while interacting with an interface. This effectively reduces the cognitive burden of the user. Mode prediction can help a user focus on the task at hand rather than thinking of modes. So, for building a modeless interface, one might need to build a system that can predict modes.

Tung et al. [269] presented a system, FlickBoard, that combined a touchpad and a keyboard into the same interaction area to reduce the switching between the keyboard and the touchpad when they are separate. Flickboard supports automatic mode switching between a typing and pointing task with 95% accuracy using Random Decision Forests (RDF).

Deming et al. [61] built a mode prediction system in a stylus-based interface. Their system uses stylus orientation and pressure to determine user intentions. For instance, if the user is selecting the objects or inking on the canvas. They used a rule-based approach to determine the appropriate mode and prompt the user in case the mode prediction fails.

Similarly, Saund and Lank [233] used properties of the stylus trajectory and the context of the trajectory to infer user intention. When the prediction is ambiguous, the user is offered a choice in the form of a pop-up button. The user can choose to ignore the pop-up and continue drawing. Predicting modes can reduce interface complexity, essentially reducing the cognitive burden of managing modes and mode-switching for the user.

6.2.5 Alternate Mode Concepts

Interface modes are often confused with the term 'states', 'task', and 'modality'. We clarify the differences while interpreting these terms.

Naively, one might draw parallels between a state machine from Automata Theory and an interface mode, but Thimbleby et al. [265] argue that they are fundamentally different. They argue that some internal system states can represent interface modes, but it is not the case for all the states.

Another common confusion is between the term 'task' and mode. A task can accommodate multiple modes as well as any combination of other sub-tasks that might not be necessarily refer to the mode concept. Often the task refers to general tasks like 'replying to an email', 'working on a presentation', 'create or edit web pages' as described in Czerwinski et al.' work [54].

Furthermore, in HCI literature, researchers use the term 'multi-modal interaction'. Here, multi-modal means multiple ways to provide an input to a system [198]. For instance, using touch, voice, gestures, feet, and so forth. Although these can be defined as 'modes', these are better understood as the mediums to communicate with a computer, but they do not refer to the interface modes. Modes can exist within one of these mediums, like using two finger input or using knuckles within a touch input mode as medium.

6.2.6 Summary

Evidently, the importance of studying modes is well understood, but there has been confusion and disagreement around their definition and the utility. To collectively make progress as an HCI community, we should strive to solidify the fundamental understanding of the terms used in our literature. Merely having a general sense of the mode concept would not help investigate interfaces thoroughly, hindering future explorations and potentially misinterpreting findings. For instance, the reasoning behind high mode-errors when an application switch occurs, compared to mode-errors within an application is not clearly known [148]. Hornbaek et al. [115] pointed out three benefits of solidifying HCI concepts. Firstly, moving from general interpretation of concepts to sharper, scientific concepts would facilitate better measurements, reasoning, and predictions. Secondly, it acts as a thinking tool. Lastly, it justifies the common baseline to reflect across the field. We share these views. So, we set out to investigate how to conduct an empirical investigation to better understand the mode concept.

6.3 Experiment

The goal of this preliminary pilot experiment is to test the task design and experimental methodology to prepare for a future full study. The ultimate goal is to test the potential of investigating interface mode concepts empirically. We compare the mean response time, mode-errors, out-of-target errors, and line crossing errors for three mode-switching interface types. This study is could be conducted in a controlled environment such as a laboratory, however, to validate the suitability of the experimental design for a much larger population, we conduct this pilot on an online platform (Amazon Mechanical Turk).

6.3.1 Participants

We recruited 12 participants from the United States using Amazon Mechanical Turk. They were informed that the task would take approximately 20 minutes and they would receive US$3.50. We required participants to have completed at least 1000 approved HITs (tasks on Mechanical Turk) and have a 95% or more approval rate. We granted qualifications to workers to ensure that they could only complete the experiment once.

6.3.2 Apparatus

Our experiment application was written in p5.js[1] and it was hosted on a local server in the university. Experiment logs were stored on the server. The application was running at 60 frames per second. We follow a common filtering practice to ask specific questions regarding the devices used by the participants at the beginning of the study. We restricted worker devices to a desktop or a laptop with a mouse to participate. The use of a trackpad or any other pointing device except mouse was prohibited. This is necessary to increase consistency across participants.

6.3.3 Task

Our experiment task is to cross two gray coloured horizontal lines in a vertical direction from top to down in single stroke. The distance between the horizontal lines was 40mm. During the experiment, the horizontal lines will flash for a 500ms duration to show a designated line drawing colour. Then, the user has to activate this colour (if not already

[1]https://p5js.org/

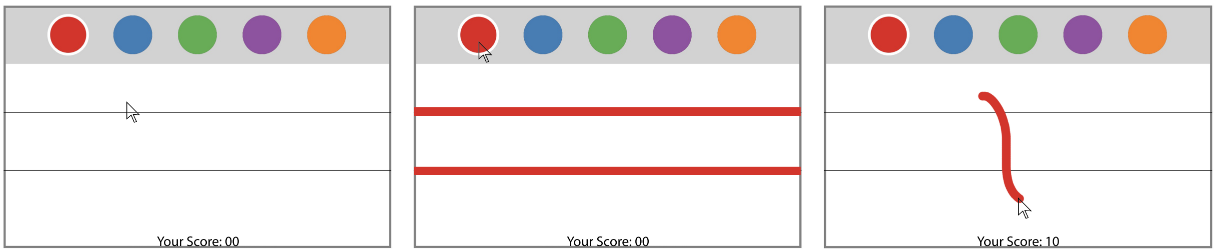


Figure 6.1: Steps involved in the line crossing task. (a) The participant waits for the horizontal lines to flash the designated colour, (b) the horizontal lines flashed the red colour, (c) the participant activates the red colour and crosses both the horizontal lines in a single stroke.

activated) using a mode selection interface, and then cross both horizontal lines in the downward motion (see Figure 6.1). We ask participants to complete the task as fast and as accurately as possible.

6.3.4 Mode Selection Interface Conditions

The position of the two gray coloured horizontal lines and the position of the buttons on top of the layouts is constant across all the interfaces. We make sure that the line crossing task across all the interface types requires similar motor movement. Miller [179] summarized past work on information capacity of humans given various stimuli, for instance, audio, haptic, and visual stimuli. Following their work, we pick a lower limit 5 as the minimum amount of capacity a user needs to process while making a decision in the moded interfaces. So, there are 5 coloured buttons in the moded interfaces described below. The order of these colours is random. All the interfaces show a score at the bottom of the interface, calculated as follows: correct line crossings increase the score by 10 and errors decrease the score by 10. Past line strokes are left on the canvas, however, as the experiment proceeds, it can clutter the canvas. Cleaning the canvas automatically might distract the participant from doing the task accurately. So, we let the participants decide when to clear the canvas by pressing the spacebar key. This does not disrupt experiment logging.

Single Mode Baseline Interface

In the BASELINE, the interface shows only one mode. We refer this interface type as un-moded interface. Tesler calls having one mode in an interface as a 'modeless' interface

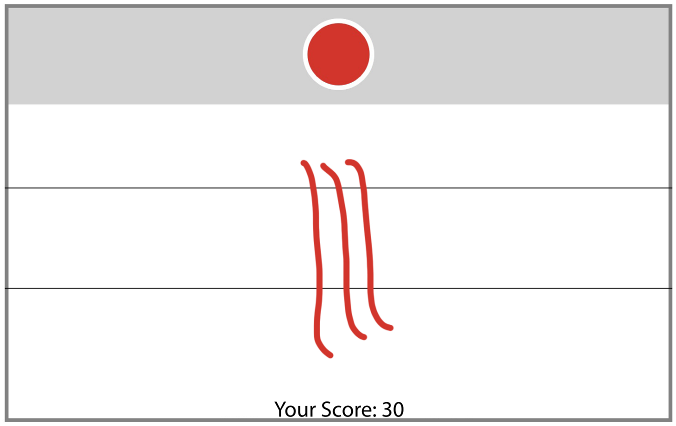


Figure 6.2: Single mode BASELINE canvas with only one button.

[263]. We use this interface as a baseline condition. The user has to press the red coloured button to draw red lines crossing the horizontal lines as shown in Figure 6.2.

Single Canvas Interface

The SINGLE canvas is an abstract representation of single artifact applications. All the commands available to the user essentially operate on this artifact. A good example of this kind of interface is the Microsoft Word application. A user is constantly operating on the word document. All the available commands are shown at the top part of the application in the form of a ribbon. A user can switch tool modes for manipulating the document from this ribbon. For instance, text style, insert images, make changes to the page style, and so forth. While there can be multiple pages in a single document, conceptually it is treated as a single artifact. Similar examples include Microsoft Powerpoint, Paint, Adobe Photoshop, and Adobe illustrtor.

The SINGLE canvas interface shows five buttons at the top. The user has to pick one of the colours during the experiment and cross the horizontal lines. The interface is shown in Figure 6.3.

Overlapping Canvas Interface

The OVERLAPPING canvas is an abstract representation of applications with multiple overlapping artifacts at the centre of the screen [27]. All the commands available to the user

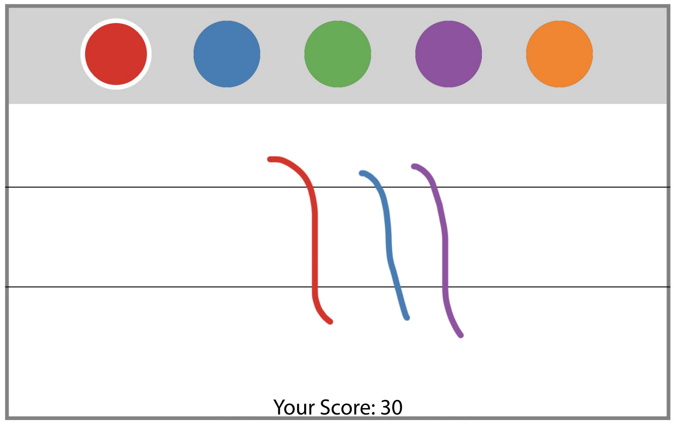


Figure 6.3: SINGLE canvas interface with 5 buttons.

operate on these documents. A good example of these kinds of interfaces is a tabbed web browser. A user is constantly operating on one of the several overlapping web pages. In order to switch, the user clicks on the tab. The current web page then replaced with the new web page associated with the tab. Conceptually, the foreground browser tab is overlapping all other tabs, the topmost tab being visible to the user and ready to accept the input.

The OVERLAPPING canvas interface shows five tabs at the top. The user has to pick one of the tabs during the experiment and cross the horizontal lines. Line colours correspond to the specific tab. For instance, if the blue coloured tab is selected, the user can draw only the blue coloured lines on the canvas. In order to switch the line colour, the user has to switch the tab. In this interface type the canvas shows the previous strokes only for that particular tab. The interface is shown in Figure 6.4.

Multiple Canvas Interface

The MULTIPLE canvas is an abstract representation of applications with non overlapping tiled artifacts spread across the display [27]. The user has to select the tile to operate on it, and the active command mode may be different for each tile. A good example is programming Interactive Development Environment (IDE) like Microsoft Visual Studio. A user can check properties tab on the right tile, while the left tile shows the solutions explorer, and at the centre the code file. The top ribbon shows the generic commands that can operate on any of the tile. By just moving the mouse over one of the tiles, a different set of operations are enabled. This refers to spacial modes and known to support

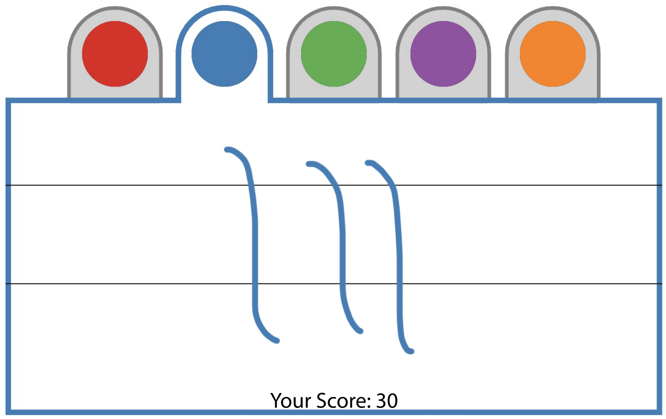


Figure 6.4: OVERLAPPING canvas interface with 5 tabs.

fast task-switching [129, 51]. More recently, knowledge workers [65] have relied on tiled window management tools like DIVVY [183] and i3 [249]. Unlike OVERLAPPING canvas, the user has access to all the artifacts, however, the space is divided among them.

