

Concert: Elementary VR Interactions Augmented By Bimanual Input

by

Futian Zhang

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirements for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2025

© Futian Zhang 2025

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	Barrett Ens Associate Professor, Department of Computer Science, Mathematics, Physics and Statistics, The University of British Columbia
Supervisors	Jian Zhao Associate Professor, Cheriton School of Computer Science, University of Waterloo
Supervisors	Keiko Katsuragawa Adjunct Assistant Professor, Cheriton School of Computer Science, University of Waterloo
Supervisors	Edward Lank Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member	Daniel Vogel Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member	Mark Hancock Professor, Department of Management Sciences, University of Waterloo
Internal-external Member	James Wallace Professor, School of Public Health Sciences, 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. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

The following list serves as a declaration of the works included in this dissertation. This material is expanded and revised from the original publication.

Portions of Chapters 3: Futian Zhang, Keiko Katsuragawa, and Edward Lank. 2022. Conductor: Intersection-Based Bimanual Pointing in Augmented and Virtual Reality. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 560 (December 2022), 15 pages. <https://doi.org/10.1145/3567713>

Portions of Chapters 4: Futian Zhang, Jiawen Stefanie Zhu, Edward Lank, Keiko Katsuragawa, and Jian Zhao. 2025. Fly the Moon to Me: Bimanual 3D Locomotion in Virtual Reality By Manipulating the Position of the Destination Object. In *Proceedings of the 51st Graphics Interface Conference (GI '25)*. Association for Computing Machinery, New York, NY, USA.

Portions of Chapters 5: Futian Zhang, Paul Kokhanov, Edward Lank, Keiko Katsuragawa, and Jian Zhao. 2025. Drum Menu: Bimanual Controller Command Access Techniques in Virtual Reality. In *Proceedings of the 51st Graphics Interface Conference (GI '25)*. Association for Computing Machinery, New York, NY, USA.

¹ <https://authors.acm.org/author-resources/author-rights>, Accessed on May 18, 2025.

ABSTRACT

Many elementary tasks in Virtual Reality (VR)—such as pointing, locomotion, and command selection—are still predominantly performed through unimanual interaction, even though the non-dominant hand often remains available. When these tasks require multiple sequential steps, their frequent use can increase overall interaction time and user effort. By engaging both hands in complementary roles, some of these tasks can be streamlined into more efficient, parallelized actions without sacrificing intuitiveness. This thesis, *Concert*, investigates how bimanual input can optimize such underexplored opportunities through three novel interaction techniques. The first, **Conductor**, is an intersection-based 3D pointing method where the dominant hand controls an aiming ray while the non-dominant hand positions an intersecting plane to determine cursor depth. Compared to Raycursor, a state-of-the-art VR pointing technique, Conductor enables faster and more accurate target selection. The second project, **Fly The Moon To Me** (Locomoontion), extends Conductor to 3D object manipulation for locomotion, integrating vertical (height) and horizontal (planar) adjustments into a single fluid operation. Users generate a preview copy of a target object, reposition it precisely in 3D space, and then align the original object and surrounding environment to the preview. In a teleportation task, Locomoontion outperformed Point & Teleport and Point & Tug, reducing completion time and physical effort. The third project, **Drum Menu**, accelerates command selection through bimanual shortcuts inspired by marking menus. Users select commands by joystick rotation, stroke drawing, or directional pointing, with the bimanual variant granting simultaneous access to two menu levels. In user studies, bimanual configurations were faster than unimanual ones for a 4-item layout, with participants preferring the bimanual joystick menu; however, 8-item layouts increased error rates among experienced users. Collectively, these studies show that many common VR tasks can be made more efficient through carefully designed bimanual interaction, providing concrete design guidelines for integrating both hands into future immersive systems.

ACKNOWLEDGMENTS

I am deeply grateful to all the people, both great and small, who made this thesis and the four-year journey possible.

First, I would like to thank my supervisors: the late Edward Lank, Jian Zhao, and Keiko Katsuragawa. I still remember that, at the end of my master's program, I was struggling with decisions about my future career, so I went to Ed for advice. It was during the COVID period, and we were discussing the possibility of pursuing a part-time PhD while working full-time in industry. Ed was incredibly supportive—he patiently walked me through the school's regulations and explained how he could help. Then he put himself in my position and offered the most generous advice: that I could quit the PhD to work in industry, and return whenever I felt ready to continue to focus on research. I will never forget his words: "These are difficult decisions, right? And so I think, you know, think through it and decide what you want to do. And I think the information that I've given you sort of gives you some of the details and risks. I think that the good news, too, Caesar, is that if you start a PhD, there's not a requirement to finish a PhD ... That looks worse for me, but it's better for you. So, it looks bad for me if I have students drop out of the PhD program, but don't worry about me. I'm old and ..." I was fortunate enough to record this conversation, and it has always reminded me—someone who usually tries to be strong and cheerful—that I still have the ability to cry. Ed sadly passed away on March 21, 2022, at the age of 49, but his influence on me is everlasting. If I have one regret from my PhD, it is that I never had the chance to take a photo with him. I am also truly grateful to Jian and Keiko, who have continually supported my PhD journey with invaluable guidance and insightful suggestions. This thesis could not have been completed without your mentorship.

I would also like to thank my committee members—Barrett Ens, Daniel Vogel, Mark Hancock, and James Wallace—for serving on my committee and approving my thesis. Defending a dissertation is never an easy task, but your thoughtful questions and feedback not only challenged me to articulate my work more clearly, but also affirmed that the research is of high quality and worthy of the title of Doctor.

I would also like to thank the URAs, Jiawen Zhu and Paul Kokhanov, who worked with me and contributed to my projects through your hard work. I hope that I was able to provide guidance to you in return and enrich your academic journey.

To advisors and friends in HCI lab, Xuye Liu, Qing Liu, Xizi Wang, Xinyu Shi, Yuzhe You, Yue Lyu, Ryan Yen, Shaikh Shawon Arefin Shimon, Liwei Wu, Ashu Adhikari, Felicia Feng, Linda Zeng, Kimi Zhong, Xingjun Li, Xuejun Du, Yuan Chen, Anastasia Kuzminykh, Bahareh Sarrafzadeh, Rina Wehbe, Jay Henderson, Jeff Avery, Damien Masson, Hanae Rateau, Nikhita Joshi, Johann Wentzel, Matthew Lakier, Ludwig Wall, Antony Irudayaraj, Nabil Hannan, Sangho Suh, Ken Jen Lee, Hemant Surale, Ana Crisan, thank you for your all help, talks and discussions. Best wishes to you all.

I am also grateful to Jialin Lu and Jingjie Zheng. Jialin was the first person to plant the idea of pursuing a PhD in my mind. I still remember being in the design classroom when you mentioned that you wanted to pursue a PhD. Now, here I am writing my PhD thesis, while Jialin has been thriving in industry for four years after completing his master's program. Jingjie's work on Finger-Aware Shortcut was one of the first HCI papers I ever read. I later had the opportunity to correspond with you via email, and your guidance helped me get several RA positions, which marked the beginning of my research journey. After I came to Canada, you welcomed me like a big brother, supporting me through difficult times and making me feel at home. Let's jam and take pictures more in the future!

I would also like to thank my friends in the UW Chinese Orchestra Club. Music has always been an important part of my life, and I am grateful to all of you for sharing the opportunity to perform with me. It was a great pleasure to rehearse and play music together. I still remember the first time I went on stage—my legs jittered like two drumsticks! That performance was a success and gave me a tremendous boost of confidence in expressing myself.

To Ben and Nini, my first clients in commercial photography: during a difficult time for me, planning Ben's proposal and capturing your precious moments truly lifted my spirits and kept me happy for a long time. It also gave me the confidence that I could make a living through photography if I don't have a job. Now that you are happily married, I wish you both all the best.

To my colleagues from industry, Swathiha, Shree, Philip, Daniel, Dean, Aryan, Milan, Timothy, Mahmood, Celine, Suyash, Jeff, Tommy, and Kyle. Great time working with you!

Thank my teachers and friends, Yusi Liu, Ruiming Zhou, Kai Wang. Now we are all Doctors! But I'm not applying for professor.

To my friends in person and online. Xiuyu, Shuai, Hani, Si, Bao, Degui, Mingliang, Gongzai, Gabriella, Mei, Kexin, Xu, Zhichao, Hongliang, Jiacy, Zhihao, Di, Guanlin, Haoqi, Ke, Shengwei, Xiaoyang, Jerry, Lucas, Xiaotao. Thank you all for accompanying and help. I am sure I am forgetting many friends who supported me and kept me motivated during difficult times, and I sincerely apologize for any omissions.

To LLM tools, ChatGPT, Grammarly, and Writeful for correcting my grammar and making me look like a professional.

I thank my friend Yuming Zhao and Yue Xu. I don't know where to start after all these years, but I hope our friendship will never end.

Finally, I want to thank my parents for their unconditional trust, support, and love. Mom, I'll still drink Coke Zero; Dad, I'm not selling my options in any company starting with "R"—that's my option!

Dedicate to you.
Just enjoy the performance!

TABLE OF CONTENTS

	List of Figures	xiv
	List of Tables	xvi
1	Introduction	1
1.1	Research Goals	3
1.1.1	Main Research Objective	3
1.1.2	Project Description and Research Questions	3
1.2	Thesis Outline	5
2	Related Work	6
2.1	Bimanual Interaction	6
2.2	Concepts of Interaction in HCI	7
2.3	Pointing	8
2.3.1	Virtual Hand	9
2.3.2	Virtual Pointing	9
2.4	VR Locomotion Techniques	10
2.4.1	Continuous Locomotion Techniques	10
2.4.2	Teleportation	11
2.4.3	VR Locomotion in 3D	11
2.5	Command Access	12
2.5.1	2D Marking Menus	12
2.6	VR Command Selection	13
2.7	Summary	14
3	Conductor	16
3.1	Introduction	17
3.2	Background and Related Work	18
3.2.1	Occluded Objects Selection Techniques	19
3.3	Intersection-Based Selection	19
3.4	Ray + Cursor	20
3.5	Conductor	20
3.5.1	Conductor versus iSith	22
3.6	Contrasting RayCursor and Conductor	23
3.6.1	Participants	23
3.6.2	Apparatus	24
3.6.3	Methodology	24

3.6.4	Performance	26
3.7	Discussion	30
3.8	Future Work and Limitations	31
3.9	Conclusion	33
4	Fly The Moon To Me	34
4.1	Introduction	34
4.2	Background and Related Work	37
4.2.1	3D Placement	37
4.3	Locomoontion Technique	39
4.4	Experiment	44
4.4.1	Experimental Task	44
4.4.2	Baseline: Point&Teleport and Bimanual Point&Tug	46
4.4.3	Participants and Apparatus	46
4.4.4	Design	46
4.4.5	Procedure	47
4.5	Performance	47
4.5.1	Task Completion Time	48
4.5.2	Task Completion Time Over Repetitions	49
4.5.3	Questionnaire Results	50
4.5.4	Interview Results	51
4.6	Discussion	52
4.6.1	Task Time Over All Conditions.	52
4.6.2	Task Time Over Repetitions.	52
4.6.3	Fine Tuning	53
4.6.4	Tutorial for Locomoontion.	54
4.7	Locomotion, or Docking Task?	54
4.8	Limitations And Future Work	55
4.8.1	Locomoontion Technique	55
4.8.2	Study Design	56
4.9	Conclusion	58
5	Drum Menu	59
5.1	Introduction	59
5.2	Background and Related Work	63
5.2.1	Rotational Command Selection	63
5.3	Drum Menu	63
5.3.1	Input Modality in VR Controllers	64
5.3.2	User Interactions with Different Drum Menu Variants	64

5.3.3	Novice and Expert Modes for Bimanual Drum Menus	66
5.4	Experiment	66
5.4.1	Participants and Apparatus	66
5.4.2	Procedure	67
5.4.3	Experimental Design	68
5.5	Performance	69
5.5.1	Task Completion Time	70
5.5.2	Error Rate	71
5.5.3	Total Hand Movement Distance	71
5.5.4	Total Hand Rotation Angle	73
5.5.5	Asynchronize Time	74
5.5.6	NASA-TLX Questionnaire	75
5.5.7	Preference Ranking	76
5.6	General Discussion	76
5.6.1	Design Implications	76
5.6.2	Limitations and Future Work	79
5.7	Conclusion	80
6	Conclusion	81
6.1	When should we consider bimanual input?	81
6.2	Future Research	81
6.2.1	Improve Conductor Performance Under Long Distance	82
6.2.2	Involve more body parts	82
6.2.3	Involve fewer body parts	83
6.2.4	Bimanual Haptics	83
6.3	Final Word	84
7	Encore: Why Concert?	86
	Bibliography	87

LIST OF FIGURES

Figure 1.1	Outline of how three projects contribute to the re- search goal. 4
Figure 3.1	Conductor introduction. 16
Figure 3.2	Conductor procedure. 20
Figure 3.3	Task completion time for target selection task. Error bars are 95% CI. 27
Figure 3.4	Task completion time for different target sizes and techniques. Error bars are 95% CI. 28
Figure 3.5	Error rate for target selection task. Error bars are 95% CI. 28
Figure 3.6	Questionnaire results. 29
Figure 4.1	Workflow of Locomoontion. 35
Figure 4.2	Preview rotation algorithm. 39
Figure 4.3	Cancel algorithm. 40
Figure 4.4	The settings of the experiment. 41
Figure 4.5	The implementation of the Baseline technique. 42
Figure 4.6	Screenshot of Locomoontion technique. 44
Figure 4.7	Task completion time for locomotion task. Error bars are 95% CI. 48
Figure 4.8	Task completion time for different heights. Error bars are 95% CI. 49
Figure 4.9	Task completion time for different distances. Error bars are 95% CI. 49
Figure 4.10	Task completion time over repetitions 50
Figure 4.11	NASA-TLX results on a 21-point scale (the lower is better). 51
Figure 4.12	Task rotation detail. 54
Figure 5.1	Drum Menu introduction. 60
Figure 5.2	Workflows of the three Drum Menus. 65
Figure 5.3	8-item layout for Drum Menus. 67
Figure 5.4	Task workflow: a) Two arrow appear in mid-air; b) Participant selects target item; (c) Selection confirma- tion. 68

Figure 5.5	Task completion time for 4-item layout.	70
Figure 5.6	Task completion time for 8-item layout.	70
Figure 5.7	Error rate for 4-item layout.	71
Figure 5.8	Error rate for 8-item layout.	72
Figure 5.9	Total Hand Movement Distance for 4-item layout.	72
Figure 5.10	Total Hand Movement Distance for 8-item layout.	73
Figure 5.11	Total Hand Rotation Angle for 4-item layout.	73
Figure 5.12	Total Hand Rotation Angle for 8-item layout.	74
Figure 5.13	Asynchronize Time for 8-item layout.	74
Figure 5.14	NASA-TLX Results.	75
Figure 6.1	Remapping the plane.	82
Figure 7.1	Concert.	86

LIST OF TABLES

Table 4.1	Task time (ms) for different height conditions	49
Table 4.2	Task completion time (ms) for different distance conditions	50
Table 5.1	User preferences for different techniques under NumberOfHand and Layout.	76

Technology alone is not enough. It's technology married with liberal arts,
married with the humanities, that yields us the result that makes our hearts
sing.

– Steve Jobs

INTRODUCTION

Have you ever noticed that behind each beautiful piece of music, almost all instruments require the coordinated use of both hands to perform complex compositions? These movements, which involve the harmonious collaboration of both hands, are referred to as bimanual interactions. In contrast to unimanual input [74], where only one hand is active, bimanual input introduces an additional dimension of control within the same timeframe. This extra dimension leverages the full potential of our natural dexterity, enabling users to intuitively perform tasks with two simultaneous dimensions of input [51, 149]. Apart from musical instruments, another common example of bimanual input improving efficiency is bimanual text input. Typing with two hands on a physical keyboard is the standard method for fast typing, and much research has been dedicated to adapting this technique to other input devices, such as touchscreens or even fingertip-based interfaces [11, 84, 98, 143], in order to improve the typing speed. As a result, bimanual interaction has significant potential to enhance efficiency.

VR has emerged as a transformative technology, offering new ways to interact with digital environments. A key advantage of VR is its ability to provide users with true depth perception and spatial awareness through stereo vision, which creates a realistic sense of three-dimensional space. Combined with immersive engagement—where users are fully surrounded by the virtual environment and can interact naturally—VR delivers a highly compelling and presence-rich experience. As the field grows, the focus has moved from just improving visual quality to creating a more complete and deeply immersive experience for users. For example, instead of creating 3D models on a 2D screen, the user can create them in VR in a more immersive and intuitive way. To achieve that, some interactions are required in VR, like **3D pointing** [7, 8, 60, 76, 97, 113, 114, 146, 148] to point at an object, **locomotion** [13, 69, 86, 120, 122, 138] to navigate in the 3D work space, and **command access** [19, 20, 47, 48, 68, 85, 110, 140, 154] to perform necessary commands. Interaction can be defined in various ways [52], such as Dialogue, Transmission, Tool Use, and other forms. In this thesis, small interactions that serve a simple goal are referred to as elementary interactions. However, such tasks can be

time-consuming in VR environments [3], partly because they require users to perform multiple sequential steps — that is, completing one step only after finishing the previous one. For example, when selecting a two-level menu with one hand, the user must first select an item in the first level, and only then select the corresponding item in the second level. By contrast, if the interface allows parallel execution, different steps can be carried out at the same time. For instance, the user could use one hand to select the first-level item while the other hand simultaneously selects the second-level item. This raises the question: Can these sequential steps be expedited by performing them in parallel using both hands?

The central goal of this thesis is to answer that question by designing new interaction techniques that enable users to perform two steps simultaneously using both hands, rather than sequentially with the dominant hand. By leveraging bimanual input, these techniques aim to reduce task time and enhance efficiency, offering a more seamless and productive experience in virtual environments.

A large body of work in VR has focused on unimanual interaction, often leaving the other hand unused. For instance, in **pointing**, single ray casting is the most popular solution for VR selection; however, it can be challenging when objects are occluded, as users must perform a second step to confirm which object on the ray should be selected [7, 109, 146, 148]. Another example is VR locomotion, where users navigate a virtual environment. Traditional **teleportation techniques** allow users to change their viewpoint in 3D space by specifying a target location, but they often lack height adjustment. This can result in time-consuming remedial interactions to reach the desired position. **Command access** in VR is also commonly implemented using unimanual techniques. Prior work has explored hierarchical menus with different layouts, such as linear panels [91] and pie menus [39, 91]. Marking menus, a command selection technique, have also been investigated in VR and shown to hold great potential as a shortcut mechanism [136]. Although these techniques enable essential VR interactions, they are still predominantly unimanual, leaving the other hand idle. This reliance on sequential input often increases task time and reduces efficiency. This motivates the focus of this thesis on bimanual interaction in VR.

This thesis investigated how bimanual interaction in VR can enhance user efficiency by augmenting dominant-hand interaction with simultaneous non-dominant hand input. The work focused on three primary objectives. First, I introduced Conductor, a novel bimanual interaction technique designed to

improve pointing, an essential task in VR environments. Second, I applied Conductor to the task of locomotion, exploring how bimanual input could facilitate more intuitive navigation in virtual spaces. Third, I examined the role of bimanual input in improving command selection tasks through the use of shortcut mechanisms. Through the development and evaluation of these techniques, this research not only demonstrated the potential of bimanual input to enhance VR efficiency but also contributed to a broader understanding of its application in immersive environments.

1.1 RESEARCH GOALS

1.1.1 *Main Research Objective*

At a high level, the research objective of this thesis is: **Explore the potential of simultaneous bimanual interaction in VR to improve user efficiency of elementary interactions achieved sequentially by one hand.** Through my work, I contribute the following:

- an understanding of the advantage of bimanual interaction and the possibility of incorporating it into VR tasks;
- designs and techniques that take advantage of bimanual input to create more efficient and smooth VR experiences.
- quantitative and qualitative findings to shed light on user experiences and inform future exploratory design endeavours.

1.1.2 *Project Description and Research Questions*

I completed three research projects to investigate different types of abstract ideas and explore how to translate them into concrete design alternatives. I briefly describe them below, and the research questions they aim to address.

CONDUCTOR - 3D POINTING In [Chapter 3](#), I explore **how to enhance 3D pointing technique in VR with bimanual interaction**. Conductor is an intuitive, plane-ray, intersection-based, 3D pointing technique where users leverage bimanual input to control a ray and an intersecting plane. Conductor allows users to use the non-dominant hand to adjust the cursor distance on the ray with a plane to "cut" the ray while pointing with the dominant hand

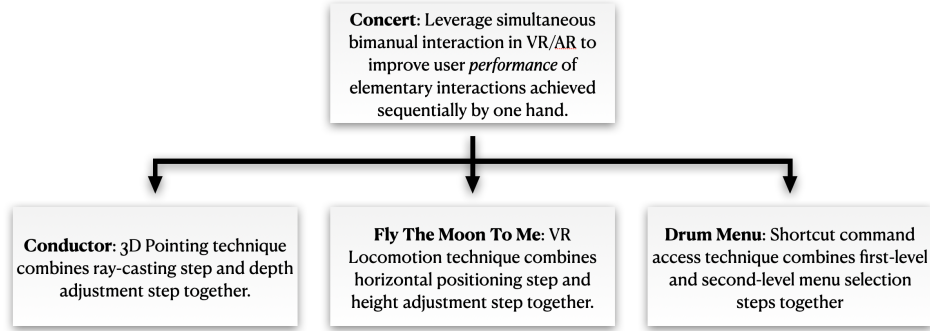


Figure 1.1: Outline of how three projects contribute to the research goal.

at the same time. **Conductor achieved the research goal by combining the ray-casting step and depth adjustment step together.** It also explores two modes, manual mode and semi-automatic mode, and provides an empirical evaluation of those two techniques. Conductor provides a fundamental mechanism of rapid pointing in 3D space using two hands, which could be applied to another project in this thesis.

FLY THE MOON TO ME - 3D LOCOMOTION 3D locomotion in VR often requires users to first navigate to a horizontal position and then adjust their height. With the introduction of Conductor, which enables rapid pointing in 3D space, can Conductor be leveraged to facilitate teleportation in 3D environments? In [Chapter 4](#), Fly the Moon to Me is proposed to answer the second research question: **how to accelerate 3D locomotion with bimanual interaction?** Instead of moving the user like most of the locomotion techniques, Fly the Moon to Me technique moves the object users want to interact with. It empowers users to create a preview copy of the object that they want to approach, manipulate it to any ideal 3D position and direction to the user using Conductor, and then seamlessly snap the original object to the preview, along with the rest of the virtual world. **Fly the Moon to Me fulfills the research goal by combining the horizontal positioning step and height adjustment step together.**

The title, *Fly the Moon to Me*, echoes the jazz standard *Fly Me to the Moon* [53]—a nod to its timeless elegance and sense of reaching beyond.

DRUM MENU - VR COMMAND ACCESS Current VR Head-Mounted Displays (HMDs) provide a few shortcut mechanisms for fast command access to control applications and environments. Command access in a menu typically requires careful visual targeting for multiple depths, which can be distracting and slow when immersed in gaming or trying to achieve efficiency on VR

HMD devices. Considering this motivation, my last research question is: **How can bimanual input accelerate command access in VR?** In [Chapter 5](#), I present Drum Menu, a bimanual, controller-based shortcut technique derived from the idea of traditional pie menus, featuring three input methods, designed for 4-item and 8-item layouts. Users can select commands by rotating the joystick, drawing a stroke, or pointing in different directions. Users first perform the selection on the non-dominant controller to select the first-level item, then perform on the dominant controller to select the second-level item. With enough rehearsal, users can enter the expert mode by making the selection on both controllers to access two levels at the same time. **Drum Menu contributes to the research goal by combining 2-level menu selection steps together.**

This thesis argues that augmenting elementary VR interactions with bimanual input can significantly reduce task completion time compared to traditional one-handed sequential interaction.

1.2 THESIS OUTLINE

The remainder of this thesis is organized as follows.

In [Chapter 2](#), I provide an overview of related work on 3D interactions, including 3D pointing, VR locomotion, and command access techniques in VR and other computing devices, as well as bimanual interaction studies.

In [Chapter 3](#), I describe the methodology and results for Conductor, a bimanual 3D pointing technique, and the design guidelines.

In [Chapter 4](#), I apply Conductor technique to locomotion, and describe the methodology and results for Fly The Moon To Me, a bimanual VR locomotion technique, with the interaction design guidelines.

In [Chapter 5](#) I describe the methodology and results for Drum Menu, a bimanual VR command access technique designed for VR controller. Three variants of Drum Menu are developed and compared with the unimanual variants. I also provide the interaction design guidelines for VR command access.

In [Chapter 6](#), I draw conclusions, summarize limitations, and suggest possible future work.

RELATED WORK

In this chapter, I review research that informs the design of my work on extending elementary VR interactions with bimanual input. Elementary tasks such as pointing, locomotion, and command access serve as the building blocks of complex VR experiences. To motivate and situate my contributions, I first introduce key concepts in bimanual interaction, which provide a theoretical basis for why two-handed input can be effective. I then examine existing approaches to pointing, locomotion, and command access in VR, highlighting both their strengths and limitations. Throughout, I emphasize how these techniques relate to the opportunities for enhancing VR interaction through bimanual designs.

2.1 BIMANUAL INTERACTION

Bimanual interaction in HCI has a fairly long history. As early as 1963, Sutherland developed Sketchpad [127], a drawing application that allowed users to press buttons with their non-dominant hand while drawing with a stylus using their dominant hand. Bimanual interaction can be divided into two categories: asymmetric bimanual interaction [44] and symmetric bimanual interaction [72].

Asymmetric bimanual interaction, described by Guiard’s kinematic chain model [44], refers to the two hands performing different but complementary roles. One hand often plays a dominant (precise) role, while the other acts in a supportive (coarse or guiding) role. For example, recent work on asymmetric bimanual gestural interaction leverages left-hand gestures to determine the input mode and right-hand actions to point or provide input [75]. Hinckley et al. [50] also found that performance was best when the non-dominant hand held an object and the dominant hand operated the tool. This provides further evidence that setting the context with the non-dominant hand and performing the action with the dominant hand is more effective than the reverse. Leganchuk et al. [74] discovered that bimanual input offers benefits such as improved time-motion efficiency due to increased degrees of freedom (DOF) and reduced cognitive load, compared to traditional unimanual

methods. When assigning tasks to both hands, functions requiring coarse or orienting actions are typically better suited to the non-dominant hand, while those requiring precision or fine control are better assigned to the dominant hand.

As for symmetric bimanual interaction, Latulipe proposed the Symmetric Interaction Model [72], in which both hands perform similar or identical roles, often in a coordinated or mirrored fashion. Typing on a keyboard is one example of this interaction style. Another common example is playing a musical instrument like the piano, where both hands contribute equally but may play different notes or rhythms in a coordinated manner. Similarly, using two joysticks to control a bimanual robotic system or operating flight simulator controls also exemplifies symmetric interaction, as both hands perform the same type of task in parallel. In digital sculpting or 3D modeling environments, users may use two styluses or controllers—one in each hand—to manipulate virtual objects simultaneously, performing sculpting or editing actions from different angles. In multi-touch interfaces, tasks such as zooming, rotating, or panning images using both hands involve coordinated, often symmetric, interactions. Even collaborative drawing tasks, where each hand operates a different brush or tool with equal importance, reinforce the principle of symmetric interaction—particularly when there is no fixed dominant or supportive role and the user switches fluidly between hands. These examples highlight that symmetric bimanual interaction emphasizes functional equivalence and temporal coordination, rather than role differentiation, and supports a more flexible and interchangeable use of the hands.

Together, this body of work establishes that bimanual interaction can increase efficiency and reduce cognitive load. However, relatively little research has examined how these benefits translate to VR’s elementary tasks. This gap motivates my investigation of bimanual input in pointing, locomotion, and command access.

2.2 CONCEPTS OF INTERACTION IN HCI

Hornbæk and Oulasvirta’s paper “What Is Interaction?” [52] provides a structured overview of prominent concepts of interaction in Human-Computer Interaction (HCI) research. The authors argue that interaction can be understood in multiple ways, each offering a distinct lens for studying and designing human-computer systems. They identify several interaction types:

- **Interaction as Dialogue:** Interaction is viewed as a cyclical communication process, where users and systems exchange information through turns or stages. This emphasizes mapping user intentions to system actions and feedback for natural, understandable exchanges.
- **Interaction as Transmission:** Here, interaction is treated as the transfer of messages over a channel, focusing on throughput, reliability, and efficient encoding and decoding of information between humans and computers.
- **Interaction as Tool Use:** Interaction is understood as using computers as instruments to accomplish tasks, emphasizing the manipulation of tools from a third-person perspective.
- **Interaction as Optimal Behavior:** This perspective sees interaction as users performing optimally within a system to achieve desired outcomes, such as minimizing errors or maximizing efficiency.
- **Interaction as Embodiment:** Emphasizing first-person experience, interaction emerges from the intentional coupling of user and environment, highlighting the influence of context and bodily engagement on how interaction unfolds.
- **Interaction as Experience:** This type centers on the user's subjective experience, including expectations, moment-to-moment feelings, and reflection, framing experience as central to interaction rather than a side effect.
- **Interaction as Control:** Interaction is framed as managing or controlling the system to achieve goals, focusing on cause-effect relationships and feedback mechanisms.

