Improving Virtual Reality Accessibility through Context-Aware Spatial Remapping

by

Johann David Wentzel

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

Waterloo, Ontario, Canada, 2024

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	Hans Gellersen Professor, Aarhus University
Supervisor	Daniel Vogel Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member	Géry Casiez Professor, Dept. of Computer Science, Université de Lille. Adjunct Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member	Jian Zhao Associate Professor, Cheriton School of Computer Science, University of Waterloo
Internal-external Member	Ville Mäkelä Assistant Professor, Stratford School of Interaction Design and Business, 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[1] is as follows:

> 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 Chapter 3:

> Johann Wentzel, Sasa Junuzovic, James Devine, John Porter, and Martez Mott. 2022. Understanding How People with Limited Mobility Use Multi-Modal Input. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 4, 1–17. `https://doi.org/10.1145/3491102.3517458`

Portions of Chapter 4:

> Johann Wentzel, Fraser Anderson, George Fitzmaurice, Tovi Grossman, and Daniel Vogel. 2024. SwitchSpace: Understanding Context-Aware Peeking Between VR and Desktop Interfaces. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 801, 1–16. `https://doi.org/10.1145/3613904.3642358`

---

[1] `https://authors.acm.org/author-resources/author-rights` – Accessed on Oct 13, 2024.

ABSTRACT

VR makes assumptions about user ability that might be impossible for a user with mobility limitations to meet, and the more gestural and motion-based input of VR means common motor accessibility solutions do not provide a sufficiently adaptable remedy. This dissertation presents the results of a three-phase research path covering a variety of research questions within accessible spatial input. First, an investigation of accessible multi-modal input setups demonstrates that designing for input categories rather than input devices is key, due to the large range of customized accessibility setups. The second project, focused on the situational impairments experienced when switching between desktop and VR, demonstrates and evaluates a solution that responds to user context to make this cross-device input easier. The final project investigates individual user ranges of motion for VR, and presents a design language for body motion inspired by 3D geometric primitives. These motion primitives are used to create a solution that enables user-customizable input remapping in a simpler and more concise way than traditional transfer functions.

ACKNOWLEDGMENTS

This thesis represents several distinct chapters of my life, spanning a global pandemic, two cross-continent moves, and countless other challenges. I feel incredibly lucky to have had the support of so many people. I'll name as many as I can, but there are truly too many to count.

First, thanks to my advisor, Daniel Vogel. It is with your endless patience, support, and guidance that I cross the finish line. Thank you for being empathetic, creative, and optimistic throughout every challenge of this degree. Also, thanks for being fine with my decision to go long-distance with you instead of long-distance with Jackie.

My Waterloo community has been nothing but supportive for the past six years, and I could not be more grateful. Thanks to the entire Exii Group for being great sounding boards, collaborators, and friends. Thanks in particular to Nikhita, Alê, Cheryl, Greg, Lisa, and Margaret for always being there to listen when I need to rant about something.

Thanks to all of my co-authors and mentors throughout my PhD for their guidance and support: Martez Mott, Sasa Junuzovic, James Devine, John Porter, Ed Cutrell, Bruno de Araujo, Jota Costa, Rahul Arora, Josh Davis, Alê Luz, Daisy Kim, Jeremy Hartmann, Lesley Istead, Fraser Anderson, George Fitzmaurice, Tovi Grossman, Matthew Lakier, Falah Shazib, and Géry Casiez.

I made it through this PhD more easily knowing my Calgary community always has my back. Thanks to Connor, Adam, Amanda, Becca, Tim, Warren, Jason, Bradley, Josh, Tori, Alex, and countless others for making sure that I take time away from academia and focus on what's really important: video games, board games, and outdoor activities.

I could not have completed this PhD without the unwavering support of my family. Johann Sr., Erika, Mada, and Colton have been invested in my success from day one. Thank you for the support, the late-night phone calls, and for making sure that I not only survived but thrived during one of the most challenging eras of my life.

Finally, I extend my heartfelt thanks to Jackie. I started the PhD as your boyfriend and ended the PhD as your husband, and I cherish every single day we have together. It is with your support and our partnership that I see this work through to the end. I can't wait to see what our future holds.

Figure 1: (left) Johann David Wentzel IV; (right) Johann David Wentzel II

To Johann David Wentzel II, Ph.D. (1927 – 2005) – my original research advisor. The best part of academia is that my publications appear on Google Scholar right next to yours.

TABLE OF CONTENTS

Appendices

LIST OF FIGURES

LIST OF TABLES

When society makes a commitment to making new technologies accessible to everyone, the focus will no longer be on what people cannot do, but rather on what skills and interests they bring to their work.
That will be as it always should have been.

– Frank Bowe

1

INTRODUCTION

The advent of consumer virtual reality (VR) and augmented reality (AR) brings into the mainstream an entirely new category of interactive content, immersive experiences, and social connections. This increased popularity brings with it plenty of developer interest, but accessibility considerations still lag behind what is necessary to fully include people with mobility limitations caused by conditions like paraplegia, Parkinson's disease, or arthritis. While current consumer headsets like the Meta Quest 3 or the Vision Pro have implemented various alternate forms of input like hand gestures or eye tracking, the most common form of input in VR or AR is in the form of *spatial gestures*: tracked physical motions which require large or precise body movements.

From a functional perspective, the accessibility of any input device or input technique depends on how well the user can match its built-in *ability assumptions*. In the language of Wobbrock et al. [143], an ability assumption is a bias ingrained within the design of an application, either explicitly or implicitly, which causes the application to require a certain level of physical ability and risks excluding people who cannot provide that level of physical ability. We define the *motor accessibility* of a given interface as how well a given user can match its ability assumptions.

At present, spatial interfaces make ability assumptions about users that might be difficult or even impossible to meet by users with mobility limitations. As an exercise in understanding common spatial input ability assumptions, consider *Beat Saber* [14], a popular VR game that utilizes large spatial gestures. In *Beat Saber*, players swing both hands independently to slash blocks placed within arms-reach, to the rhythm of a song. These slashing movements are scored based on their accuracy (hitting the block, and in the direction specified on the block), as well as the width and speed of the swing. Each in-game song is between one and five minutes long. These characteristics already reveal several ability assumptions. In-game blocks are placed at a height which requires the player to stand, as well as maintain their balance while swinging their arms. The game's scoring incentivizes wide, fast swings, requiring mobility in both arms. The game requires the player to have the grip strength necessary to grasp controllers while swinging their arms. The game requires motion speed and precision to hit blocks accurately enough to achieve a high score.

Some previous work has addressed VR accessibility issues related to vision [151], hearing [64], and motor [74, 127]. Motor accessibility work in VR often involves bespoke design prototypes for solving a single specific accessibility issue in VR, like usage in a wheelchair [48], single-handed input [146], head motion toward visual points of interest [74], or constrained spaces [127].

These solutions lack a unifying design language to aid in crossing the gap from academic research to real-life implementation.

Creating interaction techniques that center *ability* rather than *disability* demands a deep understanding of the habits and preferences of people with mobility limitations concerning both spatial and non-spatial input. At the same time, creating accessibility techniques that more easily transfer to real-world implementations demands consideration of a wide variety of impairments: *permanent* (like lack of a limb or ability to move it), *temporary* (like a broken arm in a cast), and *situational* (like having reduced precision while walking [54], or holding a cumbersome object [138]).

This thesis presents a way to operationalize ability-based design with the end goal of making spatial input more adaptable for people who, permanently, temporarily, or situationally, may not be able to provide the full range of motion that spatial applications often expect.

First, we motivate our work by understanding how people with mobility limitations overcome accessibility barriers in traditional computing platforms by combining multiple input devices. We use qualitative methods, including surveys and semi-structured interviews, to understand the configurations and associated trade-offs involved with creating more accessible configurations. The results demonstrate that accessible configurations must be designed with input *categories* in mind instead of individual input *devices*. Moreover, the discovery and creation of accessible input configurations remains a challenge for many people, motivating a solution with quick and easy configuration and iteration.

Second, we explore how cleverly utilizing context can resolve situational impairments incurred by switching between VR hardware and traditional desktop hardware. Manipulating hardware and completing tasks that require switching between VR and desktop can be cumbersome for people with and without mobility limitations. This poses a particular problem for people like VR developers, whose workflows commonly require switching between VR and desktop. We explore these issues using a formative study, which then motivates the creation and evaluation of a collection of context-aware interaction techniques to facilitate more easily switching between VR and desktop.

Third, we use both of these findings to inform and motivate the design of a customizable geometric input remapping system for VR. This system allows users with mobility limitations to create and customize input mechanisms specifically for their range of motion, and expands that range into the range of motion that a given VR application expects. We explore this with an initial formative study to understand the mobility limitations and input assumptions made by common VR games, then use these findings to motivate the design of our remapping system.

1.1 OBJECTIVE AND RESEARCH QUESTIONS

At a high level, the objective of this thesis is:

Investigate how input device combinations, mediated by context, can make spatial interfaces more accessible in a general and customizable way.

We separate this investigation into a series of research questions, along three related projects. We separate the outcomes of these projects into three distinct categories: understanding the devices that make up accessible input configurations; understanding users and their interactions; and finally providing novel insights via prototypes and associated evaluations. An overview of the grand methodology is provided in Figure 1.1.

The first set of research questions explores how users with mobility limitations create and discover multi-modal input configurations to overcome accessibility barriers. We use gaming as a framing device for our investigation due to its higher demand for speed, precision, and complex input. We explored:

(a) *How do real-life users with limited mobility alter or combine input devices to be more accessible?* We answer this question using a survey of people with limited mobility to understand which input devices and input combinations were utilized the most.

(b) *How does application or usage context affect these device or configuration choices?* We answer this question using semi-structured interviews with people with limited mobility, motivating the individual user choices underpinning these configuration choices.

(c) *How do users discover accessible setups, and what does this process reveal?* Using the finding from (b) that many users depend on social media to discover new accessible setups, we simulate this process by conducting a systematic analysis of YouTube videos to demonstrate the variety of accessibility setups that can be found online.

The second set of research questions explores a more specific application of ability-based design: addressing the situational impairments incurred while manipulating VR hardware. We explored:

(d) *What factors of VR-enabled workflows cause situational impairments or discomfort?* We answer this question using a survey and formative study eliciting workflows that involve both desktop computer and VR use, revealing their underlying situational impairments and difficulties.

(e) *How can we make use of application context to resolve these usability issues?* We propose, develop, and evaluate a prototype that uses the user's physical context as a secondary stream of input, demonstrating how UI responses can make these cross-device workflows easier and faster.

The third set of research questions explores how to categorize and accommodate a variety of ranges of motion for VR applications, with the end goal of creating a more general and customizable accessibility solution:

(f) *What kinds of motion-related interactions in VR cause accessibility issues?* We answer this question with a formative study, in which 10 people with a variety of mobility limitations tried to use 5 popular VR applications.

(g) *Can customizable input remappings resolve these accessibility issues?* We use the results from (f) to propose a customizable geometric input remapping system which allows users to create custom spatial input remappings based on their own range of motion.

1.2 CONTRIBUTIONS

We summarize the research contributions of this thesis by project, outlining the key results that form our contributions.

1.2.1 *Understanding Multi-Modal Input Configurations*

In Chapter 3, we use a multi-step methodology to understand how people with mobility limitations adapt, combine, and configure their input devices to overcome accessibility barriers. In an initial survey with a self-guided questionnaire, we found that respondents made use of a wide variety of input devices and combinations. Importantly, the vast majority of input device combinations were reported only once. This demonstrates the "long tail" of input combinations, emphasizing the need for accessibility solutions to focus on wider input categories instead of individual devices.

This need for customizability is emphasized even further in the results of semi-structured interviews. Participants reported usage of a wide variety of input combinations, as well as several different strategies for configuring their devices to overcome specific accessibility barriers. This included adapting their grip on an input device, remapping buttons, or even modifying the device hardware itself.

The interviews also revealed an important insight: input configurations are still difficult to discover. Configurations were often discovered and configured through personal trial and error, with social media providing the bulk of participants' initial inspiration for creating accessible input configurations. We simulated the social media discovery process by conducting a systematic analysis of YouTube videos to understand the variety of configurations involved, and used these results to demonstrate the importance of easy discovery.

1.2.2 *Context-Aware VR-Desktop Transitions*

In Chapter 4, we focus on one exemplar situational impairment: switching between VR and desktop interfaces as demanded by specific workflows. We begin by conducting a formative study to understand these workflows as well as the usability issues that users encounter. The results revealed a variety of usability issues associated with switching between VR and desktop interfaces, including cumbersome hardware, discomfort, or lack of precision when switching. Respondents noted that switching between these two interfaces was often quick and temporary. We use these findings to create a prototype cross-device interface between VR and desktop, making use of specific physical aspects of the user context to enable more convenient switching between VR and desktop. We found that assisting with these quick, temporary switches using our prototype made a sample cross-device task faster, more precise, and more comfortable.

1.2.3 *Modular Geometric Motion Remapping*

In Chapter 5, we directly address the issue of differing mobility and its implications on VR application usability. First, we conducted a formative study with 10 people with mobility limitations to understand the accessibility issues that they encountered while playing five popular VR games. These accessibility issues can be separated into a variety of themes, but a common theme was the difference between a given user's ability and that which is required by the application. We use the results of the formative study to motivate the concept of *motion primitives*, a concise geometric language for describing and categorizing potentially inaccessible physical motion. We develop and propose a remapping technique which uses these primitives as simple building blocks for creating more accessible motion remapping configurations. We evaluate this prototype in a second study, finding that remapping body motion in this customizable way made these physical interactions in VR easier and more comfortable to complete.

1.3 DISSERTATION OUTLINE

The remainder of this dissertation is structured as follows:

In Chapter 2, we establish the overall research area with a summary of relevant background literature on accessible design, VR motor input, and spatial input re-mapping.

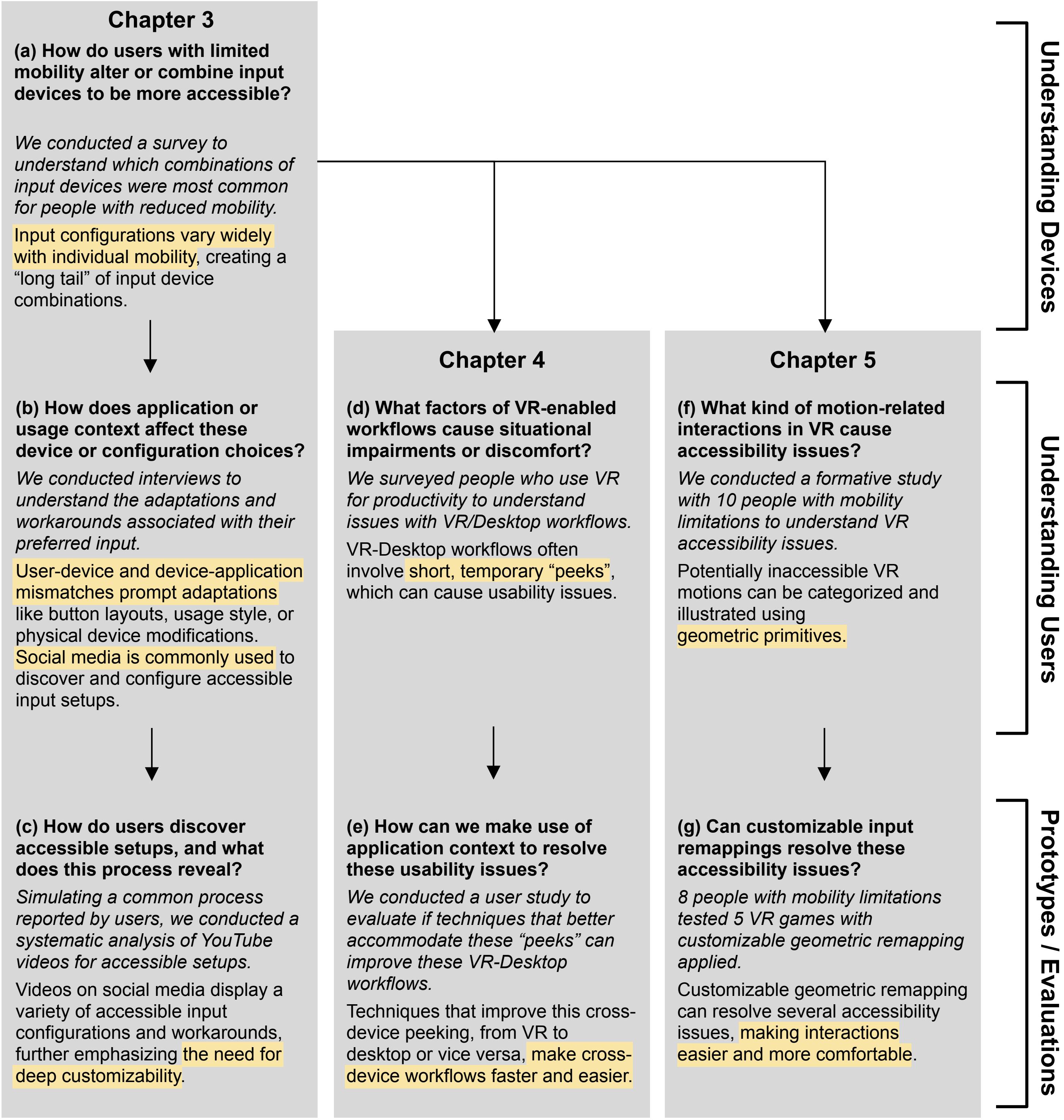
In Chapter 3, we describe our investigation into how people with limited mobility utilize and configure input devices to overcome their accessibility barriers. We discussed the results of a survey, semi-structured interviews, as well as a systematic analysis of YouTube videos. We also provide design recommendations and considerations for future qualitative accessibility investigations.

In Chapter 4, we explore how cross-device workflows between VR and desktop interfaces can be improved by interface techniques that facilitate quick, temporary "peeks" between interfaces. We present the results of a formative study that elicits usability issues with these cross-device workflows and discuss how these issues could be addressed by input techniques that use context as a secondary stream of input. We then describe *SwitchSpace*, a collection of input and display techniques that facilitates these quick, temporary switches between VR and desktop. We evaluate these techniques in a user study and discuss these results as well as implications for future cross-device work.

In Chapter 5, we explore usability issues associated with common VR applications for people with mobility limitations. We begin by analyzing and discussing the results of a formative study to motivate motion primitives, a way to describe inaccessible movements using simple 3D geometric primitives. We use motion primitives in the design of *MotionBlocks*, a prototype remapping system that enables users to easily create and configure custom transfer functions that expand their personal range of motion into the larger range of motion that the VR applications expect. We evaluate our prototype in a second study and discuss the qualitative and quantitative results.

In Chapter 6 we discuss limitations and future work, describe implementation difficulties and design considerations, and then discuss how these insights can be combined into a more generalizable accessibility solution for VR motor input.

In Chapter 7, we revisit the proposed research questions and draw overall conclusions.



Chapter 3

(a) How do users with limited mobility alter or combine input devices to be more accessible?

We conducted a survey to understand which combinations of input devices were most common for people with reduced mobility.

Input configurations vary widely with individual mobility, creating a "long tail" of input device combinations.

(b) How does application or usage context affect these device or configuration choices?

We conducted interviews to understand the adaptations and workarounds associated with their preferred input.

User-device and device-application mismatches prompt adaptations like button layouts, usage style, or physical device modifications. Social media is commonly used to discover and configure accessible input setups.

(c) How do users discover accessible setups, and what does this process reveal?

Simulating a common process reported by users, we conducted a systematic analysis of YouTube videos for accessible setups.

Videos on social media display a variety of accessible input configurations and workarounds, further emphasizing the need for deep customizability.

Chapter 4

(d) What factors of VR-enabled workflows cause situational impairments or discomfort?

We surveyed people who use VR for productivity to understand issues with VR/Desktop workflows.

VR-Desktop workflows often involve short, temporary "peeks", which can cause usability issues.

(e) How can we make use of application context to resolve these usability issues?

We conducted a user study to evaluate if techniques that better accommodate these "peeks" can improve these VR-Desktop workflows.

Techniques that improve this cross-device peeking, from VR to desktop or vice versa, make cross-device workflows faster and easier.

Chapter 5

(f) What kind of motion-related interactions in VR cause accessibility issues?

We conducted a formative study with 10 people with mobility limitations to understand VR accessibility issues.

Potentially inaccessible VR motions can be categorized and illustrated using geometric primitives.

(g) Can customizable input remappings resolve these accessibility issues?

8 people with mobility limitations tested 5 VR games with customizable geometric remapping applied.

Customizable geometric remapping can resolve several accessibility issues, making interactions easier and more comfortable.

Figure 1.1: Research path showing research questions, methodology, and main results. Bold text is the research question statement; italic text is the research methodology applied; and the final block of text is the primary results that form the contributions.

2

RELATED WORK

In this chapter, we review previous research related to our investigation. This includes conceptual frameworks of multi-modal input, cross-device input, spatial remapping, evaluating the accessibility of existing systems, as well as previous accessibility solutions for spatial input.

2.1 ABILITY-BASED DESIGN

To start, we discuss previous work in accessible design that provides a baseline structure for our investigation.

Wobbrock et al. [144] present seven principles for *ability-based design*, listed in Table 2.1. At its core, ability-based design is a philosophy which focuses on centering *ability* instead of *disability* as the central design consideration for accessible interfaces. The seven principles are organized into *stance* (the designers' overall goals), *interface* (interface-level design considerations), and *sytstem* (larger system-level design considerations). Ability-based design represents a deviation from the general definition of "assistive technology", which generally focuses more on adapting a user's ability to a system which might not have been explicitly designed to accommodate them.

The end goal of an ability-based design philosophy is a system that reduces the burden of adaptation placed on individual users, instead relying on more intelligent or proactive adaptation on the part of the system itself (Figure 2.1).

The work in this thesis adopts the same philosophy. Rather than forcing the user to adapt to potentially inaccessible movements or input actions, we propose methods that allow the system to accept whatever input device, input combination, or range of motion that the user prefers to provide. Concerning the seven principles in Table 2.1, we focus deliberately on providing system-level responses to the user's range of motion (*ability*, *accountability*), using easily accessible software with no necessary modifications to conventional hardware (*availability*). Chapter 4 describes how this proposed system can adapt to users automatically based on their momentary input and output preferences (*adaptability*, *context*), and Chapter 5 describes how users can create custom adaptations manually (*adaptability*, *transparency*).

2.2 CONCEPTUAL MODELS OF MULTI-MODALITY

Understanding multi-modal input configurations requires theoretical knowledge of how to categorize both unimodal and multi-modal input. Buxton [25] unified two taxonomies of input into a generic scheme for classifying sensing properties of input devices, establishing a tableau of continuous input devices alongside ways that these devices can be combined. Similarly, Mackinlay et

Table 2.1: The seven principles of ability-based design, from Wobbrock et al. [143].

Designer Stance	**Ability**	Focus on ability, not dis-ability. Leverage all users can do.
	Accountability	Respond to poor performance by changing systems, not users. Leave users as they are.
	Availability	Use affordable and available software, hardware, or other components acquirable through accessible means.
Interface Adaptation	**Adaptability**	Interfaces can be adapted by themselves or by users to best match users' abilities.
	Transparency	Give users awareness of adaptations as well as the means to inspect, override, discard, revert, store, retrieve, preview, and test these adaptations.
Sensing and Modeling	**Performance**	Regard users' performance, as well as monitoring, measuring, modeling, or predicting that performance to better match user ability.
	Context	Proactively sense context, anticipating and accommodating its effects on users' abilities.

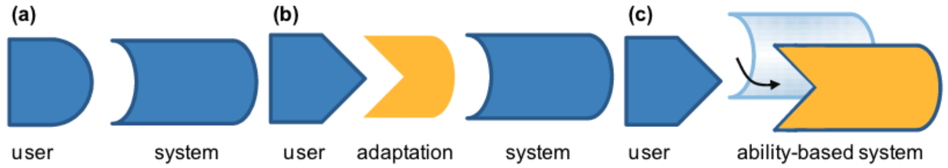


Figure 2.1: An illustration of ability-based design, from Wobbrock et al. [144]. (a) A user whose abilities match those presumed by the system. (b) A user whose abilities do not match those presumed by the system. Because the system is inflexible, the user must be adapted to it. (c) An ability-based system is designed to accommodate the user's abilities. It may adapt or be adapted to them.

al. [79] examined the inputs involved in a system, conceptualizing generic interactions as tuples including device, input domain, output domain, and system state. Oviatt [98] described a broader overview of multi-modal interaction design and provided a basic groundwork for the cognitive science behind multi-modal interaction. Expanding on the wider theory of input devices, Savidis and Stephanidis [111] discussed the Unified Interface Design Method, ultimately decomposing multi-modal input tasks into three subtasks: user tasks (what the user has to physically do), system tasks (feedback the system must provide), and physical design (the various physical interfaces upon which the user performs the task). Within accessibility, Karpov and Ronzhin [68] proposed a conceptual model of a universal assistive technology architecture, using multi-modality to span several input devices across both software and hardware. Karpov and Ronzhin used this model to propose a system with a five-layer structure that bridges computer hardware, middleware, signal processing, interaction techniques, and assistive technologies in its design, but they did not evaluate the system's real-world utility or user preference.

Conceptual models of multi-modality are important as they supply initial structure for describing and discussing multi-modal interaction. However, as implementations of these models develop, messy tradeoffs, cuts, and optimizations happen that are not necessarily present in their theoretical underpinnings. As a result, these models provide insufficient insight into the experiences, configurations, and wider design ramifications of real-world multi-modal input usage. Our work uses the structures defined in these conceptual models as a tool for describing real-world multi-modal input configurations.

2.3 MULTI-MODALITY IN PRACTICE

In addition to conceptual investigations, prior work has also evaluated a variety of instances of multi-modal interfaces. Zander et al. [150] created and evaluated a system that combined eye gaze and brain signal input applied to a search-and-select task, finding that their multi-modal interface was slower but more accurate than using eye gaze alone. Lee et al. [76] evaluated the effectiveness of a multi-modal augmented reality interface that combined freehand gestures with speech input to change the shape and color of virtual objects, and Kammerer et al. [66] evaluated the combination of speech and gaze to understand how combinations of menu designs and input devices impacted accuracy and completion time of a menu selection task. The multi-modal interfaces were not clear winners in either investigation when compared to their constituent input devices, leading the authors to suggest that cognitive load should be a prominent consideration in multi-modal interface design. This insight was corroborated by Ruiz et al. [108], who demonstrated that users choose differing input methods (in their case, redundant or complementary) depending on cognitive load. Other applications of multi-modality include combining inputs commonly found within cars for use while driving [93], combining speech and pen input to write mathematical equations [5], and combining pen and touch input using the edges, faces, and corners of a small pencil tool [42, 130].

More relevant to our work is literature that examined multi-modality for accessibility and in assistive technology. Smith et al. [117] used speech and head tracking for object selection in both general computer use as well as Augmented and Alternative Communication (AAC) systems. Similar work combined gaze-tracking with a keyboard [21] and with face-tracking [107] to emulate mouse-and-keyboard input. Keates et al. [69] combined head-tracking and joystick input to emulate keyboard-and-mouse input, finding that cognitive load and training impact the usability of a multi-modal system.

These evaluations show that cognitive load is a prominent factor in a user's preference between traditional and multi-modal input systems, and that emulation of input is a beneficial feature of accessible multi-modal systems. Additionally, work like that of Smith et al. [117] and Keates et al. [69] show that combining devices has clear usability benefits. However, these works do not describe how people use these input combinations outside of laboratory settings. There remains a gap in understanding the organic preferences, challenges, and overall experiences of people who use accessible multi-modal input in their daily lives. Our work uses the usability insights from these studies to inform our interviews with participants, and identifies additional real-world factors that affect users' preferences between traditional and multi-modal input systems.

2.4 REAL-VIRTUAL ALIGNMENT

Prior work has explored *real-virtual alignment*, aligning elements of the real world with the virtual world. We discuss real-virtual alignment with regard to output (e.g., aligning haptics or surrounding visual elements), input (e.g., aligning real and virtual representations of input devices), and workspace (e.g., aligning real and virtual desks).

Real-virtual alignment within system *output* can increase awareness and comfort, as well as provide useful illusory effects. In addition to notifying the VR user of non-VR bystanders [40, 51, 73, 96, 131], aligning real and virtual worlds can enable more comfortable interaction with real-world objects [24, 116]. Hartmann et al. [60] explored real-time rendering techniques using a headset-mounted depth camera to allow users to see real objects in VR, finding that participants could view and interact with real objects within VR without losing presence in the virtual environment. RealityLens [133] extends this idea by focusing on placing user-defined views of the real environment from VR, finding that having elements of the real world blended in VR, especially during activities or interactions that involve them, can increase presence and comfort. At the same time, strategic misalignment of real objects and virtual proxies can be used for illusory effects [8, 110] which can be exploited to provide touch feedback [32] or increase comfort [141].

Real-virtual alignment within system *input* typically involves bringing desktop input devices into VR, like using real-virtual alignment to make typing on a physical keyboard in VR faster and more comfortable [58, 84, 101].

Aligning the real and virtual *workspace* can also be beneficial. Zielasko et al. [154, 155] used a real-virtual aligned desk to evaluate desk-mounted versus in-air menu selection techniques, finding that the passive haptic feedback of tapping menu items on a real-world desk in VR slightly improved menu interaction time. Wagner Filho et al. [132] also used a real-virtual aligned desk as a tabletop interaction surface for an immersive visualization prototype, finding that their tabletop and 3D gestural interface was more engaging but made some more precise interactions slower.

Real-virtual alignment in cross-reality interactions is important because it can increase user comfort as well as mediate friction caused by input device differences. Our work synthesizes the findings of Zielasko et al. [154] and Wagner Filho et al. [132] for improving comfort and functionality, and further explores real-virtual aligned input expanding on work like McGill et al. [84]. However, we place a more explicit focus on transitioning between

VR and desktop, using real-virtual alignment as one way to make these transitions more functional and comfortable. We expand on previous findings by creating a design space of VR-desktop transitions informed by real-world use cases, and exploring the impacts of transitioning between VR and desktop specifically.

2.5 2D INTERACTIONS WITHIN 3D

Many cross-reality workflows make use of 2-dimensional input within 3-dimensional environments. For example, Kim et al. [71] explored cursor movement techniques for spatial augmented reality, finding that using a head-mounted cursor with a perspective-based targeting technique [94] was superior for long distances. Similarly, Zhou et al. [152] evaluated a depth-aware technique, interpolating cursor depth and control-display gain based on object depth relative to the user, finding that the benefits afforded depend on the level of depth complexity in the scene.

Also relevant to our work are techniques for using flat panels, like 2D displays, in 3D. Early work by Coninx et al. [33] described a 2D/3D hybrid interface, using a boom-mounted 3D display. A pinch glove provided input to flat-panel UI elements within a 3D immersive modelling task. Similarly, the Boom Chameleon [126] used a touch panel mounted to a boom arm, combining precise 2D input with free 3D movement, in a 3D annotation prototype. Later work by Surale et al. [121] and Arora et al. [6] further explored the use of tablets in 3D environments (for 2D touch gestures in VR and 2D drawing in AR, respectively), finding that the use of interactive flat-panel screens for 3D drawing can mitigate the lower accuracy of 3D spatial input. Building upon earlier cross-dimensional gestural interfaces like that of Benko et al. [15], Zhu et al. [153] used a smartphone as a more precise input surface for head-mounted AR. Participants found the content engaging, specifically the ability to transfer content from 2D to 3D. Particularly relevant to our work, Wang and Lindeman [134] evaluated a hybrid design using an arm-mounted tablet and a non-occlusive VR headset. In a design task, they found that quick interactions with a secondary 2D interface can help complete more complicated VR design tasks more easily, despite the additional complexity of learning a hybrid system.

Implementing 2-dimensional interactions in 3-dimensional environments can mitigate the inaccuracy of spatial input, and benefit overall usability [121, 134]. Previous work provides initial design insights, but does not specifically focus on the objective and subjective impacts of transitioning between standalone 2D and 3D use cases. Our implementation enables both 2D and 3D input, within 2D and 3D environments, with an explicit focus on the impact of transitioning between them.

2.6 UNDERSTANDING ACCESSIBILITY CHALLENGES

Researchers have evaluated accessibility in other computing platforms. Previous work in eliciting accessibility challenges includes surveys of individual games [148], diary studies [95], and most importantly, direct observation studies [9]. Anthony, Kim, and Findlater [4] introduced a method for identifying accessibility barriers in YouTube videos, which allows researchers to systematically analyze diverse and rich data sources. Wentzel et al. [139] used a mixed-methods approach, combining this methodology with surveys and semi-structured interviews to investigate the use of multi-modal input techniques as solutions to otherwise inaccessible input scenarios. Importantly, they found that users often combine various input devices to overcome accessibility barriers.