The MULTIPLE canvas interface shows five tiles placed side by side. The user has to move the pointer over one of the tiles during the experiment and cross the horizontal lines. Line colours correspond to the specific tile. For instance, if the pointer is over the blue coloured tile, the user can draw only the blue coloured lines on the canvas. In order to switch the line colour, the user has to move the pointer over to another tile. In this interface type the canvas shows the previous strokes for all the tiles. The interface is shown in Figure 6.5.

There are obvious benefits of each interface, and the popularity of the associated application is an attestation of their use for a productivity-based workflow. However, we are primarily interested in capturing how a user's perception and motor movement becomes affected by these interface layouts.

6.3.5 Design and Procedure

The study is a within subject design. INTERFACE type is the primary factor, with levels corresponding the four interface types described above (BASELINE, SINGLE, OVERLAPPING, MULTIPLE).

For each condition, we show a short video introducing the interface type to the participant. Then, we train the participant for 3 to 4 minutes prior to the session. To complete

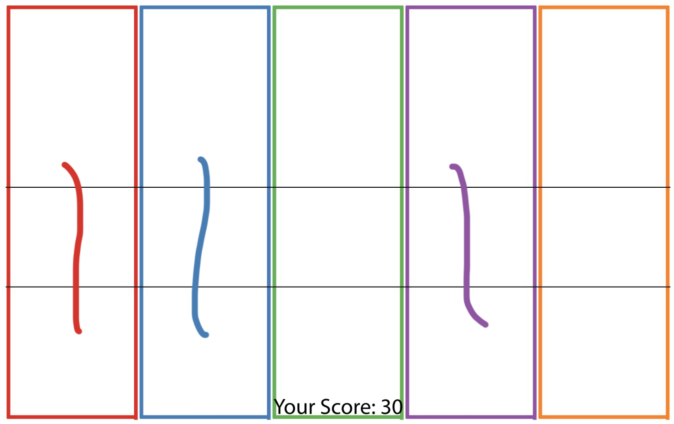


Figure 6.5: MULTIPLE canvas interface with 5 tiles.

training, the participant must complete 20 correct line crossing tasks (80% accuracy), out of 25. Otherwise, they must complete the training block again. Training blocks consist of an un-moded block (BASELINE interface) and a moded block (one of the INTERFACE types). Once the training block was completed, the participant is assigned to one of the experimental blocks.

To minimize order effects, INTERFACE was counter-balanced using a 3 × 3 Balanced Latin Square. Our block design starts and ends with an un-moded interface (BASELINE). In total, our design includes six moded interface blocks alternating between two moded and two un-moded blocks, like BBCCBBCCBBCCBB. A pair of 'C' block in this design corresponds to one of the moded interfaces. The un-moded blocks reduce a learning effect after completing one of a pair of a moded blocks. In each block the participant had to complete 25 line crossing tasks. The order of the colours is random. In case of an error, the participant had to redraw the lines using the correct colour. The order of the colours was randomized.

In sum there were: 1 single-moded BASELINE × 8 BLOCKS × 25 line crossings + 3 moded INTERFACE (SINGLE, OVERLAPPING, MULTIPLE) × 6 BLOCKS × 25 line crossings = 650 line drawings per participant.

6.3.6 Dependent Measures

Two measures were calculated based on the experimental logs.

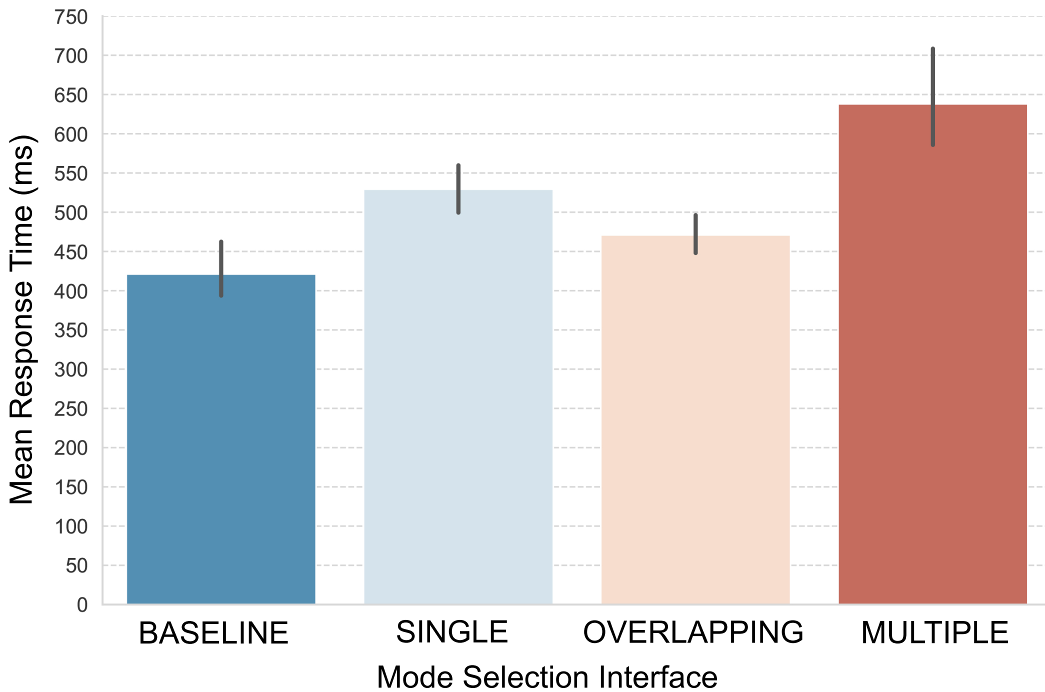


Figure 6.6: Mean Response Time by INTERFACE (error bars in all graphs are 95% CI).

Mean Response Time

Mean response time is the duration between the end of the line flash and the first click on the button (in SINGLE), or a tab (in OVERLAPPING) or a tile (in MULTIPLE). If the correct colour is already selected the response time will be the duration between the end of the line flash and the first click on the canvas. Only the correct trials were used to calculate the mean response time.

Errors and Error Rates

Like Li et al. [154], we identify three types of errors. A *mode error* occurs when the wrong colour was activated before crossing the horizontal lines. In this experiment, mode errors are the most important and other errors are not related to modes. However, we log other errors for the completeness. A *crossing error* occurs if the line stroke did not cross both horizontal lines in the correct order and direction. This captures errors related to crossing accuracy. An *out-of-target error* occurs if the user clicked on the area other than the buttons and the canvas. This most often captures a case when the participant clicked on the gray coloured background of the buttons on the top of the canvas.

Each of these error types are recorded per trial as 1 if the error occurred, and 0 otherwise. The mean value across trials produces an error rate.

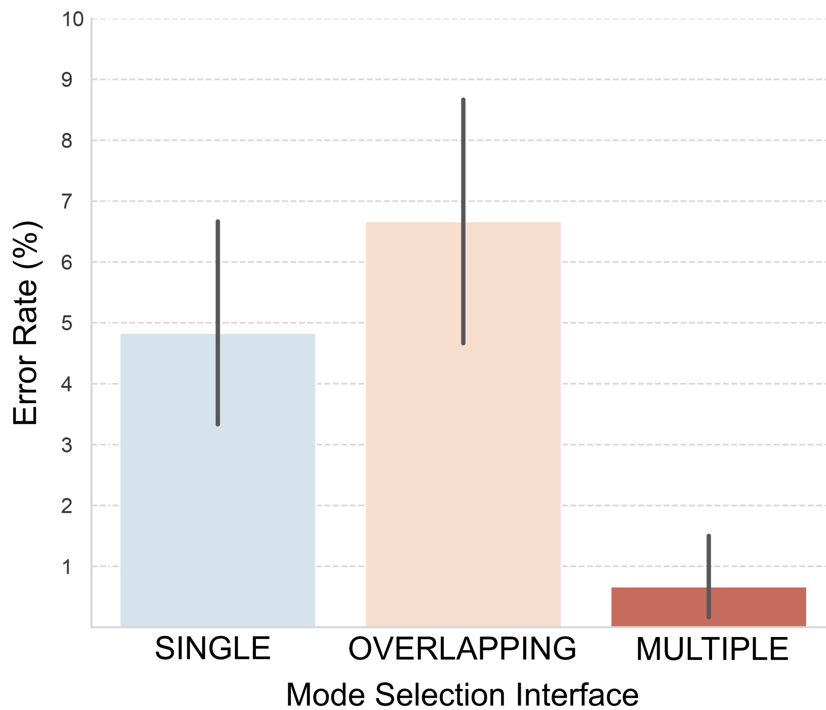


Figure 6.7: Mode Error Rates.

6.4 Results

We discuss the results based on the analysis conducted with the dependent measures.

Mean Response Time. There was a significant main effect of INTERFACE ($F_{3,3705}$ = 16.264, $p < .0001$, $\eta_G^2 = .13$). Post hoc tests found that SINGLE canvas interface was faster than MULTIPLE canvas interface ($p < .001$). Also, OVERLAPPING canvas interface was faster than MULTIPLE canvas interface ($p < .001$). BASELINE canvas interface was faster than SINGLE and MULTIPLE canvas interface ($p < .01$). The mean response time was highest for MULTIPLE canvas interface (638ms), followed by SINGLE (529ms) and OVERLAPPING canvas (471ms). The mean response time for the BASELINE canvas was the fastest (421ms). See Figure 6.6.

Mode Errors. In terms of the *mode errors*, the OVERLAPPING canvas incurred highest percentage of errors (6.6%). Followed by, SINGLE (4.8%), and MULTIPLE canvas interface (below 1%). However, we did not find any statistically significant effects across INTERFACE. See Figure 6.7.

Other Errors. crossing errors were low for all the interfaces (below 1.5%). Note in SINGLE and OVERLAPPING canvas the participants had to locate the button to switch the line colour. This space constraint is absent from the MULTIPLE canvas interface. We saw this spacial constraint to impact the *out-of-target* error. SINGLE (4.8%) and OVERLAPPING (7.1%) canvases showed higher percentage of the *out-of-taget* errors, where MULTIPLE canvas posses no such errors (0%).

6.5 Discussion

Admittedly, these results might change with a wider participant pool and a thorough investigation is required to solidify the findings. However, with the initial results, we attempt to highlight the interesting differences among commonly used mode switching interface layouts. Moreover, we also identify the important characteristics of the mode concept in the following section.

6.5.1 Mode Characteristics

Here, we summarize the identified mode characteristics. Some directly relate to the discussions found in the past literature, others less so. When possible we define each of these characteristics with the relevant work, provide an example, justify its importance while designing the interfaces, and discuss potential research directions in the context of our experiment methodology. We believe these mode characteristics can help application designers and move closer to a more unified definition of the mode concept and a single mode theory.

Mode Switching

The process of switching in and out of a mode refers to *mode switching*. We further identify two more sub-parts of mode-switching which are often overlooked: mode engagement and mode disengagement. The reason interface designers should be careful about mode engagement and disengagement strategies is because their performance can be asymmetric. For instance, for a pressure based input technique [213], mode engagement happens when a threshold value is crossed, and this requires more time compared to the mode disengagement time. From the actuator standpoint, pressure release during the mode disengagement happens faster than exerting a pressure during the mode engagement [213]. As a result, like Li et al. [153] and in our work [254, 255], we add granularity to the types of mode errors, classifying them into *mode-engagement* (mode-in) and *mode-disengagement* (mode-out) errors. An error is treated as a mode-engagement error if the user fails to make a switch to a designated mode, failure to make a switch back is considered a mode-disengagement error.