This thesis primarily adopts the dialogue perspective on interaction. Elementary interactions can be seen as short dialogues, or brief exchanges, with the computing system. Pointing, locomotion, and command access each reflect the user's intentions in these interactions. In contrast, tasks that build upon these elementary interactions and require extended engagement—such as constructing a 3D model or completing a game level—are considered complex tasks.

2.3 POINTING

Having introduced the basic models of bimanual interaction and the concept of interaction, I now turn to the elementary VR tasks where these models

may be applied. The first of these is pointing, a fundamental mechanism for selecting, manipulating, and placing objects. This section reviews 3D pointing techniques in VR, covering Virtual Hand, Virtual Pointing, occluded object selection methods, and 3D object placement.

2.3.1 *Virtual Hand*

In their survey of virtual and augmented reality-based pointing techniques, Argelaguet and Andujar [3] divided selection metaphors into two broad categories: virtual hand and virtual pointing. The virtual hand metaphor [90] uses a projected hand, typically located near the user, to support 3D object selection.

The basic virtual hand technique is relatively straightforward to implement [90]. As in real-world interactions, users can target objects that are within arm's reach; for more distant objects, users must first move near the objects before being able to select them. To address this shortcoming, techniques such as the Go-Go technique [104] combine some form of a virtual hand with an extendable arm interaction. Specifically, within a restricted range, users leverage the virtual hand technique. However, when the hand moves further than a cutoff from the body, the hand's depth, or distance from the user, is magnified via a CD Gain function to allow users to acquire more distant targets. Ninja Hands [112], a competing technique to Go-Go, leverages multiple virtual hands in VR and maps their movements to the real hand to reduce the distance to the target. This is achieved by placing these virtual hands at different locations in the virtual environment.

2.3.2 *Virtual Pointing*

In contrast to the virtual hand technique, raycasting has been the focus of significant research regarding improved performance [3], primarily due to a desire to select distant objects in virtual environments. One challenge with selection in 3D virtual environments is that objects near the user appear larger (due to perspective) and can occlude objects of interest that are "behind" nearby objects. To address occlusion, techniques such as Outline Pursuits [115] enable users to trace a moving stimulus on the outline of objects with eye gaze to select occluded objects. Some techniques utilized a two-step selection for occluded objects by spreading them into a menu and then

select, like "Flower Ray" [42] and "SQUAD" [67]. Similarly, Gyrowand [49] utilizes extra gestures to choose the targets. In their work on volumetric displays, Grossman and Balakrishnan [42] proposed four advanced raycasting techniques that enable users to select occluded objects on the same ray. One of their techniques, depthray, mimics the Go-Go technique in that users move the controller either toward or away from themselves to select objects along the ray. Finally, RayCursor [7], a recently developed pointing technique, presents significant performance benefits when targeting densely populated virtual environments. It achieves this by combining the Raycasting metaphor with depth control (as in Grossman and Balakrishnan [42]) via the touchpad located on some commercial VR controllers. Other than leveraging the cursor, previous works also explored other metaphors i.e., cone-like ray [49, 79] to improve larger range selection, or using Bubble Cursor [41] mechanism to facilitate VR target selection [83].

My work in Conductor project explores bimanual selection. I am not the first to explore this domain: different symmetric and asymmetric bimanual selection techniques have been proposed for volumetric displays, in particular, work by Ulinski et al. [130, 131]. However, Ulinski et al.'s research identified best practices in the design of bimanual selection techniques, yet the performance of these techniques in virtual environments in comparison to optimal unimanual techniques was not assessed.

2.4 VR LOCOMOTION TECHNIQUES

Beyond selection and placement, VR interaction also depends on effective locomotion, which can be understood as placing oneself within the environment. Reviewing locomotion techniques allows us to see how existing designs might be extended or rethought through bimanual control. The following subsections review representative continuous, discrete, and 3D-specific locomotion techniques.

2.4.1 *Continuous Locomotion Techniques*

A large body of VR Locomotion techniques has been explored in the past. Walking, one of the most intuitive ways to travel, provides a continuous way to navigate in VR. Some walking techniques allow users to walk in the real world but with modifications, such as rescaling the walking space [1] or altering

the direction mapping between real and virtual world [4, 71, 96, 107]. While those techniques reduce the necessary space, they still demand unobstructed areas. To enable stationary locomotion in VR, in-place locomotion techniques were developed. One of the most significant issues with in-place locomotion techniques is motion sickness [10, 55, 62] resulting from the visual and vestibular mismatch when the user remains still while the VR experience shows they are moving rapidly through a virtual environment. To mitigate motion sickness, some techniques incorporate physical movements during locomotion to compel the mind that the body is in motion. Walking-In-Place [128] is one of the most popular solutions for this design. Devices like treadmills [32, 54, 118] or cycling [35, 93] were also developed for simulating walking. Other body movements like arm rolling [87], jump [141], finger simulating walking [63, 145], or swimming [57] were also introduced for the compelling purpose.

2.4.2 Teleportation

In contrast to continuous locomotion techniques, teleportation [13], a discrete method of movement in VR, typically allows users to specify a point and then instantly move the user there. Certain teleportation methods allow users to indicate the desired direction [13, 37] after teleportation during the process of selecting the teleportation destination. Apart from being triggered by controllers, other body parts like eye [106], foot [139], and wrist [18] were created for teleportation as well as for bare hand VR teleportation. Different proxies like gate [55] or portal [34] were also introduced for teleportation. Buttussi et al. [14] compared teleportation, joystick, and leaning-based locomotion techniques in a VR travel task. The result revealed that joystick and leaning-based techniques elicited similar levels of VR motion sickness, while teleportation techniques had no significant impact on the participants' well-being. In addition to that, Frommel et al. [36] also found that teleportation provided the least discomfort and the highest scores for enjoyment, presence, and affective state.

2.4.3 VR Locomotion in 3D

Certain methods were introduced that enable users to thoroughly navigate through 3D virtual environments by incorporating height adjustments during

locomotion. Skyport [86] controls a cursor on a ray shooting from the controller, using a touchpad to control the distance of the cursor, and the user would then be teleported to the cursor’s position. However, it cannot control the direction after teleportation, and the precision is limited. Weissker et al. [135] created a preview of the users’ avatar and moved it by specifying the 2D position and elevation above the position using the controller. Specifying the 2D position and elevation could be done simultaneously, two-step, or separately, with their own advantages and disadvantages. However, the final view remains unknown, and the direction is not supported. Point-Tug [12, 24, 80] techniques, inspired by ladder-climbing motion, allow the user to grab the world and tug to move the world. Originally, it would support the x-z plane but not vertically in the y-axis. In the experiment of Fly the Moon to Me, I extended it to 3D and treated it as a refining step for the Point&Teleport technique, and combined them, comparing with Locomotion. Some approaches also explored how to use existing tools for 3D movement, like ladders [58, 70, 119], virtual stairs [70], ramps [29], elevators [132]. Actions like jumping [141], swimming [57], and flying [88, 116, 151] were introduced as well, with the benefits of reducing VR sickness. However, because most techniques require separate actions for height adjustment, they disrupt the continuity of movement and increase cognitive load. This limitation points to an opportunity for more fluid designs—such as leveraging bimanual input—to embed altitude control directly into continuous locomotion.

2.5 COMMAND ACCESS

In addition to moving and selecting objects, VR experiences often require accessing commands. Command access provides a third pillar of elementary interaction, and examining both 2D and VR-specific approaches reveals design strategies and gaps that motivate our own techniques. The following subsections review 2D marking menus, VR-specific command selection, and hierarchical menu designs to contextualize this opportunity.

2.5.1 2D Marking Menus

Marking menu is a radial menu interface that enables users to select commands by making quick gestures or marks. It supports both novice mode and expert mode. Novice mode requires a visual search to locate the item

and then select it with a gesture or stroke, which can be time-consuming and disruptive. In expert mode, selection could be done instantly without visual search [6, 68, 152, 153]. In these marking menus, the user starts in expert mode without a visible menu, but can easily switch to novice mode by displaying a GUI menu. When the user places a stylus or pointer in the input space, a radial menu appears, allowing them to select an item by stroking in the corresponding direction (novice mode). As a result, it enables smooth transitions from novice to expert usage. Kin et al. developed a bimanual marking menu technique for touch screen, which reveals the bimanual variant is 10% ~ 15% faster than the unimanual marking menu [64]. However, how to integrate bimanual interaction in 3D virtual environment and the performance remains unknown. Shengdong et al. developed Simple Marking Menu that breaks multi-level gestures from one single “zig-zag” compound mark to several simple strokes [153]. It eliminated ambiguous marks due to scale invariance and reduced space requirement.

2.6 VR COMMAND SELECTION

Alongside direct manipulation of the environment, a common need in virtual environments involves selecting commands. In modern commercial systems, it is common for VR-based systems to support command selection by displaying a command palette in front of the user and allowing the user to target individual widgets. For example, tutorials on creating user interfaces using Unity leverage the UIHelper prefab to support ray casting to control pointer position, coupled with UI Canvases to render clickable widgets.

By far the most common command selection technique in commercial VR application design leverages the linear panel menus described in the aforementioned tutorial, but other styles of interaction have been explored over the past few decades [27, 66, 136]. Specifically considering pie menus, the menu design adopted by Drum Menu, there are several variants of hierarchical pie menus for 3D virtual environments [39, 65, 94, 105]. Azai et al. introduced an on-body hierarchical menu in mixed reality [5], where menus are projected on the forearm. They include a rotation-based interaction; however, their rotation-based interaction involves using one hand to rotate the menu on the other arm, as one would rotate a bracelet. Armstrong [78] also explored the menu around the non-dominant arm and how to optimize the layout of such arm-anchored UI. However, these on-body menus are not designed for shortcuts as they require visual engagement. Monteiro et al. explored

the performance of radial and panel menus with two placements (fixed on the wall or handheld) in VR and found that a traditional panel menu with a fixed wall placement performs better than others; their technique employed a round touchpad for radial selection [91]. Santos et al. discovered that the selection time on radial menus is faster than that of linear menus with a ray cast interaction, where one uses a directional mark (as in marking menus [68]) to access menu items. Wentzel et al. [136] conducted evaluations of marking menus in VR, comparing them against raycasting and direct input techniques. They studied both single-level menus (8 and 24 items) and hierarchical two-level menus, measuring task completion time, error rate, and usability. Their results showed that marking menus are fastest for small item sets, and hierarchical organization improves performance and reduces selection complexity, even with larger sets. While their work provides a solid foundation for understanding marking menu performance in VR, it does not investigate sequential bimanual hierarchical access, rotational wrist-based shortcuts, or joystick-based shortcuts, all of which are central to our study. StickyPie [2] is a marking menu technique that allows for scale-independent input by estimating the landing positions of saccades, enabling expert-level behaviour. However, it requires additional eye-tracking hardware, which is less popular and more expensive. While many such techniques exist, none have specifically explored the effectiveness of sequential bimanual hierarchically arranged and access commands as I do in our studies.

Although a variety of 2D and VR menu designs have been explored, they generally rely on unimanual or sequential interaction, which can slow down access during complex tasks. This limitation suggests an opportunity for bimanual techniques to enable faster, more parallel, and context-sensitive command access.

2.7 SUMMARY

In summary, the literature on VR interaction demonstrates a rich set of techniques for pointing, locomotion, and command access, but these are overwhelmingly designed for unimanual input. While foundational models of bimanual interaction suggest clear benefits in efficiency and cognitive load, few VR techniques systematically explore bimanual designs for these elementary tasks. My work addresses this gap through three projects: Conductor, which investigates bimanual selection; Fly the Moon to Me and Locomotion, which extend locomotion with two-handed interaction; and Drum Menu,

which explores bimanual command access. Together, these contributions build on the reviewed literature to propose and evaluate novel ways of augmenting elementary VR interactions through bimanual input.

CONDUCTOR

Pointing is an elementary interaction in virtual and augmented reality environments, and, to effectively support selection, techniques must deal with the challenges of occlusion and depth specification. Most of the previous techniques require two explicit steps to handle occlusion. In this Chapter, we propose Conductor, an intuitive, plane-ray, intersection-based, 3D pointing technique where users leverage bimanual input to control a ray and intersecting plane. Conductor allows users to use the non-dominant hand to adjust the cursor distance on the ray while pointing with the dominant hand. We evaluate Conductor against Raycursor, a state-of-the-art VR pointing technique, and show that Conductor outperforms Raycursor for selection tasks. Given our results, we argue that bimanual selection techniques merit additional exploration to support object selection and placement within virtual environments.

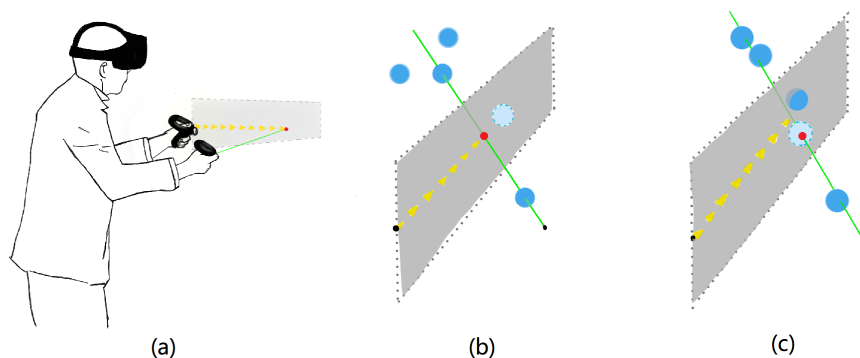


Figure 3.1: (a) Conductor uses a ray (green line) from the dominant hand and a virtual plane from the non-dominant hand to control a cursor. (b) Manual Conductor. Similar to Bubble Cursor, the target closest to the intersection of the virtual plane and the ray will be selected (light blue ball). (c) Semi-Automatic (SA) Conductor. Similar to RayCursor or Raycasting, only the target on the ray and closer to the intersection of the plane and the ray will be selected. Users could move the plane roughly close to the target together with the accurate ray pointing to quickly locate the target even with occlusion.

3.1 INTRODUCTION

As VR headsets increase in resolution and responsiveness, increasingly immersive visual experiences are available to a broad set of end-users. However, one challenge with VR is how best to interact within these environments [3]. Motivated by the increasing availability of cost-effective, high-performance VR headsets, there exists a large and ongoing body of research in interaction techniques to support target acquisition within virtual environments [8, 60, 76, 97, 113, 114, 146, 148].

Selection within VR environments is challenging because these environments, rather than being populated by a series of 2D targets as on a computer display, are populated by targets and objects of interest arranged in a 3D world where issues of depth variation and occlusion confound interaction - for instance, when nested components in a 3D CAD model obscure each other, or when avatars block props in a VR game. A significant area of research within these environments focuses, specifically, on how to support rapid, fluid targeting given partial [7] or full [101] occlusion within the 3D content contained within these spaces.

In this work, we evaluate bimanual selection in virtual environments. There are three primary premises that support bimanual targeting in virtual environments.

1. State-of-the-art AR and VR systems normally support bimanual tracking, either through bimanual controllers or some other capture mechanism;
2. When performing unimanual targeting in virtual environments, users' non-targeting hand is often not engaged in another task and so is frequently available to support targeting;
3. It is intuitive to combine actions of the non-dominant and dominant hands in real-world manipulations [43].

We acknowledge that there is a role for unimanual selection in virtual environments. In some situations, the user's non-dominant hand may be engaged in another task or the hardware may only support unimanual tracking (e.g., pointing with a personal device [60]). However, acknowledging that there is a role for unimanual targeting, we also posit that there *may* be a role for bimanual selection, but whether there is or not is unclear. More specifically, while many targeting techniques have been developed for VR environments, including both unimanual and bimanual techniques, it is not clear what the

relative performance of bimanual techniques, either existing or proposed, are compared to state-of-the-art unimanual targeting. Is there a role for bimanual targeting at all? Does it provide performance advantages over state-of-the-art unimanual techniques?

This project evaluates a bimanual selection technique, Conductor (Figure 3.1), that represents an extension of the RayCursor technique. In RayCursor, ray-based unimanual depth-aware pointing is supported via raycasting (tracking the pitch and yaw angles of a single controller), and a touchpad on the controller supports depth control along the ray. Conductor is a bimanual selection technique that combines raycasting (controlled with the yaw and pitch of the dominant hand) with depth selection (controlled via the yaw angle of the non-dominant hand) using an intersecting plane. It also describes an experiment that evaluates the performance of Conductor against the RayCursor technique [7], evaluating the speed of target acquisition within a VR environment. Conductor outperforms RayCursor, both in task completion time and in input accuracy. This argues for the viability of Conductor as a viable alternative 3D selection technique to RayCursor, focused particularly on contexts where bimanual input via dual controllers has been deemed a useful trade-off for enhanced targeting performance.

In terms of bimanual input theory, Conductor is an asymmetric technique: the dominant hand performs precise pointing, while the non-dominant hand provides coarse depth adjustment, consistent with Guiard’s kinematic chain model [44]. The two hands act largely in parallel rather than sequentially, with the non-dominant hand setting the selection plane while the dominant hand simultaneously refines the target.

The remainder of this chapter is organized as follows. We first review related work on selection in virtual and augmented reality environments. Next, we describe the design of Conductor and an evaluation of its efficacy for selection tasks. We conclude by discussing the implications of this work for interaction in virtual environments.

3.2 BACKGROUND AND RELATED WORK

In Chapter 2, we discussed related work on general VR pointing techniques and occluded VR pointing techniques. In this section, we present a deeper exploration of the background related to intersection-based selection and the Ray + Cursor technique, which forms the foundation of the Conductor technique.

3.2.1 Occluded Objects Selection Techniques

Selecting occluded objects is a challenging task in VR. Some techniques use a filter surface to divide objects and make some objects invisible to reduce occlusion [17, 30, 31, 101]. A typical approach is to use a surface and make objects closer to the user than the surface transparent, allowing the user to see through close objects to more distant objects. Other techniques allow users to look “around” objects; for example, VMirror allows users to create a mirror to find occluded objects [76]. Finally, techniques like Gimlenses enable users to look “through” occluded objects by digging holes in complex geometry [22, 102, 103]. Most of those techniques require two explicitly separated actions to access the object. Additionally, many of these techniques modify the way objects are displayed. One benefit of RayCursor is that the view of objects in the virtual environment is not transformed in any way.

3.3 INTERSECTION-BASED SELECTION

In heavily occluded 3D spaces, some mechanism is needed to support selection along varying depths. One such technique that has been proposed in the literature, and the technique that I leverage for Conductor, is a technique that I label intersection-based pointing. Within this domain, TiltCasting [101] enables intersection-based selection by rotating a virtual plane in a 3d scene and selecting the target intersecting the virtual plane. While TiltCasting effectively supports 3D targeting and placement, its primary evaluation was on traditional 2D displays, rather than in virtual environments and HMDs.

Specifically in the domain of virtual environments, and most similar to *Conductor* [149], *iSith* [142] is an intersection-based pointing technique that uses two rays emanating from the index fingers of each hand. The point that represents the shortest distance between the two rays is considered the current cursor location. Like *Conductor*, *iSith* shares the advantages of a nearly unlimited range of interaction and supports easy selection of occluded objects. However, one challenge with *iSith* is that the user must carefully align both rays; if the pitch axes of the two rays differ, the cursor can drift away from the visual intersection of the two rays, making control very difficult. As a result, while *iSith* has been proposed as an intersection-based, bimanual pointing technique in the literature, its performance against competing techniques has not, to the best of my knowledge, ever been evaluated [142].

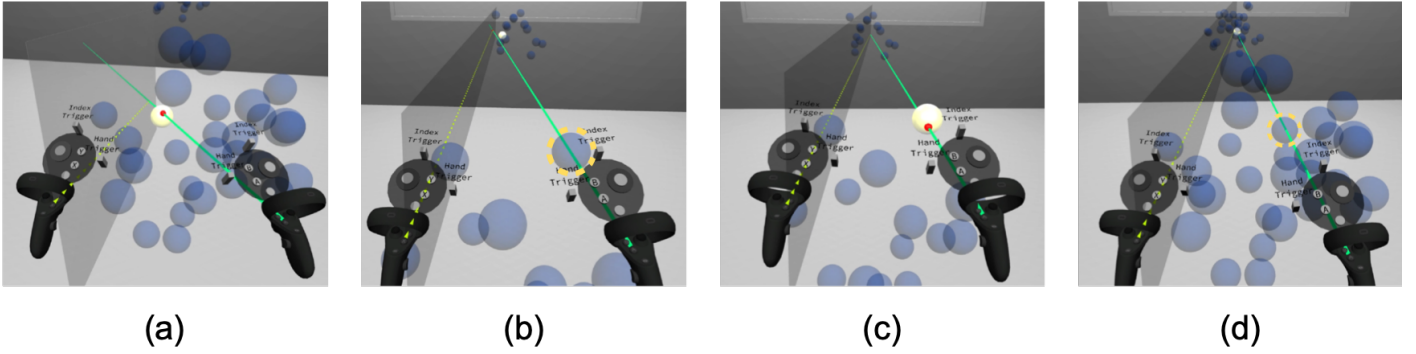


Figure 3.2: Target selection with Conductor. The grey plane is not visible to users. (a) Manual mode Conductor selects the target closest to the intersection of the ray and the plane. (b) Manual mode Conductor selects the target closest to the intersection, even if there is a target on the ray farther from the intersection (the target with a yellow dotted circle). (c) Semi-Auto (SA) mode Conductor always selects the target on the ray regardless of the distance from the intersection. (d) Semi-Auto mode Conductor selects the target on the ray closest to the plane.

3.4 RAY + CURSOR

In VR, many unimanual selection techniques have been proposed in the literature (see [3, 7, 42]). In Conductor, I focus on comparing with one specific recently proposed unimanual VR-based selection technique, RayCursor [7]. RayCursor is an improvement of Raycasting. Similar to Depth Cursor [42] and Ro et al. [109], it also attaches a cursor to the ray. RayCursor also examines three different transfer functions to map finger movement on the touchpad of a VR controller to the cursor movement along the ray. The result indicates that the best transfer function is based on the finger speed, with a maximum and minimum cutoff. RayCursor provides two modes: Manual RayCursor and Semi-Automatic RayCursor. Manual RayCursor allows users to move the cursor based on finger movement on the touchpad. Semi-Automatic RayCursor enhances Manual RayCursor by enabling the cursor to automatically move to the closest target on the ray, a variant of BubbleCursor [7, 83].

3.5 CONDUCTOR

Both Conductor and RayCursor can be viewed as techniques where a location along the ray is incorporated into pointing without modification to the scene. In RayCursor, the cursor location, i.e. the depth along the ray, is controlled via the touchpad on the controller being used for pointing. In Conductor, the cursor location is controlled via the yaw angle of the non-dominant controller.

Conceptually, we can describe Raycursor as an unimanual pointing technique where yaw and pitch represent direction, and the touchpad is used to manipulate depth. Similarly, Conductor is a plane-ray, intersection-based, bimanual pointing technique where the manipulation of the ray, as with RayCursor, is controlled by the dominant hand, and the intersecting plane, i.e. the depth, is controlled by the yaw angle of the non-dominant hand. Users direct the ray to point at a desired target and control the depth in RayCursor by manipulating the touchpad and in Conductor by moving the plane with the non-dominant hand via left-right movements, i.e., by adjusting the yaw angle of the non-dominant controller. The benefit of using a plane is that it is more intuitive compared with incrementally adjusting the cursor distance with a joystick or touchpad. The user can point to the target with the non-dominant hand while using the dominant hand's ray, rather than having to monitor the current cursor position and estimate the distance to the target. In the controller's local coordinate system, RayCursor allocates 3 DOF for position, 2 DOF for ray casting, and 1 DOF on touchpad for cursor distance. Conductor, by contrast, uses 3 DOF for position in both hands, 2 DOF for ray casting in the dominant hand, and 1 DOF for plane positioning in the non-dominant hand.

To visualize interaction, the plane can be shown:

- as an infinite vertical plane,
- as a triangular vertical plane whose point intersects the ray
- or as a line emanating from the non-dominant controller and intersecting the ray

The point along the ray, what RayCursor calls the “cursor location”, can be calculated using the distance between controllers and the angular deviation between the two controllers. Essentially, the distance calculation is performed in exactly the same way as binocular disparity is used by one's eyes to focus on different depths in a visible scene. In Figure 3.2, we show both the vertical plane (in shaded grey) and a yellow dashed line highlighting location along the ray. In default mode (and in our experiments) the dashed yellow line (see Figure 3.2) emanates from the non-dominant hand controller along the intersecting plane to the ray to provide a target location and the vertical plane is hidden. The vertical plane in Figure 3.2 is depicted for convenience here to aid the reader in visualizing Conductor.

As with RayCursor, Conductor exists in two variants: a manual variant and a Semi-Auto variant, distinguished as follows:

- *Manual Conductor*: Manual conductor selects the target closest to the intersection of the plane and the ray (Figure 3.2 (a)(b)). In the case of Figure 3.2 (b), there is a target on the ray close to the right controller. However, since there are targets closer to the intersection, the target closest to the intersection but not on the ray will be selected. Target selection is confirmed via the dominant hand trigger.
- *Semi-Auto Conductor*: Semi-Auto Conductor combines the benefits of Conductor and Raycasting. If any single target exists on the ray, that target will be selected (Figure 3.2 (c)). If multiple targets exist on the ray, the target closest to the intersection of plane and ray will be selected (Figure 3.2 (d)). Users can do precise Raycasting with their dominant hand controller and manipulate the non-dominant hand plane to approach the target at the same time. This strategy addresses issues of occlusion. Users can also short-circuit the Semi-Auto refinement and select the target closest to the intersection, as in Manual Conductor, via the non-dominant index trigger if necessary.

As well, as with Raycursor, to support smooth ray and plane movement, we used the 1€ Filter [15] with parameters set as $\text{mincutoff} = 0.1$ and $\beta = 50$ to smooth the ray for targeting. These parameters were identical to those used in RayCursor [7].

3.5.1 Conductor versus iSith

While, as we note in related work, Conductor has some similarities to the iSith technique [142], it is, in practice, quite different, and we clarify those differences here. Recall that the iSith technique uses two complementary rays controlled by two controllers and joins the closest points of two rays together with a line segment. It then places a selection point at the midpoint of this line segment. iSith can be visualized as a symmetric bimanual technique where the user manipulates two rays connected by a spring frictionlessly attached to both rays, and the user then places the midpoint of the spring on the target to be selected.

Exploring the research literature, it became clear that iSith, while proposed, had not been evaluated, so in our exploration of bimanual pointing techniques, we implemented iSith and piloted it. However, in our pilots, we observed that a minor change in controller angle, especially the pitch of one controller, could dramatically change the locations of the closest points on the two rays.

For example, imagine that to select a desired object, the user must raise the selection location upward while keeping the depth location (distance from the user) constant. If the user raises one ray, i.e. increases the pitch angle on only one controller, then the selection point moves in toward the user (or out away from the user) very rapidly rather than moving directly upward. To move the selection point upward, users must move both hands upward simultaneously. This is not too difficult if the initial pitch angle of both controllers is similar. However, if the pitch angles differ, it becomes increasingly complicated, and control of the selection location becomes unpredictable for the user.

Conductor, in contrast, uses a form of asymmetrical bimanual interaction influenced by Guiard’s kinematic chain model [44]. The non-dominant hand sets a spatial plane of reference for interaction along the ray. This reduces to 1 DOF the action of the non-dominant hand, which, in our pilots, increases the stability of the selection location. Also, and most importantly, it better allows users to predict the cursor position even when moving their two hands simultaneously.

3.6 CONTRASTING RAYCURSOR AND CONDUCTOR

The primary goal of RayCursor and Conductor is to address situations where multiple potential targets exist along a ray, i.e. when more distant targets are occluded. To accomplish this, both techniques augment the ray with a depth pointer. The primary difference is in how depth is controlled: via a touchpad unimanually, or via dual controllers and inter-controller angular deviation bimanually. Our experiment aims to compare the selection performance of RayCursor’s variants [7] to Conductor’s variants. We evaluate 4 techniques in this task: Manual RayCursor, Semi-Auto RayCursor, Manual Conductor, and Semi-Auto Conductor.

3.6.1 *Participants*

12 participants (10 male, 2 female) were recruited from the local university and the community. All were right-handed. Seven of them were run in person and five were run remotely. Nine of them experienced VR for the first time. Three of them are familiar with VR, including one is a VR researcher. Each participant received \$10 as remuneration.