Our work adopts a direct observation methodology similar to Babu et al. [9], using a contextual inquiry formative study to elicit accessibility issues, which then inform the design of a more customizable technical solution.

2.7 UNDERSTANDING VR MOTOR ACCESSIBILITY

Very relevant to our exploration is work that elicits and describes VR motor accessibility issues specifically. We summarize this work in Table 2.2, which is categorized by primary methodology, whether participants used VR directly, type of observation (directly in-person, or indirectly through interviews, videos, etc.), and participant population.

Yin et al. [147] used a survey to assemble accessibility issues with immersive content like VR and phone-based AR, but with less focus on mobility specifically. South et al. [119] explored barriers to VR accessibility for people with photosensitive epilepsy, finding various contributing factors across hardware, interfaces, applications, and individual sensitivity. Palaniappan, Zhang, and Duerstock [100] used VR and joint force calculations to identify comfort areas surrounding the body to provide objective ergonomic insights. Similarly, Cook, Dissanayake, and Kaur [34] analyzed VR controller hardware ergonomics, finding that controller designs may be inaccessible for older hands. Gerling et al. [48, 49] conducted a three-part survey and usability study evaluating VR for people in wheelchairs, followed by a discussion on general VR mobility assumptions from a theoretical perspective. Mott et al. [91, 92] discussed opportunities for accessible design and provided a general survey of the inaccessible aspects of VR hardware, after an online and in-person interview study. Six participants used VR, but the study was primarily concerned with hardware. Tian et al. [124] used a combination of surveys and interviews to create a collection of accessible freehand gestures for people with spinal muscular atrophy. Creed et al. [36, 37] elicited a collection of inaccessible aspects of VR through sandpit workshops and discussions, landing on a collection of general issues and future research directions for software and hardware.

Table 2.2: Summary and comparison of the most relevant previous works that also elicited VR motor accessibility issues. See text for more details.

	Methodology	VR Used	Observation	Participants
Our Work (Chapter 5)	Contextual Inquiry, User Study	Y	Direct	Variety of mobility limitations, detailed in Table 5.1 (N = 10). 8 also completed user study.
Creed et al. [36]	Sandpit	N	Indirect	9 disabled persons, 8 researchers, 11 academic experts, 14 stakeholders (N = 38, 4 with multiple roles)
Creed et al. [37]	(1) Sandpit	N	Indirect	15 unspecified, 7 researchers (N = 22)
	(2) Sandpit	N	Indirect	14 disabled persons, 9 researchers, 10 unspecified (N = 33)
Tian et al. [124]	Elicitation Study	N	Indirect	People with spinal muscular atrophy (N = 16)
Mott et al. [92]	Interviews (re: Hardware)	Y	Both	Variety of mobility limitations, 4 used VR (N = 16)
Gerling et al. [48]	(1) Survey	N	Indirect	All wheelchair users, 14 powered, 9 manual, 1 both (N = 25)
	(2) Usability Study	Y	Direct	All wheelchair users, 2 powered, 12 manual (N = 14)
	(3) Usability Study	Y	Direct	All wheelchair users, 3 manual, 1 manual + propulsion (N = 4)
Gerling and Spiel [49]	Theory-led Analysis	N	Indirect	Wheelchair-using participants from previous paper [48]
Yin et al. [147]	Survey	N	Indirect	One or more types of access needs, impairments, disabilities and/or long-term health conditions (N = 101)
Cook, Dissanayake, and Kaur [34]	Pilot Study	N	Direct	Older adults (unspecified)
Palaniappan, Zhang, and Duerstock [100]	Usability Study	Y	Direct	Person with tetraplegia - C4/C5 spinal cord injury (N = 1)

Six of the 9 most relevant works use indirect observations from discussions of past or imagined VR usage instead of directly observing actual VR usage by participants. Studies that did involve direct VR usage did not focus on technical solutions addressing the underlying conflict between user ability and the mobility required by applications. Time delays and inconsistent context of indirect observation (for example, workshops and interviews which involve recall or anticipation of issues) can reduce generalizability of experiential qualitative results [41]. There is an opportunity to examine and describe underlying mobility assumptions in VR applications in a concise and generative way, directly informing a customizable technical solution.

2.8 PREVIOUS APPROACHES TO IMPROVE VR ACCESSIBILITY

There exist some previous approaches to improve motor accessibility for VR applications. Thiel and Steed [122] explored "co-piloting", where a second user completes input on behalf of a primary user. They implemented techniques for a VR user to request the assistance of a remote co-pilot for inaccessible reaching actions, with an initial formative study finding such an approach feasible and helpful. Nearmi [74] compared multiple ways to re-orient the user's view toward objects of interest in VR for people with limited mobility, emphasizing a need for deep customizability in adaptive input techniques. Yamagami et al. [146] investigated how people can perform bimanual gestures using only one hand, creating a design space of bimanual motion characteristics. The authors developed prototypes within this design space, evaluating them with a video elicitation study.

We encompass and extend these previous solutions by focusing on a highly customizable motion remapping technique, which offers support for co-piloting, head movement, as well as single-handed bimanual gestures.

2.9 SUMMARY

Previous research in VR accessibility examines a wide variety of concepts, including accommodating for various forms of input, various methods for adapting input, as well as several implementations of transfer functions to directly remap spatial input. However, an overarching problem within the related work is that there is no unifying design language through which these remapping strategies can be understood.

The following chapters build upon the related work, through a multi-step strategy: qualitative work in understanding accessible input and its associated adaptations and trade-offs; quantitative work in understanding how context can be leveraged to address situational impairments; and quantitative work developing a unifying design language for accessible spatial input.

3

UNDERSTANDING HOW PEOPLE WITH LIMITED MOBILITY USE MULTI-MODAL INPUT

We begin our exploration by first motivating how people address accessibility issues in traditional computing platforms. If we are to make an adaptable and appropriate accessibility technique for spatial interfaces, we should first look to how people with mobility limitations have solved these issues for non-spatial interfaces. We focus on multi-modal input in order to understand not only input for a single device, but for multi-device configurations.

3.1 INTRODUCTION

Input devices are a crucial component of communicating with computers. At their core, input devices convert user-generated signals, such as physical movements, to input intelligible by computers [63]. For example, a mouse converts arm and wrist movement into 2D cursor movement, and finger flexion into selection. Embedded in the design of a mouse are the assumptions that users can make both coarse- and fine-grained arm movements, and that they possess the dexterity in their fingers to press buttons. Similarly, a standard video game controller enables a variety of game input but assumes that users have the strength and dexterity in their hands and fingers to interact with the joysticks, buttons, and triggers while simultaneously gripping the controller.

When users' movement abilities do not match the movement assumptions made by input devices, those devices, and the experiences they enable, may be inaccessible. Prior work has shown that people with limited mobility often encounter accessibility barriers when their abilities do not match these assumptions [143, 144]. For example, people who experience tremors might have difficulty making accurate selections with a mouse or touchscreen due to the fine movements required [69, 90, 125]. Moreover, people who use one hand might find a typical game controller inaccessible as its button and joystick placement implicitly assumes a two-handed grip.

Although an experience can be made more accessible by application developers (e.g., through accessibility options in the application) and hardware designers (e.g., adaptive devices such as the Xbox Adaptive Controller [145]), it is often up to individuals to address their own accessibility needs. In some cases, overcoming accessibility challenges involves creating a unique input device configuration with multiple devices working in tandem. This approach, known as multi-modality, is a topic of interest within the field of HCI [97, 98, 123]. Previous work has established various conceptual models of multi-modal computation for accessibility [68, 111] without exploring how these models are applied. Further work applies and evaluates these multi-modal input techniques [22, 79, 106] in lab settings. However, in addition to a lack

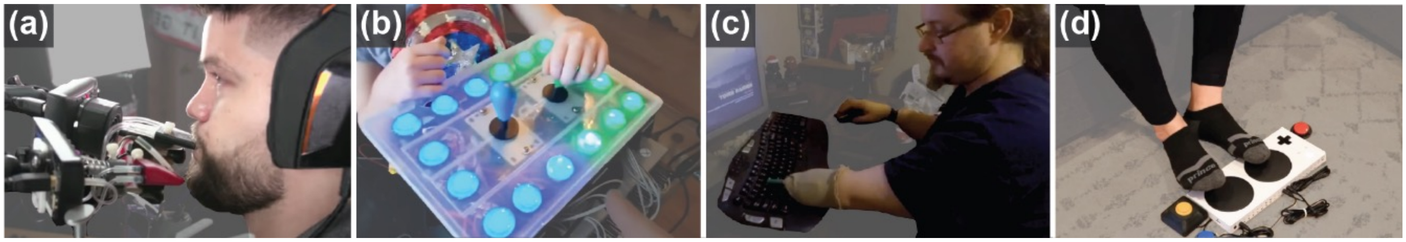


Figure 3.1: Examples of various accessible multi-modal input devices: (a) multiple QuadSticks; (b) a custom controller connected to the Xbox Adaptive controller; (c) mouse and keyboard used with a typing stick; and (d) switches connected to the Xbox Adaptive Controller (via YouTube [2, 88, 99, 115]). Images © Shot Callers Esports, ELEAGUE, MIZINO: In Over My Head, and ABSHOW, respectively.

of emphasis on people with limited mobility, these works lack a practical understanding of how people use multi-modal input and what influences their decisions when constructing input configurations in real-world scenarios. If researchers and practitioners were to know more about how people with limited mobility use, configure, and experience multi-modal input, they could create more accessible experiences that take advantage of peoples' real-world practices and preferences.

Our work contributes to the wider discussion of accessibility within cross-device computing by exploring three research questions:

(**RQ1**) *Which input devices do people with mobility limitations use, and how do they combine and configure these devices?*

(**RQ2**) *How does application or usage context affect these device or configuration choices?*

(**RQ3**) *How do users discover new accessible configurations, and how does this process inform future accessibility solutions?*

We grounded our investigation within the context of video games, as previous work shows that gaming is a common setting for both real-world multi-modal input and accessibility research [3, 95, 148].

To illustrate how people with limited mobility choose, set up, and use multi-modal input, we explored the device ecology [10, 61] of this space through a three-part investigation. First, we surveyed 43 people with limited mobility about the input devices and configurations they use. We found that multi-modal input was common and that most multi-modal device configurations were user specific. Next, we interviewed 14 people with limited mobility to gain a deeper understanding of the experiences and challenges associated with creating and personalizing multi-modal input configurations. We found that most accessibility issues with multi-modal configurations—as with input in general—occur at two key connection points: between the user and the device, and between the device and the application. Failures at each connection point have their own unique remedies, as described by participants. Additionally, participants often looked to online videos for inspiration when creating and configuring multi-modal setups (Figure 3.1). Inspired by our participants' discovery process, we performed a systematic review of 74 YouTube videos to categorize and illustrate real-world examples of multi-modal input. We found that input configurations and usage styles vary based on individual platform and device compatibility.

Our paper makes the following three contributions: (1) empirical results from a three-part investigation on the landscape of multi-modal input for people with limited mobility, including setups, compatibility challenges, and associated remedies; (2) the identification and description of users' adaptation processes for combining input devices to overcome accessibility barriers; and (3) a discussion on how our findings can influence research and practice within HCI and the wider accessibility community. Together, these findings provide a description of the ecology of multi-modal device use by people with limited mobility.

3.2 BACKGROUND

Chapter 2 discusses related work on multi-modal input and the research methods used to study it in-situ. The evaluation of multi-modal interfaces is a common topic within HCI, with previous work providing broader reviews of the area [97, 123, 128]. To inform our methods and help us examine real-world multi-modal input, we also discuss research about accessibility-oriented systematic reviews and research on accessible gaming devices.

3.2.1 *Evaluating Accessibility: Gaming and Social Media*

We examine how people who play video games use multi-modal input. Previous works frequently used gaming as a lens for evaluating accessible input devices and techniques, the insights from which can be applied to wider HCI contexts. In addition to systematic reviews [3], evaluating assistive technology's prevalence in real-world applications has involved surveys of individual games [148], diary studies [95], and using web content accessibility guidelines as a heuristic to evaluate the accessibility of individual games [109].

Evaluating technology accessibility can also involve in-situ observations. While in-situ observations can provide high ecological validity, they may not scale well and are sometimes not logistically possible. Furthermore, lab studies that include people with limited mobility often face low participant numbers or small population cross-samples. Anthony et al. [4], in a study of accessible touchscreen use in people with limited mobility, bypassed participant recruitment issues by examining user-generated YouTube videos instead of interacting with people directly. The authors presented the analysis of the videos alongside survey data, showing the effectiveness of social media for collecting in-situ examples.

Our investigation used a multifaceted approach to understanding the range of user experiences associated with accessible multi-modal input systems. As such, in conjunction with traditional qualitative analysis techniques, we explored this space using the tools described in earlier gaming, accessibility, and social media research. We combined the more applied focus of the gaming-related works with the social media approach of Anthony, Yang, and Koedinger [5] to more thoroughly describe real-world usage behaviours.

3.2.2 *Summary*

Designing accessible multi-modal input systems requires knowledge of users' real-world usage habits, including the relationship between people and their own devices. Prior work has explored conceptual models, applications, and accessibility evaluation methodologies of multi-modal input, but lack a practical understanding of multi-modal input from a real-world user perspective. This lack of understanding is especially prevalent when focusing on accessibility and people with limited mobility. Previous work has shown that describing device ecologies is a critical early step in understanding the affordances and implications of various categories of computing [23, 80]. As such, to provide a full understanding of real-world usage, we focus our investigation on describing the ecology of multi-modal input devices for users of conventional computing systems with limited mobility. We describe this ecology with the results of a three-part investigation that uses the prior conceptual models [25, 68] as a framework for analysis, the usability insights of the lab studies [21, 70, 107] to guide our interviews, and the social media approach of Anthony et al. [4] for our systematic review methodology.

3.3 A THREE-PART INVESTIGATION THROUGH VIDEO GAMES

While multi-modal input systems are a common topic within HCI research, the prevalence, user habits, and accessibility ramifications of multi-modal input systems are currently unclear. Our work illustrates these habits and contributes to the wider discussion of accessibility within cross-device computing through three topics: the prevalence of input devices in users with limited mobility, including combinations and configurations (**RQ1**); the impact of usage context on these input configurations and their associated usability (**RQ2**); and how these configurations can be better discovered and supported in conventional applications (**RQ3**).

To recruit survey respondents and interview participants, we contacted organizations and communities that focus on video game accessibility for people with physical disabilities. As a result, gaming was a common consideration in our respondents' and participants' multi-modal setups. In addition to previous work evaluating accessibility through gaming [103, 148], we believe gaming was an appropriate context for our investigation because it fits three primary criteria: it requires complex multi-modal input to cover a reasonable range of input scenarios; it demands multi-modal input frequently enough that users must prepare or adapt to it for full utility; and it uses devices that are or could be used within other computing contexts. Video games often require quick and repetitive input across several devices (e.g., mouse and keyboard in PC gaming; headset and controllers for VR), which is usually pivotal to effective control of the game. Video game input devices are used in other contexts within HCI as well as general input scenarios [82, 89, 135].

The multi-modal input situations common in games also frequently occur within general computing. For example, 3D CAD (Computer Aided Design) software such as Fusion 360 often requires complex multi-modal input combinations to manipulate objects in a 3D space - an action that occurs often in video games. Similarly, as VR gains popularity in non-gaming social and productivity contexts, the multi-modal input techniques originally developed for VR games enter the realm of conventional non-gaming VR usage [91]. We grounded our investigation within the context of video games as it is a generalizable real-world setting for multi-modal input.

3.4 SURVEY

Previous work categorized and characterized multi-modal input [22] but lacked insight into how multi-modal input is utilized in real-world scenarios. To answer **RQ1** and understand how people with limited mobility use multi-modal input, we first assessed the prevalence of their input devices and frequently utilized combinations.

We surveyed 43 respondents recruited via online postings and word-of-mouth. The survey took approximately 20 minutes to complete and respondents who completed the survey were entered into a draw for one of three $50 Amazon gift cards.

Our survey covered the following topics: respondents' demographic information and the nature of their mobility limitations; computing devices they use regularly; input devices they use regularly; and their experiences with combining multiple input devices to complete tasks. Answer formats included open-ended text responses, multiple-choice questions, and scale ratings. The supplementary materials provide more details about the survey, including all questions asked of our participants.

3.4.1 *Analysis*

We analyzed the survey data using a combination of open and axial coding [35]. The first author read and open coded responses to the open-ended survey questions, using inductive analysis [86] to identify common topics. Then, the first author and three additional authors conducted an axial coding step, creating and iterating on six thematic categories based on participants' responses. Our cooperative approach allowed us to work towards agreement on the key themes present in the data, which is a common method in qualitative data analysis [83].

3.4.2 *Demographics and Mobility Limitations*

Respondents were aged between 8 and 62 with a median age of 33. Of the 43 respondents, 32 identified as men, 8 as women, and 2 as non-binary.

Table 3.1: All mobility limitations and the number of times they were reported by survey respondents.

Mobility Limitation	**Number of Responses**
Difficulty gripping	34
Difficulty in holding	34
Low strength	26
Slow movements	23
Difficulty controlling movement distance	19
Rapid fatigue	19
Difficulty controlling movement direction	18
Poor coordination	15
Lack of sensation	11
Spasms	9
Tremors	4
Quadriplegia	2
Inability to walk	1
Hand paralysis	1
Left hand impairment	1
Extreme progressive weakness	1
Finger and wrist flexor paralysis	1
Limited to one hand	1
Limited general range of motion	1
Joint hypermobility	1
Slow reaction time	1
Difficulty typing	1
Vision issues	1

One respondent did not list their gender. As a condition of participation, all participants self-reported as having limited mobility. Respondents reported a variety of mobility limitations, the most frequent being difficulty holding (34 respondents), difficulty gripping (34), low strength (26), slow movements (23), and difficulty controlling movement distance (19). Respondents considered themselves experienced with computers, as 42 of 43 respondents rated their computer expertise 3 or higher on a 1-to-5 scale. Fifty percent of respondents started using computers at or before age 10, and 50% acquired their disability at or before age 16. When asked on a scale from 1 ("Never") to 10 ("Always") how often they use two or more devices simultaneously, 33 respondents (77%) rated 5 or higher, with 22 respondents (51%) answering 10. Table 3.1 shows the number of mobility limitations reported by survey respondents, and full respondent demographics can be found in the appendix (Table A.2).

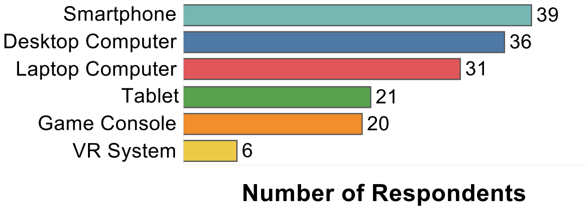


Figure 3.2: Number of respondents who reported using each computing device.

3.4.3 *Computing Devices*

To establish a baseline usage rate for each device, respondents were asked to list the computing devices they have in their home (Figure 2). Unsurprisingly, smartphones, desktops, and laptop computers were the most common. The adoption of game consoles was lower than we expected considering this population was more oriented towards gaming. One possible explanation could be that despite recent improvements to accessible game controller platforms [53], consoles' lower compatibility with third-party input devices can still present a challenge to end-users, prompting them to play games on other platforms like desktop or laptop computers.

Similarly, the low adoption of standalone VR systems speaks to the still-pervasive inaccessibility of these devices [49, 92], particularly for people with limited mobility. The reason for this difference in adoption, especially compared to devices like smartphones, could be the extremely different assumptions these devices make about users' mobility. The most pervasive mobility limitations in the survey were difficulties with gripping, holding, and general strength, all of which can present challenges when using current VR controllers [48, 49, 92]. Respondents confirmed their accessibility issues with VR in the open-ended questions, for example respondent 3: *"For normal gaming and computer usage I am fine. I would like to be able to use VR/MR technology that depends on full mobility, e.g., leg-based mobility"*.

3.4.4 *Individual Input Devices*

We also asked respondents to list input devices they use to interact with their computing devices (Figure 3.3). Mouse and keyboard are the de facto standard for desktop computer input, making their popularity unsurprising. Respondents also reported using voice input as often as a keyboard and mouse, which is similarly unsurprising given the prevalence of voice input in society (e.g., smartphone voice assistants, smart home devices). Devices specifically marketed as assistive devices (e.g., switches, mouth-controlled joysticks) only begin to appear at the 8th most common position, which could be due to cost or compatibility barriers. Alternatively, disability is a wide spectrum, meaning it could be the case that our respondents' movement abilities did not necessitate more specialized devices. Overall, the device use

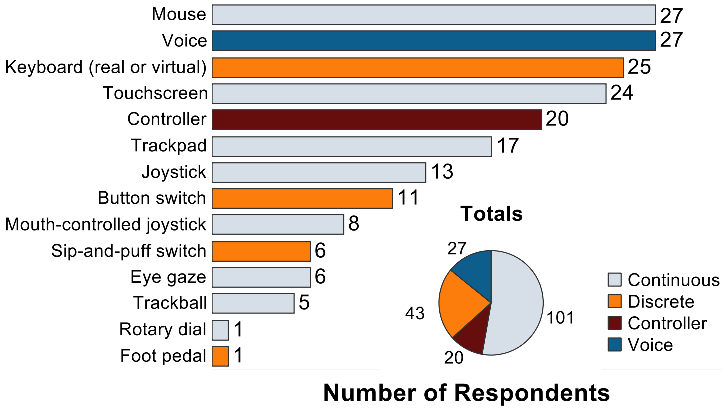


Figure 3.3: The number of respondents who reported using each individual input device.

data illustrates how individuals' abilities and preferences can lead to a wide variety of adopted input devices.

We further categorized these input devices based on the signal domain [25, 79] they support. At a sensor level, input signals are categorized as either discrete (on or off, like a button) or continuous (any intermediate point within a given input space, like a dial [25, 79]. These two categories cover most but not all devices. For instance, modern devices often combine multiple continuous and discrete inputs, like a controller with multiple joysticks and buttons, which makes these categories complicated to apply at a device level. Likewise, the broad range of natural language makes voice input hard to categorize as either discrete or continuous. As a result, we separated controllers and voice into independent categories. Other devices, like mice with several buttons, could fit this characterization, but we addressed this variance by categorizing each device's primary axis of input. For example, the mouse's two-dimensional motion sensing places it in the Continuous category.

While it is unsurprising that continuous devices are the most common based on the commonality of the mouse and touchscreen, the lack of diversity in discrete input devices is interesting. Keyboards are common, while other discrete input devices see little use. This could indicate compatibility issues or difficulty in setting up a switch-based configuration. For example, respondent 43 writes: *"complicated games need too many switches, making setups bulky to use"*.

3.4.5 *Input Combinations*

Respondents also answered questions about how they combined their input devices. They answered open-ended questions which prompted them to describe and rate each combination of input devices they use. Further questions involved describing the experience of setting up, switching to, and using these input combinations.

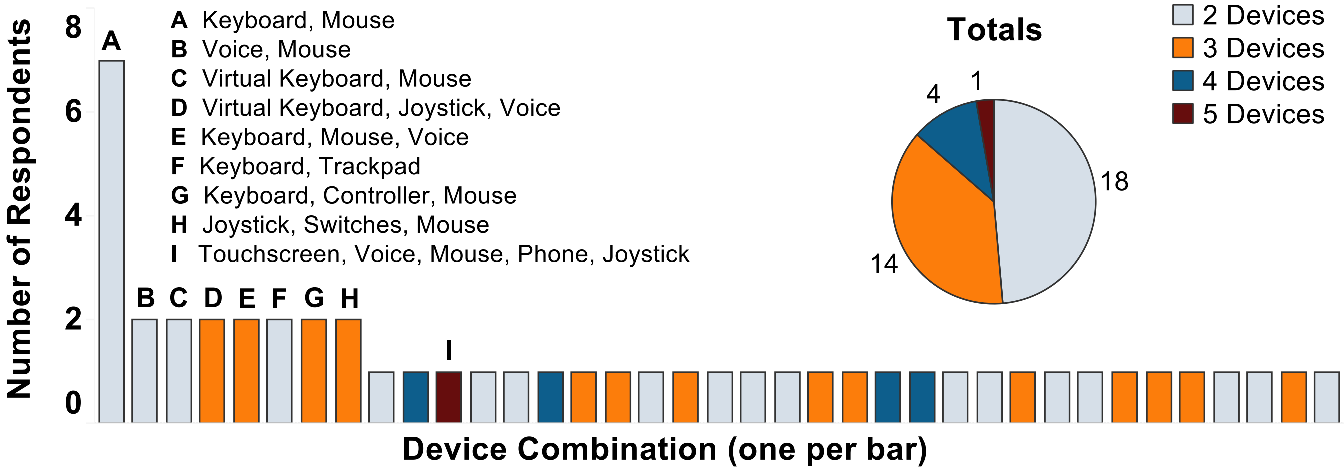


Figure 3.4: The distribution of 37 unique device combinations and the number of respondents who reported them.

Table 3.2: Categories for improving multi-device input as reported by survey respondents.

Theme	Responses
Inter-compatibility and switching between devices	21
Input difficulties and errors	16
Individual input device functionality improvements	14
Shortcut and key-binding improvements	4
Reduce discomfort or fatigue	2
Improve device availability	1

Respondents reported 37 unique combinations of 21 unique input devices, and 52 combinations in total (Figure 3.4). Most input configurations involved 2 or 3 devices, four others used 4 devices, and one combination used 5. In addition to the most popular combination of mouse and keyboard, most input combinations reported by at least 2 participants each used a pointing device (like a mouse or trackpad) alongside a typing device (like a physical or virtual keyboard). However, this commonality in configurations describes only a small part of the data set. Most device combinations occurred only once among all respondents, and 14 of 37 did not feature mice or keyboards. Detailed information on each device combination can be found in supplementary materials

When asked how to improve each input combination, respondents provided a variety of answers describing their experiences. We used a combination of open and axial coding to categorize these responses into 6 themes listed in Table 3.2.

Our device combination data showed that there is no one-size-fits-all design solution for accessible multi-modal input. When the use of the most common input devices (in this case, mice, touchscreens, and keyboards) is difficult or

Table 3.3: Demographic information for interview participants.

ID	Age	Gender	Mobility Limitations
P1	37	W	Progressive neuromuscular disorder, quadriplegia, limited finger dexterity
P2	33	W	Underdeveloped left hand, low hand dexterity
P3	36	M	Quadriplegia, limited arm and leg mobility, limited finger dexterity, muscle spasms
P4	25	W	Muscular atrophy, limited arm mobility, limited finger dexterity
P5	34	M	Cerebral palsy, paraplegia, muscle weakness, poor coordination, low dexterity
P6	30	M	Asymmetric paralysis, limited arm mobility, limited fine motor skills on right side
P7	19	M	Paralyzed in fingers, limited wrist mobility
P8	25	M	Cannot walk, limited arm/finger movement
P9	27	M	Limited arm mobility, limited leg mobility
P10	30	M	Quadriplegia, limited arm mobility, limited finger dexterity, rapid fatigue
P11	33	M	Duchenne muscular dystrophy, paralyzed from neck down, low finger movement
P12	44	M	Limited hand mobility, cannot walk, limited finger dexterity
P13	34	M	Quadriplegia, paralyzed from chest down, limited finger dexterity
P14	41	M	Limb girdle muscular dystrophy, rapid fatigue, limited arm mobility

impossible, the device combinations used to compensate can vary just as much as individual users' abilities. Respondents described this issue in detail, citing inter-compatibility and switching between input configurations as some of the main avenues for improving input combinations. As the number of devices available to users continues to grow, it will become increasingly difficult for designers to anticipate which devices people use. Designing for customization, inter-compatibility, and robustness to accommodate the extensive variety of input setups remains a priority when supporting accessible multi-modal interaction.

3.5 USER INTERVIEWS

Our survey shows that multi-modal input is common, but setups can vary widely depending on individual accessibility needs (**RQ1**). Deeper insight into these needs, including experiences, challenges, and solutions, can be useful to designers and developers supporting multi-modal input systems. To gain these insights and answer **RQ2**, we interviewed people with limited mobility about their multi-modal input configurations.

3.5.1 *Participants*

We recruited 14 people with limited mobility to participate in our study. Participant ages ranged from 19 to 44 with a median age of 33. Of the participants, 11 identified as men and 3 identified as women. Three of the interview participants also completed the survey. Table 3.3 provides full demographics and mobility limitations.

3.5.2 *Interview Protocol*

Participants and interviewers connected over Microsoft Teams. Each interview lasted approximately one hour, and participants were compensated with a $50 Amazon gift card. After a small introduction of the interviewers and explanation of the topic, participants answered demographic questions and discussed their self-identified disability. Participants then listed all computing devices they actively use. For each device, we asked participants to describe their preferred input devices, followed by open-ended questions exploring their usage experience. We took inspiration for the open-ended questions from the categories for input improvement from the survey results (Table 3.2), and a sample of the questions can be found in supplementary materials.

3.5.3 *Analysis*

We recorded and transcribed each interview, and this data is available to participants upon request. We analyzed the interviews using thematic analysis, employing a combination of open and axial coding [35]. The first author (also the primary interviewer) performed open coding, using an inductive approach to separate participants' device usage and customization practices into codes. Following the open coding step, the first author and three additional authors conducted an axial coding step, iteratively refining the codes into themes based on similarity and relevance. Similar to our analysis of the survey data, our cooperative and iterative coding approach allowed us to work towards consensus on relevant themes [83].

3.5.4 *Results*

Participant discussions focused primarily on five aspects of multi-modal input: how accessibility drives their input device and game purchase decisions, adapting individual devices to their accessibility needs, customizing their applications, configuring multi-modal input setups, and finding new multi-modal configurations. We describe the results and primary themes from the participant discussions.

3.5.4.1 *Choosing Platforms, Input Devices, and Games*

Discussion with participants involved several open-ended questions about their input devices, as well as their preferred games and game genres. Table 3.4 shows participant input device and game genre preferences, and specific game titles mentioned by participants are included in the supplementary materials.

Participants reported playing games on a variety of platforms including mobile and console, but primarily preferred PC games due to the higher level of compatibility with assistive devices. Participants often gravitated toward slower-paced games like role-playing games and adventure games because the fast reaction times needed in genres like first-person shooters was often an

Table 3.4: Input devices used and preferred game genres of interview participants.

ID	Input Devices Used	Game Genres
P1	Trackpad, joystick, virtual keyboard	Role-playing
P2	Touch screen, Nintendo Switch Joy-Cons, trackpad, physical keyboard	Role-playing, adventure
P3	Vertical mouse, touch screen, keyboard, PlayStation 4 controller, voice input	First-person shooter
P4	Mouse, virtual keyboard, touch screen, joystick	Role-playing
P5	Touchscreen, trackpad, PlayStation 4 controller, physical keyboard, voice input	Role-playing
P6	Keyboard, mouse, Xbox controller, touch screen, VR headset, Nintendo Switch Joy-Cons	N/A
P7	Touchscreen, trackball, trackpad, voice, stylus, QuadStick, Xbox Adaptive Controller, joystick, VR headset	Role-playing
P8	Mouse, virtual keyboard, voice input, chin switch, knee switch, gaze input	Strategy, racing
P9	Touchscreen, keyboard, mouse, Xbox controller, VR headset	Fighting, racing, first-person shooter, role-playing
P10	Touchscreen, voice input, stylus, custom game controller, mouse, bite switch, keyboard	First-person shooter
P11	Joystick, mouse, gaze input, voice input, QuadStick, switches, touchpad, touch screen	Platformers, Fighting
P12	Mouse, joystick, Xbox controller, keyboard, VR headset, voice input	Role-playing
P13	Keyboard, touchpad, Xbox Controller, PlayStation controller, Nintendo Switch controller	Adventure, role-playing, strategy
P14	Mouse, virtual keyboard, voice input, physical keyboard, Steam controller, VR headset, touch screen	Strategy

obstacle with their custom input configurations. Some participants avoided online games entirely due to their limitations. P1 explained: *"I have to make the choice between moving my character and [performing other in-game actions]. I don't move that fast to begin with and that can be tiring. I don't often play in groups since they usually don't understand why I'm moving slow."* 8 of the 14 participants mentioned their preference for slower-paced role-playing games.

Participants noted their use of word-of-mouth recommendations and accessibility forums as a first step in their game research process, as described by P4: *"if we [herself and similarly-abled friends on accessibility forums] find a game that works really well for us, we'll tell each other so that we can play together"*.

Participants noted making extensive use of subscription services, game trials, and return systems as a safeguard against buying games that are incompatible with their capabilities or input configurations. If a purchased game is inaccessible, participants will return these games using the refund system available in most online game stores. Game subscription services let users download unlimited games from an online catalog at a single monthly

Figure 3.5: (a) The controller P12 modified to make its joystick easier to use. (b) The custom controller created for P10 as an accessible substitute for other game controllers.

cost, meaning that trying inaccessible games has no financial consequence. P9 confirms: *"Xbox Game Pass [a game subscription service] has been great for trying games to explore their accessibility"*.