In our experiment, the mode-switching action is similar across different moded interfaces, but interestingly we see the differences in terms of the response times. For instance, the location of the colour buttons, the tabs, and the tiles is same across the interfaces. This effectively requires user to move mouse pointer in a similar way to switch modes in different interfaces. Our experimental task involves using a mouse to switch among 5 inking colours. Future studies should investigate varying mode-switching techniques to heighten the effect.

Mode Occupancy

The total amount of time spent in a particular mode is called *mode occupancy*. So far, most of the mode-switching analysis were conducted with a unit mode occupancy, meaning the user is constantly switching between modes after using a mode for only one action [154, 254, 255, 62]. Fennedy and Lee [69] investigated mode occupancy up to sixteen and reported overall reduction in mode errors when the mode occupancy is higher as well as balanced for all the modes. We speculate the results will vary given the extended use of a mode, when using a mode for multiple actions before switching to another mode.

Recently, Lee et al. [148] showed interface switching introduced higher mode-errors than using a single interface for a longer time. We believe, the mode concept relates to the abstraction a user cognitively derives based on the situation. In human psychology, "Gestalt laws of grouping" describes the way humans naturally perceive objects as organized patterns and objects [139]. In a similar vein, Buxton et al. [40] noted, "Experts and novices differ in the coarseness of granularity with which they view the constituent elements of a problem or task". They state that novice users are attentive to a more granular level of details, while experts perform the same granular tasks almost automatically. As a result, the way we perceive a smallest unit of task depends on the abstraction our mind creates while using an interface. Of course, repetition helps learning the largest chunks or compound modes (combining more than one modes) in an almost automatic fashion. However, the precise isolation of the process is not known, and should be investigated in the future.

In our experiment, mode occupancy is not random. In the best case it is 5 (consecutive colours) and 1 in the worst case (non repeating colour sequence). Past research showed higher mode-error rates when an application switch occurs [148], and our experiment has abstract representations of such applications. Within an interface, mode occupancy would potentially result in less mode-errors. An intriguing question is then: 'Do the buttons, tabs, or tiles within an interface equating to a mode, or is the whole interface a mode?'

Mode Frequency

The number of times a user switches into and out of a particular mode refers to *mode frequency*. This characteristic has implications in regard to fatigue levels observed during application use. With the frequent mode switching in our touch mode-switch study, we observed the knuckle technique received poor ratings in a sitting posture, despite having comparable error rates with the fastest technique. Certainly, the techniques involving heavy muscle use should keep mode frequency to lower values, or if unavoidable, another strategy is to design an alternate interaction that distributes the workload evenly on other muscles.

Mode frequency is inherently different from mode occupancy. Mode frequency refers to the process of switching into and from a mode; on the other hand, mode occupancy refers to the duration spent in a mode after switching.

In our experiment, the mode frequency varies from 1 (always switching) up to 4 (consecutive switches). In future, controlling the mode frequency explicitly in our experimental methodology could reveal correlations with fatigue.

Mode Pattern

A particular repetition of mode sequence refers to *mode pattern*. Note, mode-switching is different from mode pattern. While mode-switching asserts switching back to the original

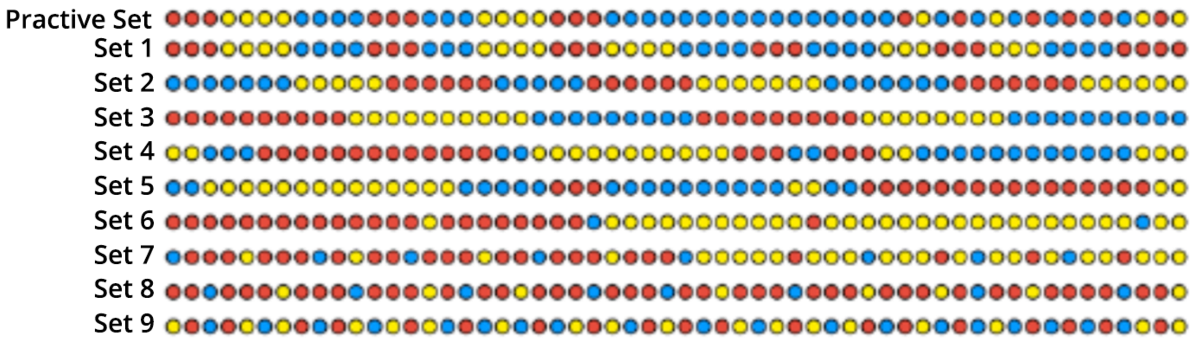


Figure 6.8: 10 sets of mode sequences, including the practice set. (from Fennedy and Lee [69])

mode, in mode pattern this requirement is relaxed. In practice, this would mean a sequence of inking, invoking commands, and gesturing in any random order without repetition. Fennedy and Lee [69] described the implications of various mode pattern usage on mode errors. They discussed 9 types of mode patterns (see Figure 6.8) with 3 modes. In set 1 to 5 the mode pattern is balanced between the three modes. We believe, a thorough investigation of such mode patterns would reveal interesting results.

Mode Chunking and Mode Phrasing

The process of separating a compound mode into two or more user actions refers to *mode chunking*. For instance, instead of using lasso selection tool to select multiple objects, directly tapping on each of the object. The process of combining two or more distinct modes in one user action is called *mode phrasing*. For instance, using a lasso selection instead of tapping on each object. Effectively phrasing multiple taps into one gesture [40]. Note the usage context might enforce the amount of mode chunking and phrasing. For example, if the objects of interests are interspersed with other objects, a user might prefer quickly tapping on objects of interest rather than drawing a lasso shape carefully [106]. Buxton et al. [40] identify mode phrasing as an expert user trait. Further, they add the process of phrasing through kinesthetic gesture. Examples include marking menus [141], spring-loaded menus [107], ToolGlass [23], and Scriboli [104]. However consider these questions: "is kinesthetic gesture the only mediator of the mode phrasing? is a pianist phrasing multiple key presses through a non-kinesthetic gesture?" We believe, further research can be conducted to answer these questions.

Mode Scaling

The total number of modes an interface can accommodate refers to the *mode scaling* of an interface. To our knowledge, Ruiz and Lank [224] conducted the first study investigating the mode scaling characteristic of an interface. The results show using a non-dominant hand can be used to increase the mode scaling of pen-tablet interface. However, mode scaling is not a straightforward task. Ruiz et al.'s work show that using concurrent mode-switching performs better as opposed to pre-mediated technique, but it is not cost-free, the task completion time with concurrent mode-switching is higher. Pre-mediated mode switching requires the action in the non-dominant hand to precede the action of the dominant hand. The concurrent technique allows mode manipulation in the non-dominant hand to overlap the action in the dominant hand. Further, the results show irrespective of the concurrent technique the time to initiate a mode increases as mode scaling increases. Mode scaling is an important characteristic of interface.

In our experiment, we test un-moded (0 modes) and moded (5 modes) interfaces. However, as described in mode chunking and mode phrasing section, we cannot be sure if one colour in the interface equates to a mode. Ruiz and Lank [224] suggest there could be a potential connection between the available mode choices and the initiation time, which could be modelled using the Hick-Hyman law. A full study could investigate this phenomenon. If the number of choices, in our case the number of colours in the interface, correlates to the initiation time, then irrespective of mode phrasing even experts would take a longer time to initiate the action. In other words, experts make the decision before starting the interaction involving two steps, versus the novice user who distributes this decision in between the steps. As a result, mode scaling depends on the result of mode phrasing and mode chunking, but this is yet to be empirically tested.

Mode chunking and phrasing, mode frequency, mode occupancy, and mode pattern characteristics described above are speculative, while others have been investigated in the past. Regardless, further research is required to investigate the impact of these speculative characteristics on application design.

6.5.2 Mode Definition

Based on the mode characteristics explored in the past and the ones speculative above, we make an attempt to define mode as follows: "a user interface mode is an elastic psychomotor ability to progressively acquire a skill."

Considering the mode characteristics we discussed, it can further be stated that this ability is controlled by a user at three different levels: at the cognitive level, at the psychomotor level, and at the motor level. One can observe the time consumed when a user is presented with equiprobable multiple choices, such phenomenon has already been investigated with the Hicks'Law. Problems at the cognitive level would result in higher task completion time or errors. At the psychomotor level, a user simultaneously makes a decision as well as executes an action. For instance, concurrent modes allow users to be faster as well support a greater number of modes [224]. Problems at the psycomotor level are the source of mistakes [194]. A mistake occurs when the user has an incorrect intent. Finally at the motor level, acquiring a skill requires training muscle memory. Problems at the motor level are the source of slips [194]. A slip occurs when the user has a correct intent, but performs a wrong action.

We hope that more iterations along this direction will make the definition more robust and help set a common reference when the term 'mode' is used in HCI literature.

6.6 Conclusion

We argue there is a lack of standardization of the mode concept. We conduct initial experimentation with abstract layouts that are commonly used. Our task design and methodology can facilitate investigation and understanding of the mode concept in a well-informed manner. Identifying mode characteristics adds value in broadening our understanding of one of the fundamental concepts in HCI. Our empirical investigation was the first step toward identifying different characteristics, standardizing the mode definition, and solidifying the experimental protocol to test the modes. However, we acknowledge that a series of more thorough investigations is needed. Specifically, investigating the role of each mode characteristic, their qualitative and quantitative impact on the usability of an interface, and converging the multifaceted concepts in the literature.

Chapter 7

Conclusion

Past research has highlighted the importance of understanding mode-switching phenomenon in mouse-based and pen-based interfaces. With this thesis, we extend this to touch-based interfaces as well to barehand mid-air interfaces. These results helped us to build an interaction vocabulary when two input modalities are used in combination. We also extend our investigation to revise the definition of interface modes, with an experimental method and task design to identify mode characteristics to facilitate future research on interface modes. Further, characterizing mode-switching revealed important aspects of how hand posture formation could influence quantitative performance as well as subjective perception. Real world implications include effective user interface design, making informed decisions especially when selecting multiple mode-switching techniques is essential, and hopefully increased user satisfaction.

Broadly, our research has the following three main implications: firstly, our empirically driven results can help application developers and researchers to make informed decisions while choosing mode-switching techniques for barehand touch and mid air input. Secondly, our experimental protocol can be used as is or extended to examine future mode-switching techniques. Lastly, we highlight the lack of strong foundation for the concept of mode, which in turn defines the limits to interpret our results and demands further research to unify the mode theory.

7.1 Future Research

There has been considerable interest in exploring novel interaction techniques for both touch-based and mid-air based interfaces. However, knowing their performance metrics and subjective perceptions are the key to their successful application. With the help of our work on mode-switching analysis, researchers and application designers will be able to make informed decisions while choosing interaction techniques and envisioning new interaction techniques. However, as discussed in the previous chapter, mode-switching is just one characteristic of an interface mode. We argue that mode is a multi-faceted concept with many more characteristics, beyond mode-switching, these are not well understood yet.

So, we believe there are several opportunities to perform additional studies to investigate different combinations of mode-switching techniques, to explore using a multi-touch smartphone in Mixed Reality interfaces, investigate implications of mode characteristics on the usability of an interface, and validate our mode definition through empirical investigations.

7.1.1 Investigations with Compound Modes

In our investigation with touch-based mode-switching techniques, all the techniques are compared against a common baseline. Our experiment required participants to frequently switch between one technique and the baseline technique. However, in real usage, frequent switching and a fixed order is highly unlikely. Also, switching to and from a common baseline technique seems may not always be necessary.

Therefore, we recommend investigating mode-switching performance when switching between arbitrary modes and associated techniques, other than a fixed baseline. For instance in a touch interface, switching from longpress to hardpress, instead of switching to one finger touch. This would help us understand the performance of the longpress and hardpress techniques when used in a sequence.