3.6.2 Apparatus

We use a replication of the study conducted in Baloup et al. [7] with minor differences in hardware. The experiment used a PC Intel Core i7-9700K PC with NVIDIA RTX-2080 Ti GPU for in-person study and an Oculus Rift S VR headset. The application was programmed in the C# language with Unity3D Version 2019.4.7. Participants manipulated a pair of Oculus Touch controllers with a joystick on each. For the RayCursor techniques, users used one controller in their dominant hand. The users used the joystick on the controller to control the depth of the cursor along the ray. For the Conductor techniques, users controlled the position and rotation of two controllers together to manipulate the ray and plane. Target selection was performed by pressing the dominant controller's trigger using the index finger.

In the original RayCursor experiments, the VR controllers were equipped with touchpads. However, Baloup et al. [7] note that RayCursor performs well regardless of whether the controller includes a touchpad or joystick. Because the Oculus Touch controllers are equipped with joysticks instead of touchpads, we set new parameters for RayCursor as per Baloup et al. to fit the joystick: $k_1 = 0.6$, $k_2 = 3$, $v_1 = 0.3$, $v_2 = 0.9$. We preserved the scale of k_1/k_2 and v_1/v_2 in the original RayCursor paper to maintain the CD gain of the original implementation. We performed an initial pilot evaluation of this RayCursor variant to ensure consistent performance with the original RayCursor implementation prior to our experiments.

3.6.3 Methodology

We replicated the environment and task in RayCursor [7] with minor modifications to create additional occlusion.

The 4 techniques were balanced by a Latin square between subjects. Following the RayCursor task protocol, we also used 2 target SIZES and 2 target DENSITIES. The 2 target sizes were 8cm ($= S_{Big}$) and 4cm ($= S_{Small}$) in diameter, and the 2 target densities were 60 targets ($= D_{High}$) and 30 targets ($= D_{Low}$).

Identical to the RayCursor protocol, all the targets were distributed inside 2 spheres of 60cm diameter, randomly in front of the participant. The closer sphere was 1m away from the participant, and the farther was 4m.

In RayCursor’s experiments, occlusion was limited, i.e., as noted by Baloup et al., “All targets were at (*sic*) sight when the user was looking at the virtual wall with the virtual display” ([7] p. 9). While targets may have been occluded from the controller’s location, targets were always at least partially visible from the perspective of the participant. Our concern was that participants could raise or lower the controller to point around targets in RayCursor. We therefore modified the experiment to create greater target occlusion. Specifically, we:

- Raised the height of two centres of the target spheres from 80cm to 140cm so that the closer group of targets could create occlusion when selecting from the farther group of targets, especially for the condition with D_{High} and S_{Big} .
- Because more distant targets are more likely to be occluded by nearer targets, we chose the farthest nine targets in each group for users to select to build additional within-group occlusion.

One challenge with full occlusion is that participants cannot visually locate targets. This may be why RayCursor limited occlusion in their experiments. To ensure participants could visually locate targets in both RayCursor and Conductor conditions, all targets in both conditions were semi-transparent blue, and the target to be selected was solid yellow.

Participants were required to select the targets alternating between the closer group and the farther group. When the participant selected the correct target, the selected target would flash green and the controller(s) vibrated for 10ms. If an incorrect target was selected, the selected target would flash red and the controller(s) would vibrate for 200ms. The trial would be marked as an error, and the participant was required to reattempt selection until it was correct. There was a virtual scoreboard in front of the participant behind the targets indicating the current error rate, and the participants were asked to maintain an error rate below 4% to balance accuracy and speed.

Each participant watched a series of short videos introducing the technique before beginning each technique. They could experiment with the technique during the instruction videos and started one practice run with S_{Big} and D_{High} to familiarize themselves with the experiment.

They then started the experiment with 3 blocks of the 4 counterbalanced (via a Balanced Latin Square) techniques combining different sizes and targets ($D_{Low} + S_{Big}$, $D_{Low} + S_{Small}$, $D_{High} + S_{Big}$, $D_{High} + S_{Small}$). Participants were allowed to rest between runs. Each combination of technique $\times$ size $\times$ density

contained 9 trials. In summary, this yields $4 \text{ TECHNIQUES} \times 2 \text{ DENSITIES} \times 2 \text{ SIZES} \times 9 \text{ trials} \times 3 \text{ blocks} = 432 \text{ data points per participant}$, or, for 12 participants, 5184 data points for analysis.

After participants completed the selection task for each technique, the participants were asked to complete a computerized form of the NASA Task Load Index (NASA-TLX) questionnaire [95] and respond to four additional questions on similar 20-point scales. They completed the questionnaire using their headset and controller. The study took about 30 minutes.

3.6.3.1 Hypotheses

We wish to evaluate the performance of our bimanual Conductor technique against RayCursor. Therefore, our primary hypothesis is that Bimanual Conductor outperforms similar RayCursor variants in time and error rate. More specifically:

- H1: Semi-Auto Conductor outperforms semi-auto RayCursor in time and error.
- H2: Manual Conductor outperforms manual RayCursor in time and error.

Finally, leveraging NASA-TLX and custom questionnaire data, our hope is that overall workload measures or user preference are not significantly impacted by switching from an unimanual to a bimanual targeting technique.

3.6.4 Performance

To analyze the data, we used a repeated measures three-way ANOVA with TECHNIQUE, DENSITY, and SIZE as independent variables. Holm-Bonferroni corrected post hoc pairwise t-tests are used for further analysis. Normality was corrected using a Box-Cox transform. When sphericity was violated, degrees of freedom were corrected using Greenhouse-Geisser ($\epsilon < 0.75$).

3.6.4.1 Task Completion Time

We removed outlier data points with selection times more than 3 standard deviations from the task mean. This removed 74 data points (1.4%) in total. Because a Shapiro-Wilk normality test revealed normality on task completion time was violated ($p < .001$), a Box-Cox transformation was applied to time ($\lambda = -0.63$), and the transformed data passed the normality test.

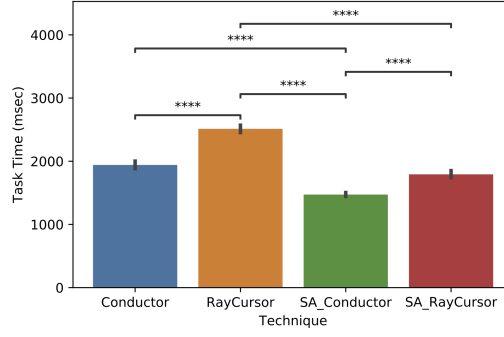


Figure 3.3: Task completion time for target selection task. Error bars are 95% CI.

Figure 3.3 shows the task completion time for target selection. There was a significant main effect for TECHNIQUE on task completion time ($F_{3,33} = 34.4$, $p < .001$, $\eta_G^2 = .50$). Overall, the Conductor techniques (SA_Conductor and manual Conductor) were faster than their corresponding RayCursor variants. Focusing, specifically, on the Semi-Auto techniques, Semi-Auto Conductor (mean = 1.46 sec) was 21% faster than the Semi-Auto RayCursor technique (SA_RayCursor) (mean = 1.77 sec, $p < 0.001$). Manual Conductor (mean = 1.89 sec) had a similar advantage over manual RayCursor (mean = 2.49 sec, $p < 0.001$), and performed approximately on-par with Semi-Auto RayCursor (mean = 1.89 sec vs 1.77 sec, ns).

Although there was no significant main effects for DENSITY nor SIZE, significant interaction effects of TECHNIQUE $\times$ DENSITY ($F_{1.35,14.9} = 12.05$, $p < .01$, $\eta_G^2 = .05$) and TECHNIQUE $\times$ SIZE ($F_{3,33} = 9.07$, $p < .001$, $\eta_G^2 = .03$) were found. Figure 3.4 shows the average task completion time for different sizes of targets. A post-hoc test revealed there were significant differences in task completion time between D_{Low} and D_{High} with the Semi-Auto RayCursor technique ($p < .001$) and between S_{Big} and S_{Small} with the Semi-Auto RayCursor technique ($p < .001$) and Semi-Auto Conductor technique ($p < .001$). As shown, Semi-Auto Conductor performed better with larger targets than with smaller targets. However, with Semi-Auto RayCursor, the larger targets slowed selection because of occlusion. For the same reason, with a higher density of targets, the selection speed with Semi-Auto RayCursor slows.

This result indicates that the Semi-Auto RayCursor technique is more prone to occlusion or, conversely, Semi-Auto Conductor is more resilient to occlusion. However, note that, overall, for selection, Semi-Auto Conductor outperforms Semi-Auto RayCursor by a statistically significant margin.

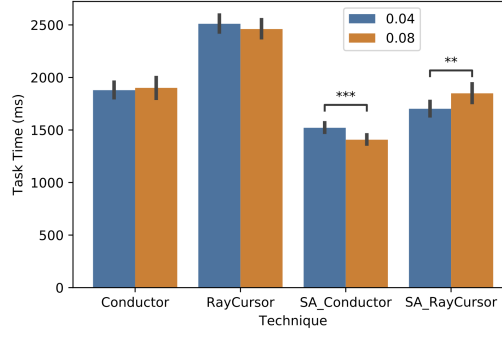


Figure 3.4: Task completion time for different target sizes and techniques. Error bars are 95% CI.

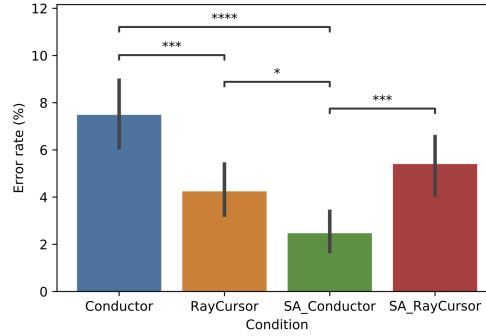


Figure 3.5: Error rate for target selection task. Error bars are 95% CI.

3.6.4.2 Error rate

A three-way ANOVA revealed significant main effect for TECHNIQUE ($F_{3,33} = 10.1$, $p < .001$, $\eta_G^2 = .13$) and SIZE ($F_{1,11} = 7.5$, $p < .05$, $\eta_G^2 = .04$) on error rate. There was no interaction effect. Figure 3.5 shows the error rate for each technique. With all four techniques, the error rate was less than 10%. The error rate of Semi-Auto Conductor was the lowest (2.5%) and it was significantly lower than Semi-Auto RayCursor (5.4%) ($p < .001$), and both manual techniques. Manual Conductor had the highest error rate, but, interestingly, while manual RayCursor outperformed manual Conductor by a statistically significant margin, Semi-Auto RayCursor did not outperform manual conductor.

3.6.4.3 Questionnaire

Our questions indicate participants had a positive impression of the Semi-Auto Conductor technique compared to other techniques. A one-way ANOVA

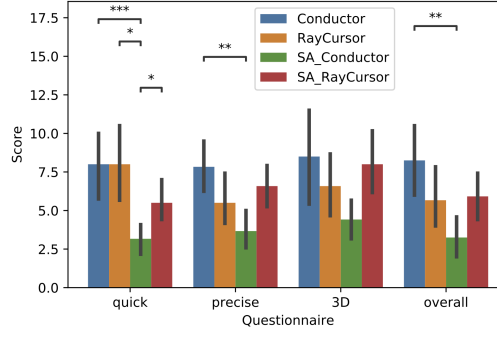


Figure 3.6: Additional questionnaire results for each technique. The questions were as follows: "quick": I can control it quickly. "Precise": I can control it precisely. "3D": I can feel that I'm controlling the cursor in all three dimensions. "Overall": Overall, I like this condition. Error bars are 95% CI.

revealed a significant main effect for TECHNIQUE ($F_{3,30} = 7.4, p < .001, \eta_G^2 = .15$) on NASA-TLX Mental workload and significant difference between Semi-Auto RayCursor and Semi-Auto Conductor techniques ($p < 0.01$).

Our additional questions also indicate that participants had a positive impression of the Semi-Auto Conductor technique compared to other techniques. A one-way ANOVA revealed a significant main effect for TECHNIQUE ($F_{3,30} = 5.9, p < .01, \eta_G^2 = .29$) on perceived speed (quick) and a significant difference between Semi-Auto RayCursor and Semi-Auto Conductor techniques ($p < .05$). We identified no statistically significant differences between the manual variants. Figure 3.6 shows the additional questions and their result. Please note that, for our custom questionnaire questions, as with the NASA-TLX, we preserved the convention that *lower* scores indicate better performance. As with the NASA-TLX, the additional questions were on a 20-point scale where left (0.0) is "Strongly Agree" and right (20.0) is "Strongly Disagree".

3.6.4.4 Synthesis

Considering our initial hypotheses for this experiment, we find that H1 is supported, i.e. the Semi-Auto variant of Conductor has performance advantages over the Semi-Auto variant of RayCursor. However, results for H2 (manual variants) is more mixed. While manual Conductor is faster than Manual RayCursor (by approximately 33% on average, a significant speed up), error rate was also higher by a statistically significant margin. Overall, the Semi-Auto Conductor technique had the lowest error rate, outperforming manual Conductor and Semi-Auto RayCursor by statistically significant margins.

Finally, questionnaire results provide evidence that Conductor is no worse than RayCursor in terms of user preference and overall workload load, i.e. that the incorporation of bimanual techniques does not prohibitively increase physical or mental workload. In particular, from custom questionnaire data, Semi-Auto Conductor is perceived to be quicker than RayCursor variants.

3.7 DISCUSSION

We find that, for target selection timing, our bimanual technique, Conductor, outperforms RayCursor by a statistically significant margin, particularly in its semi-automatic form. Therefore, we posit that, for target-aware selection tasks, bimanual selection via Conductor provides faster selection with lower error over RayCursor. We do note that in manual cases, i.e. if pointing facilitation techniques cannot be leveraged, throughput advantages may be more mixed as the higher speed of bimanual Conductor may be offset by a possible increase in error rate.

One question that can be posed in any replication experiment is whether or not our comparison was fair. We address this from two perspectives: first, whether our implementations of RayCursor was disadvantaged in some way compared to the implementation of Baloup et al. [7]; and second, whether Conductor represents a valid baseline comparison versus RayCursor.

Considering the first point, alongside evaluating relative performance, our results provide an additional opportunity to contrast our RayCursor implementation to the original implementation of Baloup et al. [7]. The average selection time of our implementation of RayCursor (mean = 2.79 sec) and Semi-Auto RayCursor (mean = 1.93 sec) in our study were virtually identical (within 70ms) of the original RayCursor implementation in Baloup et al.'s studies (mean = 2.72 sec and 1.88 sec, respectively). The error rate of RayCursor (4.7%) and Semi-Auto RayCursor (6.5%) in our study were slightly higher than the original study results (3.8% and 2.4%); Even though the error rate is small in number, only 4.1 points. However, it is relatively large: $(6.5 - 2.4) / 2.4 = 1.71 = 171\%$ increase compared with the previous study. We assume that this may be the result of higher occlusion of the targets due to the increase in target density in our experimental configuration (recall that RayCursor was never evaluated with fully occluded targets in the original study).

Second, was Conductor versus RayCursor a fair comparison. We note that there are alternative bimanual pointing techniques for virtual environments.

Some of these techniques use enhanced visualization techniques (e.g., enlarging the targets using a lens on the non-dominant hand [133], rearranging or distorting space in some way [30], or rearranging targets [42, 67]). These are, in our estimation, less of a fair unimanual-bimanual contrast against RayCursor because of the modification to the underlying scene. Many other techniques within the literature focus on both visual search and selection by making objects transparent [17, 125, 126], potentially slowing bimanual selection and thus disadvantaging bimanual techniques. Essentially, we believe that our technique represents the most direct generalization of RayCursor to bimanual interaction without added enhancements or complexity to advantage or disadvantage techniques one way or the other.

Overall, if rapid targeting is desired in virtual environments (which is the premise of our work, RayCursor, and many other proposed depth and occlusion-resistant interaction techniques [7, 42, 104, 131]), then Conductor can be a viable selection technique for incorporation into virtual environments. This is not to say that Conductor should universally replace the RayCursor technique in implementations. In some instances, for example, due to the presence of only one controller or encumbrance during interaction of the non-dominant hand controller, unimanual selection techniques such as RayCursor are needed. However, in many instances during target selection, the non-dominant controller is at rest with no assigned task. In these contexts, leveraging a bimanual technique to enhance throughput (via higher speed and lower error) may be desirable.

Beyond an analysis of individual techniques, there is a more generalized observation that our research results present. To the best of our knowledge, while unimanual and bimanual targeting techniques do exist in the literature for targeting in virtual environments, we were unable to identify studies that contrast the performance of bimanual versus unimanual techniques. However, performing such studies is important to the design of user interface systems and techniques in virtual environments, because, as we note in the introduction, if no benefit exists to bimanual techniques, then it is unclear why one would ever consider designing a bimanual technique, i.e. if unimanual performs at least on par, then there is no role for bimanual input.

3.8 FUTURE WORK AND LIMITATIONS

While our result shows that Conductor techniques outperform RayCursor in terms of target selection tasks when bimanual input is permissible, we

would also like to explore if Conductor could be applied to other interactions, including placement or docking tasks in virtual environment interaction. RayCursor, itself, has never been evaluated on these tasks, so this comparison will further our understanding of the trade-offs of unimanual and bimanual techniques in virtual environments.

Regarding the experiment design, we would also like to explore more parameters, like more target sizes and densities, or factors, like target distance. And whether there is a better algorithm could help improve the overall performance under those conditions.

Considering limitations, we note four. First, unlike in the initial RayCursor implementation, where touchpad-equipped controllers were used, our controllers incorporated joysticks. To control for this, we piloted to ensure similar performance, and we also contrast our implementation of RayCursor’s performance to performance values reported by Baloup et al. [7] in their published work. Current and previous studies’ groups exhibited similar operational times when using different devices (the difference between the two groups was within 50ms).

Understanding the performance of both touchpad and joystick controllers is valuable due to the fact that both are commonplace alternatives for VR control.

Second, due to the global health context of the experiment, suitable participants were hard to find, and some of our participants had to perform the experiment remotely using their own equipment (note that we did ensure identical hardware). While this is less ideal than a lab-based environment for careful experimental control, one advantage to VR-based experimentation is that virtual environments can be leveraged to control what is seen by participants, and controllers have similar performance across a broad range of contexts. While this is a potential limitation on the experimental validity of our results, it is controlled for via our within subjects design, and the increase in ecological validity trades off as an advantage against this potential limitation.

Third, while not a limitation of our experiment itself, we remain somewhat surprised that the incorporation of a depth cursor along a ray proposed by RayCursor in Baloup et al.’s 2019 paper is novel, but we, like Baloup et al., cannot identify any competing references. The challenge with depth incorporation may have been the identification of an appropriate gain function to make cursor/depth manipulation controllable, a significant challenge that Baloup et al. effectively solved. That said, regardless of the novelty of ray-aligned

depth manipulation, the contrast of unimanual and bimanual techniques we present remains valuable because it highlights the complementary role that these interactions can play in virtual environments.

Finally, some may question our use of RayCursor as a standard for comparison and may, instead, point to other techniques as a contrast. We chose RayCursor because it is a recent, high-performance, unimanual selection technique optimized to handle selection in environments where target occlusion occurs and without modification to the underlying scene. Other unimanual selection techniques exist (see [3] for a review) and several bimanual techniques continue to be proposed (e.g. [103]). We believe that contrasting new novel bimanual techniques with unimanual equivalents or beyond will continue to provide a valuable understanding of techniques' strengths and weaknesses.

3.9 CONCLUSION

In this chapter, we present Conductor, a bimanual technique modification of the RayCursor technique for target selection in virtual environments with two variants, Manual Conductor and Semi-Automatic Conductor. Through an initial carefully replicated experiment, we show that Conductor provides statistically significant performance enhancements to RayCursor, a state-of-the-art unimanual target selection technique for virtual environments. By distributing roles across both hands, Conductor leverages the dominant hand to control the pointing ray, while the non-dominant hand defines an intersecting plane to determine target depth at the same time. We discussed the future work and limitations of Conductor, and we believe that it could be a promising technique for VR selection.

FLY THE MOON TO ME

Teleportation - changing the point of view in 3D space by specifying a position - is one of the most common locomotion solutions in VR. However, it currently lacks a mechanism to adjust the height in 3D space, and it is difficult for users to predict the exact final view after the teleportation. Users are relocated to a place without knowing what the final view will look like. As a result, they often need to perform remedial interactions to achieve their ideal position, which can be time-consuming and effort-intensive.

In this Chapter, we present Fly the Moon to Me (Locomoontion), a novel technique that enables users to bring their destination to themselves through object manipulation. Users first create a copy of the object they want to approach as a preview by selecting it, then bring it to an ideal position and direction using existing object manipulation techniques, and then snap the original object to the preview together with the rest of the world. A controlled experiment with 18 participants via a teleportation task reveals that Locomoontion is more effective than the traditional Point&Teleport technique with grabbing the world as a remedy to adjust the final positioning.

4.1 INTRODUCTION

VR is widely used in both recreational and productive domains. Locomotion is one of the elementary interactions in VR [69], which enables users to move in VR and interact with distant virtual objects in a large VR space. For example, in a VR museum application, users can move between different collections to examine them more closely. Similarly, in a virtual house viewing application, prospective buyers can move to the windows, the sofa, or other areas to inspect the property. Walking by feet, the most immersive [120, 122] and least discomforting [56, 82] VR locomotion technique, could be limited by real-world space constraints, including limited tracking space and physical blockers in the room [46, 138]. Additionally, moving to different heights within a virtual space can be difficult when users are limited to walking on the physical floor. For example, in a VR drawing application, users may need to reach higher or lower areas to create large-scale paintings

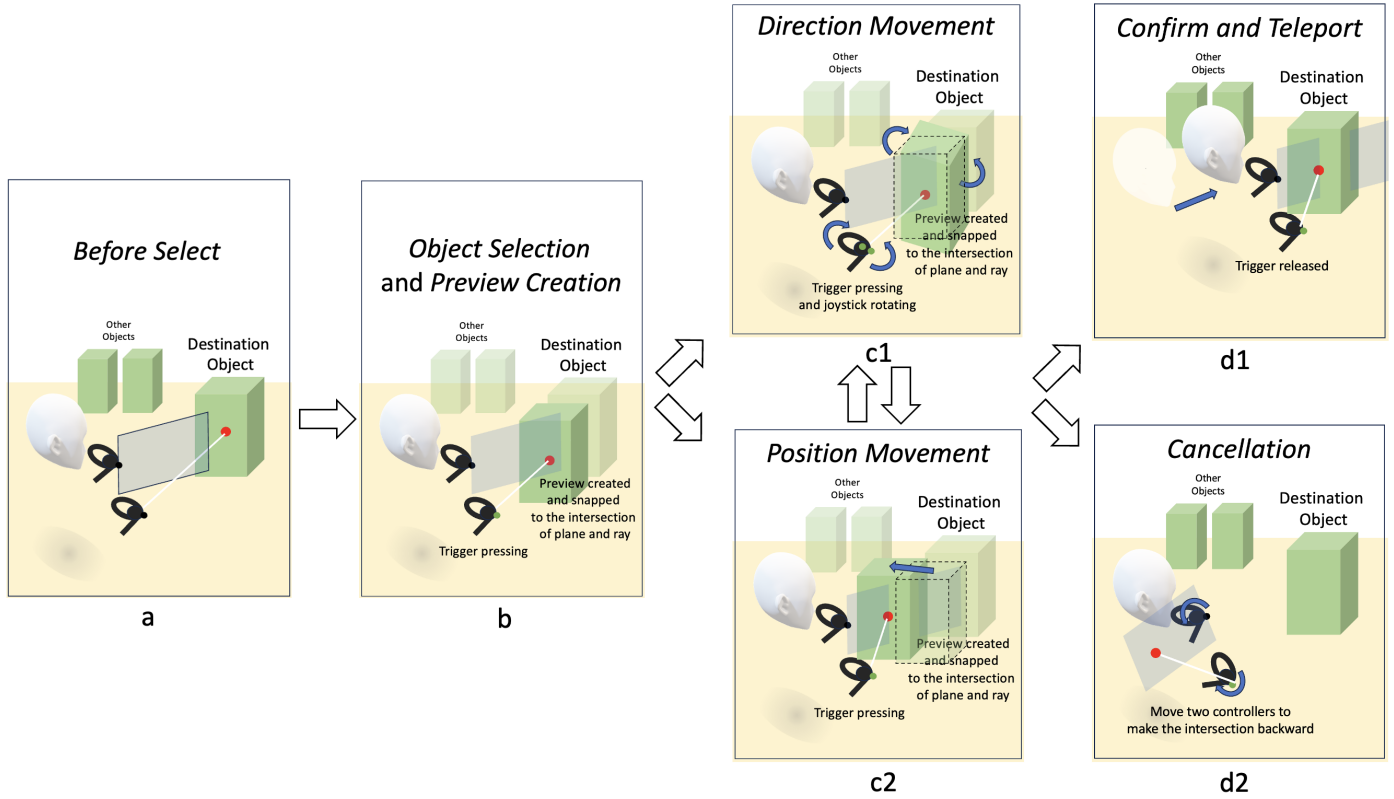


Figure 4.1: Workflow of Locomoontion, a precise 3D locomotion technique in VR. a) Find the object that the user wants to approach and point with semi-auto Conductor. b) Press the index trigger on the dominant controller to select the object. Locomoontion will make the world semi-transparent and create a solid copy of the object as a preview of the final teleportation view. The preview will be snapped to the red cursor on intersection of the plane on the non-dominant controller and ray on the dominant controller, as per manual Conductor. c) Users can manipulate the preview by c1) rotating the joystick to rotate the object or c2) moving the object in 3D space with manual Conductor. d) To finalize, users can either d1) release the index trigger, snap the object to the preview, together with the rest of the world, or d2) move the intersection backwards of the ray shooting direction to cancel Locomoontion.

or murals. Similarly, in VR architectural walkthroughs or games, users may need to inspect ceiling fixtures, lofted spaces, or elevated platforms. Several locomotion techniques have been developed to mitigate these challenges and allow users to adjust their height above the ground safely and efficiently.

Locomotion techniques could be categorised primarily as two types of movement: discrete and continuous [86]. The most popular discrete locomotion in current VR systems is teleportation [13, 18, 37, 86], which allows users to be moved instantly after a destination is selected by pointing to the ground. Other teleportation techniques would involve a preset location, such as portals [34] and gates [55]. It is easy to learn and enables fast long-distance movement, but it can be difficult to predict the user's appearance after teleportation. They assumed users could simply walk to their desired

location for short-distance adjustments, but these refinement movements can be time-consuming and potentially ergonomically problematic [137]. Such challenges are particularly pronounced for seated users or individuals with mobility impairments. There are precise continuous "remedy" interactions like Point&Tug [12, 24, 80]. However, this approach may be less efficient, requiring additional physical effort and potentially disrupting workflow. Moreover, traditional point-to-the-ground and teleportation techniques would not support 3D movement at different heights, which requires the remedy interactions to specify the height. With continuous locomotion techniques, users could move in the virtual environment continuously, but may experience more motion sickness due to the mismatch of the static physical body and the moving world. Therefore, various interactions were designed to mitigate the problem by reducing the visual area to a small area in the centre [10] or asking users to perform some actions such as walking in place [128], flying [88, 116, 151], swimming [57], and cycling [35, 93]. However, they would easily introduce fatigue, thus not being suitable for long-time use. Continuous techniques also tend to require more time for the same locomotion task compared to discrete techniques [137]. Our research question thus is: **How to enable quick and precise 3D discrete locomotion in VR?**

In this work, we present *Locomoontion*, a 3D VR locomotion technique simultaneously enabling location (xy -plane), height (z -axis), and orientation with high precision, fast speed, and low physical effort. Instead of moving the user to the desired position in VR, we employ a novel approach to "bring" the destination object to the user for previewing the teleportation results - fly the moon to me, rather than fly me to the moon. We envision this technique being particularly useful in scenarios that require precise locomotion over visible distances – for example, when the goal of movement is to position oneself for an object manipulation task. The Locomoontion technique comprises the following steps:

- **Selection And Preview:** Users select the target objects that they intend to teleport toward. A copy of the object will be generated as a visual preview, clarifying the object's post-teleportation state.
- **Direction And Position Adjustment:** Users refine the preview's 3D position and orientation to align it with their desired interaction location using Conductor, a 3D pointing technique, and direction with the joystick.

- **Teleportation:** Once users finalize the preview’s placement, the original object is moved to the preview’s confirmed location, followed by the rest of the world.

Locomoontion uses Conductor to support 3D position manipulation during teleportation. With Conductor, the dominant hand controls a pointing ray for precise target specification, while the non-dominant hand controls a cutting plane that simultaneously defines the target’s height and position context. The cursor position is given by the intersection of the ray and the plane. In this way, Locomoontion integrates height adjustment with 2D positional control into a single unified interaction, rather than treating them as separate sequential steps. From the perspective of bimanual theory, Locomoontion is an asymmetric, parallel technique: the non-dominant hand sets the spatial frame (plane + height) in parallel with the dominant hand’s precise pointing, consistent with Guiard’s kinematic chain model [44].

To assess the performance of Locomoontion, we conducted a controlled experiment with 18 participants to compare our technique with the most popular point and teleport technique for teleportation tasks. We adopted a within-subjects design involving 3 DISTANCE, 3 HEIGHT, and 4 ANGLE conditions. The result shows that Locomoontion achieved less task completion time for both overall (2.4 seconds faster) and among all conditions, also with less physical fatigue.