3.5.4.2 *Individual Device Adaptations*

Participants noted the use of several input devices (Table 3.4) and discussed their experience using these input devices as well as how these devices fit into their setups. If they found a device to be inaccessible, participants would often adapt their usage of the device for improved accessibility. Each device usage variation reported by participants fits into one of four categories:

- *Conventional usage* involves a user interacting with a system's typical input device (e.g., controller on a game console) in the most common way, albeit with potential alternate control schemes to make usage easier. For example, P14 used a mouse conventionally, but added additional software functionality to its buttons to compensate for his reduced ability to use a keyboard.
- *Adapted grip* involves using the device in a position different than conventional. For example, using an Xbox controller with only one's feet [P9] or holding a PlayStation controller sideways in one hand (P3).
- *Adapted device* usage involves physically modifying the input device to make it more accessible. For example, P12 wrapped the left joystick of his Xbox 360 controller in tape to make it easier to manipulate with reduced finger dexterity (Figure 3.5a).

- *Alternate device* usage involves substituting a system's typical input device for another device entirely. This includes using a custom-made accessible gamepad (P10, Figure 3.5b), or aftermarket assistive devices like a QuadStick[1] or Xbox Adaptive Controller[2].

Participants adopted these individual device adaptations to address mismatches between their movement abilities and devices' supported movements. As in the survey, the range of adaptations and alternate devices depended on users' unique mobility. Moreover, users reported adapting their existing devices when more accessible alternate devices were unavailable or unsupported, trying to *"[change the] ergonomics as best you can within the confines of the controller"* [P10].

3.5.4.3 *Customization*

The wide range of input devices and user adaptations means that ensuring software compatibility is a process often left up to the user. In conjunction with usage style adaptations, participants would apply software customizations to make applications more compatible with their input configurations. Participants described using remapping, or reassigning software functions to alternative controls or input devices. Several participants felt that remapping was important to accessibility. P2 elaborates: *"A lot of games have controls on different ends of the keyboard, and that's a hard thing for me to do. Games that support remapping are really great for me"*.

Two forms of remapping emerged: changing between premade control schemes provided by an application, or reassigning functionality to different inputs individually. Important to note is that premade control schemes commonly only swap controls within the same input device (e.g., alternate button mappings on a standard controller), rather than supporting the use of multiple devices. Participants noted that premade control schemes were frequently incompatible with assistive input devices thus forcing them to remap individual functions and inputs, which often required several hours of trial and error. P10 describes this increased cognitive load: *"the first two full sessions [using a remapped control layout] are just figuring it out"*. Remapping strategies were influenced by the game's controls, the game's genre (i.e., first-person shooters demand different controls than role-playing games), and even context within a single game. An example from P8: *"Overwatch [a first-person shooter] has different game mechanics and controls for each character, so my remapping settings are totally different each time"*.

Applications sometimes demanded the use of inaccessible devices, while simultaneously offering minimal support for remapping. To address this issue, participants would often use virtualization, or emulating the use of an inaccessible device through software. For example, P1's limited ability to use a keyboard prompted her to use a mouse alongside an onscreen keyboard when playing certain games, effectively virtualizing the keyboard through the

---

[1] `https://www.quadstick.com/`

[2] `https://www.xbox.com/accessories/controllers/xbox-adaptive-controller`

mouse. P4 described virtualizing a touchscreen through her computer input devices: *"a lot of games are hard for me to use through my phone, so I use [a phone companion program] to use my phone through my computer"*. Several participants mentioned using voice to virtualize keyboard input, either for typing or for individual game commands through applications like VoiceAttack[3]. Similarly, several participants reported using joystick input to virtualize keyboard input using applications like Joy2Key[4].

3.5.4.4 *Combining Multiple Devices*

Like in the survey, participants reported using a wide array of multi-device input configurations which matched their individual abilities. Most participants reported using multiple devices at the same time, such as a trackball and trackpad (P7) or a joystick and switches (P8), to do one input task. Participants used these devices in tandem to complete a typically inaccessible task, or to supplement the functionality of an existing assistive device. For example, P7 found click-and-drag interactions inaccessible, so he separated the task's components between devices, maintaining a mouse click with the trackpad in his left hand and moving the cursor with a trackball in his right hand. P4 uses a touchscreen and joystick simultaneously to interact with her smartphone, describing her usage: *"phone screens are so big these days so I can't reach [items at the top of the screen]. In those cases, I use the joystick"*.

3.5.4.5 *Discovering New Configurations*

The wide range of disability makes the process of finding an accessible and sufficiently compatible device configuration extensive and complicated. To remedy this, participants often looked to social media to find content creators with similar abilities, using creators' depicted configurations as inspiration for their own accessible device setups. P7, who plays games with a combination of a trackball, trackpad, and joystick, learned about this configuration from creators on YouTube and Twitch. Similarly, P8 noted: "usually I'll search around for setups using Google, then further research using YouTube". P10 described how reviews on YouTube and Reddit were a critical part of his process for setting up his input devices. Every six months, he and his recreational therapist consulted videos on YouTube and Reddit for gamers with his disability and observed and recreated their input setups.

Despite these processes, barriers to exploration persist. Motivation and time to learn a new input configuration is often an obstacle when searching for and creating new setups. As P6 described: *"even if it's more or less helpful, it's still another device to learn"*. Participants reported knowledge of dedicated assistive devices but were less knowledgeable in creating their personal ideal setup. As P3 described: *"I've heard of devices like the Xbox Adaptive Controller, but I'm not sure what kind of a setup would be best for me"*. Likewise, the often-arduous process of customizing accessibility settings is its own barrier. P5 explains:

---

[3] http://voiceattack.com

[4] http://joy2key.net

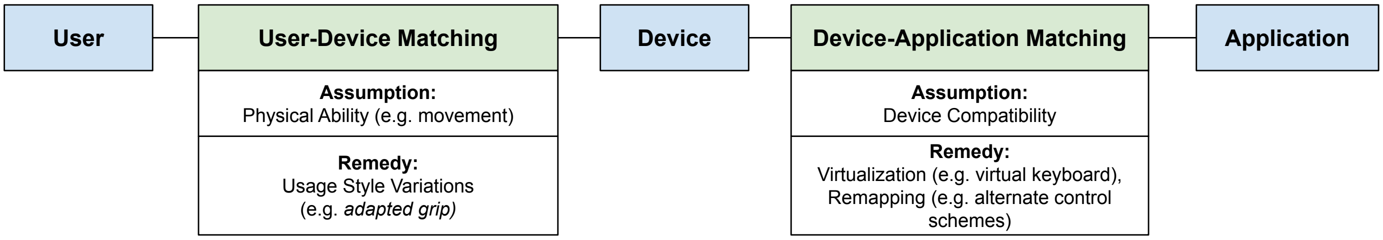


Figure 3.6: The two-stage matching process between the three components of interaction with a system. Accessibility issues reported by participants came from mismatches either between the user and device, or device and application.

> *"Even when you're looking for the accessibility options, they're often hidden. It takes you out of the immersion of the game to go hunting for the settings that make the game playable"*.

3.5.5 *Discussion*

To understand how people use multi-modality to address inaccessible computing systems, we must understand the underlying motivations of the people who use multi-modal setups. Discussions with participants revealed a common theme between the topics of adaptation, customization, and exploration.

Consider a complete interaction involving three sequential components, inspired by Savidis and Stephanidis [111]: a user, a device, and an application. The preservation of input fidelity from user to application can vary within each component. User input varies with the user's movement ability. Device input varies based on the range of user movement a device detects, and likewise the range of signal it can relay to the application. Finally, application input varies depending on the range of signals an application is prepared to receive and interpret from input devices. Each accessibility issue identified by participants fell at some point in this sequence.

This user-device-application input chain involves a two-step matching process: user-device matching, and device-application matching (Figure 3.6). We describe the matching process through its ability assumptions, similarly to Wobbrock et al. [143, 144]. User-device matching involves the central assumption that a user can manipulate a given input device with the precision and amplitude the device expects. This includes both the mobility of the user and correct placement of the device relative to the user. Mismatches occur when a user's abilities do not match the device's assumptions and cannot provide input along the device's expected degrees of freedom. Participants remedied these mismatches by adopting alternative usage styles for conventional input devices (e.g., adapted grip, adapted device) or using alternate combinations of devices (e.g., trackball and trackpad). Similarly, device-application matching involves the central assumption that a given device can provide all categories of input which the application expects. Mismatches in this step typically signify an input device which is unable to supply sufficient input fidelity, or an application not natively supporting a given input device. Participants

Table 3.5: The 60 accessibility keywords [4] and 8 gaming keywords used to construct YouTube search queries.

Accessibility Keywords
AAC, accessibility, ALS, amputation, amputee, arthritis, assistive technology, ataxia, augmentative communication, brain injury, cerebral palsy, congenital amputation, congenital amputee, disabilities, disability, disease, dystonia, essential tremor, Friedreich ataxia, Friedreich's ataxia, handicap, hemiplegia, hemiplegic, hydrocephalus, hydrocephaly, Lou Gehrig's, Lou Gehrig's disease, medical amputation, medical amputee, motor disabilities, motor impairment, MS -Microsoft, multiple sclerosis, muscular, muscular dystrophy, myopathy, paralysis, paralyzed, paraplegia, paraplegic, Parkinson's, Parkinson's disease, physical disabilities, psychomotor agitation, quadriplegia, quadriplegic, rehabilitation, sclerosis, seizure disorder, SMA, special needs, spina bifida, spinal, spinal cord injury, spinal muscular atrophy, stroke, TBI, traumatic brain injury, tremor, wheelchair
Gaming Keywords
Gaming, video games, videogames, PC gaming, console gaming, gaming setup, XBOX, PlayStation

remedied this situation by either reassigning application controls (e.g., game control remapping) or virtualizing the system's expected device (e.g., virtual keyboards).

Multi-modal configurations often involved a combination of remedies to these mismatches. Participants reported creating multi-device setups to overcome specific barriers in hardware or software. Changing existing hardware or adding more devices addressed user-device mismatches. Likewise, changing software controls or adding more software layers for added application compatibility addressed device-application mismatches. Interesting to note is that one could consider virtualizing alternative input devices as its own form of multi-modality, as it effectively assigns several categories of input to a single device. Designers and developers should consider this adaptation process to better support multi-modal systems.

3.6 SYSTEMATIC ANALYSIS: YOUTUBE (RQ3)

Designing to holistically support accessible multi-modality involves understanding how people with limited mobility search for new accessibility solutions. Participants in our interviews would often look to social media when creating a new device configuration, including Facebook, Twitch, and primarily YouTube. We borrow this process of browsing social media to answer **RQ3** and contextualize our input categorizations with additional real-world examples. We conducted a systematic review of 74 YouTube videos to find and categorize additional in-situ examples of accessible input setups.

3.6.1 *Procedure, Inclusion Criteria, and Analysis*

Following the example of Anthony et al. [4], we assembled a list of 60 accessibility-related keywords and 8 gaming-related keywords (Table 3.5). We constructed queries using one accessibility-related keyword and one gaming-related keyword, resulting in 480 total queries. Keywords containing multiple words (e.g., "assistive technology") were placed in quotation marks upon inclusion in a query. An automated script constructed all possible queries

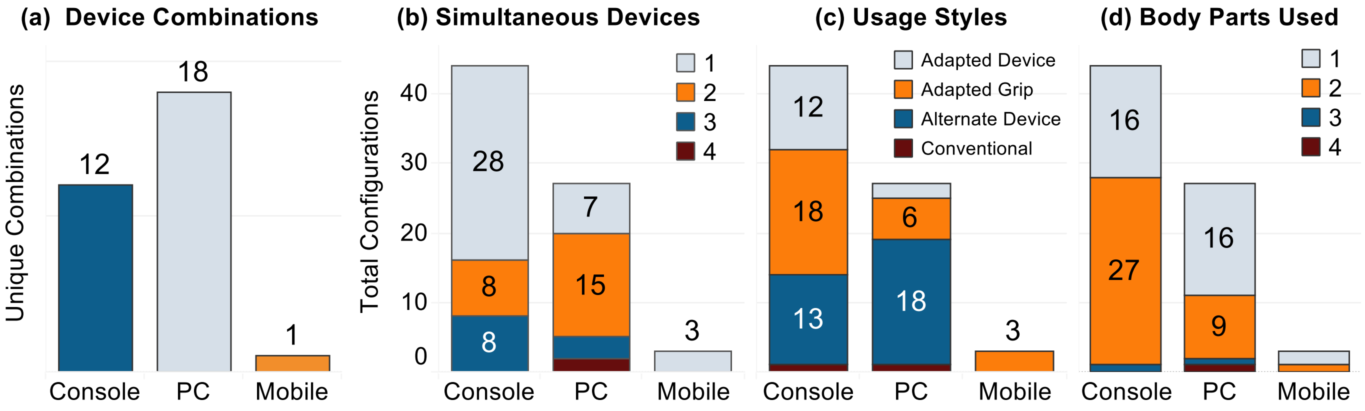


Figure 3.7: Proportions of input configuration properties over all videos from the YouTube dataset, separated by computing device used. Properties include the number of: (a) unique input device combinations; (b) simultaneous devices used; (c) usage styles adopted; and (d) body parts used simultaneously.

from each combination of two keywords, then used the YouTube Search API to construct a list of results for each query. After manual searching revealed that videos past the top 10 results showed low relevance, we limited the searches to the top 10 results for each query. After filtering duplicate results, our initial analysis set included 2061 total videos.

We filtered videos in our initial analysis set based on their relevance. Relevant videos had to: (1) show a person with limited mobility; (2) clearly display the input device(s) being used; and (3) be uploaded first-party, by either the player of the game, the player's caretaker, or an organization with the player's consent (e.g., a news interview). The third condition specifically excludes content aggregators like compilation channels, re-uploads, or unauthorized stream recordings. After filtering for relevance, our final dataset had 74 videos from 66 unique YouTube channels. The median length of the included videos was 12 minutes and 2 seconds, and we examined each video for its entire duration. The full list of videos is included in supplementary materials.

Next, we created a set of codes for video analysis. We refined and used these codes with a three-phase process. Two researchers individually coded a set of 15 videos (20% of the dataset), followed by discussion of disagreements and refinement of coding dimensions. After this, the two researchers re-coded the same set and calculated Cohen's kappa as a measure of inter-rater reliability for each dimension. Finally, one researcher coded the remaining videos using the refined coding scheme.

3.6.2 *Results*

We identify general trends in the video dataset. Where relevant, each dimension is accompanied by its Cohen's kappa (κ) as a measure of inter-rater reliability.

3.6.2.1 *Device Setup*

In our dataset, 44 videos showed users playing on a game console, 27 were on a PC, and 3 were on mobile devices ($\kappa = 0.85$, "near perfect agreement"). Videos showing console players showed 12 total unique input combinations, videos of PC players showed 18 unique combinations, and videos of mobile players showed only one (Figure 3.7a). The most common console configuration was a single game controller (22 videos), while the 3 next most common all involved the Xbox Adaptive Controller and devices that connect to it. The most common PC combination was mouse and keyboard (5 videos), followed by a single QuadStick (4 videos), controller plus switches (2 videos), and a single eye tracker (2 videos), with all other unique combinations appearing only once.

Over all setups depicted in the videos, 38 (51.3%) featured one device used, 23 (31.1%) featured two devices simultaneously, 11 (14.9%) featured three devices simultaneously, and 2 (2.7%) featured four devices simultaneously ($\kappa = 0.76$, "substantial agreement"). Most console videos showed one device used at a time, while most PC videos showed at least 2 devices used (Figure 3.7b). However, of the 11 videos showing 3 devices being used, 8 were on console and 3 were on PC. Further analysis showed that all 3-device console videos involved the Xbox Adaptive Controller and input devices that interact with it, through either hardware connection or software features like Xbox's Copilot [120]. One video showed four input devices used simultaneously on a PC, and all 3 videos showing mobile phone play used only one input device (touchscreen). Specific device combination data can be found in supplementary materials.

The dataset contained 25 videos which showed devices that were customized to the user in some way, including aftermarket hardware customizations, attachments like joystick extensions, or switch placement custom to the depicted user ($\kappa = 0.84$, "near perfect agreement"). Of these 25 videos, 16 used consoles and 9 used PC.

3.6.2.2 *User Position*

As described in the interviews, adaptations in device usage can fall into one of four categories: conventional usage (typical device, typical usage), adapted grip (typical device, held or used in a non-typical position), adapted device (typical device with hardware customizations), and alternate device (separate device entirely). In our dataset, 31 videos used an alternate device, 27 used an adapted grip, 14 used an adapted device, and only 2 used the conventional style ($\kappa = 0.82$, "near perfect agreement"). The console videos showed a more even spread between adapted device, adapted grip, and alternate device (12, 18, and 13 respectively, of 44 videos), while PC videos skewed toward alternate device with 18 of 27 total videos (Figure 3.7c). Over all videos, 11 videos showed users actively swapping between devices or setups mid-game. 71 of the users in the videos were sitting, and 3 were lying down.

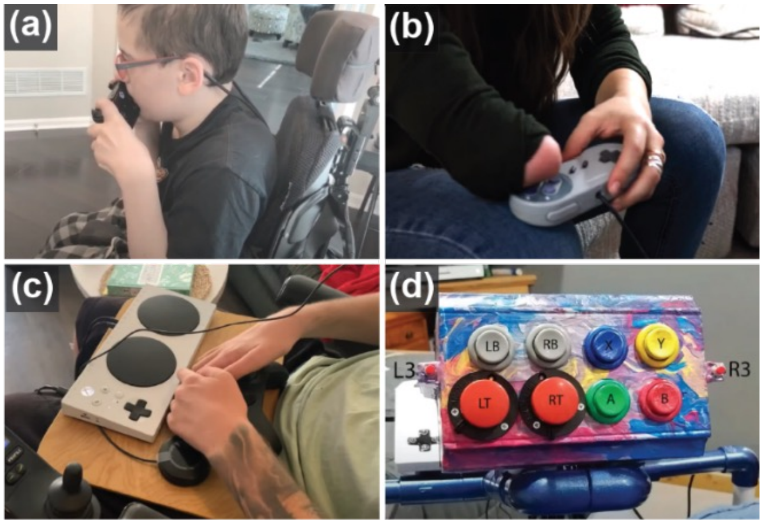


Figure 3.8: Examples of adaptations from YouTube videos: (a) a user manipulating a controller with their mouth (*adapted grip*); (b) a user manipulating a controller with their arm (*adapted grip*); (c) a controller under the palms of the user's hands with additional buttons provided by the Xbox Adaptive Controller (*adapted device*); and (d) a switch layout functioning as a completely new controller (*alternate device*). Via YouTube [1, 31, 52, 149]. Images © Gizmo XYT, ABSHOW, Zack Collie, and Charles Diaz, respectively.

Our dataset showed players using several different body parts to interact with their input devices, including fingers, palms of hands, feet, mouth, and chin ($\kappa = 0.72$, "substantial agreement"). Most console videos showed people using two body parts at the same time (e.g., chin plus palm of hand), while most PC videos showed people using one body part at a time (Figure 3.7d). Console and PC videos each had one video of 3 body parts used simultaneously, and the only video showing four body parts used simultaneously was using PC. Of the 44 console users, only 5 interacted with their devices without using hands or fingers, while 16 of 27 PC users interacted without hands or fingers. The most common non-hand interaction used the mouth (8 videos).

3.6.3 *Discussion*

Our systematic review simulated the discovery process of a person consulting YouTube for accessible device configurations and illustrates the range of input configurations that a typical user could discover. Our review found further evidence that people with mobility limitations often employ multimodal input in their gaming setups and demonstrated further just how varied accessible computing setups can be.

The interviews cited compatibility as a prominent consideration in accessible device configurations, and this effect is also evident in the Device

Setup results. Despite console-based setups appearing more often in our video dataset, their input device combinations varied less. Controllers were much more common in console videos than mouse-and-keyboard was for PC videos, suggesting that users tended to choose the typical input device for their respective platform more often on console than on PC. Consoles generally have lower compatibility with third-party input devices, and as a result, most console-based configurations used either a standard controller or the lone first-party accessible controller (Xbox Adaptive Controller). The higher compatibility of PC might be why users could use an alternate device more often. Moreover, while the PC videos tended to favor the simultaneous use of two relatively independent inputs (e.g., QuadStick and voice input), the Xbox Adaptive Controller serves as a hub for more atomic input devices like external switches or joysticks, explaining its prominence in 3-device console configurations as well as the higher representation of 3-device configurations in consoles as a whole.

The impact of device compatibility is especially clear in the User Position results. While PC videos tended to favor using a completely different device, console videos were more spread, with most users preferring to adapt their existing device or adapt their usage of it. This result suggests that console users, with less choice in devices and lower compatibility with their existing assistive inputs, remedy their accessibility issues with adapted usage styles instead of swapping out their device altogether. Figure 3.8 shows examples of these adaptations. The reduced variety of input devices for consoles could also explain the more prominent 2-body part and hand use in console players. Although console players used adapted usage styles more often, it remains true that controllers are primarily designed for use with two hands or fingers, prompting those body parts' usage. Devices that use non-hand body parts, like the mouth-controlled QuadStick or eye trackers, were more common in PC videos, explaining the higher representation of 1-body part devices in PC videos.

3.7 GENERAL DISCUSSION

Researchers and practitioners have made significant progress in making computer systems and video games more accessible to people with limited mobility. Advancements in software (e.g., VoiceAttack), hardware (e.g., Freedom Wing[5]), and multi-modal interaction techniques [22, 79, 106] have allowed people to interact with previously inaccessible systems. Although these advancements have improved access for many, people still encounter challenges with input devices and applications. Our investigation builds on previous accessibility work [4, 68, 69] to illustrate how people with limited mobility use multi-modality to overcome these challenges.

Although many of our findings were situated within the context of gaming, our findings are relevant to different applications and contexts that rely on multi-modal input. Our survey found that multi-modal input configurations

---

[5] https://ablegamers.org/charting-the-future-with-the-freedom-wing/

are a common remedy for accessibility issues, with specific setups varying widely with user ability (**RQ1**). Our interviews found that participants faced a variety of context-sensitive accessibility issues including lack of knowledge about optimal input configurations, and they remedied these issues by adapting their usage style to the device (user-device matching) or adapting their device to the application (device-application matching). Importantly, YouTube vidoes and social media were an important tool for discovering these accessible configurations (**RQ2**). Our YouTube analysis showed that these remedies and associated usage style categorizations are common in a wider real-world dataset, with prominent configuration differences depending on system compatibility as well as user and application context (**RQ3**).

We conclude by providing design recommendations informed by this investigation, discussing opportunities for researchers and practitioners to use our findings to improve the accessibility of current and emerging computing systems (extending **RQ3**), and possible limitations in our methods.

3.7.1 *Design Recommendations*

Our results provide general design guidelines for designers, developers, and researchers creating accessible multi-modal applications and games.

3.7.1.1 *Generalized Platform-Level Multi-Modality Support*

Our results show that users often remedy accessibility issues by creating multi-modal input configurations involving devices they can already use, but they vary depending on individual mobility. However, the utility of these configurations greatly depends on the ability of applications and platforms to support them. While it is true that individual developers could add support for the most common input configurations, the wide variance of combinations (as seen in Section 3.4) means that covering all combinations would be hard or even impossible. As such, we recommend that developers implement generalized input compatibility layers that consider signal domain (e.g., continuous, discrete) rather than hardware input recognition (e.g., joystick, button) in their applications. Recognizing input at a more general level allows for greater flexibility in compatible input devices, as well as greater variety of device usage contexts (e.g., adapted device, alternate device from Section 3.5.4.2).

Enhanced accessibility support often comes with a greater responsibility placed on developers, meaning that smaller development teams may place less emphasis on accessibility due to time or budget constraints. As such, we emphasize this recommendation for developers of operating systems or game platforms. We recommend that developers implement accessibility and compatibility solutions at the platform or operating system level rather than at the application level. For example, game platforms like Steam or Xbox provide some support for remapping the controls of their first-party devices and conventional input devices, the effects of which propagate to any supported application. Supporting multi-modality at a platform level allows applications to make use of multi-modal input with minimal added work for individual application developers. Wider adoption of these compatibility layers, and compatibility with more devices and input axes, can allow applications and games to support people with limited mobility more easily.

3.7.1.2 *Tools for Configuration Discovery and Sharing*

Designers and developers should consider implementing tools that allow easier discovery and implementation of user-defined input configurations. Participants in our study often use social media for discovering and sharing input configurations. However, the specific settings and adjustments needed to accurately recreate online configurations are hard to infer from social media posts alone. P6 from the interviews describes an example: *"I'm really bad at Apex Legends [a first-person shooter] because I just don't know a good way to lay out buttons in a way that works best for me"*. Game platforms like Steam have created systems for users to create and share button configurations[6], but as of yet, only for first-party controllers and conventional input devices.

One way to implement this recommendation is with the creation and use of a centralized manifest of hardware configurations, allowing users to create and share their unique input configurations on a per-application or per-device basis. This manifest could allow users to upload their device layouts and button mappings for individual applications or games (with accompanying information like usage style or additional equipment) and allow prospective users of these applications who own similar devices to more easily discover configurations that would work for them. In addition to helping end-users directly, developers who pull control schemes from such a system can make applications more responsive to individual user accessibility needs, and reduce the burden on the user to discover, create, and configure their devices individually. A common configuration discovery tool benefits application developers and end-users mutually, and can make deeper accessibility support a collective, collaborative effort.

3.7.1.3 *Consider Virtual Devices*

Our findings illustrate an input category not included within earlier device ecologies and applications: virtualized input devices. One can imagine virtualization as a "translation" of one device's input fidelity (the physical device) to another (the virtual device). As such, each combination of physical and virtual device presents different affordances and input translation fidelity. For example, consider a user with partial access to the keyboard but supplements this with a mouse-controlled virtual keyboard. Although the hardware is equivalent to traditional keyboard-and-mouse usage, usage habits and accessibility ramifications differ dramatically. Designers of cross-device applications should consider the role virtual devices play in their applications' broader user experience, and researchers in cross-device computing should consider virtual devices as a meaningful part of future device ecologies.

---

[6] `https://partner.steamgames.com/doc/features/steam_controller/browse_configs`

3.7.2 *Present Applications and Future Work*

Our work is part of a continuing effort to make applications—including games—more accessible to people with limited mobility. We discuss how practitioners can apply our current work and avenues for further exploration.

3.7.2.1 *Improving Input Customization*

As our data showed, people use a variety of input configurations, making it difficult to design control schemes that will work for everyone. Despite interview participants finding inspiration for input configurations and button mappings using social media, in the case of insufficient online search results the process of creating new configurations often involved trial and error. This situation surfaces opportunities to improve the control scheme customization experience. One approach to improve customization is to supply details about the importance and frequency of different actions (e.g., run or jump in a game). Other details, such as if a user must perform actions in rapid succession, could also be useful. With this information, users could more easily map important or common actions to controls that are comfortable and easy to access. Although our work provides insights into the challenges involved in input configuration, future work in HCI and application design should resolve these challenges through improvements to the input customization process.

3.7.2.2 *Further Understanding Device Ecologies*

Brudy et al. [22] provided an overview on the ecology of devices within a typical user space and discussed a key challenge in the area of cross-device computing: the importance of understanding the symbiosis between cross-device interfaces and human capabilities. As computing devices take new form factors and contexts, there will be a perpetual opportunity to re-evaluate user device ecologies. This relationship between users' abilities and device requirements extends to multi-modal input in a similar way. As computing devices—and thus their expected input—grow in number and variety, there are opportunities to evaluate the relationship between users' abilities and the ability assumptions made by input devices. Our findings contribute to efforts to understand users' device ecologies by describing the input configurations created by people with limited mobility.

Researchers utilize knowledge of device ecologies to develop new approaches that allow users to interact with data and applications across multiple devices [23, 80]. It is important to recognize that people with limited mobility use input configurations that researchers might not have expected or accounted for. As a result, advancements in this space might ignore the practices and preferences of people with limited mobility, which can ultimately lead to the development of inaccessible cross-device interactions. Our results give researchers useful data to consult when envisioning accessible interaction methods for different scenarios.

Our findings show that virtual devices play a critical role within the greater ecology of accessible multi-modal input. However, new input technologies such as brain-computer interfaces (BCI) could redefine the relationship between users and their devices, for example, by allowing the user to eschew physical devices entirely. It is yet to be understood how the relationship between a user and their input devices changes when minimal physical devices are involved. Moreover, the conversion of physical load to cognitive load [66, 108] for BCI could resolve accessibility issues for people with mobility limitations but introduce new ones for those with cognitive limitations. While our results provide insight into the role of virtual devices in current accessible device ecologies, future work should explore how these ecologies adapt and respond to the changing landscape of virtual devices.

3.7.2.3 *Spatial Interfaces*

Spatial interfaces, like those found in virtual reality (VR) and augmented reality (AR) systems, stretch our current understanding of multi-modal device ecologies. Previous taxonomies take into account the individual sensors involved with 6 degree-of-freedom spatial input [79], but the combination of these sensors defines a gestural language which presents unique challenges that have yet to be solved. Previous work showed that people with limited mobility encountered accessibility barriers when using these systems, which included hard-to-grasp controllers and the expectation of bodily involvement [48, 49, 92]. Interview participants were interested in VR, but as P5 described: *"it didn't feel immersive because it was clearly designed for able-bodied people. Because the movement and controls were clearly designed for people with a full range of motion, it kind of felt like being disabled all over again"*.

Understanding how users can adapt and configure their existing devices to manipulate spatial interfaces is particularly important for ensuring that the next generation of computing devices are accessible to people with limited mobility. More importantly, it can lead to more inclusive spatial interfaces where virtual representations can be customized to accurately represent users with limited mobility, and where users' input devices are incorporated as first-party input rather than remapped, adapted, or substitute options. Future work should explore how best to incorporate multi-modal input into the gestural language of spatial interfaces.

3.7.3 *Limitations*

We conducted interviews remotely due to the COVID-19 pandemic. As a result, our insights are based on what participants described instead of what we could observe directly. Although our interviews probed for as many input combinations as possible, participants could have neglected to describe prominent input combinations or priorities. Future work should include in-person studies to obtain further insights about multi-modal input setups.

Disability is a wide spectrum, so our results cannot account for all input device configurations. Although our findings highlighted several instances of how people with limited mobility use multi-modal input, there will always be perspectives that our investigation did not represent. Accessibility considerations evolve as technology advances, and as such, we present this work as the first of several future iterative investigations.

We ground our investigation within the context of video games due to its common need for multi-modal input and previous work establishing gaming as an appropriate and generalizable venue for accessibility research [106, 148]. As a result, many of our interview participants and survey respondents were members from accessible gaming communities, which resulted in several insights that were particularly focused on gaming. Although we believe our methods and findings are generalizable to different contexts that might require multi-modal input, there might be other challenges and practices for contexts that our participants have not experienced. Further work could examine people's multi-modal preferences and practices for a broader range of contexts.

As digital devices further embed themselves into every facet of modern life, interactions with computer systems will continue to grow in both number and complexity. As a result, multi-modal computing is becoming increasingly common as both a field of study and a category of everyday computer input. Without a specific focus on accessibility, and considerations for the ability assumptions embedded in multi-modal device configurations, the constant evolution of technology may leave people with limited mobility behind. While earlier works examined conceptual frameworks or individual multi-modal input techniques for the general population, little work has investigated the everyday use of multi-modality by people with limited mobility. We examined how people with limited mobility used multi-modal input as a first step toward empowering developers and designers to support accessible multi-modality more holistically.

Accessibility is, at its core, a relational and human effort. As P5 describes, *"when you're engaging with someone that might need accessibility features, there really does need to be a relational aspect that says, 'you are invited into this space"'*. As such, understanding the considerations, obstacles, and adaptations associated with accessible multi-modality is pivotal to creating inclusive input experiences that invite all users to take part. Though more work will always remain as technology evolves, we provide this investigation as one more resource to help make multi-modal computing more accessible to everyone.

4

SWITCHSPACE: CONTEXT-AWARE PEEKING BETWEEN VR AND DESKTOP

Designing for accessibility also includes designing for *situational impairments*, temporary reductions in a user's abilities based on situational factors like seating position, input devices being used, or current environment. An adaptable accessibility technique should also consider situational impairments, so we expand our investigation to deeply understand one such example.

4.1 INTRODUCTION

Switching between desktop and VR interfaces can be cumbersome. Switching from desktop to VR, for example, involves picking up VR controllers, re-adjusting headset fit, and re-orienting in the virtual environment for every switch. Likewise, switching from VR to desktop involves placing controllers down, placing the headset down, and grasping the mouse. These device transitions can take time, and impose a physical and mental context switch which can disrupt the user's focus and workflow. Switching between desktop and VR is already becoming more common and necessary as VR gains popularity for consumers and content creators.