Fennedy and Lee [69] investigated mode-switching performance of 3 modes without using a common baseline. In their experiment, the complete mode sequence is visible on the tablet screen, letting the user group taps based on the colours. However, they do not investigate compound modes and the mode-switching action is fixed to a tap. In compound modes two or more interaction techniques can be combined. For instance combining a longpress with a hardpress action.

For instance, in the ForceSelect [81] interaction technique, a user is required to perform a longpress to invoke a callout menu, then use a hardpress to select a command. The callout shows a zoomed representation of the text located under the initial touch and the five selection options. The amount of pressure against the screen would scroll through the options and lifting a finger would trigger that selection. As an example, investigating such compound modes would enable researchers to analyze how the ForceSelect technique can work alongside other mode-switching techniques in a real application.

7.1.2 Investigations with Different Modalities

Investigating the performance of mode-switching techniques beyond touch and mid-air interfaces is also important. A multimodal interface provides an opportunity to simultaneously use multiple input methods. Bolt et al. [29] conducted an early exploration of multimodal interactions. They used voice and barehand mid-air gestures together to interact with spatially anchored data. Commands like 'create', 'delete', and 'move' were initiated with voice and mid-air hand pointing specified the spatial location to operate. More recently, Srinivasan et al.'s [248] InChorus system lets the user explore the data using pen, touch, and speech input. They report 7 different types of erros, some of them are mode errors. However, they do not explicitly compare the modes and mode errors. The effectiveness of such multimodal interaction considering specifically modes and mode errors is unknown. An empirical investigation of such multimodal systems, using protocols similar to those used in this thesis would be a logical step.

7.1.3 Investigations with Different Task Types

The experiment task plays a vital role when investigating mode-switching performance. The reason is that the time spent on performing a task can affect the switching performance. In our work, we primarily target tasks related to interfaces for sketching or drawing. They incur a continuous engagement with an interface. However, results might vary for more complex tasks, like active reading [185]. Active reading involves making comments and adding annotations, navigating a document, and writing. There can be multiple ways to initiate these actions in an interface, resulting in multiple mode-switching techniques. We speculate that mode-switching performance might vary when performing such complex tasks. The challenge is to design representative tasks with a high degree of control for internal validity.

7.1.4 Investigations with Different Devices

Another important aspect is what devices are used in mode-switching studies. In our work, we focus on a tablet and a head-mounted display with a hand tracking system. However, modern information workers are surrounded by several other wearable and mobile devices

[227] such as smartphones, smartwatches, and Augmented Reality glasses. There are a plethora of interaction techniques to interact with these types of devices, but mode related issues are often overlooked.

Furthermore, investigating mode and mode related characteristics can be interesting for other forms of input devices. For instance, pen and pen-like [283, 66] input devices combined with hand postures expand the interaction vocabulary of an interface [42]. Pen input is proven to be an effective input device for mixed reality environments [203]. Recently, Reipschlager et al. [216] presented the DesignAR system for 3D solid modelling. Their system integrated pen, touch, and Augmented Reality based input into a single application. In such systems, switching between different input methods and their impact on user productivity are still unknown. Another exciting opportunity lies in investigating mode related issues in collaborative environments, where the system needs to support multiple users and multiple devices [98].

7.1.5 Investigations Beyond Mode-Switching

While a major portion of this thesis is devoted to mode-switching investigations, in the previous chapter we introduce several new mode characteristics. These characteristics can potentially impact user performance, perception, and overall productivity. As discussed earlier, our initial investigation tested three types of abstract interfaces representative of modern application layouts. However, there is a potential for testing mode characteristics in other kinds of interfaces, such as virtual and augmented reality. We discuss possible strategies for investigating them here.

Mode-occupancy, the total amount of time spent in a particular mode, likely has different impact based on the body posture of the user. For instance, if the user is interacting in VR while standing, and the mode-occupancy technique uses the arm, then body fatigue becomes a major factor [103]. If the input technique is slightly time-consuming and used frequently, having a higher mode-occupancy would make the interface unusable. However, if the application developer wants to focus on the accuracy of the task, higher mode-occupancy can potentially reduce error rate, since spending more time within a mode can reduce mode errors [148].

The mode-phrasing characteristic is known to be a trait experts possess [40]. So, while designing an interaction technique, it would be desirable to encourage the mode-phrasing and discourage mode-chunking for an expert user. On the contrary, mode-chunking should be easy when the application would primarily be used by novice users. Note that ideally an application developer might want to support both novice and the expert users, but it would be challenging to design interaction techniques that support higher mode-phrasing and higher mode-chunking at the same time. There are a few examples of such interfaces, for example Marking Menus [142] and Scriboli [104]. Yet, we are a long way from having a well-balanced chunking and phrasing interaction design in practice.

7.1.6 Modes in Aviation Psychology

Modes have received significant attention in Aviation Psychology. Miller et al. [180] investigated a formal approach to avoid mode confusion in the context of cockpit interfaces. In a similar vein, Gow et al. [84] introduced a formal model of consistency-related mode confusion. Consistency is a commonly cited principle of interface design [214]. Sarter and Woods [230] analyzed data from pilot surveys, incident reports, and pilot training observations to highlight the gaps in the pilot's understanding of the functional structure of the automation. They warn against having such gaps in pilot's mind and speculate that they might result in severe errors.

In 1995, Degani et al. [59] drew attention to four aircraft accidents backed by the analysis of the data gathered from thirty Boeing 757 and 767 type aircraft logs. They discussed the role of mode confusion as one of the primary causes of these catastrophes. One possible way to avoid this is to investigate interface design before deploying it in the real systems. Leveson et al. [152] conducted a preliminary analysis to detect error-prone automation features using formal methods. As discussed in the previous chapter, they introduce a mode definition as well as types of modes. However, they recommend additional studies to verify their approach. Note even if rare, the cost of mode errors in safety-critical system is life threatening.

Our work introduced new mode characteristics that have received little explicit attention in aviation psychology. Analyzing these characteristics in the context of building interfaces for safety-critical systems can be valuable in avoiding critical errors as well as potential mishaps.

7.2 Final Word

A major portion of the text above is devoted in understanding the impact of mode-switching in an interface. The goal of these mode-switching investigations goes beyond characterizing the performance of modern input techniques, it provides a thinking tool to analyze future interactive systems. We replicate Li et al.'s work in a touch-based interface context and update the subtraction method to analyze barehand mid-air mode-switching techniques in VR. We believe our studies can also be valuable for analyzing atomic actions such as mode-engagement and mode-disengagement, even in the interfaces that we did not test explicitly. Our studies show that however small the magnitude of the difference is among the mode-switching techniques, it does affect user experience at a greater deal. Further, we demonstrate the interaction design process when two modalities are combined together. Identifying the fundamental design dimensions when touch and mid-air interfaces are used together also helps facilitate the process of envisioning future interaction techniques. This allows researchers and practitioners to extended the expressivity of an interface. We hope, our work will motivate investigations with future mode-switching techniques and interaction vocabularies in various settings.

Toward the end, we develop a deeper understanding of the mode phenomenon. We acknowledge that we conduct a preliminary empirical evaluation, however, it does provide a glimpse of the complex nature of the mode concept. We begin to untangle the mode concept by identifying the main characteristics. These characteristics constitute a relevant contribution along with the results obtained in the mode-switching studies. This work can help researchers imagine and evaluate interfaces critically, and empower practitioners in making informed decisions when thinking about the user experience.

References

[1] Vamsi Kiran Adhikarla, Pawel Wozniak, Attila Barsi, Dave Singhal, Péter Tamás Kovács, and Tibor Balogh. Freehand interaction with large-scale 3d map data. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pages 1–4. IEEE, 2014.

[2] Roland Aigner, Daniel Wigdor, Hrvoje Benko, Michael Haller, David Lindbauer, Alexandra Ion, Shengdong Zhao, and JTKV Koh. Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci. *Microsoft Research TechReport MSR-TR-2012-111*, 2, 2012.

[3] Dzmitry Aliakseyeu, Miguel A. Nacenta, Sriram Subramanian, and Carl Gutwin. Bubble radar: Efficient pen-based interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 19–26, New York, NY, USA, 2006. ACM.

[4] Amazon. Amazon's $150 million typo is a lightning rod for a big cloud problem, 2017. https://aws.amazon.com/message/41926/.

[5] Ferran Argelaguet and Carlos Andújar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37:121–136, 2013.

[6] Ferran Argelaguet, Ludovic Hoyet, Michaël Trico, and Anatole Lécuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *Virtual Reality (VR), 2016 IEEE*, pages 3–10. IEEE, 2016.

[7] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 185. ACM, 2018.

[8] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. Experimental evaluation of sketching on surfaces in vr. In *CHI*, pages 5643–5654, 2017.

[9] Ilhan Aslan, Ida Buchwald, Philipp Koytek, and Elisabeth André. Pen + mid-air: An exploration of mid-air gestures to complement pen input on tablets. In *NordiCHI*, 2016.

[10] Rob Aspin and Kien Hoang Le. Augmenting the cave: An initial study into close focused, inward looking, exploration in ipt systems. In *Distributed Simulation and Real-Time Applications, 2007. DS-RT 2007. 11th IEEE International Symposium*, pages 217–224. IEEE, 2007.

[11] Bill Atkinson. Lisa user interface standards., 1980. https://guidebookgallery.org/articles/lisauserinterfacestandards.

[12] Teo Babic, Harald Reiterer, and Michael Haller. Gesturedrawer: one-handed interaction technique for spatial user-defined imaginary interfaces. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pages 128–137. ACM, 2017.

[13] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In *IFIP Conference on Human-Computer Interaction*, pages 248–262. Springer, 2011.

[14] BasementVR, 2017. http://store.steampowered.com/app/561390/BasementVR/.

[15] Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 717–728. ACM, 2016.

[16] Hrvoje Benko, Edward W Ishak, and Steven Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*, pages 132–140. IEEE Computer Society, 2004.

[17] Hrvoje Benko, Edward W Ishak, and Steven Feiner. Cross-dimensional gestural interaction techniques for hybrid immersive environments. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 209–216. IEEE, 2005.

[18] Hrvoje Benko, T. Scott Saponas, Dan Morris, and Desney Tan. Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 93–100, New York, NY, USA, 2009. ACM.

[19] Hrvoje Benko and Daniel Wigdor. *Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input*, pages 249–275. Springer London, London, 2010.

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

[21] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, Montréal, Québec, Canada, 2006. ACM.

[22] Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. An exploration of pen rolling for pen-based interaction. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 191–200, New York, NY, USA, 2008. ACM.

[23] Eric A Bier, Maureen C Stone, Ken Pier, William Buxton, and Tony D DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, 1993.

[24] Mark Billinghurst, Sisinio Baldis, Lydia Matheson, and Mark Philips. 3d palette: a virtual reality content creation tool. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 155–156. ACM, 1997.

[25] Mark Billinghurst, Tham Piumsomboon, and Huidong Bai. Hands in space: Gesture interaction with augmented-reality interfaces. *IEEE computer graphics and applications*, 34(1):77–80, 2014.

[26] Google Blocks, 2017. https://vr.google.com/blocks/.

[27] Sara A Bly and Jarrett K Rosenberg. A comparison of tiled and overlapping windows. *ACM SIGCHI Bulletin*, 17(4):101–106, 1986.

[28] Boeing. Human error report, 2017. https://www.boeing.com/commercial/aeromagazine/articles/qtr_2_07/AERO_Q207_article3.pdf.

[29] Richard A. Bolt. *"Put-that-there": Voice and Gesture at the Graphics Interface*, pages 262–270. SIGGRAPH '80. ACM, New York, NY, USA, 1980.

[30] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The fat thumb: Using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 39–48, New York, NY, USA, 2012. ACM.

[31] Alexander Bornik, Reinhard Beichel, Ernst Kruijff, Bernhard Reitinger, and Dieter Schmalstieg. A hybrid user interface for manipulation of volumetric medical data. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 29–36. IEEE, 2006.