4.2 BACKGROUND AND RELATED WORK

In [Chapter 2](#), we discussed related work on general VR locomotion techniques. In this section, we present a deeper exploration of the background related to object manipulation techniques, which are used for manipulating the preview object.

4.2.1 3D Placement

If we consider the virtual world, including the users themselves, as 3D objects, then locomotion essentially becomes a 3D placement, or docking task. This task involves either placing themselves somewhere within the virtual environment or placing the entire world relative to their position, which aligns with the underlying logic of *Fly The Moon To Me*. Therefore, in this section, I will provide some background on 3D placement.

In 3D placement, Berard et al. [9] evaluated the mouse against 3 DOF input devices. Somewhat surprisingly, they found that the 2D computer mouse outperformed special-purpose 3D controllers for 3D placement tasks. One challenge with generalizing their work to virtual environments is that they assume a user seated at a desk, interacting with 3D content. As virtual environments – augmented realities – become increasingly embedded into real-world environments, users often must interact with 3D content when standing or moving.

Surprisingly, despite the significant body of past research exploring targeting in 3D spaces, raycasting techniques have been rarely evaluated for placement tasks. However, many of these techniques can be used in placement as easily as in targeting. For example, Raycasting variants [42, 109], including RayCursor [7], typically have some indicator of depth attached to the ray, and this depth indicator can be freely manipulated to control position. Virtual hand techniques, also support placement in a relatively straightforward manner.

Virtual Hand techniques including Go-Go [89, 90, 104, 137], are considered one of the most natural solutions for manipulating the location and/or rotation of a distant object in VR. The User controls a 3D cursor in VR with controllers or bare hands and translates the offset to the cursor position. However, these techniques often require a significant amount of time, especially for distant objects, which can lead to increased fatigue. OrthoGaze [81] allows the gaze to manipulate the object along three orthogonal planes in VR. Yu et al. [147] also proposed four designs for gaze-supported object manipulation combined with hand motion. However, they won't support depth changes beyond the user's hand reach. In contrast, World-In-Miniature techniques [99, 123] involve scaling the world and manipulating the world to access objects.

Some 3D selection techniques also offer potential for 3D object manipulation, as they can control a cursor in 3D space, not limited to selection only. One solution involves placing a cursor on a ray and using a controller to control the cursor depth on the ray, for example, using a touchpad [7] or a smartphone [109]. There are also bimanual cursor control techniques that use a ray [61, 142] or a plane [149] on the non-dominant controller to intersect the dominant controller's ray and control the cursor accordingly.

There weren't many techniques that were designed or evaluated for docking tasks, so I would list some pointing tasks that could also be used as docking techniques. For the techniques that attach a cursor on the ray [7, 109] would be able to perform docking tasks. For the virtual hand style pointing technique,

Go-Go [104] and its improved version [90] allow users to remap the distance of the virtual hand and real hand along the ray with a non-linear function to reach further positions. Wentzel et al [137] used a Reach-Bounded Non-Linear function to improve the user's reach by 10% while maintaining the body ownership. Vuibert et al evaluated the 6 DOF docking task performance with several mid-air input techniques [133].

4.3 LOCOMOONTION TECHNIQUE

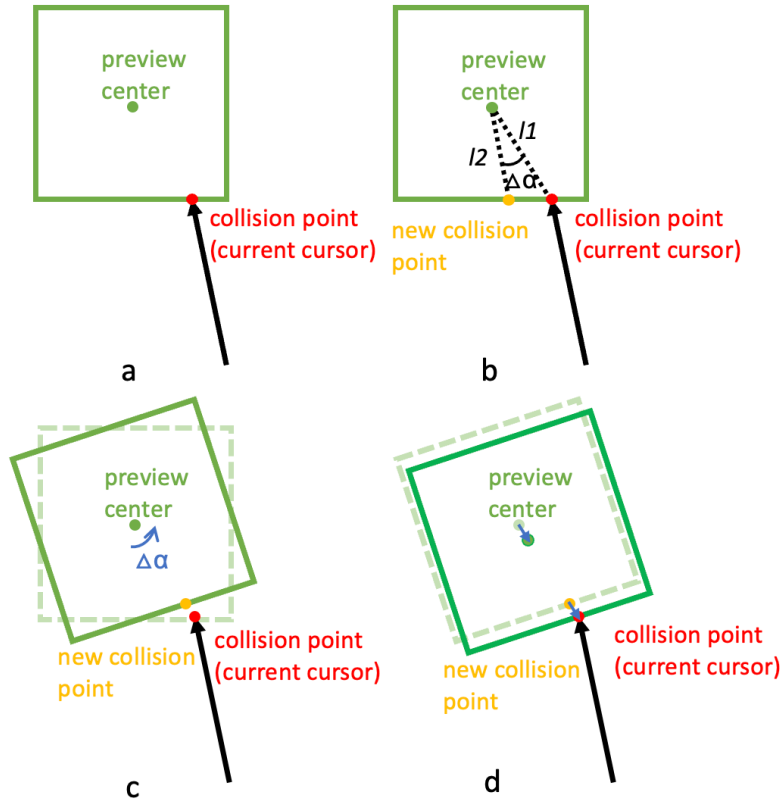


Figure 4.2: Preview rotation algorithm (a) A collision point (red) on the ray, which is also the cursor position. (b) Then, calculate the preview rotation $\Delta\alpha$. Draw a line from the centre to the collision point, and shoot another new invisible ray from the preview centre with an angle difference of $\Delta\alpha$ from the line. The new ray will intersect with the outline of the preview at a new collision point. (c) Rotate the preview from the centre by $\Delta\alpha$. The new collision point will rotate together with the preview. (d) The new collision point will rotate together with the preview, and then move the preview to the position where the new collision point is at the same position as the cursor (previous collision point).

Locomoontion is a VR locomotion technique that leverages Conductor [149] to fulfill 3D position manipulation during the process of teleportation. It consists of the following key steps shown in Figure 4.1. Our vision is that the preview-based teleportation workflow will help reduce the task time, especially when reaching different heights.

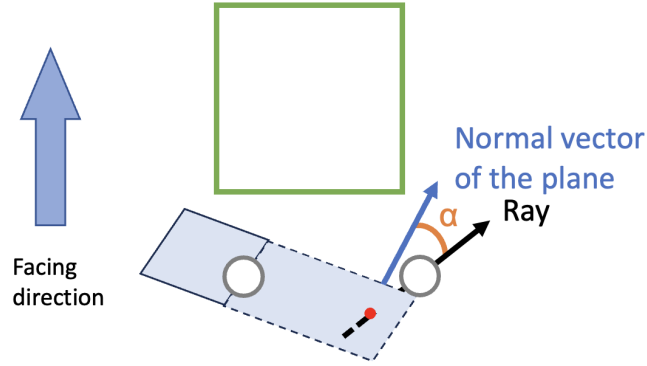


Figure 4.3: Cancel Algorithm: Compute the dot product between the plane’s normal vector and the pointing ray. If the result is greater than zero, the system enters cancellation mode.

Object Selection and Preview Creation (Figure 4.1-ab): Locomoontion utilizes Semi-Auto Conductor to select the object the user wants to interact with by pressing the dominant controller’s index trigger. It automatically creates a copy of the object as a preview. As mentioned earlier, the preview will then be attached to the red cursor, located at the first collision point of the ray and the preview. Subsequently, the cursor, along with the preview, is instantly snapped and attached to the intersection of the plane and the ray for position movement.

Position Movement (Figure 4.1-c2): Once the preview and cursor are snapped and attached to the intersection, the preview will continually follow the intersection of the plane and ray. Users can adjust the preview’s position using manual Conductor. The user keeps pressing the index trigger on the dominant controller to move the copy with Manual Conductor. The copy looks identical to the original object, while the rest of the world, including the original object, will turn semi-transparent.

Direction Movement (Figure 4.1-c1): Users can also rotate the dominant controller’s joystick clockwise or counterclockwise to change the direction of the object. To activate rotation, the user needs to push the joystick to the boundary. The user can then rotate the joystick against the boundary

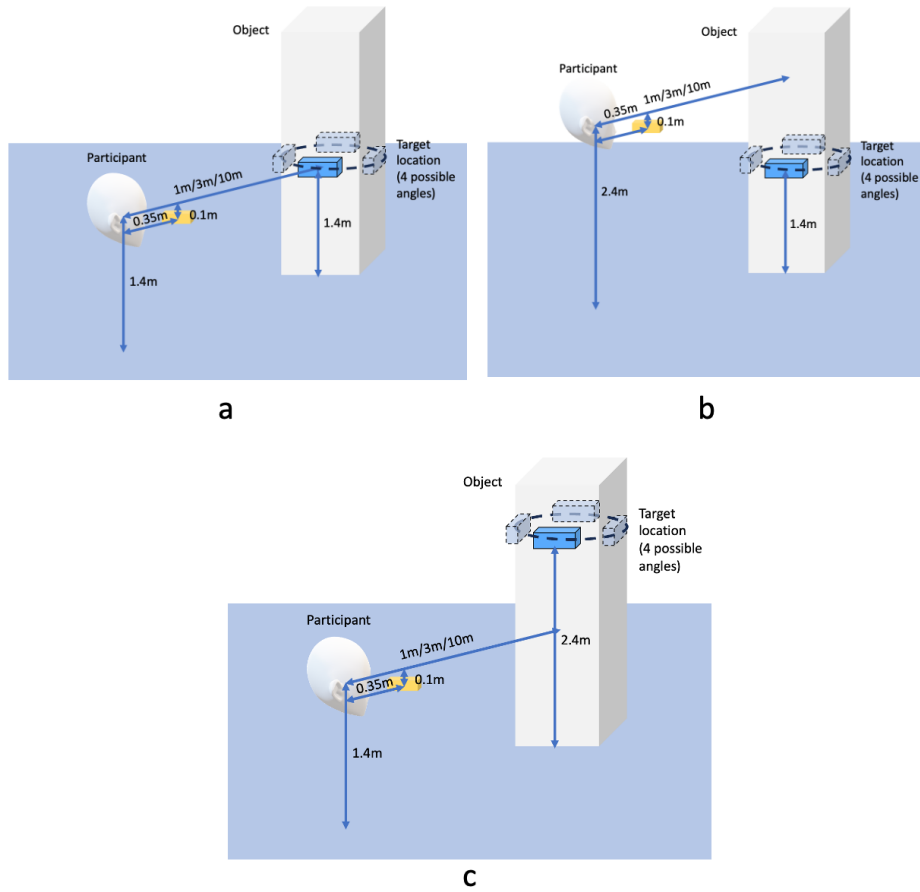


Figure 4.4: The settings of the experiment. (a) The participant is at the same height as the blue target. (b) The participant is 1m higher than the blue target. (c) The participant is 1m lower than the blue target. The solid blue box is the Front (0°) condition. The semi-transparent boxes are Left (90°), Back (180°), and Right (270°) angle conditions. Only one box would be visible in one angle condition. The distance between the user and the object could be 1m, 3m, or 10m. The yellow box attached to the user is 0.1m lower than the user's head location (VR HMD coordinate) and 0.35m in front.

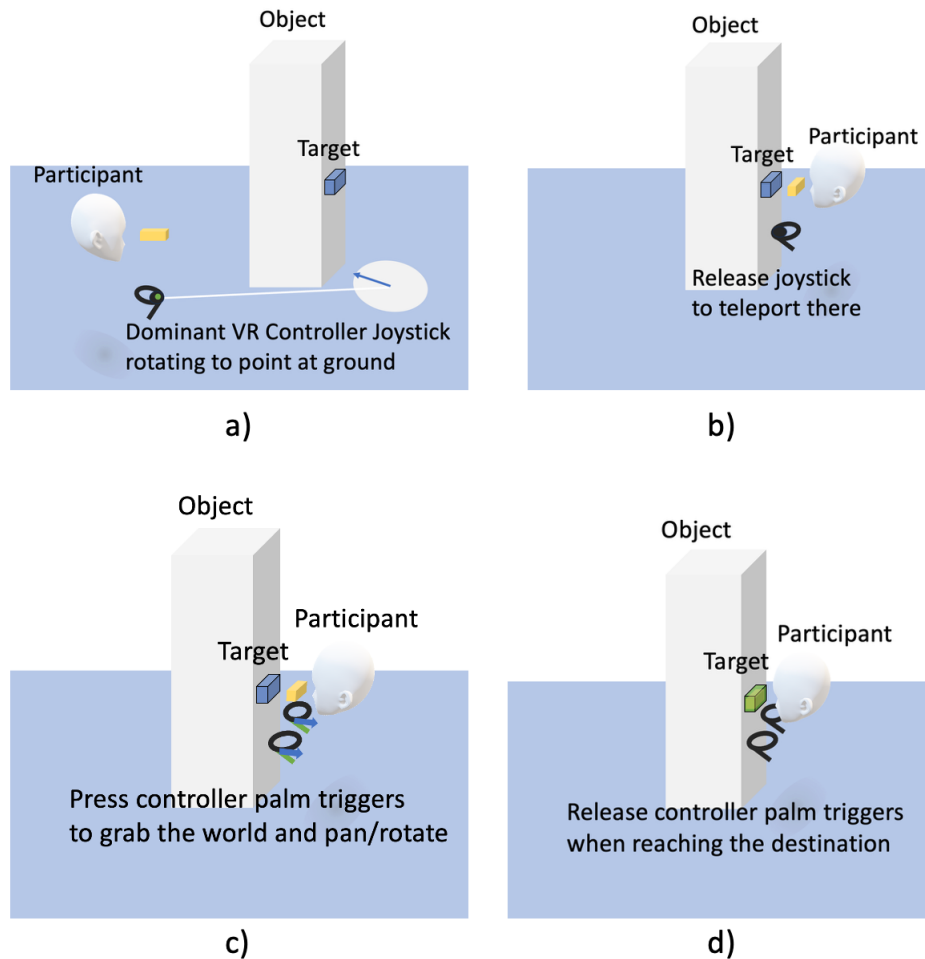


Figure 4.5: The implementation of the Baseline technique. The participant is going to reach the target position. (a) The participant is rotating the joystick to shoot a ray pointing at the ground. It would be somewhere close to the destination. (b) The participant releases the joystick to teleport there with the same height. (c) The participant presses the two-palm trigger on the controller to grab and pan or rotate the world. (d) The destination is reached.

clockwise or counterclockwise. Based on our pilot testing, the Control-Display Gain (CD gain) of the joystick rotation to object rotation is set to $1/3$, which means 3 degrees of joystick rotation would lead to one degree of preview rotation.

To keep the preview outline always snapped to the cursor, we developed an algorithm to maintain that relationship as shown in [Figure 4.2](#). To calculate the rotation angle, we first define how the preview is rotated and moved from the current frame to the next frame. In the current frame, there is a collision point on the dominant controller's ray, which is also the cursor position. Then, calculate the preview rotation $\Delta\alpha$. Draw a line from the object centre to the collision point, and shoot another new invisible ray from the preview centre with an angle difference of $\Delta\alpha$ from the line. The new ray will intersect with the outline of the preview at a new collision point. Then rotate the preview from the centre by $\Delta\alpha$. The new collision point will rotate together with the preview, and then move the preview to the position where the new collision point is at the same position as the cursor (previous collision point). To prevent jitter while pushing the joystick against the boundary, such as accidentally leaving the boundary at a certain angle, the rotation won't be completed until the thumb leaves the joystick.

Confirm and Teleport ([Figure 4.1-d1](#)): The previous two steps allow users to move and rotate the preview freely in 3D space. Once the preview is in the desired position and direction, the user can release the dominant controller's index trigger to confirm. At this point, the original object that the user selected will be relocated to the preview position and direction, along with the rest of the world. By nature, Locomoontion can also facilitate height manipulation. Users can simply select the ideal height, and move it lower or higher and close to the user, given that the object is attached to the cursor.

Cancellation ([Figure 4.1-d2](#)): When users intend to stop teleportation, they can always initiate cancellation by positioning the intersection point in a manner that contradicts the ray shooting direction, and then release the index trigger to confirm the cancellation. The algorithm is as follows ([Figure 4.3](#)). It first calculates the normal vector of the plane towards the ray side; then performs a dot product between the normal vector and the ray vector. If the dot product is greater than zero, then it is cancelling the teleportation.

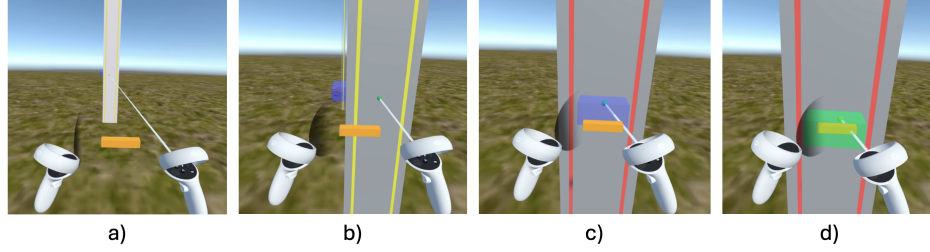


Figure 4.6: Screenshot of Locomoontion technique. a) Selecting the object with Conductor. b) Moving preview closer with Conductor. c) Rotating the preview using the joystick. d) Fine-tuning the position until the yellow box is fully inside the blue box (turn green).

4.4 EXPERIMENT

We designed a controlled experiment to evaluate the performance of Locomoontion by comparing it against the traditional Point&Teleport technique with Bimanual Point&Tug as a baseline because it is the most common teleportation technique in current VR applications and games [40, 45, 121, 129]. We designed a 3D navigation task involving positioning a box to assess their respective performance. We hypothesise that:

- **H1:** Locomoontion is faster than Point&Tug, since Locomoontion takes less time for refining the position and is able to adjust height quickly.
- **H2:** Locomoontion requires less physical demand than Point&Tug, because it requires less physical movement when adjusting the height.

4.4.1 Experimental Task

As shown in Figure 4.4, the task of the experiment asked users to move to a specific place around an object with those two techniques, which mimics the use case that they want to teleport to and interact with some part of an object. Two distinct boxes were presented: a larger blue box around the object to indicate the target location, and a smaller yellow box attached to the participant to represent their respective location. Users needed to move to an ideal location that could put the yellow box fully inside the blue box.

We selected this task to specifically highlight the refinement aspect of our experiment. While many VR studies focus on long-distance navigation challenges [37, 86, 135] — they often overlook short-distance refinement, assuming users can effortlessly "quickly navigate" to precise targets. However, even

minor positional adjustments can become time-consuming, induce ergonomic strain [137], or introduce usability barriers. This issue is amplified for seated users or individuals with mobility impairments, where subtle movements may require disproportionate physical effort or compromise accessibility.

4.4.1.1 Task Design

The task is designed to be performed in a seated position. In particular, there would be a white box (referred to as Object) in front of the user in VR, which mimics the object that the user may want to interact with. The white box is 0.3m in width and depth and 2.5m in height (0.30m, 0.30m, 2.50m). The distance between the participant and the object would be 1m, 3m, or 10m. There is also a semi-transparent blue box attached to the object, which indicates the target position that the user needs to move to. The size of the blue box was 0.15m in width and 0.07m in depth and height (0.15m, 0.07m, 0.07m). The blue target could appear in four directions: front, back, left, and right of the object, and also three heights relative to the user, which could be 1m, 0m, and -1m (relative to the user's height).

The purpose of this setting is to simulate the scenario in which the user needs to move upward, stay the same height, and move downward. There would be a yellow box that is smaller than the blue box and always attached to the user, and always in front of and 0.1m lower than the user's eyes. The yellow box indicates the user's position. The size of the yellow box is 0.05m smaller than the blue box in width, depth, and height (0.10m, 0.02m, 0.02m), which restricts both the location and orientation. The goal of the task is to put the yellow box fully inside the blue box. The blue box would change from blue to green when the yellow box is fully inside it.

4.4.1.2 Task Procedure

At the beginning of each trial, the white object turned transparent to allow the user to locate where the blue box target was, since it might be behind the white object. Participants could take as much time as they wanted to locate the blue target. Then, participants pressed the A button on the right controller to start the trial. Once A was pressed, the system recorded the start time, and the white object turned solid. The participant then performed the locomotion technique until the yellow box was fully inside the blue box. When they finished the locomotion, the trial ended, and the completion time was recorded.

4.4.2 *Baseline: Point&Teleport and Bimanual Point&Tug*

We chose the Point&Teleport and Bimanual Point&Tug, the most popular 3D teleportation technique, as the baseline (Figure 4.5). This combination has been adopted by several VR applications and games in current VR applications and games [40, 45, 121, 129]. Point&Teleport allows users to rotate the dominant hand joystick to shoot a ray pointing at the ground. The pointing position represents the location of the teleportation, and the joystick direction represents the direction that the user will be facing. Once the joystick is released, the user will be teleported to that position with the same height. In order to refine the position and direction, users could also use Bimanual Point&Tug technique. Users can press the two palm triggers to grab the whole world, and pan or rotate it in any direction, which is similar to picture manipulation on the touchscreen.

4.4.3 *Participants and Apparatus*

We recruited 18 participants from a local university via mailing list and word-of-mouth, six women and 12 men, ages 22 to 35. All of the participants were right-handed. Five of them are familiar with VR. Participants received \$25 Amazon Gift card as compensation in local currency.

We developed an experiment platform with C# in Unity 3D 2021.3.27. We use Oculus Quest 2 as the VR Headset, and the experiment platform is running as a standalone app installed inside the Oculus Quest 2.

4.4.4 *Design*

We employed a within-subjects design with three primary independent variables:

- TECHNIQUE with 2 levels (Baseline, Locomoontion);
- DISTANCE with 3 levels (1m, 3m, 10m);
- HEIGHT with 3 levels (-1m, 0m, 1m).

And a random factor DIRECTION with 4 levels (Front (0°), Left (90°), Back (180°), Right (270°)).

The order of TECHNIQUES was counterbalanced with a Latin square. Participants first finished all the trials for one TECHNIQUE and then started

the other. Within each TECHNIQUE, DIRECTIONS were combined into one block, which means one block would contain four trials with all four DIRECTIONS in a random order and the same DISTANCE and HEIGHT conditions in a block. The combination of DISTANCE and HEIGHT was also executed in random order. Once all the conditions were performed within a TECHNIQUE, we repeated the same sequence 2 times, thus 3 repetitions in total.

The dependent variable of our experiment was task completion time, which was computed from the time when participants confirmed finding the blue box position by pressing button A in the right controller, to the time when the locomotion technique finished and the yellow box was fully inside the blue box.

In summary, we collected: $2 \text{ TECHNIQUES} \times 3 \text{ HEIGHTS} \times 3 \text{ DISTANCES} \times 4 \text{ ANGLES} \times 3 \text{ REPETITIONS} = 216$ data points per participant.

4.4.5 Procedure

The researcher first went through the experimental procedure, making sure that the participants understood the experiment, and then instructed the first technique to the participants. The research took 1 block (4 trials per block) to show how the technique works, and then the participants tried it themselves for another 2 blocks. They had 1 repetition to practice and 3 repetitions for the real experiment. Participants were allowed to take a break between blocks. When they finished the first technique, they performed the same procedure again for the second technique. After they finished all the techniques, they were asked to complete a 21-point (0–20) NASA-TLX questionnaire and invited to a semi-structured interview. The overall experiment took around 75 minutes for each participant.

4.5 PERFORMANCE

For each combination of participant, TECHNIQUE, DISTANCE, HEIGHT, and ANGLE, the task completion times more than 3 standard deviations from the mean time were excluded as outliers. In total, 40 trials (1.5%) were removed.

To analyse the data, we used a repeated measures ANOVA with TECHNIQUE, DISTANCE, and HEIGHT as the independent variables, and ANGLE

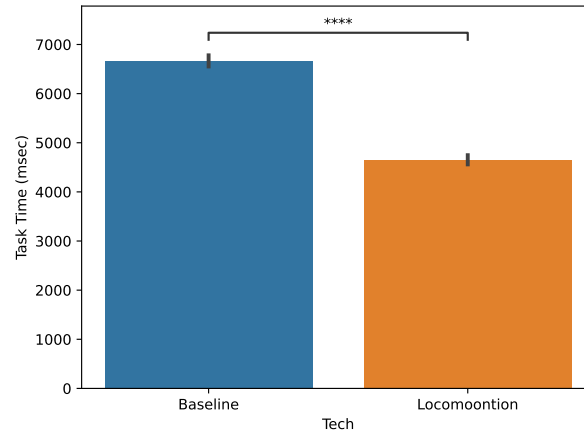


Figure 4.7: Task completion time for locomotion task. Error bars are 95% CI.

being grouped. Holm-Bonferroni corrected post-hoc pairwise t-tests were used for further analysis. Normality was corrected using a Box-Cox transform. When sphericity was violated, degrees of freedom were corrected using Greenhouse-Geisser.

4.5.1 Task Completion Time

Figure 4.7 shows the task completion time for the overall locomotion task. A repeated measures ANOVA indicated a significant main effect for TECHNIQUE on task completion time ($F_{1,17} = 32.17$, $p < .0001$, $\eta_G^2 = .38$). Overall, Locomoontion (Mean = 4652.8 ms) was faster than Baseline (Mean = 6669.3 ms).

Moreover, there were significant main effects for HEIGHT ($F_{2,34} = 17.13$, $p < .0001$, $\eta_G^2 = .03$) and DISTANCE ($F_{2,34} = 74.25$, $p < .0001$, $\eta_G^2 = .11$) on task completion times as well.

Figure 4.8 shows the task completion times for different height conditions, and Figure 4.9 shows the task completion times for different distance conditions.

Post-hoc comparisons indicate that Locomoontion was significantly faster than Baseline in all conditions for both height (Table 4.1) and distance (Table 4.2). Locomoontion also had a stable time at around 4000 ms across conditions.

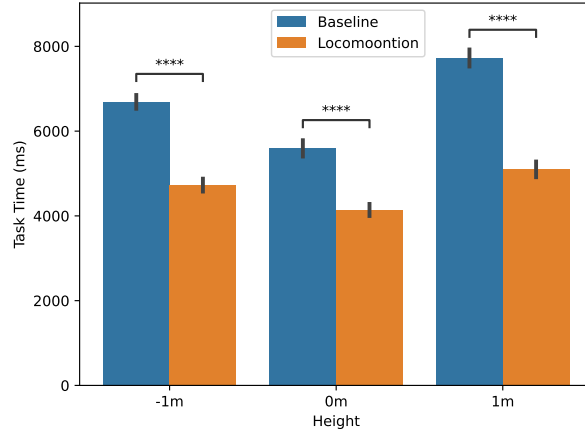


Figure 4.8: Task completion time for different heights. Error bars are 95% CI.

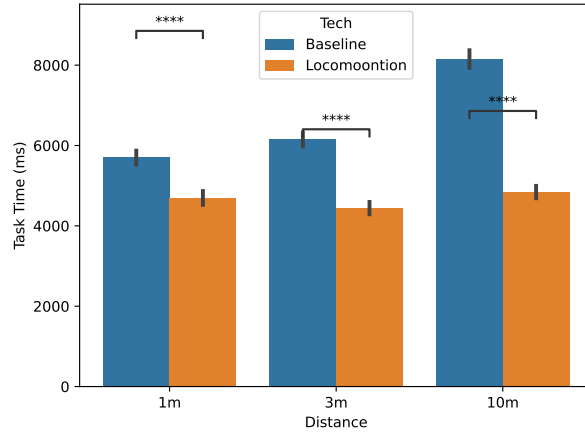


Figure 4.9: Task completion time for different distances. Error bars are 95% CI.

Table 4.1: Task time (ms) for different height conditions

Height	Locomoontion (ms)	Baseline (ms)
-1m	4727.6	6684.4
0m	4131.4	5594.6
1m	5097.2	7726.2

4.5.2 Task Completion Time Over Repetitions

Looking at task time over repetitions ([Figure 4.10](#)), we notice that both techniques showed a learning curve at some level. Even after one repetition of

Table 4.2: Task completion time (ms) for different distance conditions

Distance	Locomoontion (ms)	Baseline (ms)
1m	4684.6	5700.6
3m	4434.8	6156.6
10m	4838.6	8149.9

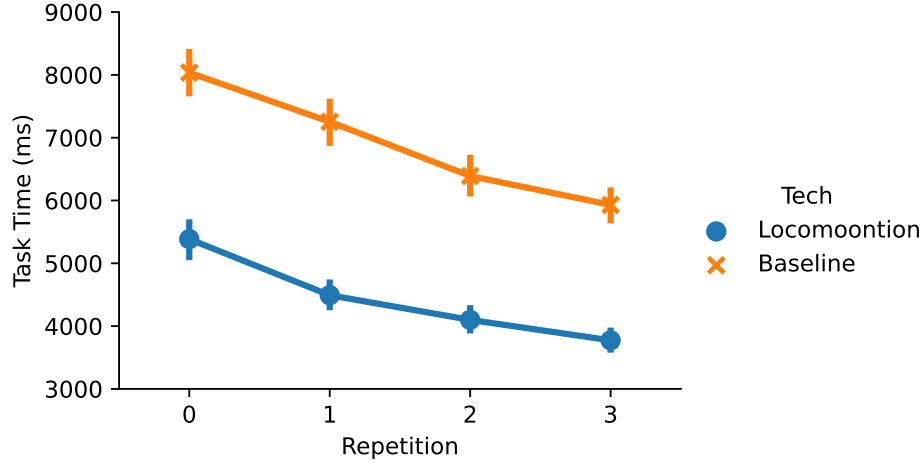


Figure 4.10: Task completion time over repetitions. Repetition 0 indicates the practice run. Error bars are 95% CI.

practicing for both techniques, the Baseline had a steeper learning curve than Locomoontion. For Baseline, the third repetition was 1277 ms faster than the first repetition; however, for Locomoontion, the difference was 625 ms. This result suggests that Locomoontion could achieve a more stable performance under the same amount of training than Baseline. We did not perform a formal ANOVA on learning effects because our main interest is the overall performance comparison, and the 95% confidence intervals already give a clear indication of learning trends, which is sufficient for interpreting relative stability between techniques.