Solutions for 2D-3D context switching typically involve secondary viewing devices integrated into the VR scene, like presenting 2D desktop interfaces within VR [33] or an articulated 2D display as a viewing window [28, 126]. Previous work also explores "cross-reality blending" approaches to unify the user's real and virtual environments, such as presenting real objects in VR [60, 84, 116, 154] or exiting VR more fluidly by fading in the real environment [72]. These systems address awareness issues with cross-device transitions, but offer little exploration of the general design space of cross-reality interfaces or the habits and preferences of users with workflows across both desktop and VR.

Our work contributes to the wider discussion of transitional interfaces by investigating context-aware transitions between VR and desktop. This work has two broad objectives, which we phrase as research questions:

(**RQ1**) *What are the challenges and preferences of VR users and content creators, when completing tasks which may require using both desktop and VR?*

(**RQ2**) *Can we use these findings to develop an effective and preferable alternative to fully transitioning between interfaces?*

A formative study of VR users and content creators found that they often need to transition between desktop and VR, but find it disorienting or tedious. These transitions often address a specific task requirement like differing functionality or input precision, or a physical issue such as fatigue. Users

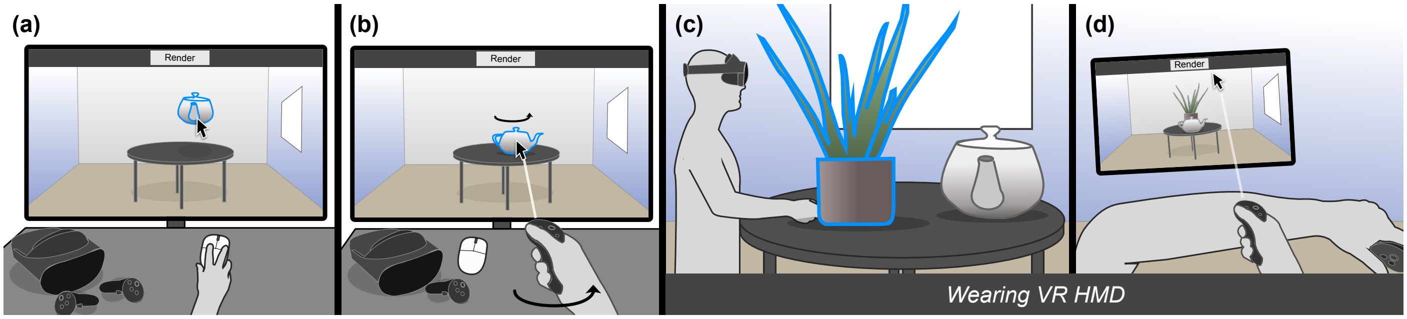


Figure 4.1: An envisioned desktop/VR 3D environment rendering application using "peeking" techniques. (a) The desktop interface, which supports interaction from the mouse and the VR controllers. The user brings the teapot into the scene using the mouse (*Desktop*). (b) The user switches to the VR controller to translate and rotate the teapot in 3D (*Desktop-to-VR input* peek). (c) The user enters VR to place the houseplant, for a better sense of scale and spacing (*VR*). (d) Instead of switching to desktop, the user summons an interactive body-mounted desktop view to save the final render (*VR-to-Desktop viewing* peek).

reported that these transitions were typically temporary, and motivated by current usage context.

To address this, we explore the idea of a "peek" between these interfaces as a temporary transition between a *primary* interface and a *secondary* interface. For example, consumer VR platforms like SteamVR and Oculus allow users to peek from VR (*primary*) to their desktop (*secondary*) by quickly activating an in-VR desktop view, interacting with it, then dismissing it. We extend this idea of contextual peeking into *SwitchSpace*, a design space for cross-reality peeking which encompasses temporary changes in both input device (mouse or VR controllers) and viewing device (desktop display or VR HMD). *SwitchSpace* is designed as a state machine which activates different peeking techniques depending on the user's current and past states. Peeking techniques using this state machine allow the user to quickly view and provide input across modalities, without having to change between input or viewing devices (Figure 4.1).

We implemented a collection of peeking techniques within our design space, then evaluated them in a controlled cross-reality task. Participants solved math problems in both VR and desktop, with each problem having a missing number in the opposite interface, prompting a peek or full transition. Peeking techniques made this cross-reality task 38% faster than a full transition, at the same time alleviating accuracy differences between the mouse and VR controllers. The ability to use peeking techniques reduced perceived workload across several NASA-TLX categories.

We make three contributions: (1) a formative study of real-world cross-reality tasks encountered by VR users; (2) a design space of VR and desktop peeking techniques for cross-reality tasks; and (3) design recommendations for cross-reality workflows motivated by the results of a user study.

4.2 BACKGROUND AND RELATED WORK

Chapter 2 discusses previous work in cross-device computing including real-virtual alignment and interaction-space conversions. We provide a more in-depth discussion specifically centred around *cross-reality interactions*, those bridging the gap between a spatial interface like VR and a non-spatial interface like traditional desktop computing.

Cross-reality interaction techniques are part of general cross-device computing, which has an extensive history within HCI. We discuss a more focused set of topics, but recommend the work of Brudy et al. [22] and Marriott et al. [81] for a wider review. We discuss specific related work for cross-reality interfaces, focusing on transitional interfaces.

4.2.1 *2D-3D Transitional Interfaces*

A major component of our work is the transitional interface between desktop and VR. Previous work has explored transitional interfaces, and provides initial evaluations.

Millette et al. [87] explored combining desktop and AR computer-aided design systems, with the AR interface controlled by hand gestures or a smartphone. An informal evaluation found context-switching helpful, but minimal visual feedback made transitioning between the smartphone and AR difficult. Similarly, Serrano et al. [114] evaluated using head gaze input in a headset-mounted AR system to transfer content and usage context across multiple devices. Participants found the system engaging but found having too many head-mounted elements distracting when primarily dealing with 2D interfaces. Grubert et al. [57] evaluated cross-display interaction techniques using AR widgets distributed over multiple devices, showing that a combination of AR and smartwatch interfaces can outperform single-device interactions. Bogdan et al. [18] evaluated transitions between 2D mouse and 3D freehand input in a 3DTV desktop interface, finding that a hybrid model enabling both 2D and 3D input was fastest. They describe several input triggers (e.g. mouse movements, freehand movements) and context triggers (e.g. camera rotation) to activate transitions automatically.

User context can provide an additional stream of input to a system. Lu et al. [78] evaluated several techniques for AR UI panels to respond to user context, finding that when users are moving around an area, UI should be as low-friction as possible. Similarly, Fender and Müller [43] evaluated state transitions for spatial augmented reality UIs based on user and object positions, allowing for the definition of states and responses ad-hoc based on their context.

Previous work also provides design guidelines for our focus on transitioning into and out of VR. Schröder et al. [112] provided several analytical lenses for transitional interfaces between desktop, tablet-based AR, and VR, and analyzed counts and frequencies of transitions as a way to characterize the use of multi-user transitional interfaces. Knibbe et al. [72] evaluated several

Table 4.1: Counts of answers from the formative study. All answers were optional, so we also show counts of non-answers.

Question	Response with Count (24 total respondents)					
Gender	*Man*	*Woman*	*Non-binary*	*Did not answer*		
	9	2	1	12		
VR Usage Time	*3 or more years*	*2 years*	*1 year or less*	*Did not answer*		
"How long have you been using VR?"	11	8	4	1		
VR Usage Per Week **"How many hours per week do you use VR?"**	*5 or fewer hours*	*6-10 hours*	*11-20 hours*	*21-30 hours*	*31-40 hours*	*41+ hours*
M = 13.2, SD = 13.86	8	7	3	3	2	1
Primary Use Case	*Recreation*	*Work*				
"For what purpose do you primarily use VR?"	17	7				
Work Usage **"If using VR for work, what are you doing with it?"**	*Interactive prototyping*	*Creating VR content*	*3D design applications*			
Of the 7 work users.	5	4	4			

remedies for disorientation when leaving VR, suggesting that systems fade in elements of the real world to make transitioning between VR and non-VR easier. George et al. [47] evaluated multiple methods for transitioning between VR and the real world, compared to standard HMD passthrough. Using a visual search task, they found that using a user-triggered AR view as an intermediate step between VR and the real world was preferable for interacting with the real world from VR without losing presence. Similarly, Grasset et al. [56] explored user transitions between AR and VR views of the same scene, for an environment exploration and search task. Transitioning from AR to VR caused disorientation, leading the authors to recommend visual aids like previews or aligned visual elements.

Pointecker et al. [102] explored four additional visual techniques for transitioning between AR and VR, finding that quickly fading between realities was preferable when switching frequently. Likewise, McGill et al. [84] found that participants preferred reality-blending techniques for interacting with real-world objects from VR instead of having to feel around or fully remove the headset.

Carvalho et al. [28] evaluated a transitional interface using three combinations of input and output techniques: desktop UI using a mouse and keyboard; stereoscopic monitor using a Wiimote for direct interaction; and a CAVE VR system using a Wiimote for raycasting. They categorize transitions based on 3 continuity properties: *perceptual* (output device), *functional* (input device), and *cognitive* (data representation). Users could transition between these combinations by explicitly choosing an option to transition. After an exploratory evaluation, the authors suggest providing additional visual feedback for transitions, and maintaining input consistency between states.

Transitional interfaces bridge the gap between disparate input and output mechanisms. George et al. [47] explore transitions between VR and reality, but do not focus on transitions between input devices or the role of context within

Table 4.2: Open-ended questions in the formative study.

1. *"Do you use applications that support both VR and non-VR usage? If so, which?"*
2. *"Why would you choose using those applications in Non-VR instead of in VR?"*
3. *"Why would you choose using those applications in VR instead of Non-VR?"*
4. *"How often do you switch between VR and Non-VR modes for these applications, in one usage session?"*
5. *"Please list any other notable experiences you have regarding using VR for extended amounts of time."*

their design space. Serrano et al. [114] provide guidelines for transitioning input devices for AR, while McGill et al. [84] and Knibbe et al. [72] provide recommendations for blending VR and reality, but little work bridges the gap between desktop and VR input and output. Moreover, while Carvalho et al. [28] provide a theoretical framework for categorizing input and output transitions, they do not explore the role of context and instead opt for manually activating transitions.

Previous work exploring transitional interfaces does not explore the interface-transitioning habits of people who use VR and desktop applications in everyday scenarios. There remains a gap in understanding the current user preferences, challenges, and objective impacts involved with real-world cross-reality tasks.

4.2.2 *Summary*

Designing generalizable cross-reality interactions requires understanding the relationship between the user's environment (VR or the real world), hardware, interaction techniques, and context. Prior work in blending real and virtual environments typically use static or manual invocations of blending techniques, instead of responding to device or context changes. Moreover, their analyses place little emphasis on the usability impact of transitioning devices or input techniques. Work exploring 2D interaction techniques within 3D environments typically focuses on blending input methods instead of transitioning between them. Prior work in transitional interfaces explores techniques for moving between VR, AR, and reality, but place little emphasis on input tasks that require the use of both VR and desktop interfaces simultaneously. Describing the real-world challenges and effects of cross-reality workflows is a critical early step in developing more effective and comfortable alternatives. As such, for a full understanding, we conduct a two-part investigation. First, we will describe the usability challenges faced by real-world VR users with cross-reality workflows. Second, we will use these insights to to develop and evaluate a system that can remedy these challenges.

4.3 FORMATIVE STUDY

We conducted a formative study with 24 frequent VR users (demographic details in Table 4.1) to better understand transitioning between VR and desktop in everyday situations (**RQ1**). Respondents ranged from 18 to 47 years old with a median age of 29. We recruited respondents via word-of-mouth and public online postings in US-based online forums. The survey took approximately 20 minutes, and respondents were entered into a draw for two $50 USD prizes. After the demographics and usage time questions, we asked participants open-ended questions (Table 4.2) about their preferences and habits between VR and desktop use.

4.3.1 *Results*

Respondents reported several factors impacting their choice of platform in cross-reality workflows.

4.3.1.1 *Choosing Desktop Over VR*

Respondents preferred desktop applications over VR applications for reduced discomfort, more convenience, reduced fatigue, and increased input precision. When asked why they would choose using cross-reality applications on desktop instead of VR, respondents noted that VR is often uncomfortable: *"Headsets need to improve and not hurt our eyes, face, etc. Current VR is medieval"* [P15].

Convenience was a contributing factor, with desktop interfaces perceived as *"less of a hassle"* [P11], with fewer *"hardware issues"* [P10] and *"more straightforward and cost less time to set up"* [P19]. Respondents preferred desktop when they need *"to jump in quickly to fix something. [It] may not warrant need to put on [the] headset if session time is limited or short"* [P21]. Similarly, respondents considered desktop to have a higher *"ability to multitask"* [P3]. Respondents valued the *"ease of multitasking while watching a movie or show"* [P9], or using desktop when *"I need to work at the same time"* [P21].

Fatigue was also a contributing factor. Respondents valued using desktop when they *"don't want to be as physical, or want to be more efficient"* [P12]. P14 agreed: *"[I get] fatigue in my arms, holding controller up in-front of me"*. Respondents considered the physical implications of using VR: *"When I'm sick I play in desktop and avoid VR entirely"* [P21]; *"I also have physical disabilities which make VR more taxing for me"* [P8].

Participants also preferred desktop for increased input precision. Desktop was preferred for *"gaming and development"* [P22], and P14 used *"precision sketch input for Gravity Sketch[1], better on [a drawing] tablet than in VR"*.

---

[1] https://www.gravitysketch.com/

4.3.1.2 *Choosing VR Over Desktop*

Respondents preferred using VR over desktop for higher immersion and better visualization. When asked why they would choose to use cross-reality applications in VR instead of desktop, respondents cited *"immersion"* [9 respondents total] and *"fun"* [P1, P11] as major factors in choosing VR. P20 noted the effect of being immersed in a comfortable VR environment: *"I was comfortable enough to actually fall asleep in a social game while hanging out with friends late one night"*.

Respondents preferred VR in situations that need spatial understanding. Respondents preferred VR for the *"better sense of scale, look, and feel of 3D designs"* [P14]. Similarly, VR offered *"better visualization compared to solid model and costs less time to build"* [P19]. Respondents valued VR for *"more flexibility of game actions"* [P10].

Respondents were excited about future 3D design applications: *"In the past [I] mostly [used VR] for review, annotations, communication, and better understanding. But I recently also got pretty excited about subdivision modelling in VR (see Gravity Sketch), promising for 3D content creation"* [P14].

4.3.1.3 *Transitioning Between VR and Desktop*

Finally, we asked respondents about their experience transitioning between desktop and VR interfaces within a single session. 5 of 7 respondents who use VR for work mentioned the need to transition between VR and desktop in their occupation workflow, often changing settings in desktop and viewing the effects in VR. P22 wrote: *"development takes lots of iterating between headset and VR"*. Transitioning is *"part of the development of the app (fixes, changes, etc.)"* [P2], or to *"check the computer system"* [P8]. 3D design and development applications also needed transitions, for example, *"In VRED [a 3D design and visualization application] I [transition] if I need to tune the scene or the visualization parameters, so it's a constant VR/non-VR change"* [P14].

Respondents who use VR for recreation reported a lesser need to *fully* transition within a single session, often choosing to *"stay in VR or [desktop] for a given session"* [P9]. However, rather than fully transitioning (taking the headset entirely off, putting down the controllers, grabbing the mouse), respondents mentioned using a feature in SteamVR or Oculus which enables an interactive view of their PC desktop in VR, often to check messages [P5, P13].

Some respondents leaving VR noted an experience similar to participants in the user study by Knibbe et al. [72], noting *"disassociation with the real world. Taking the headset off after a long session, it takes my brain a second or two to remember that I'm in my home. This happened much more frequently when I was new to VR."* [P9]. P13 agrees, having previously experienced *"visual bleeding - after a long [VR] session when I first started the real world felt unreal"*.

4.3.2 *Discussion (RQ1)*

Our first research objective (**RQ1**) was to understand the challenges, workarounds, and preferences of real-world users with cross-reality workflows. Our results validate our assumed trade-offs between VR and desktop with real-world users, as well as align with previous work in 2D-3D cross-device workflows [72, 84, 153]. The *Transitioning* feedback provides additional design considerations for improving VR-desktop cross-device interactions.

4.3.2.1 *Cross-Reality Transitions Are Temporary*

Respondents with work-related cross-reality workflows noted the need to adjust settings on the desktop then briefly view the changes in VR [e.g. P14, P22]. Likewise, respondents using VR for recreation often used an in-VR panel to temporarily view their desktop to change games or respond to messages. In these use cases, transitions between VR and desktop are rarely permanent. We see these momentary transitions as a desire to *peek* between VR and desktop.

Viewing and interacting with the PC desktop from VR is functionally an alternative to removing the headset, placing the controllers down, and transitioning to a mouse and keyboard. We can consider an in-VR view of the desktop one example of a *peeking technique*: a means to bridge the gap between VR and desktop without incurring the discomfort or cognitive demand of fully transitioning.

4.3.2.2 *Decouple Input From Output*

In VR development, quickly checking visual changes in the virtual environment may not require the use of input devices. Likewise, checking a button mapping on a VR controller may not require the use of the headset. Respondents preferred VR for its increased immersion and visual fidelity, despite lower perceived accuracy. Likewise, respondents discussed preferring desktop for its greater accuracy and comfort, despite lower immersion. A design space that separates input and output peeking techniques can make descriptions more granular and help find a compromise between conflicting design priorities like visual fidelity and input accuracy.

4.4 SWITCHSPACE DESIGN SPACE

The formative study found that most cross-reality workflows involve short, temporary movements between VR and desktop, decoupling input from output. To contextualize our findings and guide future implementations, we describe a design space of techniques which increase usability by supporting these temporary “peeks” explicitly, in addition to full transitions. This design space’s underlying state machine enables context-awareness by decoupling input and output transitions. We describe the state machine and our implementation of several examples of peeking techniques.

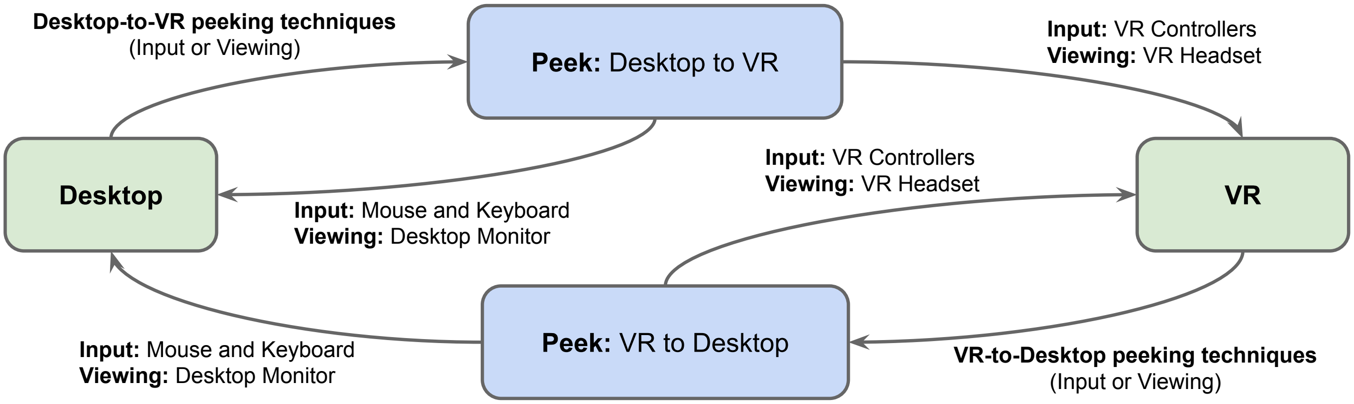


Figure 4.2: The state machine used in our description of a context-aware cross-reality interface, between primary states *Desktop* and *VR* (green). Changes in input or viewing device trigger state transitions. Changing from the standard input devices or the standard viewing device for a primary state brings the state machine to a secondary"peek" state (blue).

4.4.1 *Classifying Context*

We derive our description of context from three categories of formative study feedback: *input*, *viewing*, and *memory*.

The *input* context addresses the match between the fidelity of the user's current input device and the demands of the task. Survey respondents preferred VR and desktop input techniques for different tasks (desktop for precision, VR for 3D manipulation). Mismatches between device and task fidelity can cause usability issues within a system [139]. We consider *input* peeking techniques as techniques that address usability issues in *interacting* with content across interfaces. Our implementation triggers *input* changes by detecting mouse movements more than 1 mm or controller movements more than 5 mm in the last second, or the activation of any *input* peeking technique.

The *viewing* context addresses the match between the fidelity of the display and the task. Survey respondents preferred the VR headset for greater levels of immersion. Likewise, they preferred desktop displays when completing multiple tasks at once (e.g., working while watching a movie [P9]) or when VR headsets were physically uncomfortable. We consider *viewing* peeking techniques as techniques that address usability issues in *viewing* content across interfaces. Our implementation triggers *viewing* changes using the headset proximity sensor (to detect headset wear status) or the activation of any *viewing* peeking technique.

Peeks between interfaces are temporary. We identify the importance of *memory* to encapsulate how cross-reality workflows involve moving from a primary to a secondary interface, then returning. Including *memory* as part of a description of context allows the same combination of input and viewing devices to function differently based on the current primary and secondary interfaces, enabling a greater variety of potential designs.

4.4.2 *Context State Machine*

We represent the design space as a collection of primary and secondary states (Figure 4.2), with changes in viewing and input technique serving as transitions between them. Starting from standard configurations of input devices (mouse and keyboard for *Desktop*, controllers for *VR*) and viewing devices (desktop monitor, VR headset) for a given primary state, any change in input or viewing technique triggers a transition to a secondary "peek" state. The user can fully transition by adopting the standard input and viewing devices of the other primary state. A system-level understanding of the user's previous states enables the use of multiple peeking techniques for the same combination of input and viewing devices, and allows the system to discern between techniques based on context. Our implementation separates peeking techniques based on the direction of the associated state transition, either *VR-to-Desktop* or *Desktop-to-VR*.

Starting in *VR*, activating input or viewing peeking techniques transitions the user into a *VR-to-Desktop* peek state. The user can return to headset and controllers to re-enter the *VR* state, or fully transition to *Desktop* by removing the headset and moving the mouse. Likewise, from *Desktop*, the user can activate input and viewing peeking techniques to enter the *Desktop-to-VR* state, with a full transition to *VR* triggered by donning the headset and grabbing the controllers.

4.4.3 *Peeking Techniques*

Our design space supports many possible techniques for peeking between desktop and VR. Formative study participants reported the need to complete simple cross-reality pointing and selection tasks, like quickly summoning and interacting with the desktop view from VR. As such, we implemented a representative selection of techniques appropriate for cross-reality pointing and selection. We implemented these techniques in Unity3D with the SteamVR SDK and Vive Input Utility. We describe our techniques by direction (*Desktop* to *VR*, *VR* to *Desktop*) and category (*viewing*, *input*).

4.4.3.1 *Desktop to VR*

Peeking techniques from *Desktop* to *VR* enable the user to complete tasks in *VR* without needing to fully transition to the VR HMD and controllers.

Viewing peeking techniques from *Desktop* to *VR* involve using a simulated HMD view from the desktop, or briefly donning the HMD without controllers. From the desktop, the user can use the keyboard or an onscreen button to activate a simulated HMD view (Figure 4.3), which can be moved at multiple levels of fidelity: simple indirect translation using the WASD keys or arrow keys; directly-manipulated translation with the mouse by holding Space; or directly-manipulated rotation with the mouse by holding Shift. Without donning the HMD, the user can use a VR controller instead of the mouse to do those same camera movements in 3D. The simulated HMD view shows

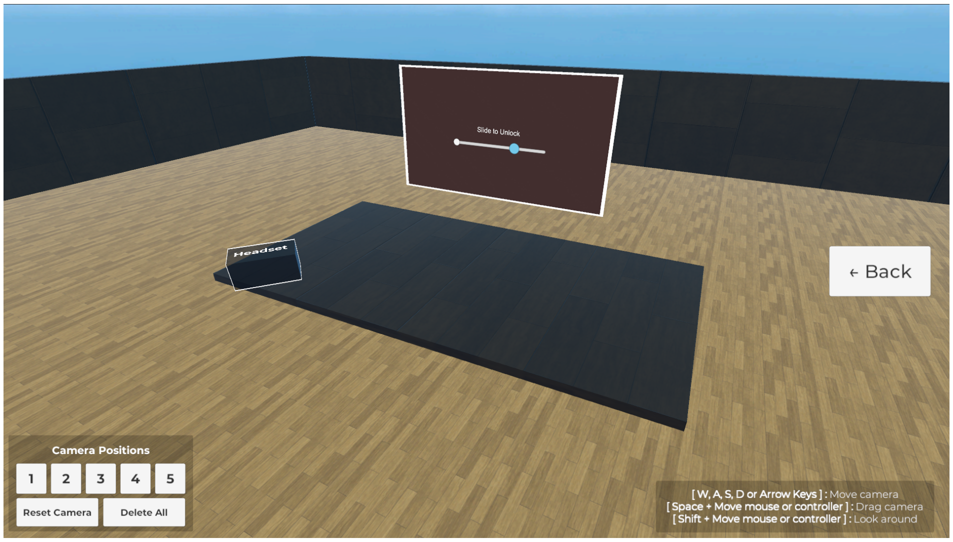


Figure 4.3: The simulated HMD view, for peeking from *Desktop* to *VR*. The real desk and HMD positions appear in the environment. The user can interact with objects and UI in the scene equivalently to VR controllers by using the mouse or by pointing the controller at the physical monitor. The user can save up to 5 camera angles for more convenient navigation (bottom left).

the position of the real HMD, and the user can save up to 5 camera angles using onscreen buttons. For added visual fidelity, the user can don the HMD without grabbing VR controllers. Two objects will appear in the HMD's view: a rectangular marker showing the position of the simulated HMD; and a cursor anchored to their view (at the depth of any hovered object) which is movable with the mouse and constrained to within the HMD's lenses.

Input peeking techniques from *Desktop* to *VR* involve using the mouse to interact as if it were a VR controller, or the VR controller to interact as if it were the mouse. While wearing the HMD or in the simulated HMD view, the mouse cursor acts as a 2 degree-of-freedom targeting mechanism for a virtual controller raycast, with left-click equivalent to the controller trigger. To the user, this functions like a standard desktop mouse but with the ability to interact in the virtual environment like a controller. Still without donning the HMD, the user can point a VR controller at the real monitor to interact via the simulated HMD view, raycasting through the real monitor into the environment (Figure 4.4). The virtual controller's ray intersects with a collision plane aligned with the real monitor, at the coplanar 2D point P_c (Figure 4.4a). That same point, but coplanar to the simulated HMD screen plane, is P_h. The simulated HMD casts a ray from its position, through P_h, into the virtual environment, returning a final collision point P_w (Figure 4.4b). The cursor is placed at P_w, and scaled to maintain visual angular size (Figure 4.4c).

If the user dons the HMD *and* picks up the controllers, they transition fully into *VR*.

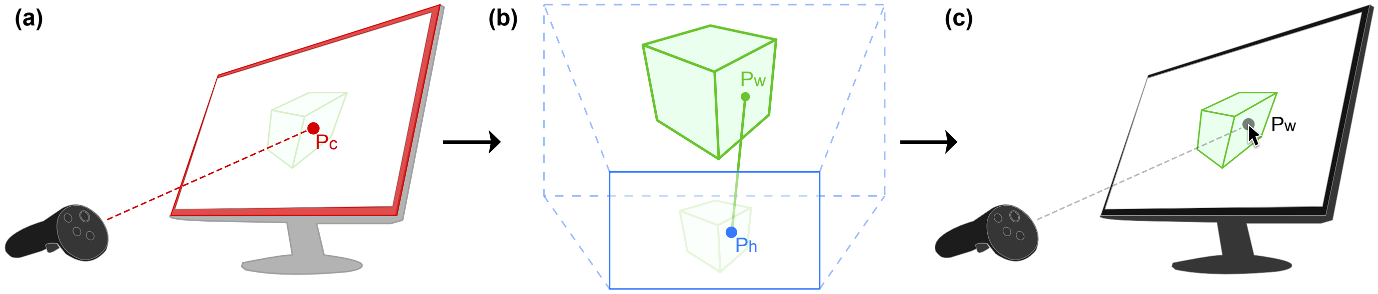


Figure 4.4: Our approach for using the VR controller to point through the real monitor, for *Desktop*-to-*VR* peeking. (a) In the virtual environment, the controller's raycast intersects a collision plane aligned with the real monitor, at point P_c. (b) P_c is converted to a screen-space point P_h, and a second raycast from the simulated HMD camera through P_h returns the world-space coordinate P_w. (c) The cursor appears at P_w, appearing to the user where the controller is physically pointing.

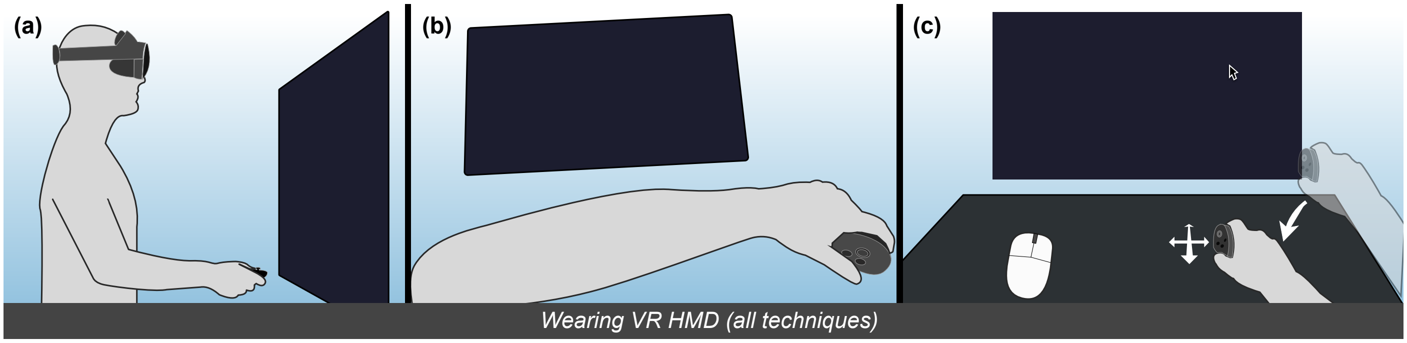


Figure 4.5: Interactive views of the desktop UI, for peeking from *VR* to *Desktop*: (a) a world-anchored view; (b) a body-anchored view; and (c) a view on the desk aligned with the real monitor, summoned by the mouse or a mouse-like movement of the *VR* controller.

4.4.3.2 *VR to Desktop*

Similarly, peeking techniques from VR to desktop allow the user to complete tasks on the desktop without needing to fully transition to the mouse or real monitor.

Viewing peeking techniques from *VR* to *Desktop* allow the user to quickly and temporarily view the monitor from *VR*. If the user is away from the desk, they can view the *Desktop* UI in two ways. First, they can summon an interactive world-anchored panel in front of them using a controller button (Figure 4.5a), similar to an existing Oculus and SteamVR feature. Alternatively, the user can raise and turn their left arm (similar to checking one's watch) to summon a smaller body-anchored panel on their arm (Figure 4.5b). If the user is at the desk, moving the cursor will summon a virtual monitor showing the desktop view, aligned with the real monitor (Figure 4.5c). If the user removes the HMD while away from the desk, UI elements will scale up for easier viewing at a distance.

Input peeking techniques from *VR* to *Desktop* involve using the controllers to move the desktop cursor. The user can manipulate the VR controllers to raycast to any active desktop view to place the cursor. The VR environment contains a real-virtual aligned representation of the desk (see Section 4.4.3.3). If at the desk, the user can also move the cursor by turning their controller sideways and sliding it on the desk like a mouse (Figure 4.5c). The user can also put down one or both controllers and use the mouse to move the cursor. Moving the mouse or sliding the controller on the desk will summon the virtual monitor, aligned with the real monitor.

Removing the HMD *and* moving the mouse will trigger a full transition back to *Desktop*.

4.4.3.3 *Real-Virtual Alignment and Calibration*

In addition to real-virtual alignment increasing spatial awareness and decreasing discomfort when transitioning [60, 72], some peeking techniques require calibration of the real monitor and desk positions. Before using the system, the user must first calibrate the positions of the real desk and monitor by touching the front of the VR controller to the front of the desk, then to the center of the real monitor.