[32] Benoît Bossavit, Asier Marzo, Oscar Ardaiz, and Alfredo Pina. Hierarchical menu selection with a body-centered remote interface. *Interacting with Computers*, 26(5):389–402, 2013.

[33] Doug A Bowman and Larry F Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages & Computing*, 10(1):37–53, 1999.

[34] Doug A Bowman, Mehdi Setareh, Márcio Serolli Pinho, Ndiwalana Ali, Alex Kalita, Yunha Lee, John Lucas, Matthew Gracey, Malini Kothapalli, Qinwei Zhu, et al. Virtual-sap: an immersive tool for visualizing the response of building structures to environmental conditions. In *Virtual Reality, 2003. Proceedings. IEEE*, pages 243–250. IEEE, 2003.

[35] Doug A Bowman and Chadwick A Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Virtual Reality, 2001. Proceedings. IEEE*, pages 149–156. IEEE, 2001.

[36] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 154–161, New York, NY, USA, 2008. ACM.

[37] Stephen A Brewster, Peter C Wright, and Alistair DN Edwards. The design and evaluation of an auditory-enhanced scrollbar. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 173–179, 1994.

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

[39] William Buxton. A three-state model of graphical input. In *Human-computer interaction-INTERACT*, volume 90, pages 449–456, 1990.

[40] William Buxton. Chunking and phrasing and the design of human-computer dialogues. In *Readings in Human–Computer Interaction*, pages 494–499. Elsevier, 1995.

[41] William Buxton and Brad Myers. A study in two-handed input. In *ACM SIGCHI Bulletin*, volume 17, pages 321–326. ACM, 1986.

[42] Drini Cami, Fabrice Matulic, Richard G Calland, Brian Vogel, and Daniel Vogel. Unimanual pen+ touch input using variations of precision grip postures. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 825–837, 2018.

[43] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 euro 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*, CHI '12, pages 2527–2530, New York, NY, USA, 2012. ACM.

[44] CBC. Taiwan plane crash., 2015. https://www.cbc.ca/news/world/taiwan-plane-crash-pilot-pulled-wrong-throttle-shut-down-engine-1.3135213.

[45] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3403–3414. ACM, 2016.

[46] Yun Suk Chang, Benjamin Nuernberger, Bo Luan, and Tobias Höllerer. Evaluating gesture-based augmented reality annotation. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*, pages 182–185. IEEE, 2017.

[47] Li-Chieh Chen, Yun-Maw Cheng, Po-Ying Chu, and Frode Eika Sandnes. The common characteristics of user-defined and mid-air gestures for rotating 3d digital contents. In *International Conference on Universal Access in Human-Computer Interaction*, pages 15–22. Springer, 2016.

[48] Michael Chen, S Joy Mountford, and Abigail Sellen. A study in interactive 3-d rotation using 2-d control devices. *ACM SIGGRAPH Computer Graphics*, 22(4):121–129, 1988.

[49] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. Claw: A multifunctional handheld haptic controller for grasping, touching, and triggering in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 654. ACM, 2018.

[50] A. Cockburn, D. Ahlström, and C. Gutwin. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *Int. J. Hum.-Comput. Stud.*, 70(3):218–233, March 2012.

[51] Ellis S Cohen, Edward T Smith, and Lee A Iverson. Constraint-based tiled windows. *IEEE computer graphics and applications*, 6(5):35–45, 1986.

[52] Ashley Colley and Jonna Häkkilä. Exploring finger specific touch screen interaction for mobile phone user interfaces. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, OzCHI '14, pages 539–548, New York, NY, USA, 2014. ACM.

[53] Paolo Cremonesi, Antonella Di Rienzo, and Franca Garzotto. Personalized interactive public screens. In *Workshop on Interacting with Smart Objects 2015*, page 10, 2015.

[54] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 175–182, 2004.

[55] Bruno R De Araùjo, Géry Casiez, and Joaquim A Jorge. Mockup builder: direct 3d modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012*, pages 173–180. Canadian Information Processing Society, 2012.

[56] Asaf Degani. *Modeling human-machine systems: On modes, error, and patterns of interaction*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of . . . , 1996.

[57] Asaf Degani. On the types of modes in human-machine interactions. In *International Symposium on Aviation Psychology, 9 th, Columbus, OH*, pages 959–964, 1997.

[58] Asaf Degani, Alex Kirlik, and Michael G Shafto. Modes in supervisory control systems: Structure and transitions. 1995.

[59] Asaf Degani, Michael Shafto, and Alex Kirlik. Mode usage in automated cockpits: Some initial observations. *IFAC Proceedings Volumes*, 28(15):345–351, 1995.

[60] Asaf Degani, Michael Shafto, and Alex Kirlik. Modes in automated cockpits: Problems, data analysis and a modelling framework. In *ISRAEL ANNUAL CONFERENCE ON AEROSPACE SCIENCES*, pages 258–266. OMANUTH PRESS LTD, 1996.

[61] Keith Deming and Edward Lank. Managing ambiguity in mode inferencing. In *Proceedings of the AAAI fall symposium series: making pen-based interaction intelligent and natural. October*, pages 21–24.

[62] R. F. Dillon, Jeff D. Edey, and Jo W. Tombaugh. Measuring the true cost of command selection: techniques and results. *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*, (April):19–26, 1990.

[63] Franciscus Cornelis Donders. About the speed of mental processes. *Research done in the Physiologisch Laboratorium der Utrechtsche Hoogeschool (1968–1869)*, 2:92–120, 1868.

[64] Haiwei Dong, Ali Danesh, Nadia Figueroa, and Abdulmotaleb El Saddik. An elicitation study on gesture preferences and memorability toward a practical hand-gesture vocabulary for smart televisions. *IEEE Access*, 3:543–555, 2015.

[65] Peter F Drucker. Tasks, responsibilities, practices, 1974.

[66] Lisa Anne Elkin. Contact-sensing input device manipulation and recall.

[67] Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. Ethereal planes: a design framework for 2d information space in 3d mixed reality environments. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pages 2–12. ACM, 2014.

[68] Alan Esenther and Kathy Ryall. Fluid dtmouse: Better mouse support for touch-based interactions. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 112–115, New York, NY, USA, 2006. ACM.

[69] Katherine Fennedy and Hyowon Lee. Modus operandi: The implications from redesigning moded experiments in multitouch user interfaces. In *Proceedings of the XX International Conference on Human Computer Interaction*, pages 1–4, 2019.

[70] Ajoy S Fernandes and Steven K Feiner. Combating vr sickness through subtle dynamic field-of-view modification. In *3D User Interfaces (3DUI), 2016 IEEE Symposium on*, pages 201–210. IEEE, 2016.

[71] George W. Fitzmaurice and William Buxton. An empirical evaluation of graspable user interfaces: Towards specialized, space-multiplexed input. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 43–50, New York, NY, USA, 1997. ACM.

[72] James D Foley, Victor L Wallace, and Peggy Chan. The human factors of computer graphics interaction techniques. *IEEE computer Graphics and Applications*, 4(11):13–48, 1984.

[73] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 647–656, New York, NY, USA, 2007. ACM.

[74] Cédric Foucault, Manfred Micaux, David Bonnet, and Michel Beaudouin-Lafon. Spad: A bimanual interaction technique for productivity applications on multi-touch tablets. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 1879–1884, New York, NY, USA, 2014. ACM.

[75] Cédric Foucault, Manfred Micaux, David Bonnet, and Michel Beaudouin-Lafon. Spad: a bimanual interaction technique for productivity applications on multi-touch tablets. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 1879–1884. ACM, 2014.

[76] Dustin Freeman, Ramadevi Vennelakanti, and Sriganesh Madhvanath. Freehand pose-based gestural interaction: Studies and implications for interface design. In *Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on*, pages 1–6. IEEE, 2012.

[77] Ina Fried. This new technology lets you control your phone with more than just your fingertips, August 2016. http://www.recode.net/2016/8/30/12639718/qeexo-phone-gestures-knuckles.

[78] Franca Garzotto and Matteo Valoriani. Touchless gestural interaction with small displays: a case study. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, page 26. ACM, 2013.

[79] Ivano Gatto and Fabio Pittarello. Prototyping a gestural interface for selecting and buying goods in a public environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 784–785. ACM, 2012.

[80] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 545–554, New York, NY, USA, 2012. ACM.

[81] Alix Goguey, Sylvain Malacria, and Carl Gutwin. Improving discoverability and expert performance in force-sensitive text selection for touch devices with mode gauges. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[82] Alix Goguey, Mathieu Nancel, Géry Casiez, and Daniel Vogel. The performance and preference of different fingers and chords for pointing, dragging, and object transformation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4250–4261, New York, NY, USA, 2016. ACM.

[83] Google. Project Soli, 2016. https://atap.google.com/soli/.

[84] Jeremy Gow, Harold Thimbleby, and Paul Cairns. Automatic critiques of interface modes. In *International Workshop on Design, Specification, and Verification of Interactive Systems*, pages 201–212. Springer, 2005.

[85] Gravitysketch, 2017. https://www.gravitysketch.com/vr/.

[86] Celeste Groenewald, Craig Anslow, Junayed Islam, Chris Rooney, Peter Passmore, and William Wong. Understanding 3d mid-air hand gestures with interactive surfaces and displays: a systematic literature review. In *Proceedings of the 30th International BCS Human Computer Interaction Conference: Fusion!*, page 43. BCS Learning & Development Ltd., 2016.

[87] Tovi Grossman and Ravin Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 3–12. ACM, 2006.

[88] Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. Hover widgets: Using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 861–870, New York, NY, USA, 2006. ACM.

[89] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 61–70, New York, NY, USA, 2004. ACM.

[90] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 159–166, 2018.

[91] Yves Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, 1987.

[92] Genesis Virtual Guide. Genesis virtual guide., 2018. https://www.youtube.com/watch?v=hDxOX8LYKHs.

[93] François Guimbretière, Andrew Martin, and Terry Winograd. Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.*, 12(3):460–476, September 2005.

[94] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3653–3656. ACM, 2015.

[95] Chris Harrison, Julia Schwarz, and Scott E Hudson. Tapsense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 627–636. ACM, 2011.

[96] Chris Harrison, Julia Schwarz, and Scott E. Hudson. Tapsense: Enhancing finger interaction on touch surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 627–636, New York, NY, USA, 2011. ACM.

[97] Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. Touchtools: Leveraging familiarity and skill with physical tools to augment touch interaction.

In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2913–2916, New York, NY, USA, 2014. ACM.

[98] Jeremy Hartmann, Hemant Bhaskar Surale, Aakar Gupta, and Daniel Vogel. Using conformity to probe interaction challenges in xr collaboration. *arXiv preprint arXiv:2004.05235*, 2020.

[99] Eiji Hayashi, Manuel Maas, and Jason I Hong. Wave to me: user identification using body lengths and natural gestures. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3453–3462. ACM, 2014.

[100] Seongkook Heo and Geehyuk Lee. Forcedrag: Using pressure as a touch input modifier. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 204–207, New York, NY, USA, 2012. ACM.

[101] Luke Hespanhol, Martin Tomitsch, Kazjon Grace, Anthony Collins, and Judy Kay. Investigating intuitiveness and effectiveness of gestures for free spatial interaction with large displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12, pages 6:1–6:6, New York, NY, USA, 2012. ACM.

[102] Otmar Hilliges, Shahram Izadi, Andrew D Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the air: adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 139–148. ACM, 2009.

[103] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072, 2014.

[104] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 451–460, New York, NY, USA, 2005. ACM.

[105] Ken Hinckley, Mary Czerwinski, and Mike Sinclair. Interaction and modeling techniques for desktop two-handed input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 49–58, New York, NY, USA, 1998. ACM.

[106] Ken Hinckley, Francois Guimbretiere, Maneesh Agrawala, Georg Apitz, and Nicholas Chen. Phrasing techniques for multi-stroke selection gestures. In *Graphics Interface*, pages 147–154, 2006.