In general, the results support H1.

4.5.3 Questionnaire Results

We also analysed participants' perceived workload using the NASA-TLX questionnaire on a 21-point scale (the lower, the better). Figure 4.11 shows the results, and Locomoontion (Mean = 7.00) significantly outperformed Baseline (Mean = 11.25) in Physical Demand, which supports H2. This indicates that

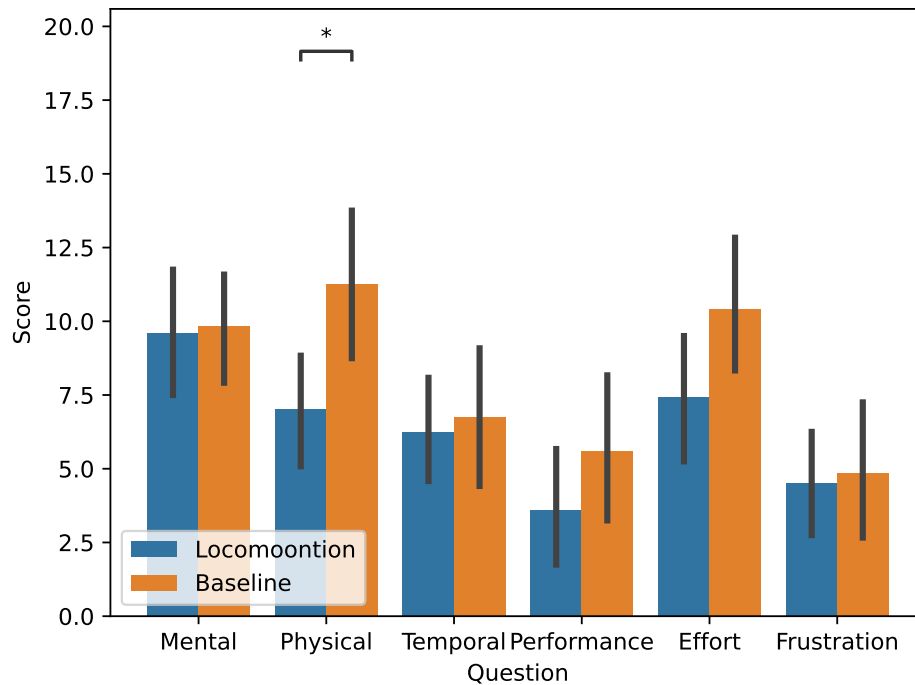


Figure 4.11: NASA-TLX results on a 21-point scale (the lower is better).

users may be less fatigued using Locomoontion to perform the teleportation task.

While there were no significant differences for all other dimensions, the average ratings of Locomotion were lower than those of Baseline.

4.5.4 Interview Results

All participants agreed that Locomoontion was easy to adjust the height. Most participants found Locomoontion was faster once they got familiar with how the technique worked and less physical movement (P1, P3, P4, P5, P7, P9, P10, P11, P12, P13, P14, P15, P18), although it might take more time to learn (P1, P6, P8). Some participants mentioned that Locomoontion was more intuitive and natural, as they could see how the object was moving, and they were controlling the object (P1, P3, P7, P12). Other participants found that the Baseline was intuitive and easy to learn, especially those with some experience with VR (P4, P5, P6, P10).

For direction control, participants said that the joystick controlling the direction could be a bit challenging in Locomoontion in terms of small precise control (P5, P8), but it got some praise for its overall performance from P4 and

P10. However, for Baseline, participants commented that direction was hard to identify in far distances since the arrow was small, or when the participant was close to the ground since the point of view was almost parallel to the ground (P10, P12). This is an inherent limitation of Point&Teleport.

As for different distance conditions, participants reported that it was hard to locate the teleportation position with Baseline. For example, pointing more than once and teleportation with the first time just approached the object closer (P7, P11). But Locomoontion performed similarly in all conditions (P7). Motion sickness was only reported by P9, who found Baseline would lead to minor dizziness.

4.6 DISCUSSION

In this section, we discussed several insights obtained from our design and evaluation of the Locomoontion technique.

4.6.1 *Task Time Over All Conditions.*

Locomoontion interestingly performs very similarly in all conditions with around 4 seconds to complete the task, since the distance and height did not affect much once after selecting the object with Semi-Auto Conductor. We only set 1m higher or lower than the user for height conditions, but Locomoontion would likely outperform much more if the height variance is larger. We look forward to the future usage of the technique in VR games, 3D authoring tools, and other applications. One of the reasons why Locomoontion is faster could be that Locomoontion has a simpler workflow compared with Baseline. Locomoontion requires locating the position, selecting the object, rotating the object, and moving the object, which is, in general, more continuous in the mindset. However, Baseline needs to locate the position, select the teleporting point, teleport and check the current place, locate the ideal position, and refine the position with rotation or height adjustment. Teleporting and checking the new place would cause a break in the workflow.

4.6.2 *Task Time Over Repetitions.*

As mentioned before, we found that Locomoontion had a smooth learning curve compared with Baseline in terms of task completion time. We asked

participants to get hands-on and practice one repetition that was not included in [Figure 4.10](#), given the fact that participants may want to explore the technique within the practice repetition in a casual way. However, even if we took the practice repetition data as an informal way to compare, we still found Baseline (Mean = 8036ms) was slower than Locomoontion (Mean = 5387 ms). This result could again provide some confidence that Locomoontion is easy enough to acquire.

4.6.3 *Fine Tuning*

During the experiment, we observed that participants could quickly move the preview close to themselves but then took additional time to make fine adjustments, especially when the direction was significantly misaligned. This could be due to the challenging nature of the task we designed, which required precise direction changes. We aimed for a highly accurate locomotion technique, resulting in a high constraint: placing a $0.1\text{m} \times 0.02\text{m} \times 0.02\text{m}$ box entirely inside a $0.15\text{m} \times 0.07\text{m} \times 0.07\text{m}$ box—just 0.05m larger along each axis. This level of precision is more demanding than the techniques in previous studies discussed in the "VR Locomotion in 3D" section, which may have had less functionality than our approach.

For the baseline condition, direction changes could be easily and intuitively accomplished with the Bimanual Point & Tug method, similar to grabbing and rotating an object in the real world. However, rotating the joystick requires precise thumb control and careful visual monitoring of the direction, particularly given the small margin for error. Once the direction is confirmed, participants need to adjust the preview's position, and the threshold for how deeply the yellow box can be placed within the blue box is significantly impacted. As illustrated in [Figure 4.12](#), when the direction deviates considerably from the object's orientation (right), the space available for the yellow box inside the blue box diminishes compared to when the orientations are parallel (left). This mechanism was intentionally designed as a penalty for directional inaccuracy.

However, in scenarios where orientation is less restricted, we anticipate that Locomoontion would demand less mental effort and be preferred over the baseline. Interestingly, some participants adjusted the direction by rotating the joystick while simultaneously modifying the position by pressing the dominant controller's index trigger. This strategy allowed them to avoid restarting the entire procedure of joystick rotation, thereby completing the

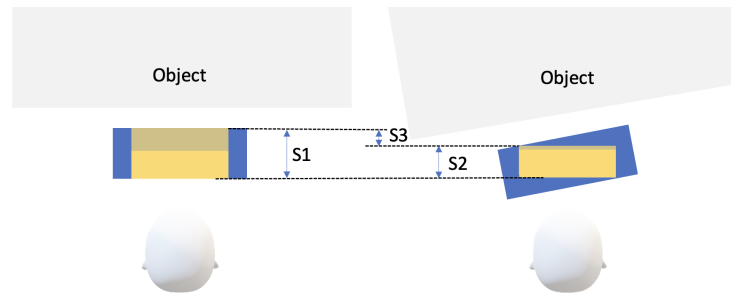


Figure 4.12: The depth of the yellow box can be put into the blue box under distinct direction differences. If the direction varies a lot from the object orientation, the room for the yellow box to be placed inside the blue box (right, denoted by S_2) is less than that with the parallel orientation (left, denoted by S_1) by $S_1 - S_2 = S_3$.

task more quickly. This approach could be included as a tip for novice users in future instructions.

4.6.4 *Tutorial for Locomoontion.*

Locomoontion and Conductor are relatively new techniques compared with the baseline. Therefore, it is important for designers and developers to provide a systematic and intuitive tutorial system to help future users quickly learn these techniques. For example, when introducing the Conductor technique—unfamiliar to most users—designers could display a large plane attached to the non-dominant controller during the tutorial to illustrate how Locomoontion works. In real-world use, this could be replaced with a smaller plane or a simple line as a visual indicator, minimizing obstruction to the view. Another useful tip for the tutorial is to encourage users to select at a height similar to the destination, which enables faster vertical alignment of objects.

4.7 LOCOMOTION, OR DOCKING TASK?

The behavior of Locomoontion is very similar to a docking task, in which users move or rotate objects to different positions or orientations. Consider the baseline technique, Point&Tug: the user points at the ground and then moves there. This is essentially a docking task, where the “object” being moved is the user themselves, relocated to the pointed location. For the “Tug”

part, the world is treated as a single object and moved accordingly. In my view, locomotion can be considered a special form of docking, in which users treat themselves as an object that can be moved or rotated to obtain the desired view. This perspective broadens the future of locomotion techniques, as any efficient docking or pointing technique in VR has the potential to improve locomotion by replacing the Conductor technique. In this sense, Locomoontion becomes a "portal" that connects these two different tasks.

4.8 LIMITATIONS AND FUTURE WORK

We designed the Locomoontion technique with scenarios that require moving to precise positions over visible distances in mind, such as navigating a confined space to manipulate objects. Our experiment focused on evaluating this process of precise movement, especially with regard to the position refinement process. We acknowledge that Locomoontion may not be suitable for all scenarios, and our study did not comprehensively explore possible use case. In this section, we discuss the limitations of the technique and experiment, and suggest directions for future work to further improve upon it.

4.8.1 *Locomoontion Technique*

We identify ways in which the locomotion technique can be further optimized.

Direction Control: As mentioned in the previous section, we found that some participants complained about the precision for direction control in Locomoontion. The current implementation is a direct $1/3$ CD gain mapping, but the performance might be improved with other CD gain functions [16, 150]. Exploring the optimal CD gain function for joystick use is worthwhile.

Scaling Factor for Different Sizes of Object and World: Due to the limited independent variable that could be analysed by ANOVA and total experiment time. SCALE condition is not included in this experiment. We would also be interested to know how Locomoontion would integrate and perform when scaling the scene is enabled. Furthermore, the scene in the current experiment setting contains only one object with one size. It is also worth exploring when the scene is more complex and the ideal object size varies. The virtual environment could be complicated, and we can only evaluate

limited conditions. And we are curious about whether Locomoontion could keep a stable performance in the other conditions.

For small and distant objects, Locomoontion might be challenging, as it is harder to select the object. However, it is also possible to optimize it by enlarging the collider of small objects or grouping small objects together to make selection easier.

Valid Teleporting Location: We did not set any constraints for the object, which means it is still possible for users to “enter” the object, which sounds unreasonable. However, we emphasise that in our experiment, we focus on investigating general teleportation tasks to one object, as the first attempt of validating our new technique. Moreover, in some scenarios, like 3D editing, constraints on users’ locations are not necessary when they are editing the objects. We do acknowledge that the teleporting location constrained by physical objects is important in some cases, such as gaming. It can be easily resolved by adding location checking in our current implementation. Additionally, Locomoontion requires at least one existing object in the scene as an anchor for teleportation. In sparse environments such as open outdoor spaces, Locomoontion would be ineffective on its own due to a lack of available anchors. In such cases, it would need to be combined with other position-based techniques to support movement across diverse environments.

Fly The Moon Away From Me: Conductor has a high precision for close position pointing, which makes it reliable when moving the preview from far to close in Locomoontion technique. However, if the user wants to move away from the object and have a bird’s-eye view of the world, the precision might be limited. That is because when the intersection is far, the angle of the plane and ray could be very small; thus, a minor change of the angle or controller position would lead to a large distance difference. One way to mitigate this issue is to remap the real plane when the plane-ray angle is large (Figure 4.7). In addition, the Locomoontion technique assumes that the users want to face the selected object after teleportation. This limitation may be mitigated in dense environments, where numerous alternative anchors are available. However, in other cases, additional refinement steps may be required to adjust the orientation.

4.8.2 Study Design

We outline aspects of our study design that may restrict the applicability of our results and point out avenues for future work.

Baseline Selection: The Locomoontion technique allows for simultaneous adjustment of location (xy -plane), height (z -axis), and orientation. To the best of our knowledge, no other such technique exists at the time of the study. Thus, when selecting the baseline for comparison, we had to combine and/or enhance existing techniques to ensure they also supported manipulation across all three dimensions. We chose Point&Teleport and Bimanual Point&Tug due to their widespread usage [40, 45, 121, 129]. While we believe this is a reasonable baseline that allows us to understand the utility of object-based teleportation techniques represented by Locomoontion, we do recognize that other options exist. One alternative is to modify the Point&Teleport variant with height adjustment outlined in [135] to allow orientation adjustment as well. The advantage of this option is that it would be a fully discrete teleportation technique, the same as Locomoontion, unlike the current baseline, which includes the continuous Point&Tug technique. On the other hand, the shortcoming is that this technique already reserves the joystick for height adjustment. Adding orientation control would require either overloading the joystick (e.g., through mode switching) or utilizing another input mechanism – both of which could significantly increase cognitive load for users. Overall, we believe there is no single best baseline, and trade-offs must be considered. Future work could focus on comparing Locomoontion with a variety of techniques to evaluate it more thoroughly.

Task Selection: The user study is designed to investigate the effectiveness of Locomoontion in scenarios where locomotion is used to position oneself for object manipulation, a type of use case often overlooked in existing locomotion studies [37]. We chose the simple task of moving to the correct position in the correct orientation in a space with a single object to comprehensively evaluate the utility of Locomoontion during refinement stages, across various relative positions to the target object. However, we acknowledge a trade-off between thoroughly evaluating simple settings and exploring more complex scenarios and tasks. The task we selected may effectively highlight this specific use case, but its applicability to more complex situations remains uncertain. To fully understand Locomoontion’s practical utility, more complex tasks – such as navigating through a labyrinth – would be necessary; this remains an avenue for future work.

Another point of interest is participants’ perception and acceptance of manipulating certain objects, such as moving a building within an outdoor city environment. While such interactions are possible through Locomoontion, it remains unknown how participants would feel about this rather unortho-

dox operation. Future work should take into account users' psychological responses and expectations, and explore adaptations of the technique to better align with user intuition and comfort, as needed.

4.9 CONCLUSION

We present a novel VR locomotion technique, Fly the Moon to Me (Locomoontion), allowing users to select an object, create a preview of the selected object to provide a clear understanding of the view after teleportation, and employ Conductor to manipulate the preview with high precision. By leveraging bimanual input, Locomoontion enables users to adjust spatial position and height simultaneously in a single coordinated action, reducing sequential corrections and accelerating locomotion tasks. The experiment conducted with 18 participants demonstrated the superior performance of Locomoontion over the traditional Point&Teleport technique. Users experienced reduced task completion times and reported less physical fatigue, indicating a more efficient and comfortable locomotion experience with Locomoontion.

In conclusion, Locomoontion shows promising potential to enhance locomotion experiences in VR, offering users the ability to navigate and interact in the virtual environment with greater ease, precision, and comfort. Further research and improvements on Locomoontion could open new avenues for immersive and seamless locomotion in VR applications.

DRUM MENU

Current VR HMDs offer limited shortcuts for rapid command access, which often require users to navigate menus through precise visual targeting at multiple depths. This process can be slow and distracting, particularly during immersive gaming or productivity tasks. While marking menus have shown effectiveness as a shortcut command access mechanism in VR [136], their potential integration with 6 DOF controllers and 2 DOF joysticks in VR environments remains largely unexplored.

In this project, we introduce the Drum Menu, a bimanual shortcut command access technique derived from the idea of traditional pie menus, featuring three input methods, designed for 4-item and 8-item layouts specifically for VR controller command access. Users can select commands by rotating the joystick, drawing a stroke, or pointing in different directions. Bimanual input enables simultaneous access to two menu levels. A controlled user study reveals that drum menus are faster than the unimanual versions for the 4-item layout. Additionally, users prefer the bimanual joystick drum menu with the 4-item layout given its short task time, low error rate, and low physical movement. For the 8-item layout, stroke drum menus are found to be less error-prone for expert users compared to the other techniques.

5.1 INTRODUCTION

Alongside the input of data or text [144], one frequent target of research innovation is command selection [140]. Besides menus and buttons [19], researchers have explored how best to access command shortcuts [20, 85], how to design command gestures [110, 140], and how to leverage rehearsal [47, 68] and cross-modal learning [48, 154] to support rich, rapid command selection for 2D displays.

Recently, VR experiences created by HMDs have been becoming more and more popular in various applications, including gaming and productivity tasks. However, it presents unique challenges for command selection. Navigating a menu and inputting shortcuts are two common ways that users utilize to select commands on computing devices. Currently, most VR applica-

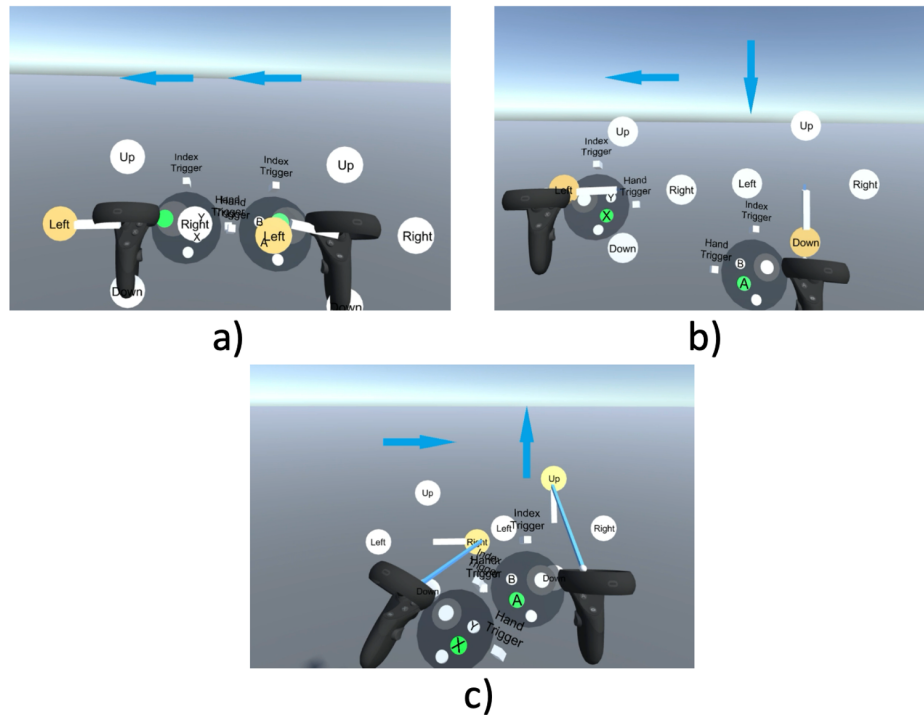


Figure 5.1: Drum Menu is a bimanual rapid command access technique in VR with three variants derived from a 2-level pie menu. a) Bimanual Joystick Drum Menu. Users begin by rotating the joystick on the non-dominant controller to different angles to select the first-level item, then the joystick on the dominant controller to select the second-level item. They could select two levels sequentially or simultaneously, which also applies to the rest of the Drum Menus. b) Bimanual Stroke. Similar to a pen-based marking menu, users can press a button to draw a stroke pointing to the item in the vertical plane. Users need to access two levels with their non-dominant controller and dominant controller. They can release the button on the dominant controller to confirm selection. c) Bimanual Rotation Drum Menu. The menu is always positioned in front of the controllers. Users can rotate the controller upward a bit to point to the upper item or leftward to select the left item, the same as the rest. They need to access two levels with their non-dominant and dominant controller. There will be a short delay between the button being pressed and the menu being visible.

tions use menus to support command selection, which highly require visual engagement and ray pointing [39]. They could be friendly and familiar to novices, but become slow and disruptive for expert users and advanced use cases [20]. Users have to spend time on finding the item and match the item with the expected action in their mind, and a visible menu will occlude the main content [68]. On the other hand, due to a lack of buttons on controllers, compared with keyboards, shortcuts are rarely available for VR HMD users.

With the emergence of productivity tasks on VR devices [33, 129], there is a demand for developing rapid command selection mechanisms.

Marking Menu is a radial menu technique allowing users to select commands by drawing a quick stroke pointing to the item. It supports both novice (GUI) and expert (gestures) users with a mechanism of transitioning from novice to expert [21]. Most of the previous marking menu techniques are developed for 2D interfaces like desktops or tablets, with single 2D input devices like a mouse and a touch screen. Mid-air marking menus were also explored [39, 77, 108], but only unimanual input was discovered, and to the best of our knowledge, expert mode performance has not been evaluated. Furthermore, all of the VR marking menu techniques were only mapping 3 positional DOF information to 2D space [39], and compared among them. However, most of the current VR devices come with two controllers with 3 DOF positional and 3 DOF directional information, plus additional 2D touchpads or joysticks. Thus, two possible avenues remain the potential to increase efficiency: 1) fully investigate the untouched input information, including 3 DOF positional, directions, and 2D touchpad/joysticks input, and 2) leverage asymmetric bimanual input to facilitate multi-level menu access.

To fill in the gaps exploring these avenues, in this Chapter, we introduce Drum Menu (Figure 5.1), a bimanual rapid command access technique derived from radial menus, tailored for VR controllers. To address the two possible avenues mentioned above, Drum Menu explores: 1) three distinct ways to map 3 DOF positional, 3 DOF directional, and 2D joystick input to a 2D marking menu layout (i.e., Stroke Drum Menu, Pointing Drum Menu, and Joystick Drum Menu), and 2) integrated with bimanual interaction. All of the Drum Menus are using a two-level radial menu layout, which is similar to marking menus. Joystick Drum Menu (Figure 5.1a) will use the joystick angle to select the menu items. Stroke Drum Menu (Figure 5.1b) is similar to the traditional marking menu, which maps the 3D position trace to a 2D stroke on a plane parallel to the VR HMD. Pointing Drum Menu (Figure 5.1c) uses the pitch and yaw dimensions of the controller, mapping the tilt angle of the controller to the pointing direction to select menu items.

Drum Menu leverages bimanual interaction to support rapid two-level command access. Each hand is associated with one level of the menu: the non-dominant hand selects the first-level menu, while the dominant hand selects the second-level menu. In contrast to Conductor, where the two hands play asymmetric roles (context vs. precision), Drum Menu is designed as a more symmetric technique, since both hands contribute equivalent actions

by selecting menu items. Furthermore, Drum Menu supports both sequential and parallel usage: novices typically perform the two selections sequentially (first-level, then second-level), while experts can transition to parallel input, selecting both levels at once. This dual mode of operation aligns with theories of bimanual skill acquisition, where sequential asymmetric coordination can gradually evolve into fluent parallel actions. Compared to unimanual marking menus, which require sequential traversal of menu levels, the bimanual design of Drum Menu has the potential to significantly reduce command selection time once users are trained.

We conducted a controlled experiment to evaluate the performance of each Drum Menu under a different number of items. To evaluate the benefits of bimanual interaction, we also developed unimanual variants in the user study, which would require users to select the two levels sequentially, just like marking menus. We designed a command access task with a 4-item menu layout and an 8-item layout and evaluated each technique based on the task completion time and error rate for general performance. To better understand each technique, we also measured total hand distance and total hand rotation, and asymmetric time, which is the time difference between selecting the first level and the second level for bimanual variants. We found that the 4-item layout is a better layout in terms of accuracy and speed; however, the 8-item layout performs much worse than the 4-item layout. Among those 4-item layouts, the Joystick Drum Menu represents an effective mechanism to support shortcut command selection in immersive virtual environments, followed by the Stroke Drum Menu. For the 8-item layout, Stroke has the lowest error rate compared with others.

In summary, the contributions of this project include:

- A bimanual rapid command access technique, called Drum Menu, in VR with three variants (Stroke Drum Menu, Pointing Drum Menu, and Joystick Drum Menu) derived from a 2-level pie menu.
- An in-depth analysis of results for participants' performance among three variants compared with the unimanual version, which further generates design implications.

5.2 BACKGROUND AND RELATED WORK

In [Chapter 2](#) we have discussed bimanual interaction, and VR command access and 2D marking menu. Here, we present a more in-depth discussion specifically centred around rotational command selection.

5.2.1 *Rotational Command Selection*

Drum Menu design involves wrist rotation command selection in Rotation variants. Wrist rotation has been explored more generally in human motor control research, and it is known that rotational targeting rotational wrist positions follows Fitts's Law (confirmed initially by [\[26\]](#) with hand unencumbered and replicated by [\[25\]](#)). Wrist rotation has been proposed in mobile interaction as a motion gesture to control a smartwatch [\[25\]](#), an interaction leveraged by several other researchers in designing systems for interaction within two-dimensional and three-dimensional spaces (e.g. [\[59, 100\]](#)). In the domain of smartphone input, wrist rotation has been used to define a motion gesture that can act as a gestural delimiter [\[110\]](#), and wrist rotation motion gestures have been commercially adopted by Motorola smartphones as the *MotoAction* to invoke the camera app. Also, the TWuiST system [\[92\]](#) uses proprioceptive displacement of a smartphone, including wrist rotation, for smartphone-based interaction, and Crossnan et al. [\[25\]](#) studies the use of wrist rotation for command selection in sitting and walking mobile contexts. There were some work has been done on the VR menu with waist rotation [\[28, 39\]](#), but they all need a CD gain factor for mapping the limited waist rotation to 360° . That is less optimal for a rehearsal-based shortcut technique, which would break the connection between visual and physical memory.

5.3 DRUM MENU

This section outlines the motivation and design of the Drum Menu, introducing three variants: Joystick, Stroke, and Pointing-based Drum Menus.

5.3.1 *Input Modality in VR Controllers*

To make full use of the potential of VR controllers, we first analyze the main input modalities of these controllers that can be used to facilitate command access. Most commercial VR controllers comprise the following inputs:

- Boolean input: Button, on-touch/release event of buttons, joystick, touchpad, and triggers
- 1D input: Trigger
- 2D input: Joystick, touchpad
- 3D input: Position, direction

Radial menus require 2D input for angular selection and a boolean input for confirmation. For instance, pen input relies on (x, y) coordinates to calculate direction and pen-lift for confirmation. In VR, 2D input can come directly from joysticks/touchpads or be derived from 3D position/tilt. Confirmation can be triggered via buttons or by transforming 1D input (e.g., trigger pressure level).

Based on these inputs and user intuitiveness, we designed the following Drum Menu variants:

- **Joystick Drum Menu:** Joystick for direction + on-touch/release for confirmation, which is a natural way to select menu items for console or VR games.
- **Stroke Drum Menu:** Position for direction + button for confirmation, which is adopted from the traditional marking menu.
- **Pointing Drum Menu:** Tilt angle for direction + button for confirmation, which is similar to ray casting in VR.

5.3.2 *User Interactions with Different Drum Menu Variants*

Each Drum Menu follows a similar flow (*i.e.*, activate, select, and confirm), but with different input modalities.

Joystick Drum Menu: 5.2a, the joystick is touched to activate the menu, which appears after 333 ms if still being touched. The menu is attached to the center top of the controllers, parallel to the user and vertical to the ground.