4.5 EXPERIMENT

The formative study demonstrated that real-world cross-reality workflows are uncomfortable and disruptive (**RQ1**) and motivated a design space of temporary, context-dependent transitions in both input and output. To quantify the effect of these transitions, and answer **RQ2** by evaluating the effectiveness of our implementation, we conducted an experiment using a "math problem" task that captures the essence of a cross-reality workflow. This compact and conceptually simple task is important for internal validity since it controls how and when participants need to transition between *Desktop* and *VR*. It is designed to be simple, while still prompting transitions that match those in the design space. We use it to systematically evaluate transitions across only viewing devices, as well as transitions across both input and viewing devices using an unlock subtask. Testing a complex real-world application, like 3D modelling, would make it difficult to control experiment factors, harder to train participants, and more challenging to systematically gather quantitative data within the time constraints of an experiment session. We discuss study designs with specific applications in Section 4.6.3.

4.5.1 *Participants*

We recruited 16 participants (ages 24 to 35, 9 men, 7 women, 0 non-binary, 13 right-handed, 3 left-handed) by word-of-mouth, and each received $15 remuneration for completing the roughly 45-minute session. This study took place in Canada, so participants were paid in Canadian dollars. 7 participants had at least moderate experience with 3D content creation applications like Blender or Unity. 14 had at least moderate experience navigating in 3D, like

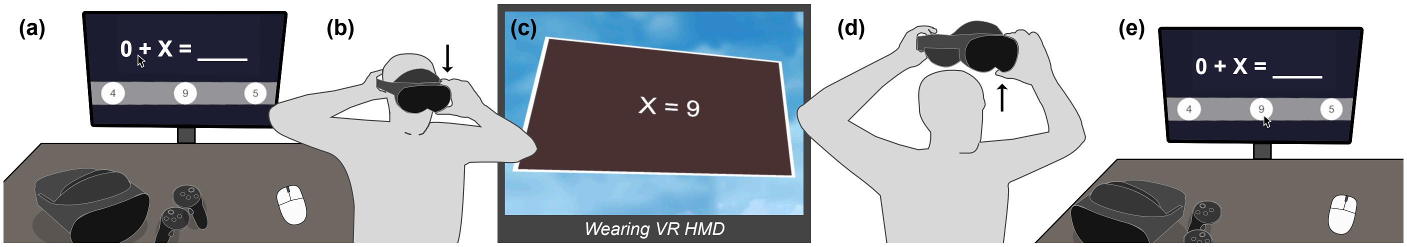


Figure 4.6: The view-only peek variation of the study task, in the baseline technique, and the desktop-vr direction. (a) The participant views the math problem on the desktop. (b) They don the headset. (c) They see the missing variable in *VR*. (d) They remove the headset. (e) They answer the math problem on the desktop.

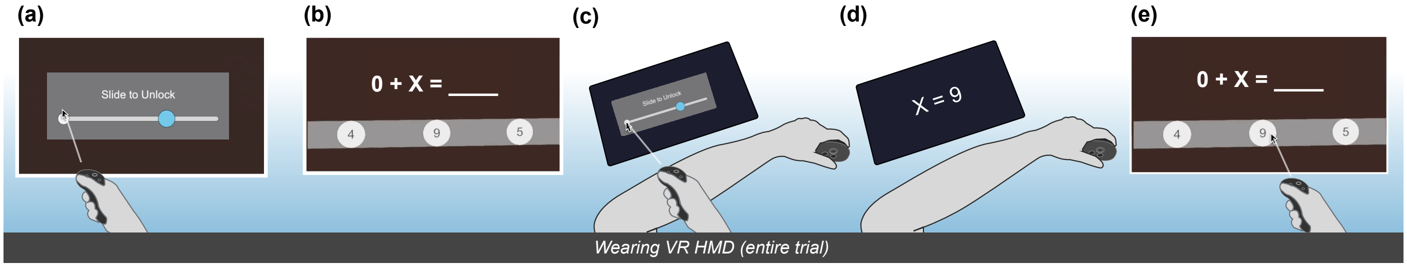


Figure 4.7: The input+view peek variation of the study task, in the switchspace technique, and the vr-desktop direction. (a) The participant unlocks the VR panel by completing the unlock subtask. (b) They see the math problem in *VR*. (c) They use a VR-to-Desktop peeking technique (this shows one of many options) to unlock the desktop panel using the controllers. (d) They see the missing variable on the desktop view. (e) They answer the math problem in *VR*.

in video games. 11 had at least moderate experience with VR. The experiment was approved by our organization's ethics review board.

4.5.2 *Apparatus*

Our implementation used a Meta Quest Pro HMD connected to a PC, which was powered by an Intel Core i7-9700k CPU and a NVIDIA RTX 3080 GPU. We calibrated the location of the monitor and the desk before each experiment session.

4.5.3 *Procedure*

Each participant completed a demographics questionnaire, then completed a system tutorial to view and try all peeking techniques, using a practice version of the task.

4.5.3.1 *Math Task*

Participants were presented with a multiple-choice addition question with a missing variable X (Figure 4.6). The math question, the value of X, and the correct multiple-choice answer were decided randomly every trial. All

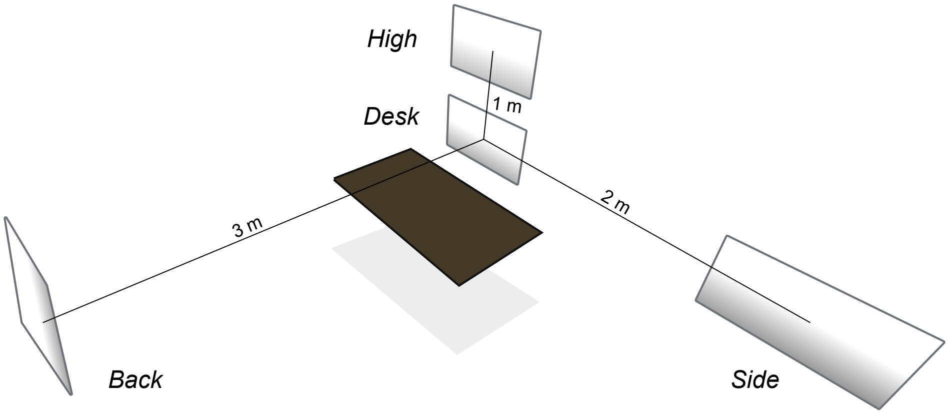


Figure 4.8: The positions of the task panels in the virtual environment: DESK, HIGH, SIDE, and BACK.

components of the math question were restricted to single digits, making the final answer 18 or lower. The math question appears in one interface (*Desktop* or *VR*) based on condition, and *X* appears in the other (*VR* or *Desktop*, respectively). Participants would see the math question in the first interface, peek or fully transition into the second interface to find *X*, then return to the first interface to select an answer. While the participant could answer the question using any available technique, the baseline condition removed all peeking techniques to prompt a full transition.

Participants had to begin a trial with the standard input and viewing devices for the current condition's starting state. For example, in conditions that required transitioning from *VR* to *Desktop*, the trial's UI would be disabled until the participant donned the HMD and grabbed the VR controllers. The trial started when the participant donned the appropriate starting input and viewing devices.

4.5.3.2 *Unlock Subtask*

In half of the trials, math questions and answers were hidden behind an initial unlock subtask (Figure 4.7), requiring that participants drag a handle from a starting position, and release on top of a blue target. This could be completed using any available input technique. We included this variation to evaluate transitioning in both input and viewing devices as opposed to only viewing.

4.5.3.3 *Task Placement*

In VR, UI panels (for either math questions or answers) were placed in one of four positions (Figure 4.8): *Desk*, at the desk, aligned with the physical monitor; *High*, 1 m above the center of the physical monitor, encouraging participants to stand; *Side*, 2 m to the right of the physical monitor, encouraging participants to move in the space; and *Back*, 3 m behind the participant when facing the physical monitor, encouraging turning around.

Table 4.3: Open-ended questions in the post-questionnaire.

1. *"Of the peeking techniques shown (viewing the VR scene from desktop, viewing the desktop display from VR, etc.), which did you prefer to use and why?"*
2. *"Did you prefer cases where you had to fully switch, or could peek between VR/Desktop?"*
3. *"Which aspects of the study felt comfortable?"*
4. *"Which aspects of the study felt uncomfortable? When did you experience the discomfort?"*
5. *"If applicable, did any techniques in the study remedy the discomfort?"*
6. *"Which felt more difficult: switching from Desktop to VR, or from VR to desktop? Why?"*
7. *"Do you have any other thoughts to share regarding these cross-modality (VR and Desktop switching) interfaces?"*

4.5.3.4 *NASA-TLX and Post-Questionnaire*

After each experiment condition, participants answered the first half of the NASA-TLX questionnaire [59] for perceived workload. At the end of the experiment, participants completed a post-questionnaire (Table 4.3) for feedback about peeking techniques, their preferences, and overall experiences.

4.5.4 *Design*

4.5.4.1 *Independent Variables*

This is a within-subject design, with two primary independent variables: TECHNIQUE with two levels (BASELINE, SWITCHSPACE) and DIRECTION with two levels (DESKTOP-VR, VR-DESKTOP). There are two secondary independent variables: POSITION with four levels (DESK, HIGH, SIDE, BACK), and PEEK with two levels (VIEW-ONLY, INPUT+VIEW). The order of DIRECTION was counterbalanced using a Latin square, as was the order of TECHNIQUE within each DIRECTION. This ordering allows participants to complete both levels of TECHNIQUE back-to-back for each DIRECTION, reducing fatigue and enabling more direct subjective comparisons between TECHNIQUE levels.

4.5.4.2 *Dependent Variables*

Dependent measures are computed from logs. *Time* is computed as the time from entering the correct state to start the trial, to correctly answering the math question. For example, trials in the VR-DESKTOP condition would start when the participant fully transitioned to *VR*, and would end upon selecting the correct answer. *Drag Error* is the average distance of the cursor from the nearest point on the unlock task's straight path while dragging the handle. This 2D Euclidean distance is calculated along an infinite hit-plane coplanar to the unlock subtask, meaning that the participant can point past the task canvas (i.e., missing the task entirely) without affecting *Drag Error* calculation. We normalize *Drag Error* to be a percentage of the unlock subtask's total length, to control for it being a different physical size depending on *viewing*

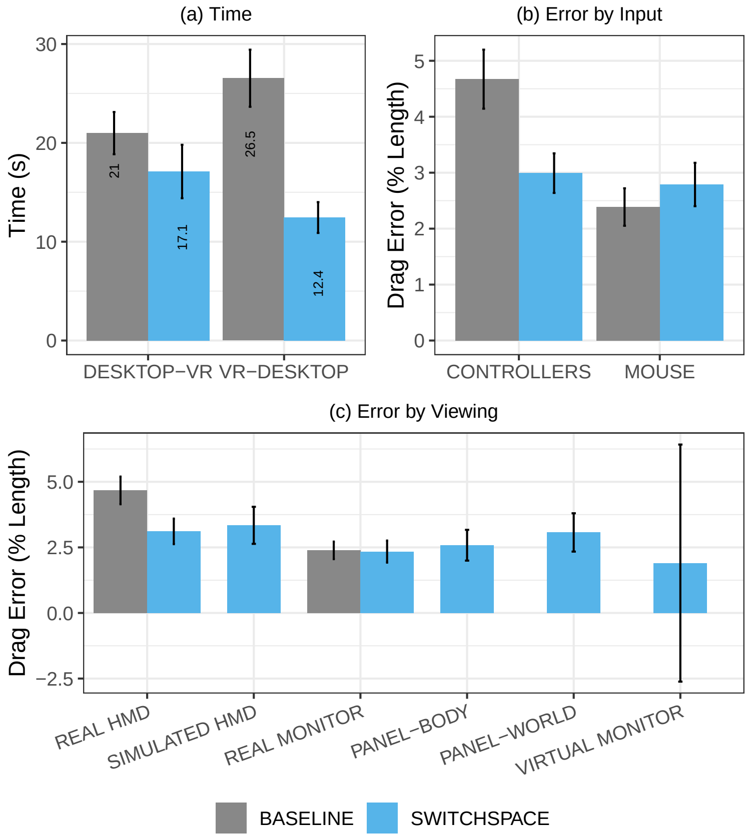


Figure 4.9: Results: (a) *Time* by TECHNIQUE and DIRECTION; (b) *Drag Error* by TECHNIQUE and INPUT; and (c) *Drag Error* by TECHNIQUE and VIEWING. Error bars represent 95% CI. BASELINE could only use REAL HMD and REAL MONITOR. VIRTUAL MONITOR was rarely chosen, resulting in high variance.

technique. To evaluate *Drag Error*, each interaction technique is a separate level of the independent variables INPUT (CONTROLLERS, MOUSE) and VIEWING (REAL HMD, SIMULATED HMD, REAL MONITOR, PANEL-BODY, PANEL-WORLD, VIRTUAL MONITOR). *Transitions Per Trial* is the number of changes in the participant's *input* technique, *viewing* technique, or overall *context* (as in Figure 4.2) in a single trial. Our implementation senses state transitions as described in Sections 4.4.1–4.4.2, and records them separately as *Input Transitions*, *Viewing Transitions*, and *Context Transitions*. *Math Error Rate* is the proportion of trials where participants incorrectly answered the math question at least once before answering correctly. *Unlock Error Rate* is the proportion of trials where participants missed the slider target (i.e., releasing the drag too early or late) in the unlock task at least once. We treat each NASA-TLX question as its own dependent variable: *Mental*, *Physical*, *Temporal*, *Performance*, *Effort*, and *Frustration*.

In summary: 2 TECHNIQUES × 2 DIRECTIONS × 4 POSITIONS × 2 PEEKS = 32 data points per participant.

4.5.5 *Results*

For each combination of participant, TECHNIQUE, and DIRECTION, we removed outliers by fully excluding from analysis any trial with *Time*, *Drag Error*, or *Transitions* more than 2 standard deviations from the mean: 28 trials (5.5%) were removed. Of the 28 trials removed, 14 were in the first trial for each condition (likely due to learning) and the rest were non-uniformly scattered along the 7 remaining trials for each condition. Examining the distribution of outliers shows that nearly all outliers were due to their *Time*. The ANOVA assumptions of homoscedasticity and normality were tested and corrected with log-transform or aligned-rank transform where noted, and we report Greenhouse-Geisser ($\epsilon < 0.75$) corrected degrees of freedom when the assumption of sphericity was violated. All pairwise comparisons use pairwise Wilcoxon signed-rank tests with Holm-Bonferroni corrections.

4.5.5.1 *Learning Effect*

We are interested in practised performance, so we remove initial slower trials due to learning effects. An initial log-transformed *Time* × TRIAL ANOVA found a significant effect ($F_{7,475} = 12.94$, $p < .001$), and pairwise comparisons showed that trials 1 – 3 were slower than trials 4 – 8. In subsequent analysis, we use trials 4 through 8 for each condition as they represent practised performance [118].

4.5.5.2 *Time*

Participants completed the task faster in all cases when they could use peeking techniques, but the effect was more pronounced when going from VR to desktop (Figure 4.9a). We analyzed log-transformed values for *Time* using a TECHNIQUE × DIRECTION × POSITION × PEEK ANOVA. We found a main effect of TECHNIQUE on *Time* ($F_{1,276} = 131.1$, $p < .001$) showing that in general, SWITCHSPACE was faster than BASELINE (14.8 s vs 23.8 s). While there was no main effect of DIRECTION, we found a TECHNIQUE × DIRECTION interaction effect ($F_{1,276} = 23.3$, $p < .001$), prompting post-hoc tests for each TECHNIQUE. We found significant effects between all combinations of TECHNIQUE and DIRECTION, but the effect of DIRECTION was more pronounced in BASELINE ($Z = 2.92$, $p < .005$) than in SWITCHSPACE ($Z = 2.32$, $p < .05$).

Participants completed tasks faster when the VR portion was near the desk. We found a main effect of POSITION on *Time* ($F_{3,276} = 5.64$, $p < .001$). Pairwise comparisons showed that conditions where POSITION was near the desk (DESK or HIGH) were completed more quickly ($Z = 2.31$, $p < .05$) than those away from the desk (17.6 s vs 21.0 s).

Participants were slower when required to peek in *input* and *viewing*. We found a main effect of PEEK on *Time* ($F_{1,276} = 149.9$, $p < .001$), showing that trials with an INPUT+VIEW peek were slower (24.5 s vs 14.6 s).

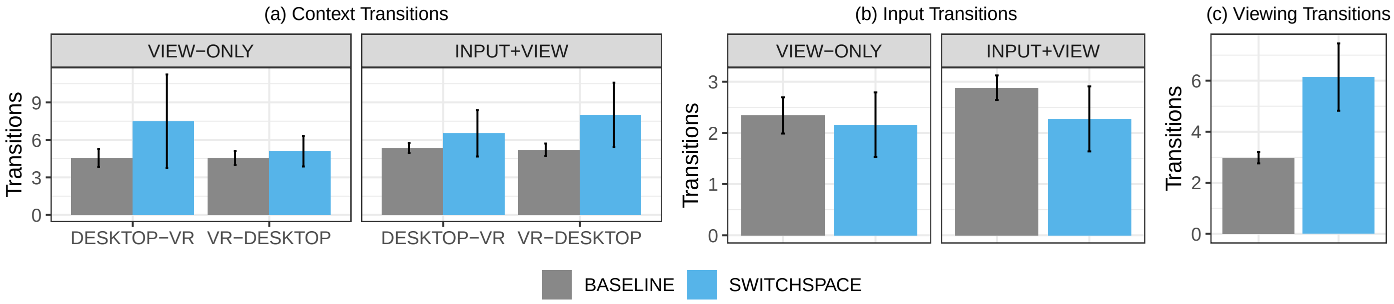


Figure 4.10: Results: (a) *Context Transitions* by TECHNIQUE, DIRECTION, and PEEK; (b) *Input Transitions* by TECHNIQUE and PEEK; and (c) *Viewing Transitions* by TECHNIQUE. Error bars represent 95% CI.

4.5.5.3 *Drag Error*

When using peeking techniques to complete the unlock task, the controllers were as accurate as the mouse (Figure 4.9b), and the VR headset was as accurate as the desktop display (Figure 4.9c). Residuals for *Drag Error* were not normally distributed, so we analyzed log-transformed values using a TECHNIQUE × INPUT × VIEWING × DIRECTION ANOVA. We found a main effect of TECHNIQUE ($F_{1,581} = 5.98$, $p < .05$), INPUT ($F_{1,581} = 26.93$, $p < .001$), and VIEWING ($F_{5,581} = 3.91$, $p < .01$) on *Time*. We found significant TECHNIQUE × INPUT ($F_{1,581} = 7.26$, $p < .01$) and TECHNIQUE × VIEWING ($F_{2,581} = 7.26$, $p < .01$) interactions, prompting separate post-hoc analysis by TECHNIQUE. While using the controllers and the HMD in BASELINE resulted in almost twice as much (95.8% more) drag error as the mouse and the desktop display ($Z = 6.09$, $p < .001$), there were no significant differences between INPUT or VIEWING devices in SWITCHSPACE.

4.5.5.4 *Transitions*

Participants changed *context* states more often when peeking techniques were available (Figure 4.10a). We analyzed transitions using a TECHNIQUE × DIRECTION × PEEK ANOVA on aligned rank-transformed values, as residuals were not normally distributed. There was a main effect of TECHNIQUE ($F_{1,461.6} = 124.7$, $p < .001$) and PEEK ($F_{1,461.4} = 108.1$, $p < .001$) on *Context Transitions*, as well as a TECHNIQUE × PEEK interaction effect ($F_{1,461.6} = 116.4$, $p < .001$), prompting separate pairwise comparisons by PEEK. Participants transitioned between context states more in the SWITCHSPACE technique than in BASELINE when the panels needed to be unlocked (7.2 times per trial vs 5.3, $Z = 8.50$, $p < .001$), but there was no significant difference when panels did not need to be unlocked. There was no significant effect of DIRECTION.

Participants changed *input* states less often in general when peeking techniques were available, and more often when the panels needed to be unlocked (Figure 4.10b). We found a main effect of TECHNIQUE on *Input Transitions* ($F_{1,228.8} = 21.6$, $p < .001$), showing that participants changed input techniques fewer times per trial in the SWITCHSPACE technique than in BASELINE (2.2

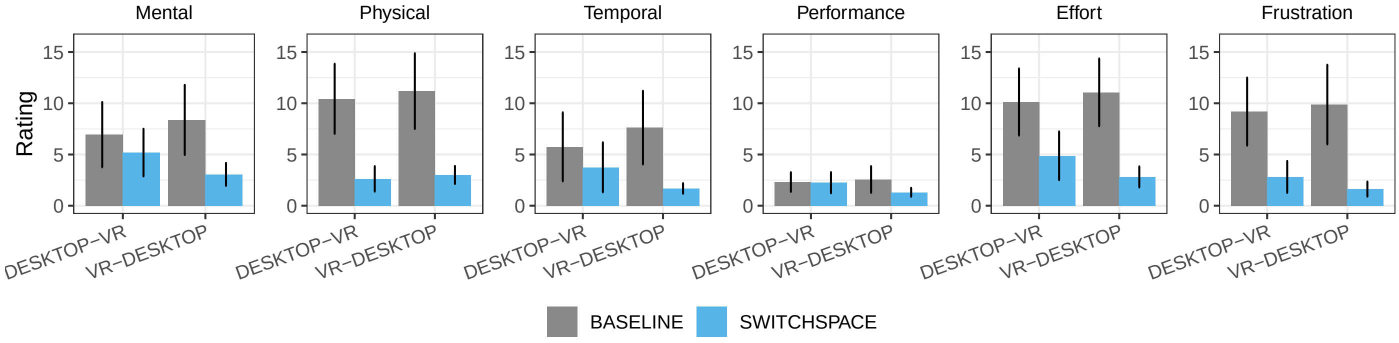


Figure 4.11: Results from the NASA-TLX questions by TECHNIQUE and DIRECTION. Error bars represent 95% CI.

times per trial vs 2.7). We also found a main effect of PEEK on *Input Transitions* ($F_{1,226.8} = 10.4$, $p < .01$) showing that participants changed input devices more often when panels required unlocking (2.8 times per trial vs 2.3).

Participants transitioned between *viewing* states more often when peeking techniques were available (Figure 4.10c). There was a main effect of TECHNIQUE on *Viewing Transitions* ($F_{1,224.9} = 52.9$, $p < .001$) showing that participants changed their viewing device more often per trial when peeking techniques were available (6.1 times per trial vs 3.0).

4.5.5.5 *Error Rates*

Participants rarely answered the math question incorrectly, and the number of unlock errors depended on task position. Residuals for *Math Error Rate* and *Unlock Error Rate* were not normally distributed, so we analyzed aligned-rank transformed error rates using a TECHNIQUE × DIRECTION × POSITION × PEEK ANOVA. We found no significant effects on *Math Error Rate*, which were below 3% for all levels of TECHNIQUE × DIRECTION. There was a main effect of POSITION on *Unlock Error Rate* ($F_{3,293.4} = 6.2$, $p < .001$), and pairwise tests found that SIDE was more error-prone than HIGH ($Z = 2.33$, $p < .05$). Mean values (± 95% CI) are $31.3\% \pm 10.2\%$ for SIDE, $14.1\% \pm 7.9\%$ for HIGH, $12.5\% \pm 8.8\%$ for DESK, and $15.6\% \pm 8.1\%$ for BACK.

4.5.5.6 *NASA-TLX*

Peeking techniques reduced perceived workload across several categories, but varied based on task direction (Figure 4.11). We analyze answers to the six NASA-TLX questions across all four combinations of TECHNIQUE and DIRECTION. Peeking techniques reduced *Mental* workload in VR-DESKTOP ($Z = 3.12$, $p < .05$), but did not affect DESKTOP-VR. Peeking techniques reduced *Physical* workload in both VR-DESKTOP ($Z = 3.21$, $p < .01$) and DESKTOP-VR ($Z = 3.16$, $p < .01$). VR-DESKTOP-SWITCHSPACE was also lower than DESKTOP-VR-BASELINE ($Z = 3.22$, $p < .01$), and DESKTOP-VR-SWITCHSPACE was lower than VR-DESKTOP-BASELINE ($Z = 3.07$, $p < .01$). Peeking techniques also reduced *Temporal* workload, but only in VR-DESKTOP ($Z = 2.81$, $p < .05$). There were no significant differences in ratings for perceived *Performance*. Peeking techniques reduced perceived *Effort* in both VR-DESKTOP ($Z = 3.49$, $p < .05$) and DESKTOP-VR ($Z = 2.98$, $p < .05$). VR-DESKTOP-SWITCHSPACE was also lower than DESKTOP-VR-BASELINE ($Z = 3.38$, $p < .05$), and DESKTOP-VR-SWITCHSPACE was lower than VR-DESKTOP-BASELINE ($Z = 2.98$, $p < .05$). Peeking techniques reduced *Frustration* in both VR-DESKTOP ($Z = 3.38$, $p < .05$) and DESKTOP-VR ($Z = 2.95$, $p < .05$). VR-DESKTOP-SWITCHSPACE was also lower than DESKTOP-VR-BASELINE ($Z = 3.45$, $p < .05$), and DESKTOP-VR-SWITCHSPACE was lower than VR-DESKTOP-BASELINE ($Z = 2.57$, $p < .05$).

4.5.5.7 *Preferences and Feedback*

Participants generally preferred configurations where they could maintain their starting input and viewing devices. If necessary, they would rather change their viewing device than their input device. 15 of 16 participants preferred peeking to fully transitioning.

Participants' preferred techniques in the post-questionnaire were the body-anchored desktop view for VR-DESKTOP, and the simulated HMD view for DESKTOP-VR. To verify, for each DIRECTION in the SWITCHSPACE conditions, we calculated the proportion of all transitions that put the user in a *peek* state. For VR-DESKTOP, the body-anchored panel view was activated the most (63% of 211 *viewing* peek transitions), followed by the world-anchored panel view (34%) and the virtual monitor (3%). For DESKTOP-VR, the simulated HMD view was activated the most (41% of 274 *viewing* peek transitions), followed by donning the headset (33%). Some participants in the DESKTOP-VR conditions would fully transition to *VR* to find *X*, then answer the question using a VR-to-desktop peeking technique. This represented 11% (world-anchored panel), 10% (body-anchored panel), and 5% (virtual monitor) of peek transitions. The transitions for *input*, generally reduced by peeking techniques (from 458 total in BASELINE to 129 in SWITCHSPACE), showed roughly even use of the mouse and the controllers. Techniques using the virtual monitor, namely moving the mouse in *VR* or the *controller-mouse* peeking technique (sliding the controller on the desk to provide mouse-like cursor input), represented 6 total transitions across all participants. This caused the large confidence interval in the VIRTUAL MONITOR *Drag Error* result (Figure 4.9c).

Participants preferred techniques that let them maintain their current input and viewing devices. In the VR-DESKTOP condition, most preferred the body-anchored desktop view for peeking to the desktop. P14 explains: *"If I'm already in VR, I like using the body interface to peek at the desktop because it's quick and not obtrusive"*. In DESKTOP-VR, most participants preferred the simulated HMD view. P4 *"caught on quickly to how it worked [...], it felt like an extension of how I already know how to use a mouse and keyboard"*. P1 agreed: *"Being able to stay in one aspect of the monitor/VR was great for extended periods of time, vs constant switching."* Participants thought *"any technique that kept me using one input modality was great"* [P10].

Some participants preferred desktop view panels at different times. In the VIEW-ONLY trials where the missing number was easily visible, some participants preferred to quickly check the body-anchored panel. In INPUT+VIEW, some participants preferred the world-anchored panel because *"[the body-anchored panel] can sometimes feel shaky or unstable since it moves with my body"* [P3].

Participants found donning and doffing the HMD uncomfortable. For example, P10 *"felt like I always had to adjust its positioning (either because the rubber sometimes got caught in my hair, or I had to make loosen and again tighten it)"*. P12 agreed: *"[I preferred] anything that lets me avoid putting down the headset (or putting it on in the first place)"*.

In BASELINE, five participants forgot the value of *X* and had to transition back to the secondary interface to complete the trial. Peeking techniques reduced this forgetfulness, or at least made transitioning again more convenient: *"I don't think I can remember X [the missing value] after taking so long to switch. [I liked peeking because] I didn't feel the pressure of remembering X and it was handy to look again if insecurity hit or if I had some memory lapse issue"* [P15].

4.5.6 *Discussion (RQ2)*

The formative study demonstrated that real-world VR users encounter usability challenges associated with their cross-reality workflows. To address this, this experiment's primary goal was to answer **RQ2** by evaluating context-aware peeking as an effective and preferable alternative to fully transitioning. Our results expand on the findings of the formative study and show that while transitioning between VR and desktop interfaces can be arduous, having a quick way to accomplish the same cross-reality task without transitioning makes it easier and faster.

The increase in speed using peeking techniques shows how the need to manipulate a headset and controllers can affect the speed of a cross-reality workflow. Especially in the baseline conditions, the need to manage the position of the HMD, mouse, and controllers caused significant time penalties. This was especially true in the trials where the task was placed in the SIDE or BACK positions, as participants either had to physically go back to the desk to place devices down or hold multiple devices in one hand when transitioning. The speed benefit was the most dramatic in VR-DESKTOP, suggesting that quickly summoning the desktop view was a faster technique than navigating the simulated HMD view, even if both provided notable improvements.

Peeking techniques decreased *Drag Error*, suggesting that at least in our synthetic task, the ability to peek between interfaces resolved accuracy differences between mouse and controllers, and between the HMD and the monitor. This may be because the sliding task was simple for both mouse pointing and controller raycasting, but the dramatic change in relative error between BASELINE and SWITCHSPACE suggests that the change was due to peeking.

The *Number of Transitions* further illustrates how peeking techniques affect usage habits. Participants changed their *context* more often when peeking techniques were available. This may be because there were more intermediary techniques available in the SWITCHSPACE condition. However, these

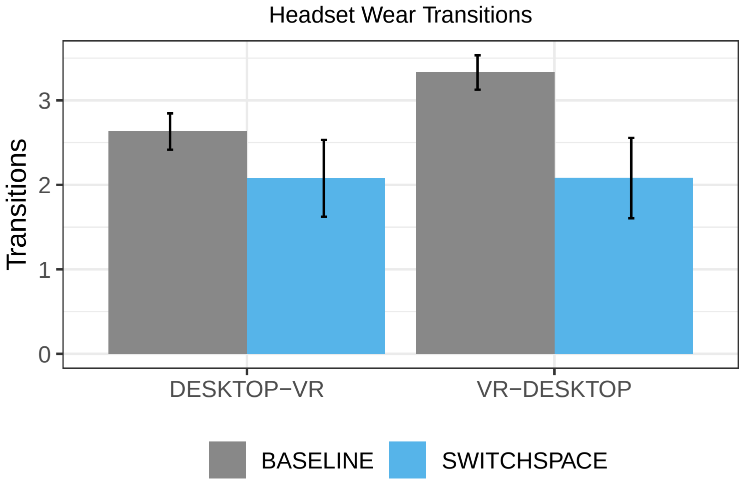


Figure 4.12: *Headset Wear Transitions* by TECHNIQUE and DIRECTION. Error bars represent 95% CI.

results become more interesting when considering the different categories of state transitions separately. *Viewing Transitions* considers activating peeking techniques such as the simulated HMD view or the *VR* desktop views, but the results prompt deeper investigation into pure hardware changes. To explore more deeply, we analyzed how often participants would don or doff the headset in a single trial (*Headset Wear Transitions*). Overall, participants donned or doffed the headset less when peeking techniques were available (Figure 4.12). We found a main effect of TECHNIQUE ($F_{1,229.3} = 63.2$, $p < .001$) and DIRECTION ($F_{1,228.3} = 30.6$, $p < .001$) on *Headset Wear Transitions*, as well as a TECHNIQUE × DIRECTION interaction effect ($F_{1,225.3} = 9.0$, $p < .01$), prompting separate comparisons by DIRECTION. People transitioned fewer times per trial using SWITCHSPACE than BASELINE, but the difference was larger in VR-DESKTOP ($Z = 8.49$, $p < .001$ for both). Overall, having peeking techniques available enabled people to transition between *VR* and *Desktop* more often, while manipulating actual hardware less.

The *Math Error Rate* results validates our choice of task, it was easy enough to be completed regardless of interface. The results for *Unlock Error Rate* suggest that while transitioning interfaces may not have affected drag accuracy in our task, the user's position relative to the dragging task may have been contributed. Participants in some trials would sacrifice accuracy for comfort by raycasting to the SIDE panel from their seated position at the desk. We chose not to set a maximum distance at which the trial could be completed, which may have caused this effect.

Peeking techniques generally reduced *NASA-TLX* scores, which is surprising considering that participants had to learn multiple different input and viewing techniques. Most participants preferred using one technique for each DIRECTION, suggesting that the perceived benefit of transitioning outweighs the effort of learning a new technique as long as the amount of learning is minimal. Peeking techniques impacted more categories of workload in VR-DESKTOP, suggesting that the gestural (body-anchored view) or single-button (world-anchored view) techniques may have been simpler to use than the 3D navigation of the simulated HMD view.