[107] Ken Hinckley, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Ed Cutrell. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, page 181, New York, New York, USA, 2006. ACM Press.

[108] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. Pre-touch sensing for mobile interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2869–2881, New York, NY, USA, 2016. ACM.

[109] Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang 'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. Sensing techniques for tablet+stylus interaction. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 605–614, New York, NY, USA, 2014. ACM.

[110] Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, Francois Guimbretiere, Marcel Gavriliu, Xiang'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. Sensing techniques for tablet+ stylus interaction. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 605–614. ACM, 2014.

[111] Ken Hinckley, Randy Pausch, Dennis Proffitt, and Neal F Kassell. Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(3):260–302, 1998.

[112] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. Pen + touch = new tools. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 27–36, New York, NY, USA, 2010. ACM.

[113] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. Pen+ touch= new tools. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 27–36. ACM, 2010.

[114] Hololense gestures, 2017. https://developer.microsoft.com/en-us/windows/mixed-reality/gestures.

[115] Kasper Hornbæk and Antti Oulasvirta. What is interaction? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 5040–5052, New York, NY, USA, 2017. Association for Computing Machinery.

[116] Mohammadreza Hosseini, Pascal Vallotton, Tomasz Bednarz, and Arcot Sowmya. A study of touchless versus touch-based interactions with bacterial biofilm images. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 285–290. ACM, 2013.

[117] Rachid Hourizi and Peter Johnson. Unmasking mode errors: A new application of task knowledge principles to the knowledge gaps in cockpit design. In *INTERACT*, pages 255–262, 2001.

[118] HTC. "introducing the logitech bridge sdk", November 2017. Retrieved Mar 31, 2018 from https://blog.vive.com/us/2017/11/02/introducing-the-logitech-bridge-sdk/.

[119] Bret Jackson and Daniel F Keefe. Lift-off: Using reference imagery and freehand sketching to create 3d models in vr. *IEEE transactions on visualization and computer graphics*, 22(4):1442–1451, 2016.

[120] Hessam Jahani and Manolya Kavakli. Exploring a user-defined gesture vocabulary for descriptive mid-air interactions. *Cognition, Technology & Work*, 20:11–22, 2017.

[121] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3328–3339, New York, NY, USA, 2017. ACM.

[122] J Johnson and G Engelbeck. Modes survey results. *ACM SIGCHI Bulletin*, 20(4):38–50, 1989.

[123] Jeff Johnson. Modes in non-computer devices. *International Journal of Man-Machine Studies*, 1990.

[124] Ricardo Jota, Miguel A Nacenta, Joaquim A Jorge, Sheelagh Carpendale, and Saul Greenberg. A comparison of ray pointing techniques for very large displays. In *Proceedings of Graphics Interface 2010*, pages 269–276. Canadian Information Processing Society, 2010.

[125] Alvin Jude, G Michael Poor, and Darren Guinness. Grasp, grab or pinch? identifying user preference for in-air gestural manipulation. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, pages 219–219. ACM, 2016.

[126] Jaemoon Jung, Hanjun Park, Dongwook Hwang, Minseok Son, Donghyun Beck, Juhee Park, and Woojin Park. A review on interaction techniques in virtual environments. In *Proc. 2014 International Conference on Industrial Engineering and Operations Management*, pages 1582–1590, 2014.

[127] Thomas Jung and Patrick Bauer. Constraint-based modeling technique for mid-air interaction. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pages 157–157. ACM, 2017.

[128] Paul Kabbash, William Buxton, and Abigail Sellen. Two-handed input in a compound task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 417–423, New York, NY, USA, 1994. ACM.

[129] Eser Kandogan and Ben Shneiderman. Elastic windows: improved spatial layout and rapid multiple window operations. In *Proceedings of the workshop on Advanced visual interfaces*, pages 29–38, 1996.

[130] Pentti Kanerva and Jeannie Howard Siegman. A text system for stanford university. *ACM SIGOA Newsletter*, 1(2):11–16, May 1980.

[131] Maria Karam et al. A study on the use of semaphoric gestures to support secondary task interactions. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pages 1961–1964. ACM, 2005.

[132] Keiko Katsuragawa, Krzysztof Pietroszek, James R Wallace, and Edward Lank. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 128–135. ACM, 2016.

[133] Daniel F Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H Laidlaw, and Joseph J LaViola Jr. Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 85–93. ACM, 2001.

[134] Daniel F Keefe, Robert C Zeleznik, and David H Laidlaw. Drawing on air: Input techniques for controlled 3d line illustration. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1067–1081, 2007.

[135] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 167–176. ACM, 2012.

[136] Ji-Sun Kim, Denis Gračanin, Krešimir Matković, and Francis Quek. Finger walking in place (fwip): A traveling technique in virtual environments. In *International Symposium on Smart Graphics*, pages 58–69. Springer, 2008.

[137] Yeji Kim, Sohyun Sim, Seoungjae Cho, Woon-woo Lee, Young-Sik Jeong, Kyungeun Cho, and Kyhyun Um. Intuitive nui for controlling virtual objects based on hand movements. In *Future Information Technology*, pages 457–461. Springer, 2014.

[138] Kinect gestures, 2017. https://support.xbox.com/en-GB/xbox-360/accessories/body-controller.

[139] Kurt Koffka. *Principles of Gestalt psychology*. Routledge, 2013.

[140] Arun Kulshreshth and Joseph J LaViola Jr. Exploring the usefulness of finger-based 3d gesture menu selection. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1093–1102. ACM, 2014.

[141] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 258–264, 1994.

[142] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94*, pages 258–264, New York, New York, USA, 1994. ACM Press.

[143] Edward Lank, Jaime Ruiz, and William Cowan. Concurrent bimanual stylus interaction: A study of non-preferred hand mode manipulation. In *Proceedings of Graphics Interface 2006*, GI '06, pages 17–24, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[144] Leap gestures, 2017. https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html.

[145] Joon Hyub Lee, Sang-Gyun An, Yongkwan Kim, and Seok-Hyung Bae. Projective windows: Bringing windows in space to the fingertip. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 218. ACM, 2018.

[146] Sang-Su Lee, Jeonghun Chae, Hyunjeong Kim, Youn-kyung Lim, and Kun-pyo Lee. Towards more natural digital content manipulation via user freehand gestural interaction in a living room. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 617–626. ACM, 2013.

[147] Sangjin Lee, Inseok Hwang, and Kenneth Leiden. Intent inference-based flight-deck human-automation mode-confusion detection. *Journal of Aerospace Information Systems*, 12(8):503–518, 2015.

[148] Sangyoon Lee, Jaeyeon Lee, and Geehyuk Lee. Diagnosing and coping with mode errors in korean-english dual-language keyboard. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[149] Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input: an experimental study. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(4):326–359, 1998.

[150] Jakob F. Leitner, Florian Perteneder, Can Liu, Christian Rendl, and Michael Haller. Kolibri: Tiny and fast gestures for large pen-based surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1789–1798, New York, NY, USA, 2013. ACM.

[151] G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. The design and evaluation of multitouch marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2233–2242, New York, NY, USA, 2010. ACM.

[152] Nancy Leveson, L Denise Pinnel, Sean David Sandys, Shuichi Koga, and Jon Damon Reese. Analyzing software specifications for mode confusion potential. In *Proceedings of a workshop on human error and system development*, pages 132–146. Glasgow Accident Analysis Group, 1997.

[153] Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, page 461, New York, New York, USA, 2005. ACM Press.

[154] Yang Li, Ken Hinckley, Ken Hinckley, Zhiwei Guan, and James A Landay. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 461–470. ACM, 2005.

[155] Yi-Chi Liao, Yen-Chiu Chen, Liwei Chan, and Bing-Yu Chen. Dwell+: Multi-level mode selection using vibrotactile cues. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 5–16. ACM, 2017.

[156] Lorraine Lin and Sophie Jörg. Need a hand?: how appearance affects the virtual hand illusion. In *Proceedings of the ACM Symposium on Applied Perception*, pages 69–76. ACM, 2016.

[157] Robert W Lindeman, John L Sibert, and James K Hahn. Hand-held windows: towards effective 2d interaction in immersive virtual environments. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 205–212. IEEE, 1999.

[158] Robert W Lindeman, John L Sibert, and James K Hahn. Towards usable vr: an empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 64–71. ACM, 1999.

[159] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. Gunslinger: Subtle arms-down mid-air interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 63–71. ACM, 2015.

[160] Andreas Löcken, Tobias Hesselmann, Martin Pielot, Niels Henze, and Susanne Boll. User-centred process for the definition of free-hand gestures applied to controlling music playback. *Multimedia systems*, 18(1):15–31, 2012.

[161] Julian Looser, Mark Billinghurst, Raphaël Grasset, and Andy Cockburn. An evaluation of virtual lenses for object selection in augmented reality. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 203–210. ACM, 2007.

[162] Paul Lubos, Gerd Bruder, Oscar Ariza, and Frank Steinicke. Touching the sphere: Leveraging joint-centered kinespheres for spatial user interaction. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, pages 13–22. ACM, 2016.

[163] Paul Lubos, Gerd Bruder, and Frank Steinicke. Analysis of direct selection in head-mounted display environments. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pages 11–18. IEEE, 2014.

[164] Yuexing Luo and Daniel Vogel. Crossing-based selection with direct touch input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2627–2636, New York, NY, USA, 2014. ACM.

[165] Yuexing Luo and Daniel Vogel. Pin-and-cross: A unimanual multitouch technique combining static touches with crossing selection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 323–332, New York, NY, USA, 2015. ACM.

[166] I Scott MacKenzie. *Human-computer interaction: An empirical research perspective.* Newnes, 2012.

[167] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1073–1082. ACM, 2014.

[168] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *IFIP Conference on Human-Computer Interaction*, pages 461–476. Springer, 2011.

[169] Nicolai Marquardt, Johannes Kiemer, David Ledo, Sebastian Boring, and Saul Greenberg. Designing user-, hand-, and handpart-aware tabletop interactions with the TouchID toolkit. In *Proc. ITS*, pages 21–30, 2011.

[170] Sebastian Marwecki, Andrew D Wilson, Eyal Ofek, Mar Gonzalez Franco, and Christian Holz. Mise-unseen: Using eye tracking to hide virtual reality scene changes in plain sight. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 777–789, 2019.

[171] MasterpieceVR, 2017. http://www.store.steampowered.com/app/504650/MasterpieceVR/.

[172] Nobuyuki Matsushita, Yuji Ayatsuka, and Jun Rekimoto. Dual touch: A two-handed interface for pen-based pdas. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 211–212, New York, NY, USA, 2000. ACM.

[173] Fabrice Matulic and Moira C. Norrie. Pen and touch gestural environment for document editing on interactive tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 41–50, New York, NY, USA, 2013. ACM.

[174] John C McClelland, Robert J Teather, and Audrey Girouard. Haptobend: shape-changing passive haptic feedback in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pages 82–90. ACM, 2017.

[175] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2143–2152. ACM, 2015.

[176] Jess McIntosh, Asier Marzo, and Mike Fraser. Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 593–597. ACM, 2017.

[177] Daniel Mendes, Filipe Relvas, Alfredo Ferreira, and Joaquim Jorge. The benefits of dof separation in mid-air 3d object manipulation. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 261–268. ACM, 2016.

[178] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[179] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[180] Steven P Miller. *Detecting mode confusion through formal modeling and analysis*. National Aeronautics and Space Administration, Langley Research Center, 1999.

[181] Minecraft. Minecraft wiki., 2018. https://minecraft.gamepedia.com/Block.

[182] Takashi Miyaki and Jun Rekimoto. Graspzoom: Zooming and scrolling control model for single-handed mobile interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 11:1–11:4, New York, NY, USA, 2009. ACM.

[183] llc mizage. Divvy: Window manager., 2020. https://mizage.com/divvy/.