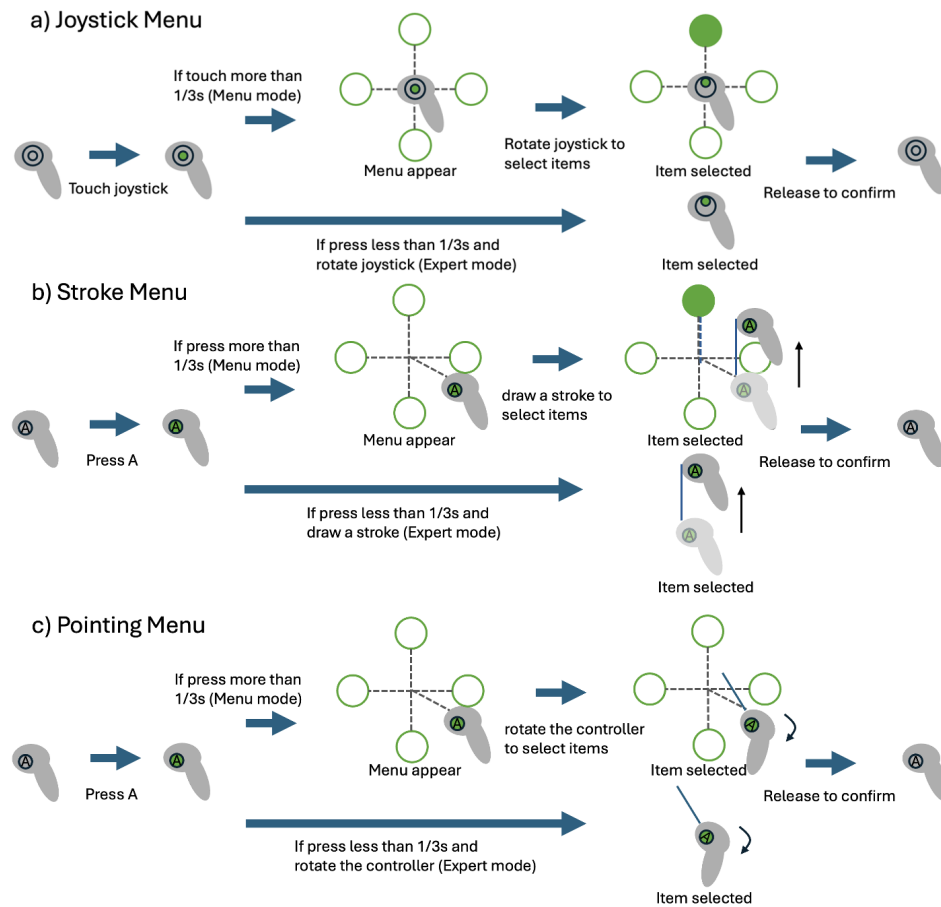


Figure 5.2: Workflows of the three Drum Menus.

Users select items by rotating the joystick, confirming by releasing. In the real interface (Figure 5.1a), joystick rotation will lead to the rotation of the white bar in the corresponding controllers. The selected item will be highlighted in yellow, which applies to the rest.

Stroke Drum Menu: As shown in Figure 5.2b, the Stroke Drum Menu is similar to the original marking menu. A button (lower button on the controller, A or X for Oculus controllers, applies to the Pointing Drum Menu as well) is pressed to activate the menu, which appears after 333 ms. Users draw a directional stroke in the air, confirmed upon button release. The menu is displayed 10 cm in front of the controller to avoid occlusion. The stroke is mapped to a 2D plane that is vertical to the ground and parallel to the VR HMD.

Pointing Drum Menu: As shown in Figure 5.2c, a button is pressed to activate the menu, which appears after 333 ms. The center of the menu is

positioned at the same height as the controller and 10 cm in front of it. It is positioned parallel to the HMD. The user rotates the controller in the pitch and yaw dimensions, and a ray is cast from the controller to help visualize the item selection. The selection is confirmed by releasing the button.

5.3.3 *Novice and Expert Modes for Bimanual Drum Menus*

All Drum Menus support novice and expert modes, differing in activation, selection, and display.

Bimanual novice mode: Users activate the first-level menu on the non-dominant controller. After 333 ms, the radial menu appears, and selection leads to the second-level menu on the dominant controller. The final command is confirmed upon selecting an item from this second-level menu.

Bimanual expert mode: A user activates the first-level menu on the non-dominant hand controller and quickly performs the same actions as in novice mode within a short period, without the menu appearing. The second-level menu is set up in real time based on the first-level selection, without explicitly confirming the first-level selection. The user then activates the second-level menu on the dominant hand controller and quickly selects an item without the menu appearing.

Users can also perform two actions simultaneously, which is even faster. We define simultaneous as the gap between the confirmation of two hands is less than 500ms, which also covers the case where the second-level menu is selected faster than the first-level menu shortly, as they operate independently.

5.4 EXPERIMENT

We conducted an experiment to compare the performance of the three Drum Menu techniques as well as their unimanual version in two different layouts (4-item and 8-item).

5.4.1 *Participants and Apparatus*

We recruited 12 right-handed participants, ages 23 to 31, 6 women and 6 men. Participants were recruited from a local university and a local community via mailing lists. They received \$20 e-gift card for their time and effort. This

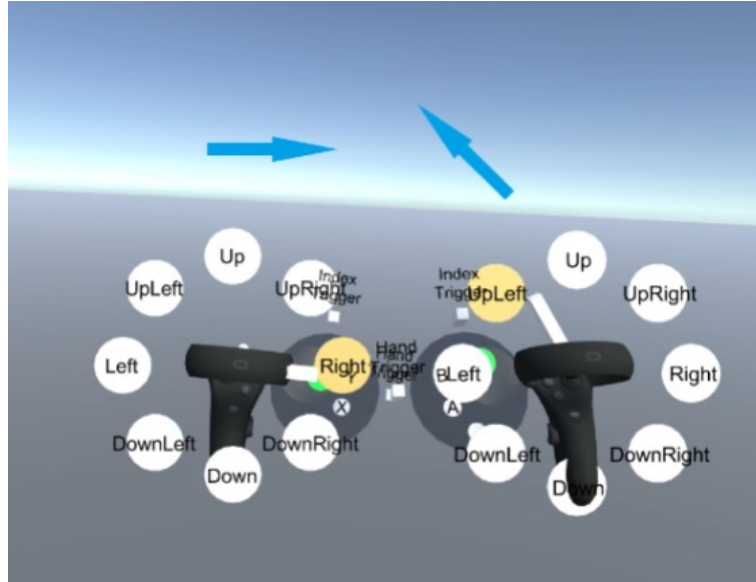


Figure 5.3: 8-item layout for Drum Menu.

study was conducted both in person and remotely. The experimental software was run on Oculus Quest 2 with Unity game engine version 2019.4.40f.

5.4.2 Procedure

Each participant first identified their dominant hand and then watched an instructional video detailing the technique. Once they understood the technique, they began a practice block with 8 trials, either in the 8-item layout or the 4-item layout. Each trial started with two arrow prompts (Figure 5.4 a) indicating the correct directions: the left arrow corresponded to the first level, and the right arrow corresponded to the second level. We avoided using word prompts to minimize the learning effect and focus on expert-level performance, as our goal was to use the marking menu as a shortcut mechanism in VR. Since all techniques were radial menus, using the same prompts was a fair approach across all techniques.

Participants then selected the item corresponding to the prompt. After each selection, feedback appeared in the air: a green checkmark for a correct selection or a red "x" for an incorrect one. Incorrect selections required participants to repeat the trial until they achieved the correct response.

Upon completing the practice block, participants advanced to the main experiment, which included 2 blocks of 16 trials each. Breaks were permitted between blocks. After finishing all trials for a specific menu condition and

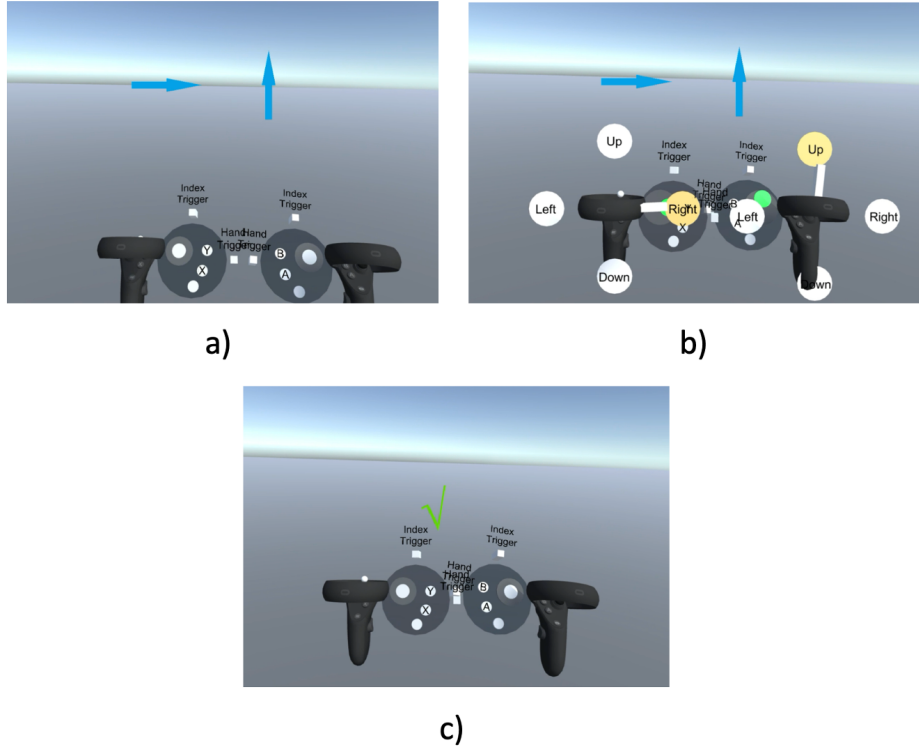


Figure 5.4: Task workflow: a) Two arrow appear in mid-air; b) Participant selects target item; (c) Selection confirmation.

layout, participants completed a questionnaire before transitioning to the next layout. Once both layouts were completed, they proceeded to the next technique. After testing all techniques, the researcher conducted a semi-structured interview with the participants, inquiring about their experiences with each technique across different layouts. They were also asked about their favourite using different numbers of hands and layouts.

Each study session lasted approximately 1 hour, including breaks.

5.4.3 Experimental Design

We adopted a within-subjects design with two independent variables:

- **TECHNIQUE** (6 levels: Bimanual-Joystick, Bimanual-Stroke, Bimanual-Rotation, Unimanual-Joystick, Unimanual-Stroke, Unimanual-Rotation),
- **LAYOUT** (2 levels: 4-item, 8-item) (see Figures 5.1 and 5.3).

Combinations of these factors are abbreviated as 4-Bi-Joy, 8-Uni-Str, etc. Trial sequences were counterbalanced via a Latin square.

Stimuli Selection: For the 4-item layout, all 16 possible items (4 directions $\times$ 4 sub-directions) were used. For the 8-item layout, following Kurtenbach’s marking menu paradigm [68], we selected 16 items evenly distributed across on-axis (Up, Down, Left, Right) and off-axis (UpLeft, UpRight, DownLeft, DownRight) directions. Four items were randomly chosen from each of four category combinations (on-on, on-off, off-on, off-off), ensuring equal representation across techniques.

Data Collection: Each participant completed 6 (Techniques) $\times$ 2 (Layouts) $\times$ 2 (Blocks) $\times$ 16 (Trials) = 384 trials.

Dependent Variables: We used the following measures in our study.

- *Task Completion Time:* Duration from prompt appearance to selection confirmation.
- *Error Rate:* Percentage of trials with more than one incorrect selection.
- *Total Hand Movement Distance:* Cumulative translation distance of both controllers.
- *Total Hand Rotation Angle:* Aggregate rotation angle of both hands.
- *Bimanual Asynchronize Time:* Temporal gap between first and second level confirmations in bimanual conditions. It could give us an idea if the participants are using the bimanual drum menu synchronously or asynchronously.
- *Preference:* Participants’ favourite technique under different numbers of items and layouts.

Additionally, subjective workload was assessed through the NASA-TLX questionnaire, providing six subscale scores: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration.

5.5 PERFORMANCE

For each combination of `TECHNIQUE` and `LAYOUT`, task completion times more than 3 standard deviations from the mean time were excluded as outliers. In total, 68 trials (1.4%) were removed.

In the analysis below, a `TECHNIQUE` $\times$ `LAYOUT` Repeated Measure two-way ANOVA with combined `TECHNIQUE` $\times$ `LAYOUT` Tukey HSD post hoc tests was used, unless noted otherwise. Normality was corrected using a Box-Cox transform. When the assumption of sphericity was violated, degrees of

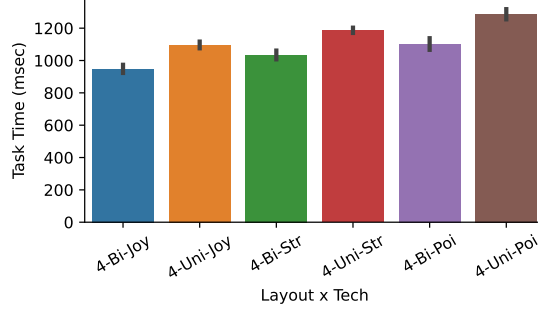


Figure 5.5: Task completion time for 4-item layout.

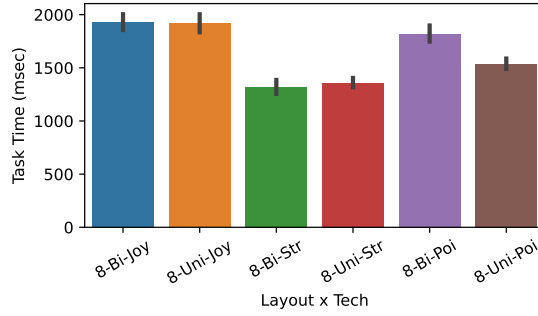


Figure 5.6: Task completion time for 8-item layout.

freedom were corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$).

5.5.1 Task Completion Time

In general, there was a significant main effect for **TECHNIQUE** on task completion time ($F_{5,55} = 5.04, p < .001, \eta_G^2 = 0.10$), **LAYOUT** ($F_{1,11} = 36.85, p < .0001, \eta_G^2 = 0.39$) and **TECHNIQUE** $\times$ **LAYOUT** ($F_{5,55} = 11.99, p < .0001, \eta_G^2 = 0.17$).

Figure 5.5 shows the task completion time for the 4-item layout (significance stars are omitted to maintain figure readability). A post hoc test revealed that Bimanual variants were faster than their Unimanual variants. Bimanual Joystick ($M = 947.8$ ms) and Bimanual Stroke ($M = 1034.1$ ms) were among the fastest techniques, with no significant difference between them. This could be because the 4-item layout provided enough space for users to select each item, and bimanual interaction allowed users to choose items in parallel without waiting to select the first level.

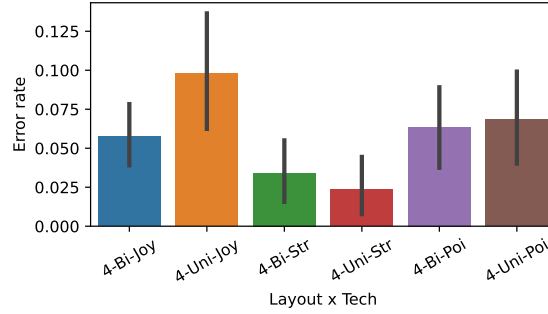


Figure 5.7: Error rate for 4-item layout.

Figure 5.6 shows the task completion time for the 8-item layout. A post hoc test revealed that only for Pointing technique Bimanual variants were slower than the Unimanual variants. Bimanual Stroke ($M = 1322.2$ ms) and Unimanual Stroke ($M = 1359.8$ ms) were the fastest ones. That could be mainly because the angle of each item is too small, which makes the users feel less confident in selecting without a visual search. However, if users draw the stroke long enough, they can still select accurately eyes-free, which turned out to be faster than the other variants.

5.5.2 Error Rate

In general, there was a significant main effect for TECHNIQUE on error rate ($F_{5,55} = 5.05$, $p < .0001$, $\eta_G^2 = 0.15$), LAYOUT ($F_{1,11} = 42.62$, $p < .001$, $\eta_G^2 = 0.30$).

Figure 5.7 shows the error rate for the 4-item layout. All of them are less than 0.1. A post hoc test revealed that no significant error rate difference was found among all techniques. Figure 5.8 shows the error rate for the 8-item layout. A post hoc test revealed that Stroke variants had the lowest error rate (0.14 for Bimanual, 0.078 for unimanual). The rest are all around or higher than 0.2. The same reason as observed in task completion time: if the stroke drawn is long enough, users can still select accurately without the menu showing up.

5.5.3 Total Hand Movement Distance

Total hand movement distance represents the movement of both hands from the prompt time to the final selection, which would impact users' physical fatigue. In general, there was a significant main effect for TECHNIQUE on

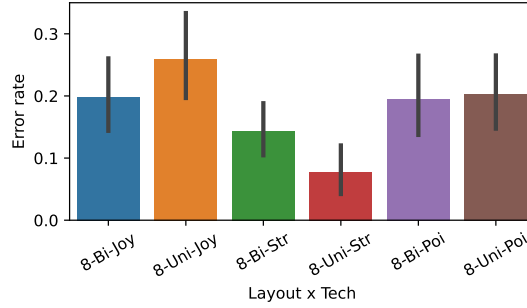


Figure 5.8: Error rate for 8-item layout.

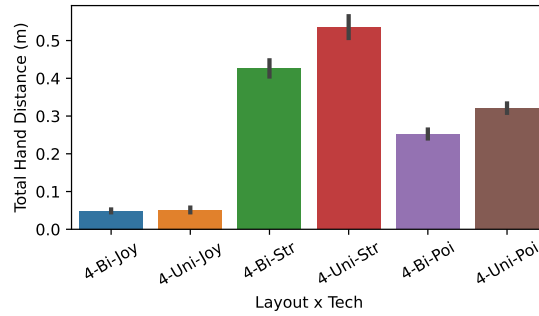


Figure 5.9: Total Hand Movement Distance for 4-item layout.

total hand movement distance ($F_{5,55} = 39.67$, $p < .0001$, $\eta_G^2 = 0.69$), LAYOUT ($F_{1,11} = 14.41$, $p < .0001$, $\eta_G^2 = 0.03$).

Figure 5.9 shows the total hand movement distance for the 4-item layout. A post hoc test revealed that Joystick variants ($M = 0.048$ m for Bimanual, $M = 0.05$ m for Unimanual) required the least hand movement. Pointing variants ($M = 0.25$ m for Bimanual, $M = 0.32$ m for Unimanual) required more distance. Stroke variants ($M = 0.42$ m for Bimanual, $M = 0.83$ m for Unimanual) moved the most. For the latter two variants, bimanual required less movement than unimanual variants.

Figure 5.10 shows the total hand movement distance for the 8-item layout, which is similar to the 4-item layout. A post hoc test revealed that Joystick variants ($M = 0.076$ m for Bimanual, $M = 0.058$ m for Unimanual) required the least hand movement. Pointing variants ($M = 0.33$ m for Bimanual, $M = 0.34$ m for Unimanual) required more distance. Stroke variants ($M = 0.52$ m for Bimanual, $M = 0.60$ m for Unimanual) moved the most.

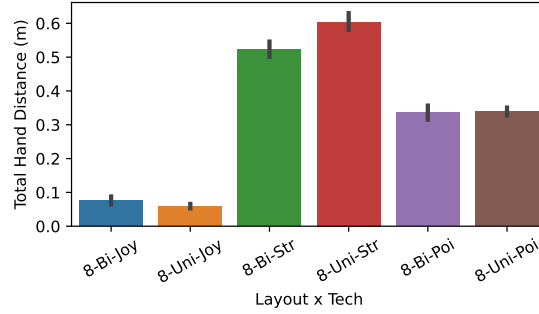


Figure 5.10: Total Hand Movement Distance for 8-item layout.

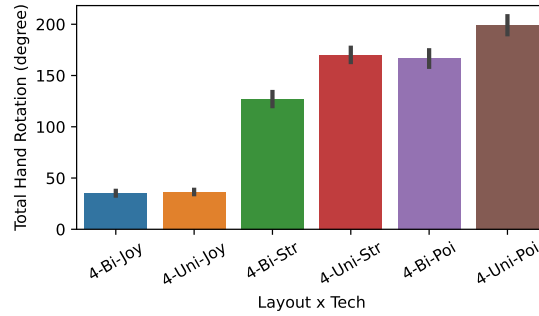


Figure 5.11: Total Hand Rotation Angle for 4-item layout.

5.5.4 Total Hand Rotation Angle

The Total Hand Rotation Angle refers to the degree of hand movement from the initial prompt to the final selection, which can influence users' physical fatigue. In general, there was a significant main effect for **TECHNIQUE** on total hand rotation angle ($F_{5,55} = 38.96$, $p < .0001$, $\eta_G^2 = 0.60$).

Figure 5.11 shows the total hand rotation angle for the 4-item layout. A post hoc test revealed that Joystick variants ($M = 35.1$ degree for Bimanual, $M = 36.3$ degree for Unimanual) required the least hand rotation. Stroke variants ($M = 126.98$ degree for Bimanual, $M = 170.19$ degree for Unimanual) required more rotation. Pointing variants ($M = 166.92$ degree for Bimanual, $M = 199.20$ degree for Unimanual) rotated the most.

Figure 5.12 shows the total hand rotation angle for the 8-item layout. A post hoc test revealed that Joystick variants ($M = 34.15$ degree for Bimanual, $M = 34.58$ degree for Unimanual) required the least hand rotation. Stroke variants ($M = 143.67$ degree for Bimanual, $M = 182.43$ degree for Unimanual)

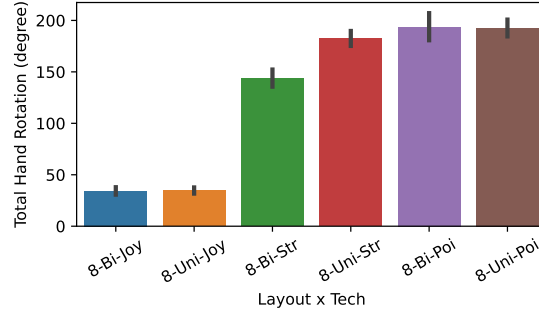


Figure 5.12: Total Hand Rotation Angle for 8-item layout.

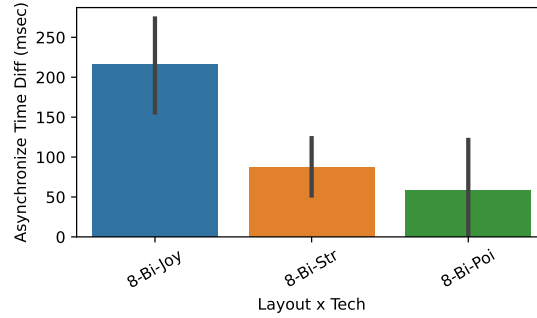


Figure 5.13: Asynchronize Time for 8-item layout. It represents how later the second level is confirmed than the first level.

required more rotation. Pointing variants ($M = 193.79$ degree for Bimanual, $M = 182.43$ degree for Unimanual) rotated the most.

5.5.5 Asynchronize Time

Asynchronize Time refers to the delay between confirming the first level and confirming the second level in bimanual drum menus. The confirmation timestamp is recorded as the moment the final item is selected. This metric provides insight into whether participants are using the bimanual drum menu synchronously or asynchronously. In general, there was a significant main effect for LAYOUT on asynchronize time ($F_{1,11} = 5.68$, $p < .05$, $\eta_G^2 = 0.09$).

For the 4-item layout, a post hoc test revealed that there is no difference among the three Bimanual techniques. The second level was slightly slower than the first level, but almost at the same time. Figure 5.13 shows asynchronize time for 8-item layout. A post hoc test revealed that Bimanual Joystick had a significantly longer asynchronize time than the others, which means the second-level confirmation time is slower.

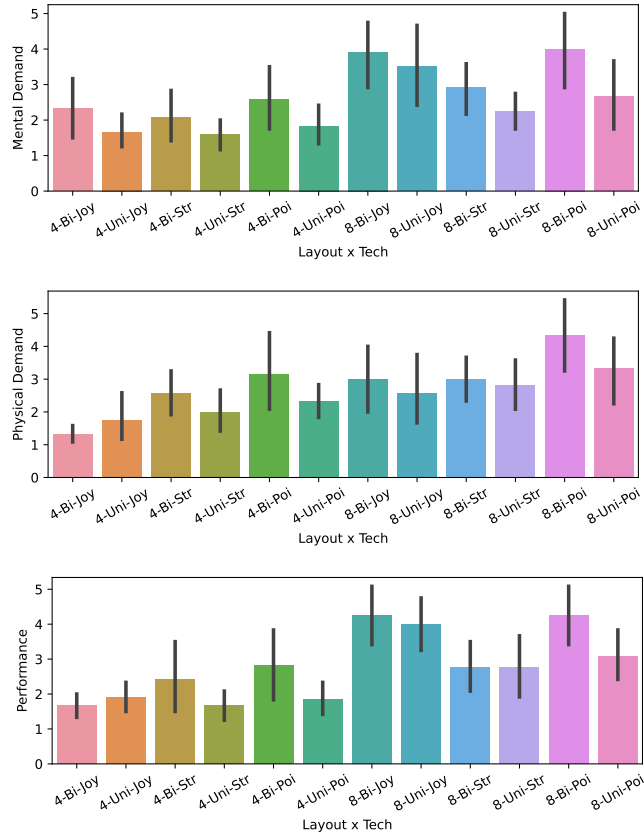


Figure 5.14: NASA-TLX Results.

5.5.6 NASA-TLX Questionnaire

We used Raw NASA-TLX to assess the perceived workload of the techniques from participants on a 7-point scale. Due to the number of techniques and questions in NASA-TLX, we highlight some key findings in the following. As shown in Figure 5.14, in general, the 4-item layout outperformed the 8-item layout. Moreover, 8-item Bimanual Pointing has significantly higher mental and physical demands than the others. That is mainly because the outward off-axis directions (down-left for left hand, down-right for right hand) were physically harder to reach, which requires more movement and focus to locate. Temporal pressures were similar across different conditions. For performance and effort questions, 8-item Bimanual Joystick and 8-item Bimanual Pointing had the lowest score. For frustration, the differences were not significant.

Table 5.1: User preferences for different techniques under NumberOfHand and Layout.

	Joystick	Stroke	Pointing
Unimanual-4Item	11	1	0
Bimanual-4Item	10	2	0
Unimanual-8Item	2	10	0
Bimanual-8Item	3	9	0

5.5.7 Preference Ranking

Table 5.1 shows that Joystick is preferred for the 4-item layout due to less physical movement, less task completion time, and less error rate. While Stroke is favored for the 8-item layout mainly because of its lower error rate compared with other techniques, even though it requires more movement than Joystick. Pointing is not preferred by any participants. It aligned with the results of Task Completion Time and Error Rate.

5.6 GENERAL DISCUSSION

5.6.1 Design Implications

In the following, we extend the discussion on the differences between many key factors characterizing the Drum Menu techniques. We derived these implications by coordinating our quantitative results with the interviews with the participants, as well as our observations during the study.

5.6.1.1 4-item v.s. 8-item layout

The quantitative results reveal the 8-item layout had a higher task completion time and error rate than the 4-item layouts. All participants reported that the 8-item layout was very difficult to select without carefully looking at the menu. One reason is that having more items reduces the range of each item, leading users to accidentally select neighbouring items. Participants also reported that off-axis items (UpLeft, UpRight, DownLeft, DownRight) were harder to select than on-axis items. They thought they were moving to a safe angle to select the off-axis items, but often ended up selecting the on-axis items instead. Due to the high error rate, participants had to activate the menu and visually locate the item, which resulted in less confidence and longer task

completion times. Only the Stroke variants were slightly better in terms of error rate. It was primarily because it allowed a larger movement range than the others. When participants moved their hands further, it was easier for them to locate the correct angle without accidentally selecting nearby items. “Moving further (using the Stroke Drum Menu) makes me feel more confident to select the correct range even without looking at the menu. For the others, I have to look at the menu.” (P5) But the error rate was still high compared to the 4-item layout.

5.6.1.2 *Bimanual v.s. Unimanual*

Given that our goal is to explore the potential of the marking menu as a shortcut, we focused more on the 4-item layout. For this layout, the bimanual variants were consistently faster than the unimanual variants, with no significant difference in error rates. Most participants preferred the bimanual approach, as it allowed them to access two levels simultaneously, saving time. One participant (P10) noted that the combined effort of two hands performing individual actions was less demanding than one hand performing two actions. However, 3 participants (P6, P9, P10) mentioned that the bimanual variants required extra mental effort to coordinate both hands. “I know I can select together, but sometimes it’s hard for me to coordinate two hands together.” (P6)

5.6.1.3 *Joystick v.s. Stroke. v.s. Pointing*

In general, the Joystick and Stroke variants were the most favored. The quantitative results also revealed a similar result, as Joystick had a shorter task completion time and Stroke had a lower error rate. Participants found it challenging to rotate to downward angles or outward angles (e.g., pointing right for the right hand, pointing left for the left hand), which required greater physical effort. Remapping the angles to more comfortable positions could possibly mitigate this issue. The Joystick was preferred for the 4-item layout due to its minimal physical demand (11 out of 12 participants for unimanual, 10 out of 12 for bimanual). However, the joystick may experience jitter issues upon release, as it can bounce back or shift angles if released too quickly. This can be mitigated by ensuring selection only occurs while the joystick is being touched or requiring it to be pushed beyond a minimum threshold. However, in practice, hardware signal inaccuracies or latency may prevent this logic from completely eliminating the jitter issue. Additionally,

the buttons themselves should also be taken into consideration, since there are only limited buttons in VR controllers and might also be occupied by other functionalities, like teleporting, which limits their use case. Stroke was preferred in the 8-item layout because of its lower error rate (10 out of 12 participants for unimanual, 9 out of 12 for bimanual), as discussed in the previous section regarding the 4-item v.s. 8-item layout.