The *Preferences and Feedback* further contextualize our results. Participants preferred peeking techniques that reduced hardware changes as well as the amount of movement. The preferences for raycast-based techniques in *VR* (PANEL-BODY, PANEL-WORLD) and mouse-based techniques in *Desktop* (SIMULATED HMD) suggest that hardware changes and physical motion are the driving factors of discomfort in transitioning. Moreover, these peeking techniques maintain the most common interaction metaphors for their associated primary state (raycasting in *VR* and mouse cursor in *Desktop*), which could explain this preference. This is evident in the results for *Drag Error*, particularly for VIRTUAL MONITOR. While all other techniques were used in at least 30 trials across all participants, the VIRTUAL MONITOR technique was used in only 3 trials, causing the large variance and resulting large error bar in Figure 4.9c. Participants forgetting the value of X mid-trial in the BASELINE technique suggests that in addition to disorientation or discomfort [72], manipulating hardware and re-orienting to the real world can impact short-term memory, possibly due to higher cognitive load. The participant comments also further contextualize the *NASA-TLX* results, suggesting that manipulating hardware, not transitioning input or viewing techniques, was the biggest contributor to the increased perceived workload in BASELINE.

4.6 GENERAL DISCUSSION

Our goal with this work was twofold: understand the challenges and preferences of VR content creators with cross-reality workflows (**RQ1**), and use these insights to develop an effective and preferable alternative to fully transitioning devices (**RQ2**). The formative study shows that users with cross-reality workflows do transition between desktop and VR, but these changes are uncomfortable, disruptive, and most importantly, temporary (**RQ1**). Focusing on the temporary nature of desktop-VR transitioning gives rise to a design space of momentary "peeking" techniques, which make cross-reality workflows faster, less cognitively demanding, and overall preferable to fully transitioning between interfaces (**RQ2**). We present design recommendations based on our results, discuss possible limitations in our methods, and discuss future applications for cross-reality peeking techniques.

4.6.1 *Design Recommendations*

Our results suggest general design guidelines for cross-reality workflows, building on earlier work in cross-reality blending [56, 72, 84] but with specific focus on transitioning between desktop and VR.

4.6.1.1 *Minimize Hardware Changes*

Participants preferred peeking techniques that allowed them to avoid manipulating hardware. Peeking techniques generally reduced both input device transitions and headset transitions, and the most common peeking techniques were the body- and world-anchored desktop view for VR-to-desktop peeking, and the simulated HMD view for desktop-to-VR, both of which avoid changing input or viewing hardware. Hardware changes while in VR can be especially difficult, since without reality-blending techniques [60, 84] or appropriate real-virtual aligned models [116], there is no visual way to find the mouse or keyboard in the real world. Cross-reality peeking allows users to circumvent hardware changes, and future designers should consider avoiding hardware changes in cross-reality interface design.

4.6.1.2 *Design for Physical Space*

Our findings illustrate that peeking techniques also reduce physical movements. Our experiment required participants to walk around a space to interact with panels in the VR scene, as well as return to a physical desk. As before, the most common peeking techniques allowed users to avoid movement, since the simulated HMD view was done at the desk, and the body- and world-anchored desktop views could be summoned anywhere in the VR environment. Some peeking techniques were not used by participants, likely due to the requirement to move back to the desk. For example, the VR-to-desktop techniques which summoned the desk-aligned virtual monitor (requiring the user to move back to the desk) were almost completely unused. This may have been a result of our task design, as other cross-reality tasks (e.g., using the virtual monitor and mouse to control secondary a camera within the VR scene) could make returning to the desk more worthwhile. Peeking techniques that minimize real-world movement can make tasks faster, especially in more constrained real-world environments.

4.6.1.3 *Use Familiar Interface Metaphors*

The most frequently-used peeking techniques maintained the typical interface metaphor for their primary state. For desktop-to-VR peeking, the simulated HMD view utilized standard WIMP mouse interaction [28]. Likewise, for VR-to-desktop, the body- and world-anchored desktop views used raycasting, a common interaction technique in consumer VR. Pointing the VR controller at the physical monitor or using the controller on the desk like a mouse were underutilized, perhaps because they introduced new metaphors. Designers should consider peeking techniques which maintain the most common input mechanism of the primary state.

4.6.2 *Limitations*

4.6.2.1 *Task Choice*

The math task was designed to be simple and experimentally controllable, while still prompting transitions faithful to those described in the formative study and design space. It uses 2D selection on a 3D plane, as well as 6DOF raycast input for some desktop and VR input methods. Previous work found differences between 2D and 3D pointing and manipulation tasks [11–13, 18], meaning task-specific performance results such as our 2D *Drag Error* may not apply for all implementations. However, our emphasis on transitions within a simple task means that our relative results for *Time*, *Number of Transitions*, *NASA-TLX*, and *Preferences and Feedback* likely provide generalized insights.

4.6.2.2 *Other Usage Contexts*

Our work did not consider all variations of cross-reality interaction, like responding to bystanders [40, 73] or different approaches to reality blending [96]. We focus on workflows involving a single person with basic reality-blending techniques to render their physical desk and monitor in VR. Future work could explore VR-desktop peeking techniques that consider specialized scenarios with bystanders and more complex reality blending.

4.6.2.3 *Technique Choice*

Our experiment shows that context-aware peeking is effective and preferable, but more specialized 3D tasks may require extending our current set of peeking techniques. We chose our peeking techniques to be best suited for cross-reality pointing and selection because they are essential and common in both desktop and VR interactions, and the formative study found related tasks like quickly changing settings on desktop then viewing the effects in VR. However, while our techniques support some 3D interaction, like 6DOF pointing and 3D camera manipulation, more complex 3D manipulations like docking would warrant extending our techniques for more complex direct manipulation. Designing input peeking techniques to support more complex 3D tasks would likely be simple extensions within our design space. For example, *SwitchSpace*'s context-awareness could easily trigger mode-switches between simple and more complex input, like the 2D-3D transitions of Bogdan et al. [18]. Future work should evaluate more complex tasks with a larger set of peeking techniques.

4.6.2.4 *Hardware*

The Quest Pro has a small gap between the user's nose and the headset, which some participants wanted to look through in order to see the desktop. We instructed participants to fully remove the headset when transitioning to desktop to control for inconsistencies between headset models and gather more general insights. This could be considered another form of peeking technique, especially for non-occlusive headsets [56]. Similarly, some study participants wanted to rest the headset on their forehead when checking the desktop display. We used the built-in proximity sensor to sense when the headset is put on or taken off, so resting the headset on the forehead can cause the system to function as if the headset is still worn. We instructed participants to remove the headset entirely when transitioning to desktop, but this may have resulted in slightly longer *Time* measures in BASELINE. However, the mean decreases in *Time* (3.9 s in DESKTOP-VR and 14.1 s in VR-DESKTOP) suggest that peeking techniques would still be faster. A more custom hardware implementation could use sensors other than headset proximity to avoid this issue. Another hardware limitation is that some VR systems depend on the sensors in the HMD to track the controllers. As a result, peeking techniques where controllers are tracked independently may not work for all VR systems.

4.6.3 *Future Work*

Our work is part of a continuing effort to make spatial computing more interoperable with other categories of computing.

4.6.3.1 *Additional Cross-Reality Tasks*

Our design space provides insights for a simple cross-reality pointing task, but more detailed comparisons extending our techniques toward more complex tasks is an interesting avenue of future work. For example, our through-the-monitor raycast technique, extended by adding simple swipe motions or a contextual mode-switch between 3D manipulation and 2D pointing [18], could allow the VR controllers to be more suitable for use in desktop-based 3D design programs without the need to fully enter VR. Similarly, in-VR locomotion (e.g., teleportation [20] or walking-in-place [75, 129]) could extend our *input* peeking technique of using the mouse in VR to select a new in-VR position without fully having to grab VR controllers.

4.6.3.2 *Alternate Context Signals*

We designed our state machine and peeking techniques around a definition of context which includes input and viewing devices. However, other factors in a user's environment could be repurposed into state machine input. For example, context aware systems could track whether a user is at or away from their desk, or whether they are seated or standing. Future work could examine additional sources of context.

4.6.3.3 *Passthrough for Peeking*

The cameras on current VR HMDs enable a "passthrough" mode for viewing the real world while wearing the HMD. While the resolutions of cameras in current passthrough systems are generally too low for detailed cross-reality work, future work could evaluate higher-fidelity passthrough functions as another peeking technique, extending and further evaluating implementations like George et al. [47] or Do et al. [40].

4.6.3.4 *Addressing Situational Impairments*

A large body of accessibility work focuses on *situational impairments*, temporary degradation of the user's capabilities based on their current circumstances. For example, walking can affect typing accuracy [54]. Some work addresses situational impairments in augmented reality using gaze-based interfaces [50], but little work examines how context can be used to overcome additional situational impairments for VR interfaces. We consider this work one example of of a design to overcome a specific situational impairment, but other cases and situational impairments might exist. Future work should examine other hardware based situational impairments and unique ways to address them.

4.7 CONCLUSION

Informed by a formative study with VR users and content creators, we presented *SwitchSpace*, a design space for context-aware UI which enables quick and temporary "peeking" between VR and desktop interfaces. Peeking techniques in *SwitchSpace* are determined by changes in input device (mouse or VR controllers) and viewing device (desktop display or VR HMD). Peeking techniques enabled users to transition between VR and desktop more often, but manipulate hardware less. A user study of peeking interaction techniques found they made a controlled cross-reality workflow faster, more comfortable, and less cognitively demanding.

Interface changes are inevitable in modern VR workflows. Our design space allows VR applications to use these changes as input, rather than as distractions or hindrances to productivity, helping make cross-reality interfaces more comfortable, more accessible, and more fluid for extended use.

5

MOTIONBLOCKS: MODULAR GEOMETRIC MOTION REMAPPING

In the previous chapters, we provide a motivational understanding of multimodal input, and demonstrate how context can also be used to provide secondary input to a VR application. With these findings as the foundation, we can now more thoroughly investigate how to leverage multi-modal input and contextual factors to design and implement a robust and customizable motor accessibility solution.

5.1 INTRODUCTION

Virtual reality (VR) makes implicit assumptions about user abilities [143, 144] that may be difficult or even impossible for people with mobility limitations to meet. Previous work in VR motor accessibility has focused on broader design considerations [36, 37, 48, 49], but lacks a detailed first-hand understanding of the underlying conflict between the physical motions required by an application's interaction design and users' ability to perform them. Similarly, previous implementations of VR accessibility techniques focus on specific categories of motor disability [46, 74, 146] without consideration for a more generally-applicable solution.

As spatial computing devices embed themselves further into the mainstream, UIs that rely purely on motion are becoming increasingly common. This increasing popularity and more diverse audience makes the demand for a detailed and generalizable method for making motion accessible increasingly critical. As a step toward more accessible spatial motion input, our work explores three research questions:

(**RQ1**) *What kinds of motion-related interactions in VR games cause accessibility issues?*

(**RQ2**) *How do we represent these interactions in a way that is easy to identify, and easy to adapt into more accessible spatial input mappings?*

(**RQ3**) *Do these custom input mappings make common VR applications easier to use?*

To answer these questions, we first conducted a formative study with 10 people with a diverse range of mobility limitations. Participants tried five VR applications that covered a variety of motion requirements for interaction and locomotion. An analysis of participant comments and our observations produced detailed sub-categories of findings both within and outside those defined by previous work, as well as prompted a wider discussion of mobility requirements.

A key outcome of the formative study is the concept of *motion primitives*, a way to specify both the range of motion for a user and the range of motion

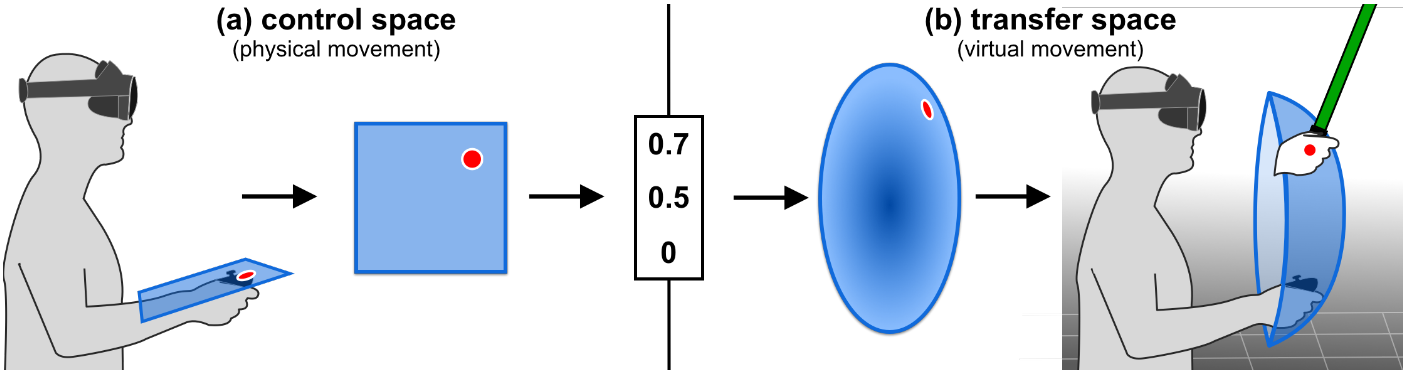


Figure 5.1: *MotionBlocks* is an approach for constructing complex 3D input using more accessible ranges of motion or simpler input devices. MotionBlocks uses geometric primitives to capture and reconstruct motion, bringing input from *control space* (the range of motion the user can accomplish) to *transfer space* (the range of motion necessary to interact with the VR application). In this example, a user's physical hand moves within a control-space *Plane* primitive, which is remapped to a transfer-space *Hemisphere* primitive.

required by a VR application, using a set of geometric primitives that require fewer dimensions of input to traverse. Describing motion with geometric primitives gives designers a concise language to illustrate the body motions necessary to complete a given 3D interaction, and can be useful as a theoretical framework for categorizing and addressing 3D input accessibility issues. Using this concept, we developed *MotionBlocks* (Figure 5.1), a modular input remapping approach using motion primitives to change how users control their input and how it is represented within the VR application. This modular remapping design enables the user to perform large 3D interactions using a more comfortable range of motion, possibly with lower dimensionality, in a different coordinate space, or even using an alternative input device. We evaluated this concept in a user study, in which 8 of the formative study participants created their own customized motion primitive remapping configurations, then played the same VR games both with and without remapping applied. This enabled users to complete tasks in VR more comfortably and more effectively than with a standard input configuration.

We make three contributions: (1) a detailed description and characterization of accessibility issues with spatial input, based on the results of a formative study; (2) a description of how geometric primitives can be used as a way to categorize and address accessibility issues within spatial input; and (3) the results of a user study showing that customizable geometric input remapping can make VR motion more accessible.

5.2 RELATED WORK

Chapter 2 provides a detailed discussion on previous work focusing on eliciting accessibility challenges in computing platforms and previous VR accessibility solutions. Here, we focus specifically on previous technical approaches for spatial input remapping.

5.2.1 *Spatial Input Remapping*

Related to our general approach are methods that scale the movement of a controller to amplify the movement of a corresponding virtual hand position. Classic unbounded methods, like Go-Go [104] or HOMER [19], extend the user's reach far beyond the length of their physical arm. Alternatively, bounded remapping techniques apply scaling to the input motion, but keep the virtual hand within a realistic "arms-reach" distance from the body. For example, Tseng et al. [127] explored using fingertip movement to control VR hand motion. This technique was focused on more comfortable motion with minimal physical movement for constrained spaces. Motor accessibility was not tested, but the method significantly reduced fatigue. For the more conventional remapping of physical to virtual hand positions, Erg-O [89] used a simulated annealing approach to create dynamic ergonomic transfer functions based on known targets near the user's body. RNL amplification [141] is an alternative approach using a non-linear transfer function to make arms-reach VR input more ergonomic. Both techniques reduced fatigue, which in theory could improve accessibility. Of particular interest to our approach, tests of RNL found that participants maintained performance even at high levels of amplification as long as reach was kept within realistic bounds.

Previous work proposing spatial input remapping has focused on creating input adaptations without an explicit focus on accessibility. Our more general approach for input remapping is explicitly focused on accessibility.

5.2.2 *Summary*

Unlike most previous work investigating VR motor accessibility, we use direct observation to produce more specific insights than interviews about previous experiences or anticipated difficulties, and then use these insights to inform a concise geometric language for designers to examine the relative accessibility of spatial applications. We use this geometric language to propose a new bounded input remapping technique inspired by Tseng et al., Erg-O, and RNL, but with an emphasis on customizability inspired by Nearmi.

5.3 FORMATIVE STUDY

Previous work in VR accessibility places relatively little emphasis on understanding the underlying structure of body motions that result in inaccessible VR motion. We answer **RQ1** by conducting a contextual inquiry study [16] focusing on direct observation of participant VR usage. Contextual inquiries can motivate VR design [136], and using unmodified VR applications establishes a similar context for observation as if they bought the hardware themselves.

Table 5.1: Demographic information for study participants.

ID	Age	Gender	Self-Reported Mobility Limitation
P1	26	W	Dwarfism, paralyzed from the waist down, uses a power wheelchair
P2	17	M	Spastic diplegic cerebral palsy, uses wheelchair outside of house and crawls in house
P3	18	M	Cerebral palsy
P4	24	W	Genetic condition resulting in physical development delay
P5	15	M	Tri-plegia cerebral palsy
P6	50	M	T4 complete spinal cord injury
P7	31	M	Spina bifida
P8	68	M	Parkinson's syndrome: tremors in hands/arms, lack of balance, vertigo
P9	63	W	Psoriatic arthritis and osteoarthritis of wrist, hand joints, knees, and ankles
P10	15	M	Left-sided weakness in arm and leg, reduced elbow bending ability

5.3.1 *Participants*

We recruited 10 people with mobility limitations (7 identify as men, 3 as women, age 15–68, median age 25) to participate in the study. Table 5.1 provides full demographics and mobility limitations. Seven of 10 participants reported having little to no VR experience, but 6 of 10 reported at least moderate experience with video games. Sessions took place at a local disability outreach foundation, or in the participants' homes. The protocol was approved by an ethics review board.

5.3.2 *Apparatus and Applications*

Our study used a Meta Quest 3 HMD connected to a PC powered by an Intel Core i7-10875 CPU and a NVIDIA RTX 2080 GPU. The facilitator used the PC to start and stop each application, monitoring the onscreen headset view to provide assistance.

Participants used five VR applications (Table 5.2). All applications used controllers and were selected to have a variety of locomotion techniques, interaction techniques, and levels of activity. We categorized applications by distance of menu interaction, 3D selection and manipulation, and locomotion. Matching Tian et al. [124], we classified interactions as *Near* or *Far* based on distance away from the participant's arm's reach. For example, *Near* locomotion actions include physically bending, jumping, or joystick-based smooth locomotion, while *Far* locomotion actions involve teleportation. We added an additional category of bimanual interaction.

5.3.3 *Protocol*

Each session lasted approximately one hour, and participants were compensated with a $50 Amazon gift card. After providing informed consent,

Table 5.2: Applications used by study participants. Categorizations are from Tian et al [124].

Title	Description	Menu Interaction	Selection & Manipulation	Locomotion	Bimanual
TheBlu [142]	Exploration of an underwater scene, uses smooth joystick locomotion.	Near	Near	Near	None
Tilt Brush [55]	3D drawing, needs bimanual input and teleport locomotion.	Near	Near	Near, Far	Required
Walkabout Mini Golf [85]	Mini golf game, uses precise arm motion and teleport locomotion.	Far	Near	Near, Far	None
Space Pirate Trainer [62]	First-person shooter game, needs large body motions to dodge bullets.	Far	Far	Near	Optional
Beat Saber [14]	Rhythm game requiring arm swings, body motion to dodge hazards.	Far	Near	Near	Required

participants completed a pre-questionnaire about their VR and gaming experience, and a semi-structured interview about their experience with motion input.

The facilitator introduced the participant to the VR hardware, explaining that they could wear their glasses under the headset if necessary. Participants with balance or stability issues were offered the option to remain seated during the session. Following a brief tutorial on adjusting the headset straps and lenses, the participant put on the headset, which initially displayed the real environment in "passthrough mode" to ease the transition into VR.

For each application, the participant completed a short assisted walkthrough on accessing the application's main interactions or gameplay loop. Next, participants used the application for at least 10 minutes. Participants could remove the headset at any time. If the participant tried an application and decided that their level of mobility or discomfort prevented them from using the application entirely, it was skipped. This occurred 9 times across 3 participants, which we discuss in results.

Participants were instructed to think aloud while using each application and to describe any difficulties they encountered. After each application participants completed a short debrief interview about their difficulties. Experimenters recorded audio and took notes throughout the study.

5.3.4 *Results (RQ1)*

We analyzed session data using open and axial coding [35]. The first author read and open coded the participant dialogue and facilitator notes, using inductive analysis [86] to identify common themes. Two authors independently coded the dataset, and discussed disagreements to refine the codes.

Table 5.3: Motor accessibility issues from the study, including Cohen's kappa (κ) as a measure of inter-rater reliability. $\kappa \geq 0.6$ is substantial agreement.

Category	**Theme**	**Participants**	**Cohen's Kappa (κ)**
Spatial Input	Lateral Body Movement	8	**0.74**
	Virtual Locomotion	8	**0.62**
	Bending and Crouching	7	**0.74**
	Reaching	6	**0.62**
	Movement Speed	3	**0.78**
	Two-Handed	4	**1.00**
	Shakiness and Tremors	4	**0.78**
	Balance	5	**0.80**
	Co-pilot Compatibility	2	**1.00**
Application Design	Game Difficulty	2	**1.00**
	Setup Time	3	**0.73**
Hardware Ergonomics	Headset Adjustment	6	**1.00**
	Grasping Controllers	5	**0.80**
	Wheelchair Conflict	3	**0.78**
	Controller Buttons	5	**0.80**
	Passthrough	3	**1.00**

We divide the themes into three categories (Table 5.3), and discuss them in the subsections below.

5.3.4.1 *Spatial Input*

Themes in this category addressed conflicts between participant body motion ranges and the motion demanded by the application.

LATERAL BODY MOVEMENT Eight participants had issues with applications requiring lateral body movements, like sidestepping, turning, jumping, or leaning. P8, who completed the study while seated, struggled with *"the physicality of moving the whole body. With a chair you can only move so much"*. P7 agreed that the large lateral movements for dodging bullets in *Space Pirate Trainer* were difficult, and similar movements for slicing blocks and avoiding hazards in *Beat Saber* caused him to lean far out of his wheelchair (Figure 5.2). P7 speculated that this is due to the game's recognition of upper body motion: *"it comes down to how well trunk movement is registered as opposed to physically sidestepping"*. P7 suggested a transfer function for head motion, where *"half as much movement equates to the same amount in-game"*.

Figure 5.2: P7 leaning out of their wheelchair to avoid hazards in *Beat Saber*.

VIRTUAL LOCOMOTION Traversing the VR environment, and the small corrective motions associated with it, presented challenges for 8 participants. While standing participants could easily correct their physical position after teleporting (for example, aligning themselves for a putt in *Walkabout Mini Golf*), those seated or in wheelchairs had difficulty making precise movements, often resorting to multiple small corrective teleports. Despite these adjustments, some still found their positioning inaccurate. Some participants preferred smooth locomotion for this reason, like P7: *"if I could walk with a joystick it's way easier. I very often have to reset my view and go backwards"*. P5 suggested an arm-swinging metaphor [129], but *"can't do it with only one arm"*.

BENDING AND CROUCHING Similarly, 7 participants had issues with bending over or crouching. P5 noted issues with hazards in *Beat Saber* requiring a deep crouch: *"this is as far as I can go, and then I still have to get back up"*. P9 agreed: *"the moment I have to do fast movement with knee bending or crouching I can't do that"*.

REACHING Six participants had issues with the required amount of reach. P1 struggled with *Beat Saber*: *"farther blocks were hard to slash because my arms weren't long enough and my seated reach isn't high enough"*. P8 struggled playing *Walkaround Mini Golf*, noting that being seated caused his legs and the chair to be in the way of reaching downward to make a putt.

MOVEMENT SPEED Similarly, 3 participants had difficulties with the speed of motion required. Participants with difficulty bending their elbows or raising their arms had trouble with faster required movements. P5 explained: *"I'm not very fast so getting the arrows [in Beat Saber] is hard"*. P8, playing seated,

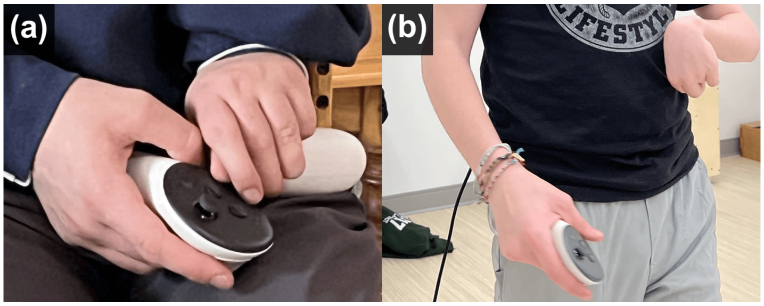


Figure 5.3: Examples of controller usage strategies for bimanual applications: (a) P10 uses his lap to keep his left controller steady for the bimanual interactions in *Tilt Brush*; (b) P5 plays *Beat Saber* with one arm, implicitly required to play the one-handed challenge mode.

used his non-dominant arm to support himself during quick arm movements because it was *"less fatiguing when I had the chair for support"*.

TWO-HANDED Four participants had issues with applications requiring the use of both hands simultaneously. P10 explained: *"my grip strength [in my left hand] means I can choose to either use motion or use buttons"*. *Tilt Brush* uses a "painter's palette" metaphor for virtual menus, requiring bimanual input which was challenging for P5 and P10. To complete these interactions, P10 braced the left controller against his lap (Figure 5.3a), and the facilitator held the left controller near P5's body. Similar difficulties were found in *Beat Saber*, which does support one-handed play (Figure 5.3b), but only as a challenge mode at the hardest difficulty instead of as an accessibility setting.

SHAKINESS AND TREMORS Four participants had issues with shakiness and tremors during input and menu selection. P8 described his experience: *"sometimes if my tremors are more pronounced it's really hard to tap [accurately]"*. P2 agreed, finding *"aiming [in Space Pirate Trainer] is hard, especially through the scope [on top of the player's guns]"*. P8 noted a feedback loop effect: *"stress [like selecting incorrectly] aggravates the tremors, making accuracy worse"*.

BALANCE Five participants had issues with body stability and balance. P9 suggested: *"I want to see my feet in VR. Without that I have no connection to the ground and have worse balance"*. P8 emphasized this idea: *"once you start falling there's no recovery. Standing up, I'd be on the floor. I have no visual frame of reference [for the floor], so it feels risky to stand up, especially in [virtual] terrain I'm unfamiliar with"*. This caused nausea in two participants, prompting some application demos to be skipped. P10 said faster-paced games *"probably would have been easier standing up, but I'd probably fall over"*.

CO-PILOT COMPATIBILITY Two participants required the use of a co-pilot to fully navigate some VR applications. The co-pilot relied on the PC rendering of the VR user's view to see, causing difficulty in selection due to head instability or lack of depth perception.

5.3.4.2 *Application Design*

Themes in this category describe how mobility issues can arise within central components of an application's design.

GAME DIFFICULTY Two participants commented on game difficulty making it hard to feel confident making large motions. As above, P5 had to play *Beat Saber* at the hardest difficulty: *"all other game modes are locked in one-handed mode so it restricts the fun and challenge"*. Both one-handed participants tried *Beat Saber* in two-handed modes (including "no fail" mode where missed targets are not counted against score), but the experience of missing half of the moving targets (and the targets subsequently passing close by the player's face) was uncomfortable.

Participants enjoyed game mechanics that reduce the amount of movement necessary to succeed. In *Space Pirate Trainer*, for example, the player can collect an item that spawns shields on their left and right for blocking bullets. P7 enjoyed this, but *"wanted to hop between the left and the right shield, through either button feedback or leaning"*.

SETUP TIME Three participants commented on the lack of "setup time" before the action begins, like during large environment changes or launching applications: *"it would be nice to be able to linger and not have to [start moving] so fast"* [P8].

5.3.4.3 *Hardware Ergonomics*

Themes in this category describe participants' issues with VR hardware, in particular the tension between VR hardware design and participants' abilities.

HEADSET ADJUSTMENT Six participants had issues fitting and adjusting the headset. Tightening the headstrap of the Quest 3 involves pulling apart two fabric pieces at the base of the skull, which was difficult for some participants.

GRASPING CONTROLLERS Five participants had difficulty grasping controllers, modifying how they were holding them as a result. P9's limited finger dexterity caused hand discomfort after a few minutes, causing some application demos to be skipped. P10's limited hand dexterity and discomfort caused him to hold the controller with the buttons rotated away.

WHEELCHAIR CONFLICT Three participants noted conflicts with their assistive devices. For two of these participants, this involved the choice between maintaining hold of the VR controller, or manipulating their wheelchair (and dropping the controller). For P1, larger arm movements caused a risk of striking her power chair's joystick. All three of these participants noted challenges manipulating their wheelchair while the headset was obscuring their vision.

CONTROLLER BUTTONS Five participants had issues pressing buttons or manipulating joysticks on the VR controllers. P2 required assistance to navigate *TheBlu*, which requires a two-button gesture for raycast selection: the middle-finger grip button to activate the ray, and the index finger trigger to make a selection.

PASSTHROUGH Three participants had difficulty activating the headset's "passthrough" (real world view) mode. On the Quest 3, users can activate passthrough mode with a double-tap gesture on the side of the headset. However, these participants had difficulty tapping on the headset firmly enough to activate this feature.

5.3.5 *Discussion*

Our analysis found 16 themes of specific issues which we presented in three categories: issues with application motion requirements, issues with 3D application design, and hardware challenges. Importantly, these results are the product of directly observing participants actually experiencing VR. This validates and significantly extends general motor-related challenges elicited in prior work, with more emphasis on specific actionable issues informing a technical solution. Several issues were encountered when large movements of the head, body, or arms were required. Some work has proposed techniques to address some aspects of these and related issues, such as two-handed use [146], use in constrained spaces [127], and more accessible freehand gestures [124], but little work proposes a general method for overcoming these motor issues in a highly-customizable way.

We believe a more concise description and categorization of 3D spatial input could be useful for generalized understanding and identification, and to generate new VR accessibility techniques. Developing accessible input for VR demands a concise conceptual understanding of both the motion that is possible for a given user, as well as the motion required by a given application.

5.4 MOTIONBLOCKS

Inspired by previous work in 3D input remapping [67, 89, 104, 141], we propose *MotionBlocks*, a modular approach for creating customizable 3D input mappings from smaller input spaces or less-complex input devices. MotionBlocks allows users to define their comfortable range of motion and then remap that motion to fit the various larger spatial motions required by a VR application. A critical component in this approach is the use of *motion primitives*, simplified geometric representations of complex 3D movement. Analogous to geometric primitives in 3D graphics, motion primitives are simple geometric shapes and surfaces representing different types of spatial

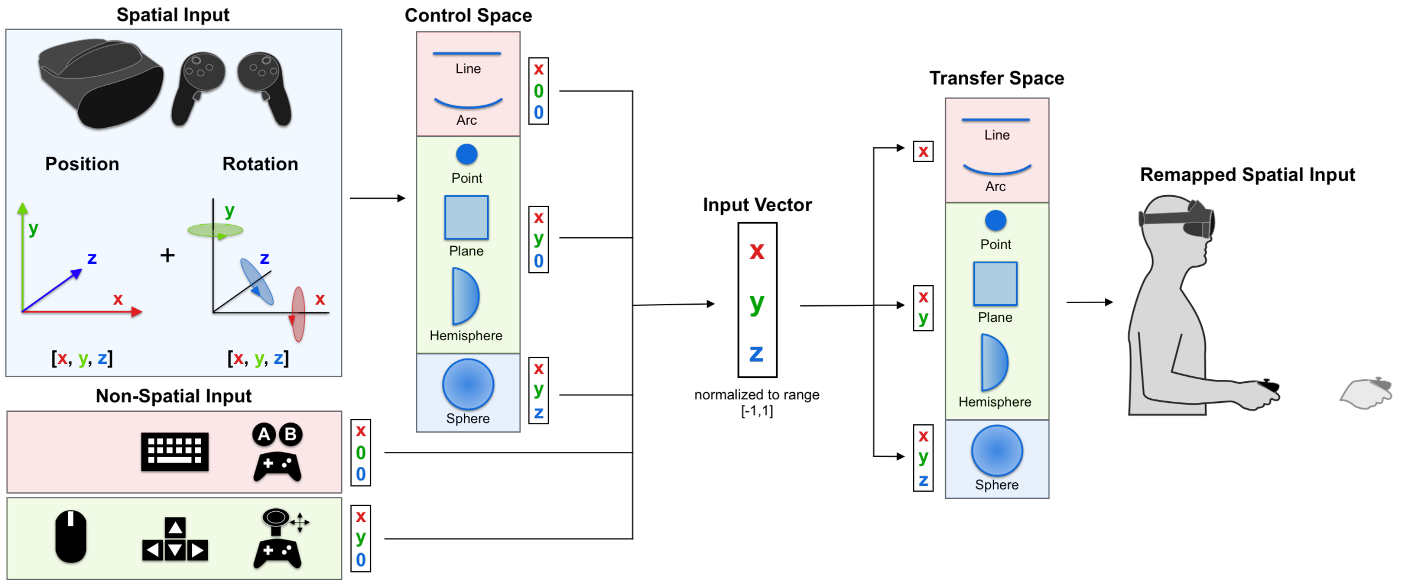


Figure 5.4: Our implementation of MotionBlocks for customizable VR input remapping. Physical motion along a control-space motion primitive is collapsed into a normalized input vector, which is then mapped to movement along a transfer-space motion primitive. The input vector normalization step also enables non-spatial 1-dimensional or 2-dimensional input devices to be remapped to transfer-space primitives.

input movements. These form a descriptive, concise, and generative language through which VR motion can be expressed and categorized (**RQ2**). While transfer functions and modifications of control-display gain have a long history within HCI (e.g. [30, 44, 104, 113, 141]), describing motion using geometric primitives is a new approach. Using motion primitives to describe spatial movement creates a simple common categorization of motor spaces across VR applications, enabling input space configurations that are easy to describe and easy to customize by developers and end users.