[184] Andrew Monk. Mode errors: A user-centred analysis and some preventative measures using keying-contingent sound. *International journal of man-machine studies*, 24(4):313–327, 1986.

[185] Meredith Ringel Morris, AJ Bernheim Brush, and Brian R Meyers. Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*, pages 79–86. IEEE, 2007.

[186] Myo gestures, 2017. https://support.getmyo.com/hc/en-us/articles/202647853-What-gestures-does-the-Myo-armband-recognize-.

[187] Mathieu Nancel, Daniel Vogel, Bruno De Araujo, Ricardo Jota, and Géry Casiez. Next-point prediction metrics for perceived spatial errors. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 271–285, New York, NY, USA, 2016. ACM.

[188] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 177–186. ACM, 2011.

[189] CBA News. Taiwan plane crash, 2017. Accessed: 2020-02-20, http://tinyurl.com/qg3bje8.

[190] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. Designing for low-latency direct-touch input. In *Proc. UIST*, pages 453–464, 2012.

[191] Tao Ni, Doug A Bowman, Chris North, and Ryan P McMahan. Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. *International Journal of Human-Computer Studies*, 69(9):551–562, 2011.

[192] Nngroup. Modal and nonmodal dialogs, 2017. Accessed: 2020-02-20.

[193] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[194] Don Norman. *The design of everyday things*. Verlag Franz Vahlen GmbH, 2016.

[195] Donald A Norman. Categorization of action slips. *Psychological review*, 88(1):1, 1981.

[196] NYTimes. How to handle the dreaded 'reply all moment', 2017. Accessed: 2020-02-20.

[197] Oculus. Oculus locomotion., 2018. https://developer.oculus.com/design/latest/concepts/bp-locomotion.

[198] Sharon Oviatt. Multimodal interfaces. In *The human-computer interaction handbook*, pages 439–458. CRC press, 2007.

[199] Paintlab, 2017. https://www.paintlabvr.com/.

[200] Anna Pereira, Juan P Wachs, Kunwoo Park, and David Rempel. A user-developed 3-d hand gesture set for human–computer interaction. *Human factors*, 57(4):607–621, 2015.

[201] Thies Pfeiffer. Towards gaze interaction in immersive virtual reality: Evaluation of a monocular eye tracking set-up. In *Virtuelle und Erweiterte Realität-Fünfter Workshop der GI-Fachgruppe VR/AR*, pages 81–92, 2008.

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

[203] Duc-Minh Pham and Wolfgang Stuerzlinger. Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11, 2019.

[204] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*, pages 282–299. Springer, 2013.

[205] Thammathip Piumsomboon, Adrian J. Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *INTERACT*, 2013.

[206] Ondrej Polacek, Martin Klima, Adam J Sporka, Pavel Zak, Michal Hradis, Pavel Zemcik, and Vaclav Prochazka. A comparative study on distant free-hand pointing. In *Proceedings of the 10th European conference on Interactive tv and video*, pages 139–142. ACM, 2012.

[207] Merle F Poller and Susan K Garter. The effects of modes on text editing by experienced editor users. *Human Factors*, 26(4):449–462, 1984.

[208] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996.

[209] Ivan Poupyrev, T Ichikawa, S Weghorst, and M Billinghurst. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, volume 17, pages 41–52. Wiley Online Library, 1998.

[210] Ivan Poupyrev and Tadao Ichikawa. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages & Computing*, 10(1):19–35, 1999.

[211] Ivan Poupyrev, Numada Tomokazu, and Suzanne Weghorst. Virtual notepad: handwriting in immersive vr. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 126–132. IEEE, 1998.

[212] Gonzalo Ramos and Ravin Balakrishnan. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 105–114, Vancouver, Canada, 2003. ACM.

[213] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494, 2004.

[214] Jef Raskin. *The humane interface: new directions for designing interactive systems.* Addison-Wesley Professional, 2000.

[215] Guido Maria Re and Monica Bordegoni. A natural user interface for navigating in organized 3d virtual contents. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 93–104. Springer, 2014.

[216] Patrick Reipschläger and Raimund Dachselt. Designar: Immersive 3d-modeling combining augmented reality with interactive displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pages 29–41, 2019.

[217] Gerhard Reitmayr and Dieter Schmalstieg. Mobile collaborative augmented reality. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 114–123. IEEE, 2001.

[218] Christian Rendl, Patrick Greindl, Kathrin Probst, Martin Behrens, and Michael Haller. Presstures: Exploring pressure-sensitive multi-touch gestures on trackpads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 431–434, New York, NY, USA, 2014. ACM.

[219] MIT Tech Review. Amazon's $150 million typo is a lightning rod for a big cloud problem, 2017. https://www.technologyreview.com/s/603784/amazons-150-million-typo-is-a-lightning-rod-for-a-big-cloud-problem/.

[220] Anne Roudaut, Eric Lecolinet, and Yves Guiard. Microrolls: Expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 927–936, New York, NY, USA, 2009. ACM.

[221] Élisabeth Rousset, François Bérard, and Michaël Ortega. Two-finger 3d rotations for novice users: surjective and integral interactions. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 217–224. ACM, 2014.

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

[223] Jaime Ruiz, Andrea Bunt, and Edward Lank. A model of non-preferred hand mode switching. In *Proceedings of Graphics Interface 2008*, GI '08, pages 49–56, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[224] Jaime Ruiz and Edward Lank. A study on the scalability of non-preferred hand mode manipulation. In *Proceedings of the 9th international conference on Multimodal interfaces*, pages 170–177. ACM, 2007.

[225] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 197–206. ACM, 2011.

[226] Jaime Ruiz and Daniel Vogel. Soft-constraints to reduce legacy and performance bias to elicit whole-body gestures with low arm fatigue. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3347–3350. ACM, 2015.

[227] Stephanie Santosa and Daniel Wigdor. A field study of multi-device workflows in distributed workspaces. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 63–72, 2013.

[228] Markus Sareika and Dieter Schmalstieg. Bimanual handheld mixed reality interfaces for urban planning. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 189–196. ACM, 2010.

[229] Nadine B Sarter. *"Strong silent and 'out-of-the-loop'": properties of advanced (cockpit) automation and their impact on human-automation interaction.* PhD thesis, The Ohio State University, 1994.

[230] Nadine B Sarter and David D Woods. Pilot interaction with cockpit automation ii: An experimental study of pilots' model and awareness of the flight management system. *The International Journal of Aviation Psychology*, 4(1):1–28, 1994.

[231] Nadine B Sarter, David D Woods, Charles E Billings, et al. Automation surprises. *Handbook of human factors and ergonomics*, 2:1926–1943, 1997.

[232] Farzan Sasangohar, I Scott MacKenzie, and Stacey D Scott. Evaluation of mouse and touch input for a tabletop display using fitts' reciprocal tapping task. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 53, pages 839–843. SAGE Publications, 2009.

[233] Eric Saund and Edward Lank. Stylus input and editing without prior selection of mode. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 213–216, New York, NY, USA, 2003. ACM.

[234] Steven Schkolne, Michael Pruett, and Peter Schröder. Surface drawing: creating organic 3d shapes with the hand and tangible tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 261–268. ACM, 2001.

[235] Valentin Schwind, Pascal Knierim, Cagri Tasci, Patrick Franczak, Nico Haas, and Niels Henze. These are not my hands!: Effect of gender on the perception of avatar hands in virtual reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1577–1582. ACM, 2017.

[236] Abigail J Sellen, Gordon P Kurtenbach, and William AS Buxton. The prevention of mode errors through sensory feedback. *Human-computer interaction*, 7(2):141–164, 1992.

[237] Kyeongeun Seo and Hyeonjoong Cho. Airpincher: A handheld device for recognizing delicate mid-air hand gestures. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST'14 Adjunct, pages 83–84, New York, NY, USA, 2014. ACM.

[238] Junchen Shen, Yanlin Luo, Xingce Wang, Zhongke Wu, and Mingquan Zhou. Gpu-based realtime hand gesture interaction and rendering for volume datasets using leap motion. In *Cyberworlds (CW), 2014 International Conference on*, pages 85–92. IEEE, 2014.

[239] Ben Shneiderman and Catherine Plaisant. *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010.

[240] Itiro Siio and Hitomi Tsujita. Mobile interaction using paperweight metaphor. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1325–1330, New York, NY, USA, 2006. ACM.

[241] Sketchfab, 2017. https://sketchfab.com/virtual-reality.

[242] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. Grips and gestures on a multi-touch pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1323–1332, New York, NY, USA, 2011. ACM.

[243] Peng Song, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1297–1306. ACM, 2012.

[244] Peng Song, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1297–1306. ACM, 2012.

[245] Peng Song, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1297–1306. ACM, 2012.

[246] Charles F Spencer Jr. Cockpit automation and mode confusion: The use of auditory inputs for error mitigation. Technical report, AIR COMMAND AND STAFF COLL MAXWELL AFB AL, 2000.

[247] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of*

the 33rd Annual ACM Conference on Human Factors in Computing Systems, pages 3643–3652. ACM, 2015.

[248] Arjun Srinivasan, Bongshin Lee, Nathalie Henry Riche, Steven M Drucker, and Ken Hinckley. Inchorus: Designing consistent multimodal interactions for data visualization on tablet devices. *arXiv preprint arXiv:2001.06423*, 2020.

[249] Michael Stapelberg. i3: tiling window manager., 2020. https://i3wm.org/.

[250] Sophie Stellmach, Markus Jüttner, Christian Nywelt, Jens Schneider, and Raimund Dachselt. Investigating freehand pan and zoom. In *Mensch & computer*, pages 303–312, 2012.

[251] Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co., 1995.

[252] Evan Strasnick, Christian Holz, Eyal Ofek, Mike Sinclair, and Hrvoje Benko. Haptic links: Bimanual haptics for virtual reality using variable stiffness actuation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 644. ACM, 2018.

[253] Hemant Bhaskar Surale, Aakar Gupta, Mark Hancock, and Daniel Vogel. Tabletinvr: Exploring the design space for using a multi-touch tablet in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[254] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. Experimental analysis of mode switching techniques in touch-based user interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3267–3280. ACM, 2017.

[255] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. Experimental analysis of barehand mid-air mode-switching techniques in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[256] Ivan E Sutherland. The ultimate display. *Multimedia: From Wagner to virtual reality*, pages 506–508, 1965.

[257] Zsolt Szalavári and Michael Gervautz. The personal interaction panel–a two-handed interface for augmented reality. In *Computer graphics forum*, volume 16. Wiley Online Library, 1997.

[258] Anthony Talvas. *Bimanual haptic interaction with virtual environments*. PhD thesis, INSA de Rennes, 2014.

[259] Justin H Tan, Cherng Chao, Mazen Zawaideh, Anne C Roberts, and Thomas B Kinney. Informatics in radiology: Developing a touchless user interface for intra-operative image control during interventional radiology procedures. *Radiographics*, 33(2):E61–E70, 2013.

[260] Shun'ichi Tano, Shinya Yamamoto, Junko Ichino, Tomonori Hashiyama, and Mitsuru Iwata. Truly useful 3d drawing system for professional designer by "life-sized and operable" feature and new interaction. In *IFIP Conference on Human-Computer Interaction*, pages 37–55. Springer, 2013.

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

[262] MIT technology review. 150 million dollar typo., 2017. https://www.technologyreview.com/s/603784/amazons-150-million-typo-is-a-lightning-rod-for-a-big-cloud-problem/.

[263] Larry Tesler. The smalltalk environment. *Byte*, 6(8), 1981.

[264] Larry Tesler. A personal history of modeless text editing and cut/copy-paste. *interactions*, 19(4):70–75, 2012.

[265] Harold Thimbleby. Character level ambiguity: consequences for user interface design. *International Journal of Man-Machine Studies*, 16(2):211–225, 1982.

[266] Tiltbrush, 2017. https://www.tiltbrush.com/.

[267] Alex Todd. Apple 3d touch vs huawei press touch vs zte force touch, February 2016.