5.6.1.4 *Design Recommendation*

In comparison to the existing desktop marking menus [6, 68, 152, 153], a key advantage of the Drum Menu lies in its support for bimanual input, enabling users to access commands across two levels simultaneously. Given the result, bimanual input helps users access commands faster.

As a conclusion of the discussion above, for shortcut purposes, we recommend the 4-item layout with the Bimanual Joystick Drum Menu due to its short task completion time and lower physical effort. In real-world applications, the joystick is [38] still a common way to activate pie menus. But if joysticks have been used by other functions, Stroke variants could also be a good choice.

When an 8-item layout is required to accommodate a larger command set, we recommend either the Bimanual or Unimanual Stroke Drum Menu. These variants provide the necessary flexibility to handle more commands while maintaining reasonable usability. However, it is important to note that the increased complexity of an 8-item layout may demand a higher level of visual engagement from users, as they may need to rely more on visual feedback to accurately select the desired command. This trade-off between command density and user attention should be carefully evaluated based on the specific context of use. This study primarily focuses on evaluating expert-level command selection time, which provides valuable insights into the efficiency of the proposed techniques. However, it is important to recognize that increasing the number of menu items may introduce additional challenges, such as memorization issues [152].

We recommend that designers carefully evaluate the complexity of the task and the cognitive load imposed on users when selecting an appropriate layout. For simpler tasks or applications with a limited command set — for example, a customized shortcut menu containing 16 commands — a 4-item layout may offer an optimal balance between efficiency and usability, minimizing the need for extensive memorization while maintaining fast selection times.

On the other hand, for more complex tasks requiring a larger command set, an 8-item layout might be justified, but designers should consider implementing additional visual cues, progressive disclosure techniques, or training mechanisms to support user learning and retention.

Ultimately, the choice of layout should align with the specific requirements of the application, the frequency of use, and the expected user expertise. Designers should also consider conducting user studies to assess memorability and error rates across different layouts, ensuring that the chosen design not only supports expert-level performance but also accommodates the needs of all users, regardless of their familiarity with the system. This holistic approach will help create more intuitive and effective interaction designs that enhance the overall user experience.

5.6.2 *Limitations and Future Work*

In our study, we used arrows as prompts to reduce the learning effect and minimize experiment time. However, this approach only measured performance in a scenario where all menu locations were fully memorized. As a result, we could not assess how the learning effect would manifest for users who gradually familiarize themselves with the Drum Menus over time. This limitation restricts our ability to generalize the findings to real-world usage, where users may not have prior knowledge of all item locations. Future studies should investigate the learning curve associated with Drum Menus, exploring how performance evolves as users become more proficient with both unimanual and bimanual input methods in various menu layouts.

Moreover, we discovered some 6-item layout marking menus. We did not include this in our experiment because there were already too many conditions. However, a 6-item layout may still be worth exploring, given the significant performance gap between the 4-item and 8-item layouts. Participants could consistently perform expert behavior for the 4-item layout but not for the 8-item layout due to the high error rate and the resulting lack of confidence. We are curious about how the 6-item layout would perform for each Drum Menu variant.

Third, we only tested with the two-level menu. However, a three-level menu was also explored in the past [153]. Since the 4-Item layout only contains $4 \times 4 = 16$ items in total, the three-level menu could expand this to $4 \times 4 \times 4 = 64$ items. However, the interaction design may need adjustment, and memorability should also be taken into consideration.

Lastly, in this study, we did not investigate the experience of transit. More studies may be needed to further examine how users who are already familiar with the unimanual menu would transition the experience to bimanual variants. It could be faster because of the existing knowledge, or slower due to the inertia. We would like to start with the unimanual Drum Menu and then switch to the bimanual Drum Menu and compare the performance. Additionally, we only recruited 12 participants in the experiment. Although we reported the effect sizes, most of which are considered large ($\eta_G^2 \geq 0.14$) [23], the findings could be further strengthened by additional future studies.

5.7 CONCLUSION

In this chapter, we present the Drum menu, which combines two-level menu selection through the coordinated use of both hands. We investigate three different ways to invoke the marking menu including Joystick, Stoke, and Pointing. Unimanual and bimanual variants were also compared. We tested their performance against two layouts, the 4-item layout and the 8-item layout. In general, we recommend the Bimanual Joystick variant with the 4-item layout as the shortcut mechanism in VR due to its short task completion time, low error rate, and low physical movement.

CONCLUSION

In conclusion, while unimanual input is the predominant choice in VR systems with HMDs, those interactions could be accelerated by executing them with two hands together. Leveraging the simultaneous capabilities of both hands, bimanual input expands the interaction dimension, complementing existing unimanual methods for elementary interactions such as 3D pointing, locomotion, and command access.

6.1 WHEN SHOULD WE CONSIDER BIMANUAL INPUT?

The three projects demonstrate the potential of bimanual input to improve interaction efficiency. At the same time, it is important to consider the best use cases for bimanual input and its trade-offs compared to unimanual input. Bimanual input can accelerate interactions by allowing an additional input dimension to execute another step simultaneously. However, it may come with a higher learning curve, as it requires coordinating both hands. Users must think about the sequence and timing of actions for each hand. This is similar to learning the violin: teachers often have students practice the left hand for fingering and the right hand for bowing separately. Once sufficient muscle memory is developed, the student can perform the piece harmoniously. Therefore, designers should treat bimanual input as an advanced interaction technique. It is ideal for users who need higher efficiency, while unimanual input may be easier for beginners to pick up. Additionally, the coordination challenge of bimanual input can be framed as a feature, appealing to users—such as gamers—who enjoy mastering difficult, skill-based interactions.

6.2 FUTURE RESEARCH

I outline future research opportunities, offering preliminary ideas for their design, technical implementation, and user studies to explore how people perceive the proposed interaction techniques.

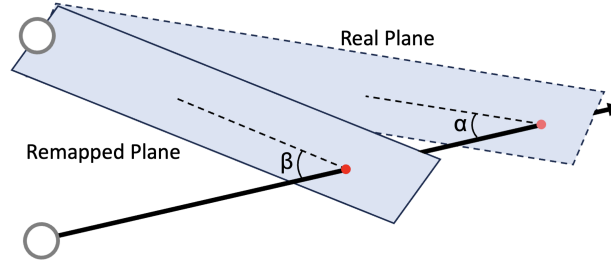


Figure 6.1: Remapping the plane to mitigate the far distance imprecision issue. The plane with a dashed line is the real plane with a small plane-ray angle α . The plane with a solid line is the remapped visible plane with a large plane-ray β .

6.2.1 Improve Conductor Performance Under Long Distance

In Chapter 3, the maximum distance evaluated was 10 m. However, VR spaces are theoretically unbounded. This implies that the conductor's effective range could also be unlimited (i.e., when the plane and the ray are parallel). Nevertheless, precision tends to degrade at longer distances. For instance, if the user's hands are 0.5 m apart and the angle between them is 1° , the intersection point lies approximately 14.3 m away. A slight reduction of just 0.5° —a small, natural jitter—can shift the intersection to 28.7 m. This sensitivity raises a key question: how can we mitigate such instability?

One practical approach is to be less ambitious with range. Although VR environments have no physical bounds, users typically interact with objects that are within close proximity—just like in the real world. Therefore, it is reasonable to trade some range for greater control precision. A potential solution involves remapping the interaction plane when the angle between the plane and ray becomes too small (Figure 6.1). Crucially, by keeping the ray fixed and only adjusting the plane, we affect only one dimension of the cursor's movement (i.e., its distance along the ray). This preserves more intuitive control compared to altering the ray itself.

6.2.2 Involve more body parts

Although the theme of this thesis is bimanual interaction, the ultimate goal is to use the non-dominant hand to support and facilitate the dominant hand.

But can we involve additional body parts beyond the non-dominant hand to introduce new dimensions of input? For example, Wagner et al. [134] explored the use of gaze as an additional input in a VR selection task—using eye gaze for pointing and the dominant hand to control distance. I believe other body parts, such as head movement or feet, could also be integrated to work simultaneously with the dominant hand. So far, only interactions involving two body parts were primarily discussed. But what if more than two are used? Imagine gaze and both hands working in concert—something that already feels intuitive in everyday life. Typing, which I’m doing right now, is a good example: both hands press the keyboard while the eyes monitor the screen to ensure accuracy. Playing the piano takes this even further, incorporating the feet to control the soft pedal. These naturally coordinated actions in everyday tasks suggest rich, underexplored potential for multi-part interaction in virtual environments.

6.2.3 *Involve fewer body parts*

While I have discussed the possibility of incorporating more body parts, it is also worth considering the opposite: reducing the number of body parts involved. Can we accomplish the task using only one hand by enhancing its expressiveness? When I talk about using one hand, I refer to the palm, the back of the hand, the five fingers, and potentially the forearm. If you’re a Marvel fan like me, you will recall how Iron Man equips his hands with beams from the palms, missiles from the back of the hand, and plasma torches from the fingers. With the increasing capabilities of hand-tracking technology in modern VR HMDs, it is worth exploring how a single hand, through its dexterity, can perform tasks traditionally requiring two hands. For instance, Bird [117] investigated one-handed pointing in 3D environments by determining depth through the hand’s opening size. These explorations open new possibilities for simplifying interaction design while maintaining rich expressiveness and control.

6.2.4 *Bimanual Haptics*

So far, I have primarily discussed input into the virtual environment. However, hands are not only input tools—they are also vital sensory organs for perceiving the physical world. For example, Haptic Links [124] explored

electro-mechanical connectors that physically link two VR controllers to provide dynamic and controllable stiffness and force feedback. Another approach is pseudo-haptics [73, 111], which creates a sensation of force feedback through a mismatch between the real hand and the visual representation of the virtual hand. For instance, if a virtual object held in the hand moves less than the actual hand, users perceive it as heavy. In bimanual interaction, holding a heavy object with both hands typically feels lighter than holding it with one, and the weight distribution across the object can further affect the perceived force in each hand. If the left side of the object is heavier, the left hand should feel more load than the right. Understanding and simulating these tactile perceptions can significantly enhance the realism and intuitiveness of bimanual interactions in virtual environments.

6.3 FINAL WORD

The inspiration for this thesis stemmed from a simple yet profound observation: the human body is naturally adept at coordinated bimanual actions—something I see in everyday tasks like typing, playing musical instruments, or even tying shoelaces. Yet, much of our interaction with virtual environments continues to rely predominantly on one hand, leaving a vast reservoir of potential untapped. This work has been a journey toward unlocking that potential within the domain of VR.

Through the development of Conductor, Fly the Moon to Me, and Drum Menu, this thesis explored how simultaneous bimanual interaction can simplify and accelerate fundamental tasks like pointing, navigating, and accessing commands in 3D spaces. Each project demonstrated that by rethinking the assumptions of sequential, unimanual interaction, we can create more fluid, intuitive, and efficient user experiences.

Of course, bimanual input is not a panacea—it introduces its own design challenges and cognitive load considerations. However, this work has shown that when applied thoughtfully, bimanual techniques can significantly enhance the usability of immersive environments. More importantly, these contributions lay the groundwork for future exploration of richer input models—those that integrate multiple limbs, senses, and even modalities, such as gaze or voice, into a cohesive interaction framework.

As VR technologies continue to evolve and find their way into more aspects of daily life, the insights and design approaches from this thesis aim to inspire new ways of thinking about interaction, not just as a technical necessity, but

as a deeply human activity. My hope is that this work encourages others to look beyond convention, to experiment boldly, and to continue pushing the boundaries of what is possible in immersive interaction.

ENCORE: WHY CONCERT?

As this thesis reaches its final note, one may ask: why is it named **Concert**? The answer resonates in its very architecture (see [Figure 7.1](#)). **Conductor** represents the musician, **Fly the Moon to Me** embodies the musical piece, and **Drum Menu** serves as the instrument. Together, these works reflect the core elements of a concert: musician, piece, and instrument—united in expressive harmony. Just as the **Performance** sections within each work measured its execution, so too does this thesis—as a whole—perform its central theme: exploring the harmonious potential of virtual experiences shaped by our two hands.

Thank you, my wonderful reader—or **audience**—for attending this performance. May the echoes of this concert linger with you.

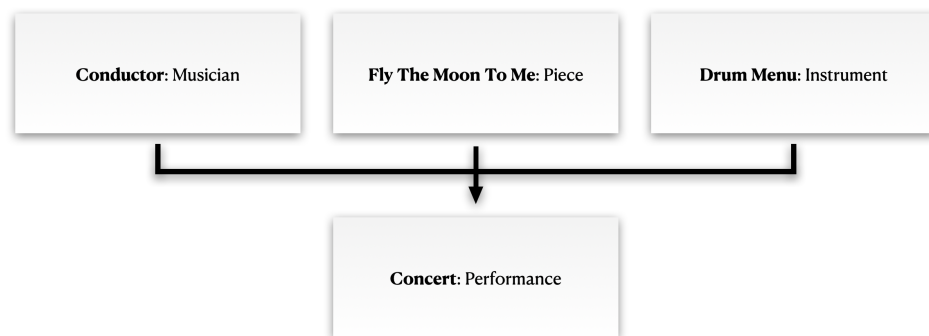


Figure 7.1: The Concert (the performance) is composed of three projects: Conductor (the musician), Fly the Moon to Me (the piece), and Drum Menu (the instrument).

BIBLIOGRAPHY

- [1] Parastoo Abtahi, Mar Gonzalez-Franco, Eyal Ofek, and Anthony Steed. "I'm a Giant: Walking in Large Virtual Environments at High Speed Gains." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–13. ISBN: 9781450359702. DOI: [10.1145/3290605.3300752](https://doi.org/10.1145/3290605.3300752). URL: <https://doi.org/10.1145/3290605.3300752>.
- [2] Sunggeun Ahn, Stephanie Santosa, Mark Parent, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. "StickyPie: A Gaze-Based, Scale-Invariant Marking Menu Optimized for AR/VR." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–16.
- [3] Ferran Argelaguet and Carlos Andujar. "A survey of 3D object selection techniques for virtual environments." In: *Computers & Graphics* 37:3 (2013), pp. 121–136.
- [4] Jonas Auda, Max Pascher, and Stefan Schneegass. "Around the (Virtual) World: Infinite Walking in Virtual Reality Using Electrical Muscle Stimulation." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–8. ISBN: 9781450359702. DOI: [10.1145/3290605.3300661](https://doi.org/10.1145/3290605.3300661). URL: <https://doi.org/10.1145/3290605.3300661>.
- [5] Takumi Azai, Shuhei Ogawa, Mai Otsuki, Fumihisa Shibata, and Asako Kimura. "Selection and Manipulation Methods for a Menu Widget on the Human Forearm." In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 357–360. ISBN: 9781450346566. DOI: [10.1145/3027063.3052959](https://doi.org/10.1145/3027063.3052959). URL: <https://doi.org/10.1145/3027063.3052959>.
- [6] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. "Flower Menus: A New Type of Marking Menu with Large Menu Breadth, within Groups and Efficient Expert Mode Memorization." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '08. Napoli, Italy: As-

- sociation for Computing Machinery, 2008, 15–22. ISBN: 9781605581415. DOI: [10.1145/1385569.1385575](https://doi.org/10.1145/1385569.1385575). URL: <https://doi.org/10.1145/1385569.1385575>.
- [7] Marc Baloup, Thomas Pietrzak, and Géry Casiez. “RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–12. ISBN: 9781450359702. DOI: [10.1145/3290605.3300331](https://doi.org/10.1145/3290605.3300331). URL: <https://doi.org/10.1145/3290605.3300331>.
- [8] Anil Ufuk Batmaz and Wolfgang Stuerzlinger. “Effect of Fixed and Infinite Ray Length on Distal 3D Pointing in Virtual Reality.” In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–10. ISBN: 9781450368193. DOI: [10.1145/3334480.3382796](https://doi.org/10.1145/3334480.3382796). URL: <https://doi.org/10.1145/3334480.3382796>.
- [9] François Bérard, Jessica Ip, Mitchel Benovoy, Dalia El-Shimy, Jeffrey R Blum, and Jeremy R Cooperstock. “Did “Minority Report” get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task.” In: *IFIP Conference on Human-Computer Interaction*. Springer. 2009, pp. 400–414.
- [10] Jiwan Bhandari, Paul MacNeilage, and Eelke Folmer. “Teleportation without Spatial Disorientation Using Optical Flow Cues.” In: *Proceedings of Graphics Interface 2018*. GI 2018. Toronto, Ontario: Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, 2018, pp. 162–167. ISBN: 978-0-9947868-3-8. DOI: [10.20380/GI2018.22](https://doi.org/10.20380/GI2018.22).
- [11] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. “Bimanual gesture keyboard.” In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST ’12. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2012, 137–146. ISBN: 9781450315807. DOI: [10.1145/2380116.2380136](https://doi.org/10.1145/2380116.2380136). URL: <https://doi.org/10.1145/2380116.2380136>.
- [12] David Bond and Madelein Nyblom. *Evaluation of four different virtual locomotion techniques in an interactive environment*. 2019.

- [13] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. "Point & Teleport Locomotion Technique for Virtual Reality." In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '16. Austin, Texas, USA: Association for Computing Machinery, 2016, 205–216. ISBN: 9781450344562. DOI: [10.1145/2967934.2968105](https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2967934.2968105). URL: <https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2967934.2968105>.
- [14] Fabio Buttussi and Luca Chittaro. "Locomotion in place in virtual reality: A comparative evaluation of joystick, teleport, and leaning." In: *IEEE transactions on visualization and computer graphics* 27.1 (2019), pp. 125–136.
- [15] Géry Casiez, Nicolas Roussel, and Daniel Vogel. "1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 2527–2530.
- [16] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. "The impact of control-display gain on user performance in pointing tasks." In: *Human-computer interaction* 23.3 (2008), pp. 215–250.
- [17] Yuan Chen, Junwei Sun, Qiang Xu, Edward Lank, Pourang Irani, and Wei Li. "Global Scene Filtering, Exploration, and Pointing in Occluded Virtual Space." In: *Human-Computer Interaction – INTERACT 2021*. Cham: Springer International Publishing, 2021, pp. 156–176. ISBN: 978-3-030-85607-6.
- [18] Sohan Chowdhury, A K M Amanat Ullah, Nathan Bruce Pelmore, Pourang Irani, and Khalad Hasan. "WriArm: Leveraging Wrist Movement to Design Wrist+Arm Based Teleportation in VR." In: *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2022, pp. 317–325. DOI: [10.1109/ISMAR55827.2022.00047](https://doi.org/10.1109/ISMAR55827.2022.00047).
- [19] Andy Cockburn, Carl Gutwin, and Saul Greenberg. "A Predictive Model of Menu Performance." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: Association for Computing Machinery, 2007, 627–636. ISBN: 9781595935939. DOI: [10.1145/1240624.1240723](https://doi.org/10.1145/1240624.1240723). URL: <https://doi.org/10.1145/1240624.1240723>.

- [20] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. "Supporting Novice to Expert Transitions in User Interfaces." In: *ACM Comput. Surv.* 47.2 (Nov. 2014). ISSN: 0360-0300. DOI: [10.1145/2659796](https://doi.org/10.1145/2659796). URL: <https://doi.org/10.1145/2659796>.
- [21] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. "Supporting Novice to Expert Transitions in User Interfaces." In: *ACM Comput. Surv.* 47.2 (2014). ISSN: 0360-0300. DOI: [10.1145/2659796](https://doi.org/10.1145/2659796). URL: <https://doi.org/10.1145/2659796>.
- [22] Christopher Coffin and Tobias Höllerer. "Interactive Perspective Cut-away Views for General 3D Scenes." In: *3D User Interfaces (3DUI'06)* (2006), pp. 25–28.
- [23] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. routledge, 2013.
- [24] Noah Coomer, Sadler Bullard, William Clinton, and Betsy Williams-Sanders. "Evaluating the effects of four VR locomotion methods: joystick, arm-cycling, point-tugging, and teleporting." In: *Proceedings of the 15th ACM symposium on applied perception*. 2018, pp. 1–8.
- [25] Andrew Crossan, John Williamson, Stephen Brewster, and Rod Murray-Smith. "Wrist Rotation for Interaction in Mobile Contexts." In: *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI '08. Amsterdam, The Netherlands: Association for Computing Machinery, 2008, 435–438. ISBN: 9781595939524. DOI: [10.1145/1409240.1409307](https://doi.org/10.1145/1409240.1409307). URL: <https://doi.org/10.1145/1409240.1409307>.
- [26] E. R. F. W. Crossman and P. J. Goodeve. "Feedback Control of Hand-Movement and Fitts' Law." In: *The Quarterly Journal of Experimental Psychology Section A* 35.2 (1983), pp. 251–278. DOI: [10.1080/14640748308402133](https://doi.org/10.1080/14640748308402133).
- [27] Raimund Dachsel and Anett Hübner. "Three-dimensional menus: A survey and taxonomy." In: *Computers & Graphics* 31.1 (2007), pp. 53–65.
- [28] Gabriel Di Domenico, Natan Berwaldt, Mauren W. D'Avila, and Cesar Pozzer. "Radial Menu for Virtual Reality Based on Wrist Rotation." In: *Proceedings of the 25th Symposium on Virtual and Augmented Reality*. SVR '23. Rio Grande, Brazil: Association for Computing Machinery, 2024, 238–242. ISBN: 9798400709432. DOI: [10.1145/3625008.3625022](https://doi.org/10.1145/3625008.3625022). URL: <https://doi.org/10.1145/3625008.3625022>.

- [29] Jose L. Dorado and Pablo A. Figueroa. "Ramps are better than stairs to reduce cybersickness in applications based on a HMD and a Gamepad." In: *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. 2014, pp. 47–50. DOI: [10.1109/3DUI.2014.6798841](https://doi.org/10.1109/3DUI.2014.6798841).
- [30] Niklas Elmqvist. "BalloonProbe: reducing occlusion in 3D using interactive space distortion." In: *VRST '05*. 2005.
- [31] Niklas Elmqvist and Mihail Eduard Tudoreanu. "Occlusion Management in Immersive and Desktop 3D Virtual Environments: Theory and Evaluation." In: *Int. J. Virtual Real.* 6 (2007), pp. 21–32.
- [32] Jeff Feasel, Mary C Whitton, Laura Kassler, Frederick P Brooks, and Michael D Lewek. "The integrated virtual environment rehabilitation treadmill system." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.3 (2011), pp. 290–297.
- [33] Blender Foundation. *Home of the blender project - free and open 3D creation software*. URL: <https://www.blender.org/>.
- [34] Sebastian Freitag, Dominik Rausch, and Torsten Kuhlen. "Reorientation in virtual environments using interactive portals." In: *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. 2014, pp. 119–122. DOI: [10.1109/3DUI.2014.6798852](https://doi.org/10.1109/3DUI.2014.6798852).
- [35] Jann Philipp Freiwald, Oscar Ariza, Omar Janeh, and Frank Steinicke. "Walking by Cycling: A Novel In-Place Locomotion User Interface for Seated Virtual Reality Experiences." In: *CHI*. 2020, pp. 1–12.
- [36] Julian Frommel, Sven Sonntag, and Michael Weber. "Effects of Controller-Based Locomotion on Player Experience in a Virtual Reality Exploration Game." In: *Proceedings of the 12th International Conference on the Foundations of Digital Games*. FDG '17. Hyannis, Massachusetts: Association for Computing Machinery, 2017. ISBN: 9781450353199. DOI: [10.1145/3102071.3102082](https://doi.org/10.1145/3102071.3102082). URL: <https://doi.org/10.1145/3102071.3102082>.
- [37] Markus Funk, Florian Müller, Marco Fendrich, Megan Shene, Moritz Kolvenbach, Niclas Dobbartin, Sebastian Günther, and Max Mühlhäuser. "Assessing the Accuracy of Point & Teleport Locomotion with Orientation Indication for Virtual Reality Using Curved Trajectories." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for

- Computing Machinery, 2019, 1–12. ISBN: 9781450359702. DOI: [10.1145/3290605.3300377](https://doi.org/10.1145/3290605.3300377). URL: <https://doi.org/10.1145/3290605.3300377>.
- [38] Epic Games. *XR - Unreal Engine*. <https://www.unrealengine.com/en-US/xr>. Accessed: 2024-10-17. n.d.
- [39] S. Gebhardt, S. Pick, F. Leithold, B. Hentschel, and T. Kuhlen. “Extended Pie Menus for Immersive Virtual Environments.” In: *IEEE Transactions on Visualization and Computer Graphics* 19.4 (2013), pp. 644–651.
- [40] Gravity Sketch. *Gravity Sketch: Collaborative 3D Design Platform*. <https://gravitiesketch.com/>. 3D design and collaboration software. 2016.
- [41] Tovi Grossman and Ravin Balakrishnan. “The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor’s activation area.” In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2005, pp. 281–290.
- [42] 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*. 2006, pp. 3–12.
- [43] Yves Guiard. “Asymmetric Division of Labor in Human Skilled Bimanual Action.” In: *Journal of Motor Behavior* 19.4 (1987). PMID: 15136274, pp. 486–517. DOI: [10.1080/00222895.1987.10735426](https://doi.org/10.1080/00222895.1987.10735426). eprint: <https://doi.org/10.1080/00222895.1987.10735426>. URL: <https://doi.org/10.1080/00222895.1987.10735426>.
- [44] Yves Guiard. “The kinematic chain as a model for human asymmetrical bimanual cooperation.” In: *Advances in Psychology*. Vol. 55. Elsevier, 1988, pp. 205–228.
- [45] *HalfLife Alyx*. 2020. URL: https://store.steampowered.com/app/546560/HalfLife_Alyx/.
- [46] Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D Wilson. “Realitycheck: Blending virtual environments with situated physical reality.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12.
- [47] Jay Henderson, Sylvain Malacria, Mathieu Nancel, and Edward Lank. “Investigating the Necessity of Delay in Marking Menu Invocation.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI’20)*. 2020, p. 13.

- [48] Jay Henderson, Sachi Mizobuchi, Wei Li, and Edward Lank. “Exploring Cross-Modal Training via Touch to Learn a Mid-Air Marking Menu Gesture Set.” In: *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. Mobile-HCI '19. Taipei, Taiwan: Association for Computing Machinery, 2019. ISBN: 9781450368254. DOI: [10.1145/3338286.3340119](https://doi.org/10.1145/3338286.3340119). URL: <https://doi.org/10.1145/3338286.3340119>.
- [49] Juan David Hincapié-Ramos, Kasim Özacar, Pourang P Irani, and Yoshifumi Kitamura. “GyroWand: An approach to IMU-based raycasting for augmented reality.” In: *IEEE Computer Graphics and Applications* 36.2 (2016), pp. 90–96.
- [50] Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten, and Neal Kassell. “Cooperative bimanual action.” In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Atlanta, Georgia, USA: Association for Computing Machinery, 1997, 27–34. ISBN: 0897918029. DOI: [10.1145/258549.258571](https://doi.org/10.1145/258549.258571). URL: <https://doi.org/10.1145/258549.258571>.
- [51] 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 23rd Annual ACM Symposium on User Interface Software and Technology*. UIST '10. New York, New York, USA: Association for Computing Machinery, 2010, 27–36. ISBN: 9781450302715. DOI: [10.1145/1866029.1866036](https://doi.org/10.1145/1866029.1866036). URL: <https://doi.org/10.1145/1866029.1866036>.
- [52] Kasper Hornbæk and Antti Oulasvirta. “What Is Interaction?” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 5040–5052. ISBN: 9781450346559. DOI: [10.1145/3025453.3025765](https://doi.org/10.1145/3025453.3025765). URL: <https://doi.org/10.1145/3025453.3025765>.
- [53] Bart Howard. *Fly Me to the Moon*. Audio recording. Performed by Frank Sinatra. Originally published in 1954. From the album *It Might As Well Be Swing*. Reprise Records. 1964.
- [54] Hiroo Iwata. “Walking about virtual environments on an infinite floor.” In: *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. IEEE. 1999, pp. 286–293.