The MotionBlocks approach (Figure 5.4) maps 3D input between two egocentric coordinate spaces: *control space*, a comfortable range of motion as defined by the user's capabilities; and *transfer space*, the task-appropriate range of motion defined by the developer for the VR application. In typical VR usage (standing, with full use of both arms), control space and transfer space match. However, mobility limitations can cause mismatches between control and transfer space. MotionBlocks uses motion primitives to describe these spaces and their relationship, in order to create adjustable mappings between them.

5.4.1 *Motion Primitives (RQ2)*

We describe six motion primitives derived from VR body motions observed in our study (Figure 5.5), along with the dimensionality of each primitive's associated input space.

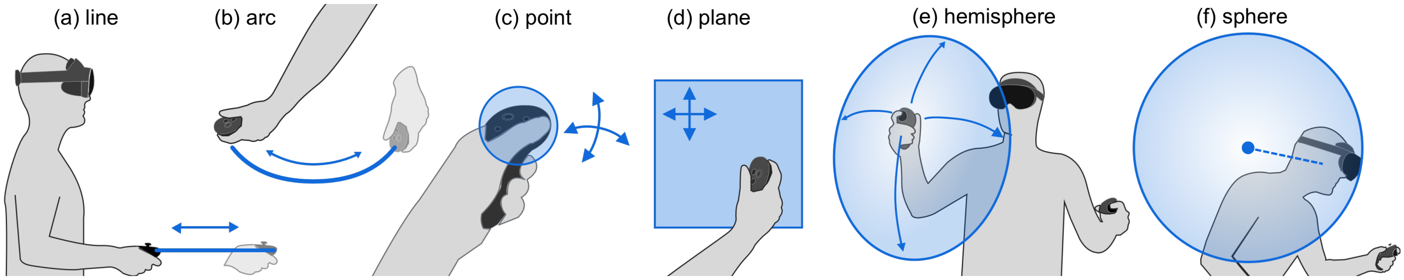


Figure 5.5: Motion primitives are geometric representations of potentially inaccessible movements in VR applications: (a) *Line* for 1D translation toward an object; (b) *Arc* for 1D rotation around a point with a given radius; (c) *Point* for 2D rotation without translation; (d) *Plane* for bounded 2D translation; (e) *Hemisphere* for 2D rotation around a point with a given radius; (f) *Sphere* for 3D translation.

Line: translation along a line segment defined by two 3D points: an origin and a target. A line represents 1D input calculated from the proportion of the line traversed from origin to target. For example, some participants had difficulty reaching directly toward a menu in *TheBlu*. This motion can be described with a *Line*.

Arc: translation along a curved line segment defined by a radius and arc length. An *Arc* represents 1D input: the proportion of the arc shape traversed from the start to the end. For example, P8 noted a potentially-challenging wide swing of the arm in *Walkabout Mini Golf*, which can be described as an *Arc* around the shoulder with an arc length of 90 degrees.

Point: 2D rotation around a fixed position. Traversing a point requires 2D input mapping to rotations in the X-axis (pitch) and Y-axis (yaw). For example, P1 preferred raycasting at menus using only rotation of the wrist, which can be captured by a *Point*.

Plane: 2D translation over a bounded flat surface. Traversing a plane requires 2D input, mapping to a local (X, Y) coordinate along the plane. Lateral upward reaches (without depth) can be captured using a *Plane*. Formative study participants who had trouble reaching upward (e.g. P8, P9) could still create planar motions by reaching forward and sideways along their lap or the surface of a table.

Hemisphere: rotational motion around a point and a given radius, resulting in motion along the surface of a spherical cap. A hemisphere is defined by a radius and two arc lengths mapping to total X-axis (pitch) and Y-axis (yaw) rotation. Traversing a hemisphere requires 2D input, mapping to rotation along the X and Y axes. For example, some participants playing *Beat Saber* (e.g. P9) had issues with the multi-directional large arm swings around the elbows. These motions can be captured by two *Hemisphere* primitives, one per hand.

Sphere: 3D motion from a given starting point. Traversing a sphere requires 3D input representing a 3D position relative to the centre of the sphere.

> For example, head movements while seated or standing (like ducking under hazards in *Beat Saber*) can be captured by a *Sphere*.

Each of these primitives, in addition to their geometric configurations above, has its own size, position, and rotation in 3D space for placement relative to the user. In our implementation, control- and transfer-space motion primitives for hands defined their position and rotation within the headset's coordinate space, moving relative to any headset motion.

5.4.2 *Input Vector*

A key benefit of motion primitives is their ability to simplify complex 3D motor interactions, reducing their required input dimensions [27, 79] in a way that is easier to configure than traditional transfer functions. Each motion primitive requires a specific number of dimensions to traverse. These dimensions are captured as input within the control-space primitive, and transformed into an input vector with a maximum of three dimensions. This input vector is then normalized to the range $[-1, 1]$ for each component, ensuring compatibility when mapping to any transfer-space primitive. Another advantage of using a normalized input vector is the ability to integrate non-spatial input devices. For instance, joysticks on VR controllers, as well as traditional game controllers, naturally provide input within the $[-1, 1]$ range. This accommodates input from any device capable of supplying a vector, such as controller joysticks (2D), mice (2D), game controller buttons (1D), or keyboards (1D).

5.4.2.1 *Mismatched Input Vectors*

The use of a normalized input vector as the central link between control space and transfer space allows any combination of primitives to be applied. However, this flexibility introduces the challenge of dimension mismatches when the input vector from the control-space primitive does not match the dimensional requirements of the transfer-space primitive.

When the input vector has more dimensions than required, the extra dimensions can be simply ignored. For instance, if the user's control space is a *Sphere* (3D) mapped to a transfer-space *Hemisphere* (2D), only the X and Y components from the input vector are used, disregarding the Z component. We make a simplifying assumption that motion primitive axes are assigned in a canonical intuitive way, with a consistent ordering of X, Y, Z relative to the dominant dimensions to traverse the primitive.

A more challenging issue arises when the transfer-space primitive requires more input dimensions than are provided by the control-space input vector. For example, how should the system map the 2D input vector from a control-space *Plane* to the 3D input required by a transfer-space *Sphere*? For our study, we made the simplifying assumption that the control-space dimensionality is equal to or greater than the transfer-space dimensionality. This matches the most common use case for MotionBlocks.

5.4.3 *Configuring Primitives*

Choosing the appropriate motion primitive for a given application requires a careful configuration process. In a real-world implementation, configuring control-space primitives would likely involve a user tool capable of measuring each individual's range of motion and automatically selecting the most suitable primitive. Likewise, a VR developer could provide a predefined set of transfer-space primitives that are dynamically activated or deactivated based on specific application logic. For the purpose of testing the general approach, in our study, we implemented a Unity environment where the facilitator creates control-space and transfer-space primitives as requested by the participant, manually moving and adjusting them relative to the participant's VR position. Participants could also move and resize primitives directly within the Unity application.

5.4.4 *System*

Our implementation of MotionBlocks depends on a custom-developed SteamVR input driver, similar to the approach taken by OpenVR Input Emulator[1]. This driver uses DLL injection of the SteamVR controller driver via MinHook[2] to obtain the physical positions, rotations, and button inputs of the standard VR controllers. This information is sent via named pipe to a Unity application on the same system which contains the control-space and transfer-space motion primitives, and performs the input vector remapping between them. Our implementation optionally applied the 1€ filter [29] to the input vector to smooth shaky motions. The remapped controller positions and rotations are sent back to the SteamVR input driver, and subsequently provided to SteamVR via its standard API methods. This driver-level remapping allows MotionBlocks to be applied to system-level spatial input, enabling accessible remappings within any SteamVR game.

5.5 MOTIONBLOCKS STUDY

To test the MotionBlocks approach and answer **RQ3**, we conducted a user study with the same consumer VR applications, but this time using an implementation of the MotionBlocks approach. We recruited a subset of participants from the formative study, 5 months after formative study completion.

5.5.1 *Participants*

We re-recruited 8 participants from the formative study (all but P3 and P4). Their mobility limitations can be found in Table 5.1. Sessions took place at a local disability outreach foundation, or in the participants' homes. The protocol was approved by an ethics review board.

---

[1] `https://github.com/matzman666/OpenVR-InputEmulator`

[2] `https://github.com/TsudaKageyu/minhook`

5.5.2 *Protocol*

Each session lasted one hour, and participants received another $50 Amazon gift card upon completion. This study used the same PC and VR hardware as the formative study.

5.5.2.1 *Task*

Following informed consent and a brief re-introduction to the VR hardware, participants donned the headset and entered a simplified virtual environment for control space calibration. During this process, participants collaborated with the facilitator to identify the control-space primitive that best aligned with their natural range of motion. Once the configuration was complete, participants were given time to familiarize themselves with how their smaller, more comfortable movements translated into larger inputs within the VR environment. Participants then used their choice of three of the five applications, with facilitators enabling and disabling the remapping system as necessary for UI navigation. During testing sessions, participants would also provide feedback with regard to both the control-space and transfer-space primitives, with the facilitators able to make real-time adjustments to remapping configurations as necessary. Participants could swap out control- and transfer-space primitives for alternate primitives if desired. To facilitate direct comparison, participants completed rounds of gameplay both with the remapping system applied and in a baseline condition without remapping. Participants could request that the facilitators enable and disable the MotionBlocks remapping at any point during the study.

At the end of the study, participants completed the first half of the NASA-TLX questionnaire [59] to assess perceived workload. This was done for both the MotionBlocks-enabled condition (MOTIONBLOCKS) and the baseline condition (BASELINE), with participants rating each dimension on a scale from 1 to 7. The six dimensions of the NASA-TLX (*Mental*, *Physical*, *Temporal*, *Performance*, *Effort*, and *Frustration*) were treated as independent variables in our analysis.

5.5.2.2 *Applications and Primitives*

A core element of our approach involves applying motion primitives within the application's transfer space, which defines the types of movements the application is designed to accommodate. In this study, facilitators took on the role of application designers, pre-defining a set of transfer-space motion primitives tailored to the primary motion requirements of each application. This study used the same VR applications as the formative study (Table 5.2): *TheBlu*, *Tilt Brush*, *Walkabout Mini Golf*, *Space Pirate Trainer*, and *Beat Saber*.

TheBlu is an underwater experience focused on locomotion within the virtual environment. However, participants reported enjoying reaching toward objects to enhance immersion. To support this, we implemented *Line* primitives for the hands, designed to accommodate large reaching motions. Each

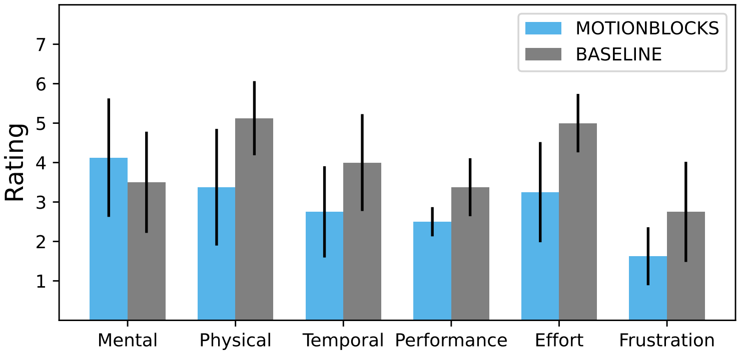


Figure 5.6: Results from the NASA-TLX questions for MOTIONBLOCKS and BASELINE (no input remapping). Error bars represent 95% CI.

Line primitive started 10cm in front of the user at shoulder height, extending 1m forward relative to their position.

In *Tilt Brush*, painting requires large arm movements to create 3D brushstrokes, and preserving the direction of user movement between input and transfer spaces is crucial. To achieve this, we implemented *Sphere* primitives for both the hands and head, allowing for easier examination of 3D drawings. These 1m-radius *Sphere* primitives tracked the hands and head, amplifying motion based on the distance from the activation point when triggered.

Walkabout Mini Golf involves downward-facing arm swings along a single axis, close to the body, to execute putting motions. For this, we implemented *Arc* primitives for both the left and right hands. These *Arc* primitives were positioned 20cm in front of the body, with the bottom of the arc reaching the floor. This setup allowed for one-dimensional input to generate lateral, arcing putt motions across the body in both directions.

In *Space Pirate Trainer*, players must aim precisely while also making large body movements to dodge incoming enemy fire. To accommodate these requirements, we implemented *Sphere* primitives for both the head and hands, functioning similarly to their use in *Tilt Brush*.

In *Beat Saber*, players must make large arm swings, both vertically and horizontally, around the elbow. To support these motions, we implemented *Hemisphere* primitives with a 0.5m radius, positioned slightly to the left and right in front of the user. A 1m-radius *Sphere* primitive for the head was added to facilitate easier ducking and dodging of in-game hazards.

During testing sessions, participants were encouraged to provide feedback as to the size or shape of the transfer-space primitives as needed, with facilitators adjusting these parameters in real time. Any modifications made during the study were documented, particularly if they deviated from the pre-specified configurations.

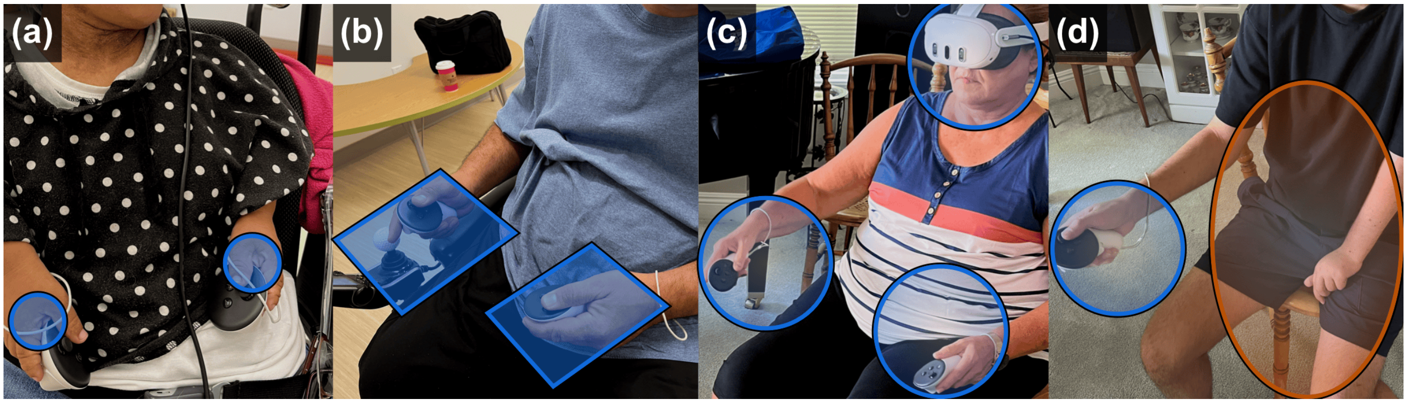


Figure 5.7: Examples of control-space motion primitive configurations: (a) P1 using *Point* primitives tracking 2D wrist rotation; (b) P6 using *Plane* primitives tracking 2D translation forward and sideways across their lap; (c) P9 using *Sphere* primitives tracking smaller, more comfortable 3D movements of the head and hands; (d) P10 using a *Sphere* tracking motion in their right hand, mapped to a transfer-space primitive as normal, but additionally the joystick in the right controller provides input to a transfer-space *Hemisphere* (orange) for the left hand, enabling bimanual input.

5.5.3 *Results (RQ3)*

We discuss the MotionBlocks configurations created by participants, as well as the effects of the MotionBlocks approach on VR application usability. Experimenters completed the same thematic analysis process as in the formative study. After the NASA-TLX results, we present user feedback results organized by discovered themes. Cohen's Kappa (κ) is provided as a measure of inter-rater agreement ($\kappa \geq 0.6$ is substantial agreement).

5.5.3.1 *NASA-TLX*

Configurable input remapping reduced perceived workload across several categories (Figure 5.6). A pairwise Wilcoxon signed-rank test showed that input remapping with MOTIONBLOCKS significantly reduced *Physical* workload ($W = 2.5$, $p < .05$), *Temporal* workload ($W = 0.0$, $p < .05$), as well as perceived *Effort* ($W = 2.5$, $p < .05$) relative to BASELINE. There were no significant differences between ratings for *Mental* workload, *Performance*, or *Frustration*.

5.5.3.2 *Easier Hand and Arm Motions*

Participants configured their control-space motion primitives based on their ranges of motion and comfort ($\kappa = 1.0$). For example, P1 initially started with an *Arc* configuration to track their motion for playing *Walkabout Mini Golf*, to match small 1D wrist swinging motions they were comfortable making. In other games, they switched to a *Point* primitive specifically tracking 2D wrist rotation for a wider but still comfortable range of motion (Figure 5.7a). Other participants configured their control-space motion primitives based on what they believed to be easiest for their current body position. For example, P6 used a *Plane* primitive aligned with his lap to provide input to play *TheBlu* while seated (Figure 5.7b). Other participants who could provide smaller 3-dimensional motions chose to use *Sphere* primitives to amplify their comfortable motions (Figure 5.7c).

Across control-space and transfer-space motion primitives, participants often preferred configurations that utilized multiple dimensions of motion. While 1-dimensional motion primitives like *Line* and *Arc* provided a way for very simple motions to produce interactions for slower-paced applications, our participants mostly preferred providing two-dimensional or three-dimensional input which MotionBlocks then upscaled. As a result, the most common control-space primitives for the hands were *Sphere* (3-dimensional), *Plane* (2-dimensional), and *Point* (2-dimensional), and the most common transfer-space primitives were *Sphere* (3-dimensional) for body-accurate upscaled 3D motion, *Hemisphere* (2-dimensional) for large arcing motions, and *Point* (2-dimensional) for fine aiming motions.

Participants enjoyed the additional mobility that MotionBlocks provided, as well as the configurability of those motions. For example, P1's reduced range of arm motion prompted her to create a configuration that would track her wrist rotations using a *Point* primitive, and map those motions to a *Hemisphere* primitive in-game. Using a control-space *Point* primitive allowed her to achieve comfortable interactions without any conflicts with her power chair joystick. P1 elaborates on her experience playing *Beat Saber* using this configuration: *"this is much easier since it's using my wrist rotation. I can keep my hands closer to my body and still make big swipes"*. When exploring *TheBlu* using a control-space *Plane* mapped to a transfer-space *Line*, P6 was surprised: *"I don't even have to move my chair [to interact]"*. When these primitives were disabled, P6 noted that *"I'd definitely prefer having more range"*. Participants noted that using this motion remapping enabled longer play sessions: *"I feel like I get low on stamina when playing really involved games and I didn't this time"* [P6].

5.5.3.3 *Head Motion and Leaning*

Participants who played the VR games seated commonly selected a smaller control-space *Sphere* primitive for the head, mapped to a larger concentric transfer-space *Sphere*, ultimately creating a simple bounded motion amplification [141] that made large ducking and dodging movements safer and easier ($\kappa = 0.6$). Participants tried other control-space motion primitives (namely *Plane* and *Point*), but found concentric *Sphere* primitives to be the most comfortable for extended play sessions. Participants who enabled head position remapping commonly did so due to factors like balance issues (e.g. P8, P10) or restriction in ability to lean (e.g. P1). Because the control-space and transfer-space *Spheres* were concentric, participants' neutral seated head position would look and feel normal, only visibly remapping when making leaning or dodging motions.

Head position remapping enabled a wider range of motion and actions. For example, P5 felt he had *"so much more range"* while making fine leaning motions to look closely at drawings in *Tilt Brush*. Similarly, *Beat Saber* and *Space Pirate Trainer* required significant body motion to dodge in-game hazards, which posed significant challenges for people who could not make large leaning or crouching motions. Using our approach allowed them to evade more successfully: *"[head amplification] makes it so much easier to dodge"* [P2].

P7 describes the benefits of this approach: *"[without it,] I'm physically leaning a lot more because I have to. Games that require leaning movements really don't consider how much harder it is to do in a chair or with impairments – if [extended leaning] was a setting in my Quest 2 today I would turn that on immediately"*. P9, who experiences balance issues and fine motor instability due to Parkinson's, found that playing *Space Pirate Trainer* with *Sphere* remapping *"gave me the confidence to think that I could actually accomplish more. Hitting more targets, avoid incoming fire, it just seemed to add to what I was doing. It didn't do things for me, it just expanded my ability"*.

5.5.3.4 *Alternate Input Devices*

When necessary, participants prioritized configurations that would enable alternate input devices ($\kappa = 1.0$). Participants 5 and 10, who could only provide input using one hand, compensated by assigning the joystick on the VR controller in their fully mobile hand to a *Hemisphere* primitive for the other hand (Figure 5.7d). This meant that their final control scheme was a combination of input methods on the same physical device: physical motion to control one hand, and joystick input in that same hand to control the VR representation of the other. The configuration process for these participants involved working through several iterations of symmetric, in-phase [146] techniques for controlling both VR hands with the motion of one physical hand, but ultimately participants preferred a combination of motion and joystick. P9 also initially experimented with using the joysticks on an Xbox controller to control both VR hands, noting that *"if I took the time to learn this it would be the easiest of all, but doing physical motions is easier to start"*.

5.5.3.5 *Game-Specific Adjustments*

Participants varied their configurations based on game mechanics ($\kappa = 0.6$). Many participants chose *Sphere* primitives for both the head and hands to maintain an accurate spatial relationship between them, particularly in games that required more precise motions. However, for slower-paced applications, participants more often chose more aggressive motion remapping which needed less physical motion. For example, while P6 used a *Line* primitive for forward reaches in the slower-paced exploration in *TheBlu*, he used *Hemisphere* primitives when making large swiping motions in *Beat Saber* and *Sphere* primitives for precise strokes in *Walkabout Mini Golf*.

5.5.3.6 *Learning New Configurations*

Participants noted that while ultimately motion remapping was helpful, learning the remapped motions required some time ($\kappa = 0.6$). As an example, P9, using *Point* primitives mapped to *Hemisphere* primitives for *Beat Saber*, remarked that *"it's a bit of a learning process but I'm getting the hang of it"*. P2, playing *Space Pirate Trainer* using both control-space and transfer-space *Sphere* primitives on the head and hands, noted that *"there were some parts that were tough to learn but overall it felt easier"*. P10 noted that overlapping primitives *"get confusing sometimes; it's more confusing in games like Beat Saber since your hands can get crossed"*.

5.5.4 *Discussion*

The formative study demonstrated that people with differences in motor capability experience difficulties using their own motion range to complete actions in VR applications. To address this, this study's primary goal was to answer **RQ3** by evaluating MotionBlocks as a method to make VR hand and head motion easier. Our results show that a modular, versatile way to represent 3D input can make VR motions easier, safer, and in general, more accessible. We discuss the impact of this motion-remapping approach within the context of the themes discovered in the formative study.

A positive impact was particularly evident when looking at the spatial input issues that we identified in the formative study. Hand remapping using MotionBlocks allowed for increased reach and also provided an alternative to two-handed input for one-handed use. Participants with shakiness and tremors noticed fewer issues, especially considering that input remapped using our technique could also be filtered. Our technique for remapping head movement resolved many reported issues with bending and crouching, movement speed, and balance. A secondary effect of the remapping techniques was that because some participants felt more capable of achieving effective input while seated, the balance issues associated with standing were less of a concern in the second study. Participants found that making fine locomotion adjustments within the game was easier when remapping techniques were enabled. Support for additional hardware outside of VR controllers can also introduce better compatibility for copilots, as was required by some formative study participants.

Our approach remedied some issues with application design as well. Participants in the formative study noted issues with game difficulty. P9 explained in the results that he felt more confident avoiding hazards and focusing on the task at hand within the game. Similarly, participants in the formative study reacted positively to game mechanics that reduced the amount of motion that was necessary. As such, it is understandable that a game-agnostic technique for reducing motion requirements was regarded positively.

Similarly, our approach addressed several of the issues with VR hardware elicited in our study. Because users no longer had to make large reaches away from the body, issues with maintaining a firm grip on the controllers were less prominent. Issues with wheelchair conflict were less prominent since enabling VR actions with smaller motions made striking the joystick of a power chair less of a concern.

5.6 GENERAL DISCUSSION

Although many of our findings are situated within the context of VR games, our findings are relevant to spatial input as a whole. Our formative study elicited several areas within VR spatial input and application design that pose accessibility issues for people with limited mobility (**RQ1**). We represent these interactions with a concise design language which reduces interaction down to simple geometric primitives (**RQ2**). This design language underpins the *MotionBlocks* approach for creating highly customizable accessible input remappings. Our study shows that input remapping using this approach provides an effective way to address several of the elicited VR accessibility issues (**RQ3**).

5.6.1 *Design Recommendations*

5.6.1.1 *Design for Differences*

In the set-up process for the study, we played the role of application designer by pre-specifying an initial set of transfer-space motion primitives for each application. Surprisingly, we quickly found that users had unique preferences for how their virtual hand should move in relation to their real hand or the input they otherwise provide. For example, many users of *Walkabout Mini Golf* preferred to have the movement direction of their hand preserved by using transfer-space *Sphere* primitives instead of the *Arc* primitives that we pre-specified. Applications built using motion primitives should provide options to quickly and gracefully handle different preferences for remapped 3D input, and make the process of tuning such a remapping system as simple and accessible as possible.

5.6.1.2 *Understand the Comfort-Precision Trade-off*

Participants in the study often preferred different motion remapping configurations depending on the pace of the application. Slower and more exploratory applications like *TheBlu* prompted participants to choose more comfortable configurations (e.g. P6's lap-aligned *Plane* primitives) at the cost of precision. Similarly, participants who played *Space Pirate Trainer* often preferred configurations that would allow for an increased amount of precision, even if that meant that they had to physically move more than in other applications. Designers of motion primitive configurations should consider the trade-offs between comfort and precision in deciding the recommended motion primitives for an application.

5.6.2 *Limitations*

5.6.2.1 *Manual Primitive Selection*

Our study relied on active facilitator intervention, specifically for enabling, disabling, or configuring motion primitives. As a result, the reported usability and cognitive load might differ from implementations where motion primitives are activated or deactivated automatically, as well as from implementations where the user is responsible for triggering them. Future work should focus on implementations that de-emphasize facilitator intervention.

5.6.2.2 *Participant and Application Diversity*

Our motion primitives are based on a formative study with 10 participants. Naturally, the wide variety of disabilities and levels of mobility could mean that other motion primitives might be more appropriate for other users or applications. Disability is a wide spectrum, so our results cannot describe all individual circumstances and VR input configurations. While our contextual inquiry and study probed for as many circumstances and use cases as reasonably possible, we present this work as one part of a deeper design investigation. Likewise, selecting alternate games or hardware might prompt alternate input accessibility issues.

5.6.2.3 *Full-Body Input*

Our study focuses on hand and head movements but does not address other potentially relevant interaction types, such as full-body tracking or leg-based movements, which could be important for certain VR experiences like exercise or physical therapy applications. This may limit the generalizability of the findings for applications requiring full-body input.

5.6.3 *Future Work*

Our work is part of a continuing effort to make spatial computing accessible for people with any level of mobility.

5.6.3.1 *Native Implementation*

The implementation in our study relied on a DLL injection of a current SteamVR Input driver, which could lead to instability or reduced functionality as the SteamVR Input API evolves. We propose our approach in the hopes that makers of VR headsets and software adopt this functionality natively within their products.

5.6.3.2 *Community Configurations*

Our implementation relied on a set of predefined transfer-space motion primitives for each application, with room to adjust these configurations as necessary. This configuration step imposes a greater setup time for users with disabilities to be able to use a given application – an issue which was made explicit in the formative study as a barrier to VR usage. Real-life implementations of this system could use a community-based approach, where users with varying mobility share the motion configurations that work for them. This could reduce setup time or inspire more accessible configurations.

5.6.3.3 *Context-Sensitive Configurations*

The implementation in our study relied on manually switching the motion primitive on and off for motion remapping. However, to streamline usage and respond to changes in a given application, future implementations could use context cues like body position or proximity to objects [60, 138] as triggers for enabling a variety of motion primitives.

5.6.3.4 *Classifying Configurations*

Our implementation depends on manually calibrating both the control-space motion primitives for the user's range of motion and the transfer-space motion primitives used by the application. Future work could improve this process by intelligently selecting appropriate motion primitives depending on the user's natural motion. Such implementations could use RANSAC [45] or Iterative Closest Point [17] to intelligently match user movement point clouds to appropriate motion primitives, or even use a neural network approach like PointNet [105] to classify point clouds into their most appropriate primitive.

A further extension of this work would be to break away from geometric descriptions entirely, focusing on entirely custom motion range remapping. For example, future implementations could construct a manifold from the user's movement points over a set amount of time, then determine a bijective mapping between the user's motion manifold and a manifold constructed from a given application's typical motion data. This dynamic, irregular remapping could present a more flexible and accurate solution for dynamic input remapping, but the effect on user behaviour remains to be explored.

5.6.3.5 *Missing Input Axes*

Because motion primitives might differ in input complexity between control space and transfer space, some configurations might incur input vector mismatches. Our implementation made the simplifying assumption that the input spaces matched, subsequently discarding or imputing zero for extraneous or missing values respectively. Future work should examine ways to dynamically infer missing values based on application context. For example, when matching a control-space *Plane* to a transfer-space *Sphere*, the missing third dimension of input (typically Z-axis depth) could be inferred based on nearby objects in the scene.

5.7 CONCLUSION

As spatial interfaces embed themselves further into the mainstream, a lack of consideration for motor accessibility entrenches recreational, social, and economic barriers for people with mobility limitations. We present the results of a formative contextual inquiry study, examining the VR accessibility barriers experienced by 10 people with limited mobility. The results motivate the concept of *motion primitives*, a method to describe complex body motion using simpler input dimensions. Motion primitives enable a concise design language for identifying and categorizing VR movements, inspiring alternative spatial interaction designs and techniques. We use motion primitives to design and evaluate *MotionBlocks*, an approach for creating customizable geometric input remappings that enable complex 3D input using smaller ranges of motion or simpler input devices. This approach addressed several of the issues found in the formative study, reducing the effort necessary for meaningful and effective VR input.

VR provides an opportunity for people to experience environments and social interactions completely outside their norm. As P7 describes: *"it's an equalizer, it opens up a seemingly mundane thing [like mini golf] and lets you say 'oh, I can do that now"'*. *MotionBlocks* provides a highly configurable way to make spatial interactions easier, safer, and more inclusive for all levels of mobility. Our work is a specific step toward a more mobility-focused view of accessible spatial input, echoing P7: *"if you're going to expect a range of motion, help everyone get there"*.

6 DISCUSSION

The previous three chapters discussed ways to understand, design, and evaluate accessibility solutions for a variety of VR input scenarios. In this chapter, we discuss limitations in our methods and associated future work, then describe a possible design for a combined system that could integrate all insights from our work.

6.1 LIMITATIONS

Motor disability is a wide spectrum, and our results may not apply to people with all forms of disability. While we focused our efforts on making a solution as generalizable as possible, specific scenarios might exist that our explorations might not cover. Future work should examine these scenarios and expand our accessibility findings, or otherwise provide more specific adaptations of our general methodology.

The W3C Working Group XR Accessibility User Requirements[1] state that user interfaces should support a variety of input modalities, often simultaneously. Our explorations focus on motor input methods, but other input methodologies should also be considered in future work. For example, how could our motion primitive approach be combined with speech input? How might eye-tracking support change a system's definition of user context? Future work should examine a wider set of input modalities as defined by W3C.