[268] Huawei Tu, Xing-Dong Yang, Feng Wang, Feng Tian, and Xiangshi Ren. Mode switching techniques through pen and device profiles. In *Proceedings of the 10th asia pacific conference on Computer human interaction - APCHI '12*, page 169, New York, New York, USA, 2012. ACM Press.

[269] Ying-Chao Tung, Ta Yang Cheng, Neng-Hao Yu, Chiuan Wang, and Mike Y Chen. Flickboard: enabling trackpad interaction with automatic mode switching on a capacitive-sensing keyboard. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1847–1850. ACM, 2015.

[270] Apivan Tuntakurn, Saowapak S Thongvigitmanee, Vera Sa-Ing, Shoichi Hasegawa, and Stanislav S Makhanov. Natural interactive 3d medical image viewer based on finger and arm gestures. In *Biomedical Engineering International Conference (BMEiCON), 2013 6th*, pages 1–5. IEEE, 2013.

[271] Florian van de Camp, Alexander Schick, and Rainer Stiefelhagen. How to click in mid-air. In *International Conference on Distributed, Ambient, and Pervasive Interactions*, pages 78–86. Springer, 2013.

[272] Lode Vanacken, Tovi Grossman, and Karin Coninx. Multimodal selection techniques for dense and occluded 3d virtual environments. *International Journal of Human-Computer Studies*, 67(3):237–255, 2009.

[273] Radu-Daniel Vatavu. User-defined gestures for free-hand tv control. In *Proceedings of the 10th European conference on Interactive tv and video*, pages 45–48. ACM, 2012.

[274] Radu-Daniel Vatavu. A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. *Journal of Ambient Intelligence and Smart Environments*, 5(2):187–211, 2013.

[275] Radu-Daniel Vatavu and Ionut-Alexandru Zaiti. Leap gestures for tv: insights from an elicitation study. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*, pages 131–138. ACM, 2014.

[276] Eduardo Velloso, Dominik Schmidt, Jason Alexander, Hans Gellersen, and Andreas Bulling. The feet in human–computer interaction: A survey of foot-based interaction. *ACM Computing Surveys (CSUR)*, 48(2):21, 2015.

[277] HTC VIVE. Htc custom tracker., 2018. https://dl.vive.com/Tracker/Guideline/HTC_Vive_Tracker_Developer_Guidelines_v1.3.pdf.

[278] Amilcar Meneses Viveros and Erika Hernández Rubio. Kinect©, as interaction device with a tiled display. In *International Conference on Human-Computer Interaction*, pages 301–311. Springer, 2013.

[279] Bahtijar Vogel, Oskar Pettersson, Arianit Kurti, and Andreas Simonsson Huck. Utilizing gesture based interaction for supporting collaborative explorations of visualizations in tel. In *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on*, pages 177–181. IEEE, 2012.

[280] Daniel Vogel and Ravin Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 137–146. ACM, 2004.

[281] Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 33–42, New York, NY, USA, 2005. ACM.

[282] Daniel Vogel and Géry Casiez. Conté: Multimodal input inspired by an artist's crayon. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 357–366, New York, NY, USA, 2011. ACM.

[283] Daniel Vogel and Géry Casiez. Conté: multimodal input inspired by an artist's crayon. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 357–366, 2011.

[284] Julie Wagner, Stéphane Huot, and Wendy Mackay. BiTouch and BiPad: designing bimanual interaction for hand-held tablets. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, page 2317, New York, New York, USA, 2012. ACM Press.

[285] Julie Wagner, Stéphane Huot, and Wendy Mackay. Bitouch and bipad: designing bimanual interaction for hand-held tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2317–2326. ACM, 2012.

[286] Robert Walter, Gilles Bailly, and Jörg Müller. Strikeapose: revealing mid-air gestures on public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 841–850. ACM, 2013.

[287] Robert Walter, Gilles Bailly, Nina Valkanova, and Jörg Müller. Cuenesics: using mid-air gestures to select items on interactive public displays. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pages 299–308. ACM, 2014.

[288] Jia Wang and Robert Lindeman. Coordinated 3d interaction in tablet-and hmd-based hybrid virtual environments. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pages 70–79. ACM, 2014.

[289] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 86. ACM, 2018.

[290] Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. Rock & rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1581–1590, 2011.

[291] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 143–146, 2011.

[292] Jacob O Wobbrock, Meredith Ringel Morris, and Andrew D Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1083–1092. ACM, 2009.

[293] Huiyue Wu and Jianmin Wang. User-defined body gestures for tv-based applications. In *Digital Home (ICDH), 2012 Fourth International Conference on*, pages 415–420. IEEE, 2012.

[294] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, Vancouver, Canada, 2003. ACM.

[295] Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society. ACM ID: 1110635.

[296] Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *First*

IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'06), pages 8–pp. IEEE, 2006.

[297] Siju Wu, Amine Chellali, Samir Otmane, and Guillaume Moreau. Touchsketch: a touch-based interface for 3d object manipulation and editing. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, pages 59–68. ACM, 2015.

[298] Haijun Xia, Tovi Grossman, and George Fitzmaurice. Nanostylus: Enhancing input on ultra-small displays with a finger-mounted stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology*, UIST 2015, pages 447–456, New York, NY, USA, 2015. ACM.

[299] Yukang Yan, Chun Yu, Xiaojuan Ma, Shuai Huang, Hasan Iqbal, and Yuanchun Shi. Eyes-free target acquisition in interaction space around the body for virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 42. ACM, 2018.

[300] Ionuţ-Alexandru Zaiţi, Ştefan-Gheorghe Pentiuc, and Radu-Daniel Vatavu. On free-hand tv control: experimental results on user-elicited gestures with leap motion. *Personal and Ubiquitous Computing*, 19(5-6):821–838, 2015.

[301] Mario Martínez Zarzuela, Francisco J Díaz Pernas, Sergio Martín Calzón, David González Ortega, and Miriam Antón Rodríguez. Educational tourism through a virtual reality platform. *Procedia Computer Science*, 25:382–388, 2013.

[302] Shumin Zhai, Jing Kong, and Xiangshi Ren. Speed–accuracy tradeoff in Fitts' law tasks—on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, 61(6):823–856, December 2004.

[303] Jingjie Zheng, Xiaojun Bi, Kun Li, Yang Li, and Shumin Zhai. M3 gesture menu: Design and experimental analyses of marking menus for touchscreen mobile interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 249. ACM, 2018.

[304] Jamie Zigelbaum, Alan Browning, Daniel Leithinger, Olivier Bau, and Hiroshi Ishii. G-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 261–264. ACM, 2010.

APPENDICES

Appendix A

Questionnaires: Touch-Based Mode-Switching Investigation

This appendix includes the full questionnaire that was used in Section 3.4.

A.1 Questionnaires

The pre-experiment questionnaire included six unique questions and the post-experiment questionnaire included rating each mode-switching technique across six different dimensions.

PARTICIPANT # _______________ DATE ________________________

Pre-experiment Questionnaire

1. Gender: ______
2. Age: ______
3. What hand do you typically use to control a touchpad, mouse, or touch screen?

   Left *Right*

4. Do you use a tablet?

   Yes *No*

5. If yes to question 4, how many hours per week on average do you use a tablet?

   ________ *hours per week*

6. If yes to question 4, how often do you hold a tablet in one arm or hand while interacting with the other?

Frequently	*Often*	*Moderate*	*Sometimes*	*Never*

PARTICIPANT # ________________ DATE ________________________

Post-experiment Questionnaire

1. Please fill out the following questionnaire in the scale from 1 to 5 (with 1 being the worst and 5 being the best).

    - **Thumb on finger:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		Operation speed	
Score	Comments	Score	Comments
Eye Fatigue		Hand Fatigue	
Score	Comments	Score	Comments

PARTICIPANT # ________________   DATE _________________________

- **Knuckle:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		**Speed**	
Score	Comments	Score	Comments
Eye Fatigue		**Hand Fatigue**	
Score	Comments	Score	Comments

- **Hard press:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		**Speed**	
Score	Comments	Score	Comments
Eye Fatigue		**Hand Fatigue**	
Score	Comments	Score	Comments

PARTICIPANT # _______________ DATE ________________________

- **Other hand:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		Speed	
Score	Comments	Score	Comments
Eye Fatigue		Hand Fatigue	
Score	Comments	Score	Comments

- **Long press:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		Speed	
Score	Comments	Score	Comments
Eye Fatigue		Hand Fatigue	
Score	Comments	Score	Comments

PARTICIPANT # _______________ DATE ________________________

- **Two fingers:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		**Speed**	
Score	Comments	Score	Comments
Eye Fatigue		**Hand Fatigue**	
Score	Comments	Score	Comments

---

1. Which mode-switching technique do you LIKE overall?

   Thumb on finger   Knuckle   Hard press   Other hand   Long press   Two fingers

2. Which mode-switching technique do you DISLIKE overall?

   Thumb on Finger   Knuckle   Hard press   Other hand   Long press   Two fingers

3. Do you have any additional comments?

Appendix B

Questionnaires: Barehand Mid-Air Mode-Switching Investigation

This appendix includes the full questionnaire that was used in Section 4.4 and 4.5.

B.1 Questionnaires

The pre-experiment questionnaire included seven unique questions and the post-experiment questionnaire included scales to rate mode-switching techniques.

PARTICIPANT # _______________   DATE ________________________

Pre-experiment Questionnaire

1. Gender: ______
2. Age: ______
3. What hand do you typically use to control a touchpad, mouse, or touch screen?

   Left   Right

4. *How many hours on an average do you use a computer or a laptop each day?* ____________
5. *Have you used any of the Virtual Reality headsets, like Oculus Rift, HTV Vive, Samsung Gear VR, or Google DayDream?*

   Yes   No

6. If you have answered yes to the above question, how many hours per week on average do you use it?

   ________ *hours per week*

7. Do you frequently play fast-paced games like first person shooters or car racing?

Frequently	*Often*	*Moderate*	*Sometimes*	*Never*

PARTICIPANT # ________________ DATE _________________________

Post-experiment Questionnaire

1. Please fill out the following questionnaire in the scale from 1 to 5 (with 1 being the worst and 5 being the best).

    - **[Mode-Switching Technique]:**

Ease of learning		Ease of use	
Score	Comments	Score	Comments
Accuracy		**Operation speed**	
Score	Comments	Score	Comments
Eye Fatigue		**Hand Fatigue**	
Score	Comments	Score	Comments

1. Which mode-switching technique do you LIKE overall?

    ____

2. Which mode-switching technique do you DISLIKE overall?

    ____

3. Do you have any additional comments?

    ____

Appendix C

Questionnaires: Touch and Mid-Air Input used in Combination

This appendix includes the full questionnaire that was used in Section 5.5.

C.1 Questionnaires

The pre-experiment questionnaire included five unique questions and the post-experiment questionnaire included six questions.

PARTICIPANT # _______________ DATE ________________________

Pre-experiment Questionnaire

1. Gender: ______
2. Age: ______
3. Have you used any 3D modelling tools? Which ones?

   ____

4. *Have you used any of the Virtual Reality headsets, like Oculus Rift, HTV Vive, Samsung Gear VR, or Google DayDream?*

   Yes *No*

5. If you have answered yes to the above question, how many hours per week on average do you use it?

   ________ *hours per week*

PARTICIPANT # _______________ DATE ________________________

Post-experiment Questionnaire

1. Describe how easy or difficult it was to learn and use the following features of TabletInVR system?
    - Creating objects
    - Selecting and deselecting objects
    - Deleting objects
    - Transforming the objects (i.e. rotate, translate, and scale)
    - Modifying the geometry of an object (i.e. extrude and slicing)
    - Menu Navigation (i.e. scrolling through the list of objects on the back of the tablet)
    - Navigation
    - Other
2. How was your overall experience of replicating the target model?
3. How was your overall experience of the freeform design task?
4. Which feature did you LIKE the most?

   ____

5. Which feature did you DISLIKE the most?

   ____

6. Do you have any additional comments?

   ____