- [55] M. P. Jacob Habgood, David Moore, David Wilson, and Sergio Alapont. “Rapid, Continuous Movement Between Nodes as an Accessible Virtual Reality Locomotion Technique.” In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2018, pp. 371–378. DOI: [10.1109/VR.2018.8446130](https://doi.org/10.1109/VR.2018.8446130).
- [56] Beverly K Jaeger and Ronald R Mourant. “Comparison of simulator sickness using static and dynamic walking simulators.” In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 45. 27. SAGE Publications Sage CA: Los Angeles, CA. 2001, pp. 1896–1900.
- [57] Dhruv Jain, Misha Sra, Jingru Guo, Rodrigo Marques, Raymond Wu, Justin Chiu, and Chris Schmandt. “Immersive Terrestrial Scuba Diving Using Virtual Reality.” In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’16. San Jose, California, USA: Association for Computing Machinery, 2016, 1563–1569. ISBN: 9781450340823. DOI: [10.1145/2851581.2892503](https://doi.org/10.1145/2851581.2892503). URL: <https://doi.org/10.1145/2851581.2892503>.
- [58] Vineet Kamboj, Tuhin Bhuyan, and Jayesh S. Pillai. “Vertical Locomotion in VR Using Full Body Gestures.” In: *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*. VRST ’19. Parramatta, NSW, Australia: Association for Computing Machinery, 2019. ISBN: 9781450370011. DOI: [10.1145/3359996.3364770](https://doi.org/10.1145/3359996.3364770). URL: <https://doi.org/10.1145/3359996.3364770>.
- [59] 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*. 2016, pp. 128–135.
- [60] Daniel Kharlamov, Brandon Woodard, Liudmila Tahai, and Krzysztof Pietroszek. “TickTockRay: Smartwatch-Based 3D Pointing for Smartphone-Based Virtual Reality.” In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. VRST ’16. Munich, Germany: Association for Computing Machinery, 2016, 365–366. ISBN: 9781450344913. DOI: [10.1145/2993369.2996311](https://doi.org/10.1145/2993369.2996311). URL: <https://doi.org/10.1145/2993369.2996311>.
- [61] DongHoon Kim, Dongyun Han, Siyeon Bak, and Isaac Cho. “Crossing Rays: Evaluation of Bimanual Mid-air Selection Techniques in an Immersive Environment.” In: *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Los Alamitos, CA, USA: IEEE

- Computer Society, Oct. 2024, pp. 329–338. DOI: [10.1109/ISMAR62088.2024.00047](https://doi.org/10.1109/ISMAR62088.2024.00047). URL: <https://doi.ieeecomputersociety.org/10.1109/ISMAR62088.2024.00047>.
- [62] Hyun K Kim, Jaehyun Park, Yeongcheol Choi, and Mungyeong Choe. “Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment.” In: *Applied ergonomics* 69 (2018), pp. 66–73.
- [63] Ji-Sun Kim, Denis Gračanin, Krešimir Matković, and Francis Quek. “Finger Walking in Place (FWIP): A Traveling Technique in Virtual Environments.” In: *Proceedings of the 9th International Symposium on Smart Graphics*. SG ’08. Rennes, France: Springer-Verlag, 2008, 58–69. ISBN: 9783540854104. DOI: [10.1007/978-3-540-85412-8_6](https://doi.org/10.1007/978-3-540-85412-8_6). URL: https://doi.org/10.1007/978-3-540-85412-8_6.
- [64] Kenrick Kin, Björn Hartmann, and Maneesh Agrawala. “Two-handed marking menus for multitouch devices.” In: *ACM Trans. Comput.-Hum. Interact.* 18.3 (Aug. 2011). ISSN: 1073-0516. DOI: [10.1145/1993060.1993066](https://doi.org/10.1145/1993060.1993066). URL: <https://doi.org/10.1145/1993060.1993066>.
- [65] Konstantin Klamka, Patrick Reipschläger, and Raimund Dachsel. “CHARM: Cord-Based Haptic Augmented Reality Manipulation.” In: June 2019, pp. 96–114. ISBN: 978-3-030-21606-1. DOI: [10.1007/978-3-030-21607-8_8](https://doi.org/10.1007/978-3-030-21607-8_8).
- [66] Rick Komerska and Colin Ware. “A study of haptic linear and pie menus in a 3D fish tank VR environment.” In: *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS ’04. Proceedings.* (2004), pp. 224–231. URL: <https://api.semanticscholar.org/CorpusID:15234153>.
- [67] Regis Kopper, Felipe Bacim, and Doug A Bowman. “Rapid and accurate 3D selection by progressive refinement.” In: *2011 IEEE symposium on 3D user interfaces (3DUI)*. IEEE. 2011, pp. 67–74.
- [68] Gordon Paul Kurtenbach. “The Design and Evaluation of Marking Menus.” PhD thesis. CAN, 1993.
- [69] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

- [70] Chengyuan Lai, Ryan P. McMahan, and James Hall. "March-and-Reach: A realistic ladder climbing technique." In: *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 2015, pp. 15–18. DOI: [10.1109/3DUI.2015.7131719](https://doi.org/10.1109/3DUI.2015.7131719).
- [71] Eike Langbehn, Paul Lubos, Gerd Bruder, and Frank Steinicke. "Bending the Curve: Sensitivity to Bending of Curved Paths and Application in Room-Scale VR." In: *IEEE Transactions on Visualization and Computer Graphics* 23.4 (2017), pp. 1389–1398. DOI: [10.1109/TVCG.2017.2657220](https://doi.org/10.1109/TVCG.2017.2657220).
- [72] Celine Latulipe. "Symmetric interaction in the user interface." In: *Companion Proceedings of UIST* (2004), pp. 63–66.
- [73] A. Lecuyer, S. Coquillart, A. Kheddar, P. Richard, and P. Coiffet. "Pseudo-haptic feedback: can isometric input devices simulate force feedback?" In: *Proceedings IEEE Virtual Reality 2000 (Cat. No.00CB37048)*. 2000, pp. 83–90. DOI: [10.1109/VR.2000.840369](https://doi.org/10.1109/VR.2000.840369).
- [74] Andrea Leganchuk, Shumin Zhai, and William Buxton. "Manual and cognitive benefits of two-handed input: an experimental study." In: *ACM Trans. Comput.-Hum. Interact.* 5.4 (Dec. 1998), 326–359. ISSN: 1073-0516. DOI: [10.1145/300520.300522](https://doi.org/10.1145/300520.300522). URL: <https://doi.org/10.1145/300520.300522>.
- [75] Julien-Charles Lévesque, Denis Laurendeau, and Marielle Mokhtari. "An asymmetric bimanual gestural interface for immersive virtual environments." In: *International Conference on Virtual, Augmented and Mixed Reality*. Springer. 2013, pp. 192–201.
- [76] Nianlong Li, Zhengquan Zhang, Can Liu, Zengyao Yang, Yinan Fu, Feng Tian, Teng Han, and Mingming Fan. "vMirror: Enhancing the Interaction with Occluded or Distant Objects in VR with Virtual Mirrors." In: 2021.
- [77] Wenmin Li, Xueyi Wan, Yanwei Shi, Nailang Yao, Ci Wang, and Zhaifeng Gao. "Depth and breadth of pie menus for mid-air gesture interaction." In: *International Journal of Human-Computer Interaction* 37.2 (2021), pp. 131–140.
- [78] Zhen Li, Joannes Chan, Joshua Walton, Hrvoje Benko, Daniel Wigdor, and Michael Glueck. "Armstrong: An empirical examination of pointing at non-dominant arm-anchored UIs in virtual reality." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14.

- [79] Jiandong Liang and Mark Green. “Geometric modeling using six degrees of freedom input devices.” In: *Proc. 3rd International Conference on CAD and Computer Graphics*. Citeseer. 1993, pp. 217–222.
- [80] Donghae Lim, Shizuka Shirai, Jason Orlosky, Photchara Ratsamee, Yuki Uranishi, and Haruo Takemura. “Evaluation of User Interfaces for Three-Dimensional Locomotion in Virtual Reality.” In: *Proceedings of the 2022 ACM Symposium on Spatial User Interaction*. 2022, pp. 1–9.
- [81] Chang Liu, Alexander Plopski, and Jason Orlosky. “OrthoGaze: Gaze-based three-dimensional object manipulation using orthogonal planes.” In: *Computers Graphics* 89 (2020), pp. 1–10. ISSN: 0097-8493. DOI: <https://doi.org/10.1016/j.cag.2020.04.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0097849320300431>.
- [82] Gerard Llorach, Alun Evans, and Josep Blat. “Simulator sickness and presence using HMDs: comparing use of a game controller and a position estimation system.” In: *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*. 2014, pp. 137–140.
- [83] Yiqin Lu, Chun Yu, and Yuanchun Shi. “Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality.” In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2020, pp. 35–43. DOI: [10.1109/VR46266.2020.00021](https://doi.org/10.1109/VR46266.2020.00021).
- [84] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. “BlindType: Eyes-Free Text Entry on Handheld Touchpad by Leveraging Thumb’s Muscle Memory.” In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1.2 (June 2017). DOI: [10.1145/3090083](https://doi.org/10.1145/3090083). URL: <https://doi.org/10.1145/3090083>.
- [85] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. “Promoting hotkey use through rehearsal with exposehk.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 573–582.
- [86] Andrii Matviienko, Florian Müller, Martin Schmitz, Marco Fendrich, and Max Mühlhäuser. “SkyPort: Investigating 3D Teleportation Methods in Virtual Environments.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (2022).
- [87] Morgan McCullough, Hong Xu, Joel Michelson, Matthew Jackoski, Wyatt Pease, William Cobb, William Kalescky, Joshua Ladd, and Betsy Williams. “Myo Arm: Swinging to Explore a VE.” In: *Proceedings*

- of the ACM SIGGRAPH Symposium on Applied Perception. SAP '15. Tübingen, Germany: Association for Computing Machinery, 2015, 107–113. ISBN: 9781450338127. DOI: [10.1145/2804408.2804416](https://doi.org/10.1145/2804408.2804416). URL: <https://doi.org/10.1145/2804408.2804416>.
- [88] Daniel Medeiros, Mauricio Sousa, Alberto Raposo, and Joaquim Jorge. “Magic Carpet: Interaction Fidelity for Flying in VR.” In: *IEEE Transactions on Visualization and Computer Graphics* 26.9 (2020), pp. 2793–2804. DOI: [10.1109/TVCG.2019.2905200](https://doi.org/10.1109/TVCG.2019.2905200).
 - [89] Daniel Mendes, Fabio Marco Caputo, Andrea Giachetti, Alfredo Ferreira, and Joaquim Jorge. “A survey on 3d virtual object manipulation: From the desktop to immersive virtual environments.” In: *Computer graphics forum*. Vol. 38. 1. Wiley Online Library. 2019, pp. 21–45.
 - [90] Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. “Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction.” In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 1997, 19–26. ISBN: 0897918967. DOI: [10.1145/258734.258747](https://doi.org/10.1145/258734.258747). URL: <https://doi.org/10.1145/258734.258747>.
 - [91] P. Monteiro, H. Coelho, G. Gonçalves, M. Melo, and M. Bessa. “Comparison of Radial and Panel Menus in Virtual Reality.” In: *IEEE Access* 7 (2019), pp. 116370–116379.
 - [92] Tony Morelli and Eelke Folmer. “Twuist: A discrete tactile-proprioceptive display for eye and ear free output on mobile devices.” In: *2012 IEEE Haptics Symposium (HAPTICS)*. IEEE. 2012, pp. 443–450.
 - [93] Yann Moullec, Mélanie Cogné, Justine Saint-Aubert, and Anatole Lécuyer. “Assisted walking-in-place: Introducing assisted motion to walking-by-cycling in embodied virtual reality.” In: *IEEE Transactions on Visualization and Computer Graphics* 29.5 (2023), pp. 2796–2805.
 - [94] Martin Mundt and Tintu Mathew. “An evaluation of pie menus for system control in virtual reality.” In: *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*. 2020, pp. 1–8.
 - [95] NASA-TLX Questionnaire. <https://humansystems.arc.nasa.gov/groups/TLX/>.

- [96] Niels Christian Nilsson, Tabitha Peck, Gerd Bruder, Eri Hodgson, Stefania Serafin, Mary Whitton, Frank Steinicke, and Evan Suma Rosenberg. "15 Years of Research on Redirected Walking in Immersive Virtual Environments." In: *IEEE Computer Graphics and Applications* 38.2 (2018), pp. 44–56. DOI: [10.1109/MCG.2018.111125628](https://doi.org/10.1109/MCG.2018.111125628).
- [97] Tomi Nukarinen, Jari Kangas, Jussi Rantala, Olli Koskinen, and Roope Raisamo. "Evaluating Ray Casting and Two Gaze-Based Pointing Techniques for Object Selection in Virtual Reality." In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. Tokyo, Japan: Association for Computing Machinery, 2018. ISBN: 9781450360869. DOI: [10.1145/3281505.3283382](https://doi.org/10.1145/3281505.3283382). URL: <https://doi.org/10.1145/3281505.3283382>.
- [98] Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, Myroslav Bachynskyi, Keith Vertanen, and Per Ola Kristensson. "Improving two-thumb text entry on touchscreen devices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, 2765–2774. ISBN: 9781450318990. DOI: [10.1145/2470654.2481383](https://doi.org/10.1145/2470654.2481383). URL: <https://doi.org/10.1145/2470654.2481383>.
- [99] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. "Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments." In: *Proceedings of the 1999 Symposium on Interactive 3D Graphics*. I3D '99. Atlanta, Georgia, USA: Association for Computing Machinery, 1999, 141–145. ISBN: 1581130821. DOI: [10.1145/300523.300540](https://doi.org/10.1145/300523.300540). URL: <https://doi.org/10.1145/300523.300540>.
- [100] Krzysztof Pietroszek, Liudmila Tahai, James R Wallace, and Edward Lank. "Watchcasting: Freehand 3D interaction with off-the-shelf smart-watch." In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2017, pp. 172–175.
- [101] Krzysztof Pietroszek, James R Wallace, and Edward Lank. "Tiltcasting: 3D interaction on large displays using a mobile device." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 57–62.
- [102] Cyprien Pindat, Emmanuel Pietriga, Olivier Chapuis, and Claude Puech. "Drilling into complex 3D models with gimlenses." In: *VRST '13*. 2013.

- [103] Carole Plasson, Dominique Cunin, Yann Laurillau, and Laurence Nigay. "A Lens-Based Extension of Raycasting for Accurate Selection in Dense 3D Environments." In: *Human-Computer Interaction – INTERACT 2021*. Ed. by Carmelo Ardito, Rosa Lanzilotti, Alessio Malizia, Helen Petrie, Antonio Piccinno, Giuseppe Desolda, and Kori Inkpen. Cham: Springer International Publishing, 2021, pp. 501–524. ISBN: 978-3-030-85610-6.
- [104] 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*. UIST '96. Seattle, Washington, USA: Association for Computing Machinery, 1996, 79–80. ISBN: 0897917987. DOI: [10.1145/237091.237102](https://doi.org/10.1145/237091.237102). URL: <https://doi.org/10.1145/237091.237102>.
- [105] Majid Pourmemar and Charalambos Poullis. "Visualizing and Interacting with Hierarchical Menus in Immersive Augmented Reality." In: *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '19. Brisbane, QLD, Australia: Association for Computing Machinery, 2019. ISBN: 9781450370028. DOI: [10.1145/3359997.3365693](https://doi.org/10.1145/3359997.3365693). URL: <https://doi.org/10.1145/3359997.3365693>.
- [106] Yuan Yuan Qian and Robert J. Teather. "Look to Go: An Empirical Evaluation of Eye-Based Travel in Virtual Reality." In: *Proceedings of the 2018 ACM Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: Association for Computing Machinery, 2018, 130–140. ISBN: 9781450357081. DOI: [10.1145/3267782.3267798](https://doi.org/10.1145/3267782.3267798). URL: <https://doi.org/10.1145/3267782.3267798>.
- [107] Sharif Razzaque. "Redirected Walking." AAI3190299. PhD thesis. USA, 2005. ISBN: 0542340534.
- [108] Gang Ren and Eamonn O'Neill. "3D marking menu selection with freehand gestures." In: *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2012, pp. 61–68.
- [109] Hyocheol Ro, Seungcho Chae, Inhwon Kim, Junghyun Byun, Yoonsik Yang, Yoonjung Park, and Tackdon Han. "A dynamic depth-variable ray-casting interface for object manipulation in ar environments." In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017, pp. 2873–2878. DOI: [10.1109/SMC.2017.8123063](https://doi.org/10.1109/SMC.2017.8123063).

- [110] 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*. 2011, pp. 197–206.
- [111] Majed Samad, Elia Gatti, Anne Hermes, Hrvoje Benko, and Cesare V. Parise. "Pseudo-Haptic Weight: Changing the Perceived Weight of Virtual Objects By Manipulating Control-Display Ratio." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019). URL: <https://api.semanticscholar.org/CorpusID:140224865>.
- [112] Jonas Schjerlund, Kasper Hornbæk, and Joanna Bergström. "Ninja Hands: Using Many Hands to Improve Target Selection in VR." In: 2021.
- [113] Valentin Schwind, Sven Mayer, Alexandre Comeau-Vermeersch, Robin Schweigert, and Niels Henze. "Up to the Finger Tip: The Effect of Avatars on Mid-Air Pointing Accuracy in Virtual Reality." In: *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '18. Melbourne, VIC, Australia: Association for Computing Machinery, 2018, 477–488. ISBN: 9781450356244. DOI: [10.1145/3242671.3242675](https://doi.org/10.1145/3242671.3242675). URL: <https://doi.org/10.1145/3242671.3242675>.
- [114] John Sermarini, Jr. Joseph T. Kider, Jr. Joseph J. LaViola, and Daniel S. McConnell. "A Kinematic Evaluation of Linear and Parabolic Pointing in Virtual Reality." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 65.1 (2021), pp. 86–90. DOI: [10.1177/1071181321651131](https://doi.org/10.1177/1071181321651131). eprint: <https://doi.org/10.1177/1071181321651131>. URL: <https://doi.org/10.1177/1071181321651131>.
- [115] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. "Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–13.
- [116] Erik Sikström, Amalia de Götzen, and Stefania Serafin. "Wings and flying in immersive VR — Controller type, sound effects and experienced ownership and agency." In: *2015 IEEE Virtual Reality (VR)*. 2015, pp. 281–282. DOI: [10.1109/VR.2015.7223405](https://doi.org/10.1109/VR.2015.7223405).
- [117] Aubrey Simonson, Dana W Gretton, and Casper Harteveld. "Bird: A Point Cursor for Virtual Immersive Environments." In: *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in*

- Computing Systems* (2025). URL: <https://api.semanticscholar.org/CorpusID:278035024>.
- [118] Nancy A Skopp, Derek J Smolenski, Melinda J Metzger-Abamukong, Albert A Rizzo, and Greg M Reger. "A pilot study of the virtusphere as a virtual reality enhancement." In: *International Journal of Human-Computer Interaction* 30.1 (2014), pp. 24–31.
 - [119] Mel Slater, Martin Usoh, and Anthony Steed. "Steps and Ladders in Virtual Reality." In: *Proceedings of the Conference on Virtual Reality Software and Technology*. VRST '94. Singapore, Singapore: World Scientific Publishing Co., Inc., 1994, 45–54. ISBN: 9810218672.
 - [120] Mel Slater, Martin Usoh, and Anthony Steed. "Taking steps: the influence of a walking technique on presence in virtual reality." In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 2.3 (1995), pp. 201–219.
 - [121] *Steam VR home*. 2020. URL: <https://store.steampowered.com/news/app/250820/view/2898585530113860152>.
 - [122] Frank Steinicke, Yon Visell, Jennifer Campos, and Anatole Lécuyer. *Human walking in virtual environments*. Vol. 56. 7. Springer, 2013.
 - [123] 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*. CHI '95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, 265–272. ISBN: 0201847051. DOI: [10.1145/223904.223938](https://doi.org/10.1145/223904.223938). URL: <https://doi.org/10.1145/223904.223938>.
 - [124] 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*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–12. ISBN: 9781450356206. DOI: [10.1145/3173574.3174218](https://doi.org/10.1145/3173574.3174218). URL: <https://doi.org/10.1145/3173574.3174218>.
 - [125] Chengzheng Sun. "Xpointer: an x-ray telepointer for relaxed-space-time wysiwis and unconstrained collaborative 3d design systems." In: *Proceedings of the 2013 conference on Computer supported cooperative work*. 2013, pp. 729–740.

- [126] Junwei Sun and Wolfgang Stuerzlinger. "Selecting Invisible Objects." In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2018, pp. 697–698.
- [127] Ivan E Sutherland. "Sketch pad a man-machine graphical communication system." In: *Proceedings of the SHARE design automation workshop*. 1964, pp. 6–329.
- [128] James N. Templeman, Patricia S. Denbrook, and Linda E. Sibert. "Virtual Locomotion: Walking in Place through Virtual Environments." In: *Presence: Teleoperators and Virtual Environments* 8.6 (Dec. 1999), pp. 598–617. DOI: [10.1162/105474699566512](https://doi.org/10.1162/105474699566512). eprint: <https://direct.mit.edu/pvar/article-pdf/8/6/598/1623326/105474699566512.pdf>. URL: <https://doi.org/10.1162/105474699566512>.
- [129] *Tilt Brush*. 2014. URL: <https://www.tiltbrush.com/>.
- [130] Amy C Ulinski, Zachary Wartell, Paula Goolkasian, Evan A Suma, and Larry F Hodges. "Selection performance based on classes of bimanual actions." In: *2009 IEEE Symposium on 3D User Interfaces*. IEEE. 2009, pp. 51–58.
- [131] Amy Ulinski, Catherine Zambaka, Zachary Wartell, Paula Goolkasian, and Larry F Hodges. "Two handed selection techniques for volumetric data." In: *2007 IEEE Symposium on 3D User Interfaces*. IEEE. 2007.
- [132] Khrystyna Vasylevska and Hannes Kaufmann. "Influence of Metaphors for Vertical Navigation on Presence." In: Mar. 2014.
- [133] Vanessa Vuibert, Wolfgang Stuerzlinger, and Jeremy R. Cooperstock. "Evaluation of Docking Task Performance Using Mid-Air Interaction Techniques." In: *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. SUI '15. Los Angeles, California, USA: Association for Computing Machinery, 2015, 44–52. ISBN: 9781450337038. DOI: [10.1145/2788940.2788950](https://doi.org/10.1145/2788940.2788950). URL: <https://doi.org/10.1145/2788940.2788950>.
- [134] Uta Wagner, Matthias Albrecht, Andreas Asferg Jacobsen, Haopeng Wang, Hans Gellersen, and Ken Pfeuffer. "Gaze, Wall, and Racket: Combining Gaze and Hand-Controlled Plane for 3D Selection in Virtual Reality." In: *Proceedings of the ACM on Human-Computer Interaction* 8 (2024), pp. 189 –213. URL: <https://api.semanticscholar.org/CorpusID:273605984>.

- [135] Tim Weissker, Pauline Bimberg, Aalok Shashidhar Gokhale, Torsten Kuhlen, and Bernd Froehlich. “Gaining the High Ground: Teleportation to Mid-Air Targets in Immersive Virtual Environments.” In: *IEEE Transactions on Visualization and Computer Graphics* 29.5 (2023), pp. 2467–2477. DOI: [10.1109/TVCG.2023.3247114](https://doi.org/10.1109/TVCG.2023.3247114).
- [136] Johann Wentzel, Matthew Lakier, Jeremy Hartmann, Falah Shazib, Géry Casiez, and Daniel Vogel. “A Comparison of Virtual Reality Menu Archetypes: Raycasting, Direct Input, and Marking Menus.” In: *IEEE Transactions on Visualization and Computer Graphics* 31.9 (2025), pp. 4868–4882. DOI: [10.1109/TVCG.2024.3420236](https://doi.org/10.1109/TVCG.2024.3420236).
- [137] Johann Wentzel, Greg d’Eon, and Daniel Vogel. “Improving Virtual Reality Ergonomics Through Reach-Bounded Non-Linear Input Amplification.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–12. ISBN: 9781450367080. DOI: [10.1145/3313831.3376687](https://doi.org/10.1145/3313831.3376687). URL: <https://doi.org/10.1145/3313831.3376687>.
- [138] Betsy Williams, Gayathri Narasimham, Bjoern Rump, Timothy P McNamara, Thomas H Carr, John Rieser, and Bobby Bodenheimer. “Exploring large virtual environments with an HMD when physical space is limited.” In: *Proceedings of the 4th symposium on Applied perception in graphics and visualization*. 2007, pp. 41–48.
- [139] Julius von Willich, Martin Schmitz, Florian Müller, Daniel Schmitt, and Max Mühlhäuser. “Podoportation: Foot-Based Locomotion in Virtual Reality.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–14. ISBN: 9781450367080. DOI: [10.1145/3313831.3376626](https://doi.org/10.1145/3313831.3376626). URL: <https://doi.org/10.1145/3313831.3376626>.
- [140] 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*. 2009, pp. 1083–1092.
- [141] Dennis Wolf, Katja Rogers, Christoph Kunder, and Enrico Rukzio. “JumpVR: Jump-Based Locomotion Augmentation for Virtual Reality.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–12. ISBN: 9781450367080. DOI: [10.1145/3313831.3376243](https://doi.org/10.1145/3313831.3376243). URL: <https://doi.org/10.1145/3313831.3376243>.

- [142] Hans Peter Wyss, R. Blach, and M. Bues. “iSith - Intersection-based Spatial Interaction for Two Hands.” In: *3D User Interfaces (3DUI’06)* (2006), pp. 59–61.
- [143] Zheer Xu, Weihao Chen, Dongyang Zhao, Jiehui Luo, Te-Yen Wu, Jun Gong, Sicheng Yin, Jialun Zhai, and Xing-Dong Yang. “BiTipText: Bimanual Eyes-Free Text Entry on a Fingertip Keyboard.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–13. ISBN: 9781450367080. DOI: [10.1145/3313831.3376306](https://doi.org/10.1145/3313831.3376306). URL: <https://doi.org/10.1145/3313831.3376306>.
- [144] 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*. CHI ’18. Montreal QC, Canada: Association for Computing Machinery, 2018. ISBN: 9781450356206. DOI: [10.1145/3173574.3173616](https://doi.org/10.1145/3173574.3173616). URL: <https://doi.org/10.1145/3173574.3173616>.
- [145] Zhixin Yan, Robert W. Lindeman, and Arindam Dey. “Let your fingers do the walking: A unified approach for efficient short-, medium-, and long-distance travel in VR.” In: *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. 2016, pp. 27–30. DOI: [10.1109/3DUI.2016.7460027](https://doi.org/10.1109/3DUI.2016.7460027).
- [146] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. “Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments.” In: *ACM Trans. Graph.* 38.6 (2019). ISSN: 0730-0301. DOI: [10.1145/3355089.3356544](https://doi.org/10.1145/3355089.3356544). URL: <https://doi.org/10.1145/3355089.3356544>.
- [147] Difeng Yu, Xueshi Lu, Rongkai Shi, Hai-Ning Liang, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. “Gaze-Supported 3D Object Manipulation in Virtual Reality.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI ’21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: [10.1145/3411764.3445343](https://doi.org/10.1145/3411764.3445343). URL: <https://doi.org/10.1145/3411764.3445343>.
- [148] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and Jorge Gonçalves. “Fully-Occluded Target Selection in Virtual Reality.” In: *IEEE Transactions on Visualization and Computer Graphics* 26 (2020), pp. 3402–3413.

- [149] Futian Zhang, Keiko Katsuragawa, and Edward Lank. "Conductor: Intersection-Based Bimanual Pointing in Augmented and Virtual Reality." In: *Proc. ACM Hum.-Comput. Interact.* 6.ISS (2022). DOI: [10.1145/3567713](https://doi.org/10.1145/3567713). URL: <https://doi.org/10.1145/3567713>.
- [150] Futian Zhang, Sachi Mizobuchi, Wei Zhou, Taslim Arefin Khan, Wei Li, and Edward Lank. "Leveraging CD Gain for Precise Barehand Video Timeline Browsing on Smart Displays." In: *Human-Computer Interaction—INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part IV* 18. Springer. 2021, pp. 72–91.
- [151] Yaying Zhang, Bernhard E. Riecke, Thecla Schiphorst, and Carman Neustaedter. "Perch to Fly: Embodied Virtual Reality Flying Locomotion with a Flexible Perching Stance." In: *Proceedings of the 2019 on Designing Interactive Systems Conference*. DIS '19. San Diego, CA, USA: Association for Computing Machinery, 2019, 253–264. ISBN: 9781450358507. DOI: [10.1145/3322276.3322357](https://doi.org/10.1145/3322276.3322357). URL: <https://doi.org/10.1145/3322276.3322357>.
- [152] Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. "Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-Stroke Marking Menus." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: Association for Computing Machinery, 2006, 1077–1086. ISBN: 1595933727. DOI: [10.1145/1124772.1124933](https://doi.org/10.1145/1124772.1124933). URL: <https://doi.org/10.1145/1124772.1124933>.
- [153] Shengdong Zhao and Ravin Balakrishnan. "Simple vs. Compound Mark Hierarchical Marking Menus." In: *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. UIST '04. Santa Fe, NM, USA: Association for Computing Machinery, 2004, 33–42. ISBN: 1581139578. DOI: [10.1145/1029632.1029639](https://doi.org/10.1145/1029632.1029639). URL: <https://doi.org/10.1145/1029632.1029639>.
- [154] Suwen Zhu, Jingjie Zheng, Shumin Zhai, and Xiaojun Bi. "I'sFree: Eyes-Free Gesture Typing via a Touch-Enabled Remote Control." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019. ISBN: 9781450359702. DOI: [10.1145/3290605.3300678](https://doi.org/10.1145/3290605.3300678). URL: <https://doi.org/10.1145/3290605.3300678>.