6.2 FUTURE WORK

This work is an initial step in a much wider area of study. Developing accessible systems and techniques for spatial input, as well as making them feasible for real-world implementation, requires more research than possible in a single PhD. We discuss future research directions and implementations based on our findings.

6.2.1 *Evaluating VR UI Motor Accessibility*

Our investigation focuses primarily on the usability implications of specific interactions within software and hardware for VR. However, less work examines the accessibility implications of the most common VR interfaces. While some guidelines[2] exist for general VR user accessibility, and previous work examines general usability and performance without explicit focus on ac-

---

[1] `https://www.w3.org/TR/xaur/` – Accessed Oct 14, 2024.

[2] See footnote 1.

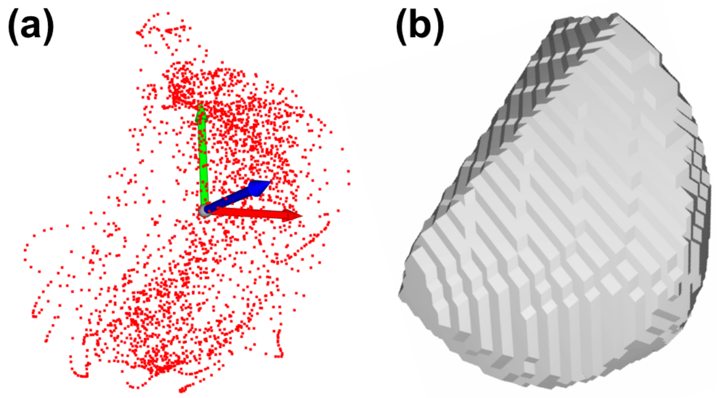


Figure 6.1: Visualizations of user movement of the right hand during one gameplay round of *Beat Saber*: (a) 3D point cloud; (b) convex hull.

cessibility [140], there remains a gap in understanding how common spatial interface design affects accessibility, evaluated in-situ with real people with mobility limitations. These evaluations could use both the formative study methodology and the general accessibility testing methodology we introduce in Chapters 4 and 5, respectively. Some potential evaluations could include general usability and performance, how well a given VR menu adheres to W3C guidelines, or a variety of other accessibility metrics [26, 148].

6.2.2 *Hardware Accommodations for VR Situational Impairments*

Chapter 4 investigates how user interfaces can use context to better address situational impairments. However, solving for accessibility solutions through software might be insufficient for people whose main accessibility issues with VR concern aspects of its physical hardware [92]. In such a situation, more involved physical accommodations might be necessary. Previous hardware approaches have augmented commodity articulated desktop monitor mounts with custom hardware [38], to mount VR heasdets to a desk, enabling developers to quickly switch between desktop and VR usage without putting on a headset. Based on the findings in this thesis, there exists an opportunity for researchers to examine other physical accommodations which might address similar situational impairments. For example, future work could examine additional ways to instrument VR hardware to reduce the amount of physical manipulation necessary, like mounting the headset to an instrumented chair or desk. The solutions might make the headset less mobile but better suited for specific workflows or accessibility needs.

6.2.3 *Classifying Context and Motion Ranges*

The discussion of future work in Chapter 5 mentions using techniques like RANSAC or Iterative Closest Point to automatically assign and configure motion primitives based on a user's range of motion. We can expand this idea

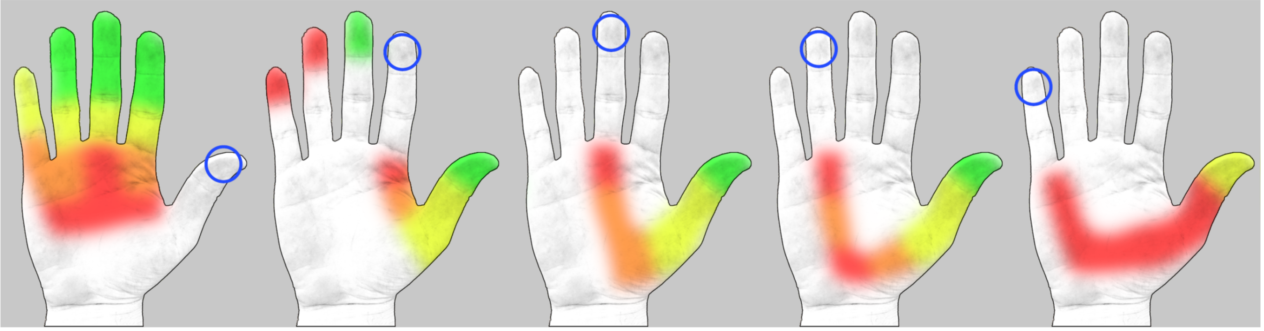


Figure 6.2: (From Dewitz, Steinicke, and Geiger [39]) Heatmaps generated from ratings of comfort for different active parts (blue circle). Green (4): very easy to reach, yellow (3): slightly less comfortable, orange (2): major difficulties, red (0,1): not reliably reachable, white: not considered.

by considering the outcomes of all three previous chapters. We can visualize typical motion within VR applications using point clouds (Figure 6.1a), which can then be run through mesh generation algorithms like Marching Cubes [77] to create essentially a manifold of user movement (Figure 6.1b). In addition to these computer vision techniques which utilize large point clouds, it might be possible to train a classifier for motion primitives based on simple feature engineering. Instead of point clouds, a classifier could be trained on simpler features like: proximity to certain furniture like a desk or chair; input device configuration (including individual devices as well as combinations [79, 137]); or even use some form of scene understanding [7] to determine the current room in which the VR usage is taking place. These classifications could be used to dynamically reassign motion ranges, provide more timely and useful UI responses to user context changes, or even signal to an application that the user might need additional accessibility accommodation.

6.2.4 *Remapping Pinch Gestures*

The hand tracking employed by modern headsets like the Quest 3 or Vision Pro often relies on a pinch gesture (touching the tips of one's thumb and index finger) to select items within the virtual environment. This fine motion might be difficult for people with conditions which reduce motor precision like tremors or arthritis. Building from the previous three chapters, future research could explore software augmentations that might make these pinch gestures easier.

One possible exploration of pinch gestures might include understanding the individual comfort and usability of completing the pinch action by touching together different finger joints or using different fingers entirely. Within the realm of contextual constraints (i.e., situational impairments), previous work explores how hand posture and hand location can affect usability, suggesting various contextual microgestures along the middle, ring, and pinky fingers [65]. Likewise, other work explores the relative usability of one-handed interactions across various positions on the hand (Figure 6.2),

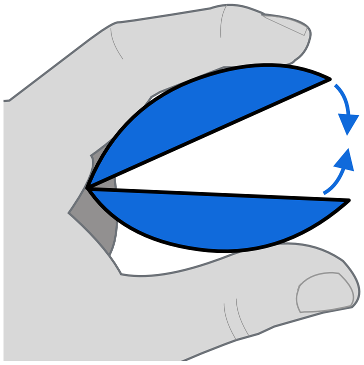


Figure 6.3: A proposed 'castanet' hardware augmentation to make pinch gestures easier. Closing the castanet to trigger a pinch gesture could require a customizable amount of tension, making pinch gestures more intentional and precise for people with involuntary tremors.

demonstrating the feasibility of tap gestures outside of the traditional thumb and index tip configuration [39]. Future work could extend this specifically for spatial input accessibility, examining the relative performance and accessibility of pinch gestures for targeting and selection along combinations of joints other than the thumb and index finger tips. For example, consider a user like P8 in Chapter 5, whose tremors made fine selection difficult. If his middle finger experiences fewer tremors than his index, could an alternative solution simply be to remap the traditional pinch gesture such that the thumb touches the middle fingertip instead of the index? Alternatively, a system could trigger a pinch gesture when the user's thumb tip touches the connection between the proximal and middle phalanges of the index finger (the middle of the three knuckles on the finger). Modern hand-tracking systems like those by Meta or Apple provide joint positions and rotations to developers, making arbitrary fingertip-to-joint pinch gestures feasible to explore in future work.

6.2.5 *Hardware Augmentations for Non-Controller Input*

Chapter 3 describes the variety of ways that users with mobility limitations augment their input devices to make interacting with them more accessible. As headsets like the Vision Pro move away from traditional controllers and input devices, *adapted device* modifications like adjusting the height of a joystick become less feasible to implement. Keeping with the spirit of adapting input devices, what hardware-based physical adaptations for hands and tracked hand input could make motions like pinch or grip gestures more accessible? For example, let us revisit P8 in Chapter 5, whose tremors could cause pinch gesture difficulties. Augmenting his thumb and index finger with a device that provides some amount of resistance or tension could provide a form of physical filtering for these tremors, or even become a way to provide low-cost haptic feedback. One possible physical augmentation could resemble a castanet (Figure 6.3), a small percussive instrument played by pinching the index and thumb together. Increasing the tension necessary to pinch the castanet closed, and instrumenting it to provide simple pinch gesture input to a VR system, could allow pinch gestures to be completed more easily and accessibly.

6.3 WALK-THROUGH: MOTOR ACCESSIBILITY CALIBRATION

The three research projects in this thesis form the foundation of a larger accessibility solution. Chapter 3 shows how physical input devices can be reconfigured and combined into more accessible setups. Chapter 4 demonstrates that interfaces can leverage context to overcome certain situational impairments, and Chapter 5 demonstrates that customized input remapping can address other situational or permanent impairments. The input techniques developed in this thesis were designed with the ideal end goal of native implementation within consumer VR headset operating systems. To demonstrate how a solution like ours could be implemented and integrated into current systems, we now discuss a user walk-through of an imagined combined accessibility solution.

To start, the user dons the headset. Of course, this assumes that the user can put the physical headset on without issues, which may not necessarily be the case [92]. The headset begins an initial setup process.

MOTION CALIBRATION At this point, the headset would prompt the user to move their hands as much as they are physically capable of doing. Using some kind of spatial tracking like built-in hand tracking, the headset would then measure the total motion range that the user prefers. The headset could then analyze the resulting point cloud, calculating descriptive statistics such as speed, maximum motion distance, or detection of tremors for adjusting input filtering. Importantly, such a point cloud could also be used to classify the user’s motion range into control-space motion primitives.

INPUT DEVICE CALIBRATION After a baseline measurement of the user’s range of motion, the headset could conduct some form of input calibration. If the user’s preference for input devices is the typical VR controller packaged with the system, this process is skipped. However, if the user prefers to combine or otherwise reconfigure an alternate input device, the system could provide initial suggestions for control mappings through generic means, namely treating each input as a class defined by dimensionality [79]. For example, a trackpad (2-dimensional continuous input) and a button switch (1-dimensional discrete input) could be treated in a combined “$2D + 1$” class [137], containing 2 continuous dimensions plus 1 discrete dimension. The system could modify its UI in response to the dimensionality of input available, like enlarging targets or rearranging items dynamically.

CONTEXTUAL CALIBRATION After baseline physical calibration, using either manual indication or some form of scene understanding [7], the user could specify proximity- or input-based context triggers that might prompt different amounts of motion compensation on the part of the system. For example, if the user is lying on their bed, the system might adapt by turning all UI such that input planes are parallel with the ceiling. Alternatively, if the user is seated at a table, the system might recognize hand motion across the surface of the table as a secondary form of two-dimensional continuous input, similar to the *controller-mouse* input technique described in Chapter 4.

APPLICATION RESPONSE Naturally, this multi-step calibration process could lead to a wide variance of input types to which an application must respond. Ideally, applications following the principles of ability-based design should be designed to anticipate and provide useful adaptations which reduce the burden on the user to manually adjust. However, as a fallback, developers could specify its necessary range of motion as a set of transfer-space motion primitives, to which the control-space primitives determined earlier could be mapped. The transfer space primitives might vary depending on application context, just as control-space primitives might vary based on user context.

Improving the motor accessibility of spatial input involves addressing a variety of design and research questions which are yet to be explored. We hope that the explorations and insights in this thesis can serve as the foundation for a variety of deeper future investigations.

7

CONCLUSION

This thesis explores ways to make spatial input more accessible for people with a variety of temporary and permanent mobility impairments. This was conducted through three primary projects: understanding the input devices and usage scenarios that people with mobility limitations experience; understanding and accommodating people with a variety of situational impairments by cleverly making use of context; and understanding and accommodating spatial input for people with a variety of permanent impairments. These projects are the result of a deliberate effort to pursue our high-level research objective:

Investigate how input device combinations, mediated by context, can make spatial interfaces more accessible in a general and customizable way.

In this final chapter, we summarize the work in the thesis by revisiting the proposed research questions and conclude with some final thoughts.

SUMMARY

We investigate ways that a system can utilize a wider set of input and context scenarios to make spatial interfaces more adaptable and accessible. This work is one step in the process of operationalizing ability-based design [144], with the overall goal of establishing AR/VR spatial input as a mature and accessible tool.

In Chapter 3, we used qualitative methods to explore how people with mobility limitations remap, reconfigure, and combine input devices to overcome accessibility barriers within traditional computing.

(a) *How do real-life users with limited mobility alter or combine input devices to be more accessible?*

A survey of people with mobility limitations shows that input devices and configurations vary as widely as people's own abilities. It is impossible to anticipate and design for every possible individual device and preference, so it is critical to design solutions for input *categories* instead of individual *devices*.

(b) *How does application or usage context affect these device or configuration choices?*

Semi-structured interviews demonstrate that individual users remap, reconfigure, combine and even alter their input devices in order to create custom accessibility solutions that are best tailored to their own input. Additionally, discovering new input configurations is a challenge so providing simple configuration and easy discovery is of utmost importance.

(c) *How do users discover accessible setups, and what does this process reveal?*

Many accessible input setups are found using social media. Social media displays an incredibly wide array of configurations, and the availability and visibility of accessible configurations directly impacts user engagement with these platforms.

This project provides overall motivation: designing more accessible spatial input is not as simple as individual button remapping schemes. Designing for context and a variety of input categories is the best way to ensure that an accessibility solution is robust to the wide variety of disability.

In Chapter 4, we expanded our investigation to include situational impairments. Specifically, we focus on the situational impairment of transitioning between VR and desktop. We use this specific situational impairment as a way to demonstrate how context can be utilized as an alternate stream of input to make interfaces more responsive to a wide variety of input situations.

(d) *What factors of VR-enabled workflows cause situational impairments or discomfort?*

A survey and formative study of people with workflows involving both desktop and VR usage shows that switching between VR devices and desktop interfaces introduces both mental and physical roadblocks. Importantly, we found that these specific switches are quick and temporary "peeks", from one interface to the other, then back.

(e) *How can we make use of application context to resolve these usability issues?*

We propose *SwitchSpace*, a collection of UI response techniques that utilize context to better accommodate these quick and temporary “peeks” between VR and desktop. A user study that evaluated SwitchSpace with a sample cross-reality task demonstrated that clever responses to context can provide an effective and comfortable remedy to this class of situational impairment.

This project demonstrates that context can be used as a secondary stream of input when primary input may be uncomfortable or insufficient for the task at hand.

In Chapter 5, we focus on solving the primary problem in our scope: providing a simple, customizable method to increase VR accessibility which is compatible with a wide variety of devices and device categories.

(f) *What kinds of motion-related interactions in VR cause accessibility issues?*

An initial formative study had users with mobility limitations try five popular VR applications, revealing a wide variety of accessibility issues within software and hardware, primarily concerning VR’s motion requirements relative to their own ability.

(g) *Can customizable input remappings resolve these accessibility issues?*

We use the results and accessibility issues from (f) to design *MotionBlocks*, a customizable geometric input remapping system which allows users to create custom spatial input remappings based on their own range of motion. We brought the same participants back to try the same games but with their custom *MotionBlocks* remappings applied, and found immediate improvements and positive results concerning usability and physical load.

FINAL WORD

Historically, the adoption of technology follows a predictable pattern[1] of discovery, hype, disillusionment, enlightenment, and eventually, productivity. Through it all, the devices, platforms, and input mechanisms that last the test of time and become truly embedded into our shared technical lexicon are the ones whose designers make a concerted and deliberate effort to include everyone. Early motor accessibility techniques for graphical interfaces paved the way for the wider adoption of the mouse. The smartphone would not be in its dominant position without researchers working to understand the accessibility implications of touch interaction. The maturity and wider adoption of spatial interfaces depends on making spatial input accessibility research a top priority. This work represents another step toward making AR and VR transcend the tech hype cycle and truly become mature.

We hope our work will motivate further explorations into the ability assumptions of spatial interfaces, and inspire ongoing effort to make AR and VR more accessible. If spatial interaction is the future, then it is up to us to create a future that includes everyone.

---

[1] See the Gartner Hype Cycle: `https://en.wikipedia.org/wiki/Gartner_hype_cycle`

BIBLIOGRAPHY

[1] ABSHOW. *How I Play Video Games With 5 Fingers! (One Handed Video Gamer Girl)*. Jan. 2018. URL: https://www.youtube.com/watch?v=FQVF03JG4mQ (visited on 09/22/2024).

[2] ABSHOW. *Xbox Adaptive Controller ONE Handed | Gaming With My Feet???* Mar. 2019. URL: https://www.youtube.com/watch?v=Yk6Oeix0AH0 (visited on 09/22/2024).

[3] Juan Aguado-Delgado, José-María Gutiérrez-Martínez, José R. Hilera, Luis de Marcos, and Salvador Otón. "Accessibility in video games: a systematic review." en. In: *Universal Access in the Information Society* 19.1 (Mar. 2020), pp. 169–193. ISSN: 1615-5297. DOI: 10.1007/s10209-018-0628-2. URL: https://doi.org/10.1007/s10209-018-0628-2 (visited on 01/20/2023).

[4] Lisa Anthony, YooJin Kim, and Leah Findlater. "Analyzing user-generated youtube videos to understand touchscreen use by people with motor impairments." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. New York, NY, USA: Association for Computing Machinery, Apr. 2013, pp. 1223–1232. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2466158. URL: https://doi.org/10.1145/2470654.2466158 (visited on 03/15/2023).

[5] Lisa Anthony, Jie Yang, and Kenneth R. Koedinger. "Evaluation of multimodal input for entering mathematical equations on the computer." In: *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '05. New York, NY, USA: Association for Computing Machinery, Apr. 2005, pp. 1184–1187. ISBN: 978-1-59593-002-6. DOI: 10.1145/1056808.1056872. URL: https://doi.org/10.1145/1056808.1056872 (visited on 09/22/2024).

[6] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. "SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1–15. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173759. URL: https://doi.org/10.1145/3173574.3173759 (visited on 01/24/2023).

[7] Armen Avetisyan et al. *SceneScript: Reconstructing Scenes With An Autoregressive Structured Language Model*. arXiv:2403.13064. Mar. 2024. DOI: 10.48550/arXiv.2403.13064. URL: http://arxiv.org/abs/2403.13064 (visited on 10/14/2024).

[8] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. “Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences.” en. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, May 2016, pp. 1968–1979. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858226. URL: https://dl.acm.org/doi/10.1145/2858036.2858226 (visited on 09/06/2023).

[9] Rakesh Babu, Rahul Singh, The University of North Carolina at Greensboro, USA, Jai Ganesh, and Infosys Technologies Limited. “Understanding Blind Users’ Web Accessibility and Usability Problems.” en. In: *AIS Transactions on Human-Computer Interaction* 2.3 (Sept. 2010), pp. 73–94. ISSN: 19443900. DOI: 10.17705/1thci.00015. URL: https://aisel.aisnet.org/thci/vol2/iss3/1 (visited on 04/23/2024).

[10] Till Ballendat, Nicolai Marquardt, and Saul Greenberg. “Proxemic interaction: designing for a proximity and orientation-aware environment.” In: *ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’10. New York, NY, USA: Association for Computing Machinery, Nov. 2010, pp. 121–130. ISBN: 978-1-4503-0399-6. DOI: 10.1145/1936652.1936676. URL: https://doi.org/10.1145/1936652.1936676 (visited on 09/22/2024).

[11] Mayra Donaji Barrera Machuca and Wolfgang Stuerzlinger. “The Effect of Stereo Display Deficiencies on Virtual Hand Pointing.” en. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, May 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300437. URL: https://dl.acm.org/doi/10.1145/3290605.3300437 (visited on 10/31/2023).

[12] Anil Ufuk Batmaz, Mayra Donaji Barrera Machuca, Duc Minh Pham, and Wolfgang Stuerzlinger. “Do Head-Mounted Display Stereo Deficiencies Affect 3D Pointing Tasks in AR and VR?” In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. ISSN: 2642-5254. Mar. 2019, pp. 585–592. DOI: 10.1109/VR.2019.8797975. URL: https://ieeexplore.ieee.org/document/8797975 (visited on 10/31/2023).

[13] Anil Ufuk Batmaz, Rumeysa Turkmen, Mine Sarac, Mayra Donaji Barrera Machuca, and Wolfgang Stuerzlinger. “Re-investigating the Effect of the Vergence-Accommodation Conflict on 3D Pointing.” In: *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology*. VRST ’23. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 1–10. ISBN: 9798400703287. DOI: 10.1145/3611659.3615686. URL: https://dl.acm.org/doi/10.1145/3611659.3615686 (visited on 10/31/2023).

[14] Beat Games. *Beat Saber - VR rhythm game*. URL: https://beatsaber.com/ (visited on 04/19/2024).

[15] H. Benko, E.W. Ishak, and S. Feiner. "Cross-dimensional gestural interaction techniques for hybrid immersive environments." en. In: *IEEE Proceedings. VR 2005. Virtual Reality, 2005.* Bonn, Germany: IEEE, 2005, pp. 209–327. ISBN: 978-0-7803-8929-8. DOI: 10.1109/VR.2005.1492776. URL: http://ieeexplore.ieee.org/document/1492776/ (visited on 01/24/2023).

[16] Erik Berndt, Dominic Furniss, and Ann Blandford. "Learning Contextual Inquiry and Distributed Cognition: a case study on technology use in anaesthesia." en. In: *Cognition, Technology & Work* 17.3 (Aug. 2015), pp. 431–449. ISSN: 1435-5566. DOI: 10.1007/s10111-014-0314-y. URL: https://doi.org/10.1007/s10111-014-0314-y.

[17] P.J. Besl and Neil D. McKay. "A method for registration of 3-D shapes." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 239–256. ISSN: 1939-3539. DOI: 10.1109/34.121791. URL: https://ieeexplore.ieee.org/document/121791 (visited on 09/11/2024).

[18] Natalia Bogdan, Tovi Grossman, and George Fitzmaurice. "HybridSpace: Integrating 3D freehand input and stereo viewing into traditional desktop applications." In: *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. Mar. 2014, pp. 51–58. DOI: 10.1109/3DUI.2014.6798842.

[19] Doug A. Bowman and Larry F. Hodges. "An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments." In: *Proceedings of the 1997 symposium on Interactive 3D graphics*. I3D '97. New York, NY, USA: Association for Computing Machinery, Apr. 1997, 35–ff. ISBN: 978-0-89791-884-8. DOI: 10.1145/253284.253301. URL: https://dl.acm.org/doi/10.1145/253284.253301 (visited on 09/11/2024).

[20] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. "Point & Teleport Locomotion Technique for Virtual Reality." en. In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. Austin Texas USA: ACM, Oct. 2016, pp. 205–216. ISBN: 978-1-4503-4456-2. DOI: 10.1145/2967934.2968105. URL: https://dl.acm.org/doi/10.1145/2967934.2968105 (visited on 07/29/2022).

[21] Luisa Brinkschulte, Robert Mertens, Leon Strapper, Sebastian Pospiech, and Lars Knipping. "A Multi Modal Interaction Paradigm Combining Gaze Tracking and Keyboard." In: *2017 IEEE International Symposium on Multimedia (ISM)*. Dec. 2017, pp. 370–371. DOI: 10.1109/ISM.2017.72.

[22] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. "Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019,

pp. 1–28. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300792. URL: https://doi.org/10.1145/3290605.3300792 (visited on 01/24/2023).

[23] Frederik Brudy et al. “SurfaceFleet: Exploring Distributed Interactions Unbounded from Device, Application, User, and Time.” In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST ’20. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 7–21. ISBN: 978-1-4503-7514-6. DOI: 10.1145/3379337.3415874. URL: https://dl.acm.org/doi/10.1145/3379337.3415874 (visited on 09/22/2024).

[24] Pulkit Budhiraja, Rajinder Sodhi, Brett Jones, Kevin Karsch, Brian Bailey, and David Forsyth. “Where’s My Drink? Enabling Peripheral Real World Interactions While Using HMDs.” In: (Feb. 2015).

[25] William Buxton. “Lexical and pragmatic considerations of input structures.” In: *ACM SIGGRAPH Computer Graphics* 17.1 (Jan. 1983), pp. 31–37. ISSN: 0097-8930. DOI: 10.1145/988584.988586. URL: https://doi.org/10.1145/988584.988586 (visited on 07/03/2022).

[26] *Can I Play That?* en-US. Mar. 2023. URL: https://caniplaythat.com/ (visited on 03/30/2023).

[27] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. “A morphological analysis of the design space of input devices.” In: *ACM Transactions on Information Systems* 9.2 (Apr. 1991), pp. 99–122. ISSN: 1046-8188. DOI: 10.1145/123078.128726. URL: https://doi.org/10.1145/123078.128726 (visited on 07/20/2022).

[28] Felipe G. Carvalho, Daniela G. Trevisan, and Alberto Raposo. “Toward the design of transitional interfaces: an exploratory study on a semi-immersive hybrid user interface.” en. In: *Virtual Reality* 16.4 (Nov. 2012), pp. 271–288. ISSN: 1434-9957. DOI: 10.1007/s10055-011-0205-y. URL: https://doi.org/10.1007/s10055-011-0205-y (visited on 01/24/2023).

[29] 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*. CHI ’12. New York, NY, USA: Association for Computing Machinery, May 2012, pp. 2527–2530. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208639. URL: https://dl.acm.org/doi/10.1145/2207676.2208639 (visited on 09/11/2024).

[30] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. “The Impact of Control-Display Gain on User Performance in Pointing Tasks.” EN. In: *HUMAN–COMPUTER INTERACTION* (Aug. 2008). Publisher: Taylor & Francis Group. URL: https://www.tandfonline.com/doi/full/10.1080/07370020802278163 (visited on 09/12/2024).

[31] Charles Diaz. *How a quadriplegic can play shooter video games.* Oct. 2019. URL: https://www.youtube.com/watch?v=vy1-D3Df36M (visited on 09/22/2024).

[32] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. “Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17. New York, NY, USA: Association for Computing Machinery, May 2017, pp. 3718–3728. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025753. URL: https://dl.acm.org/doi/10.1145/3025453.3025753 (visited on 09/05/2023).

[33] K. Coninx, F. Van Reeth, and E. Flerackers. “A hybrid 2D/3D user interface for immersive object modeling.” In: *Proceedings Computer Graphics International*. June 1997, pp. 47–55. DOI: 10.1109/CGI.1997.601270.

[34] David M. Cook, Derani Dissanayake, and Kulwinder Kaur. “Virtual Reality and Older Hands: Dexterity and accessibility in hand-held VR Control.” en. In: *Proceedings of the 5th International ACM In-Cooperation HCI and UX Conference on - CHIuXiD’19*. Jakarta, Surabaya, Bali, Indonesia: ACM Press, 2019, pp. 147–151. ISBN: 978-1-4503-6187-3. DOI: 10.1145/3328243.3328262. URL: http://dl.acm.org/citation.cfm?doid=3328243.3328262 (visited on 07/18/2022).

[35] Juliet Corbin and Anselm Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. en. Google-Books-ID: hZ6kBQAAQBAJ. SAGE Publications, Nov. 2014. ISBN: 978-1-4833-1568-3.

[36] Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarcar, and Ian Williams. “Inclusive AR/VR: accessibility barriers for immersive technologies.” en. In: *Universal Access in the Information Society* 23.1 (Feb. 2023), pp. 59–73. ISSN: 1615-5297. DOI: 10.1007/s10209-023-00969-0. URL: https://doi.org/10.1007/s10209-023-00969-0 (visited on 04/12/2024).

[37] Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarcar, and Ian Williams. “Inclusive Augmented and Virtual Reality: A Research Agenda.” en. In: *International Journal of Human–Computer Interaction* (Aug. 2023), pp. 1–20. ISSN: 1044-7318, 1532-7590. DOI: 10.1080/10447318.2023.2247614. URL: https://www.tandfonline.com/doi/full/10.1080/10447318.2023.2247614 (visited on 04/12/2024).

[38] Danny Yaroslavski [@dannyaroslavski]. *Version 2.0 of #vrdev Quest2 arm mount: Snaps on, attaches to most monitor arms, slides in/out for quick release, no camera occlusion. 1/ https://t.co/jdelS2jY03*. en. Tweet. Mar. 2021. URL: https://x.com/dannyaroslavski/status/1369308893059239937 (visited on 10/14/2024).

[39] Bastian Dewitz, Frank Steinicke, and Christian Geiger. “Functional Workspace for One-Handed Tap and Swipe Microgestures.” en. In: (2019). Publisher: Gesellschaft für Informatik e.V. DOI: 10.18420/MUC2019-WS-440. URL: http://dl.gi.de/handle/20.500.12116/25193 (visited on 10/27/2024).

[40] Youngwook Do, Frederik Brudy, George W. Fitzmaurice, and Fraser Anderson. "Vice VRsa: Balancing Bystander's and VR user's Privacy through Awareness Cues Inside and Outside VR." en. In: Apr. 2023. URL: https://openreview.net/forum?id=gItvr7Xl66 (visited on 09/10/2023).

[41] Kevin Doherty and Gavin Doherty. "The construal of experience in HCI: Understanding self-reports." In: *International Journal of Human-Computer Studies* 110 (Feb. 2018), pp. 63–74. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2017.10.006. URL: https://www.sciencedirect.com/science/article/pii/S1071581917301441.

[42] Lisa A. Elkin, Jean-Baptiste Beau, Géry Casiez, and Daniel Vogel. "Manipulation, Learning, and Recall with Tangible Pen-Like Input." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–12. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376772. URL: https://dl.acm.org/doi/10.1145/3313831.3376772 (visited on 09/22/2024).

[43] Andreas Fender and Jörg Müller. "SpaceState: Ad-Hoc Definition and Recognition of Hierarchical Room States for Smart Environments." In: *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. ISS '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 303–314. ISBN: 978-1-4503-6891-9. DOI: 10.1145/3343055.3359715. URL: https://doi.org/10.1145/3343055.3359715 (visited on 01/24/2023).

[44] Tiare Feuchtner and Jörg Müller. "Ownershift: Facilitating Overhead Interaction in Virtual Reality with an Ownership-Preserving Hand Space Shift." In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 31–43. ISBN: 978-1-4503-5948-1. DOI: 10.1145/3242587.3242594. URL: https://dl.acm.org/doi/10.1145/3242587.3242594 (visited on 09/11/2024).

[45] Martin A. Fischler and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://dl.acm.org/doi/10.1145/358669.358692 (visited on 09/11/2024).

[46] Rachel L. Franz, Jinghan Yu, and Jacob O. Wobbrock. "Comparing Locomotion Techniques in Virtual Reality for People with Upper-Body Motor Impairments." en. In: *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*. New York NY USA: ACM, Oct. 2023, pp. 1–15. ISBN: 9798400702204. DOI: 10.1145/3597638.3608394. URL: https://dl.acm.org/doi/10.1145/3597638.3608394 (visited on 04/12/2024).

[47] Ceenu George, An Ngo Tien, and Heinrich Hussmann. "Seamless, Bi-directional Transitions along the Reality-Virtuality Continuum: A Conceptualization and Prototype Exploration." In: *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. ISSN: 1554-7868. Nov. 2020, pp. 412–424. DOI: 10.1109/ISMAR50242.2020.00067. URL: https://ieeexplore.ieee.org/document/9284807 (visited on 11/11/2023).

[48] Kathrin Gerling, Patrick Dickinson, Kieran Hicks, Liam Mason, Adalberto L. Simeone, and Katta Spiel. "Virtual Reality Games for People Using Wheelchairs." en. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, Apr. 2020, pp. 1–11. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376265. URL: https://dl.acm.org/doi/10.1145/3313831.3376265 (visited on 01/25/2023).

[49] Kathrin Gerling and Katta Spiel. "A Critical Examination of Virtual Reality Technology in the Context of the Minority Body." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–14. ISBN: 978-1-4503-8096-6. DOI: 10.1145/3411764.3445196. URL: http://doi.org/10.1145/3411764.3445196 (visited on 09/14/2022).

[50] Yalda Ghasemi and Heejin Jeong. "Using Gaze-based Interaction to Alleviate Situational Mobility Impairment in Extended Reality." en. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 66.1 (Sept. 2022). Publisher: SAGE Publications Inc, pp. 435–439. ISSN: 2169-5067. DOI: 10.1177/1071181322661224. URL: https://doi.org/10.1177/1071181322661224 (visited on 09/01/2023).

[51] Sarthak Ghosh, Lauren Winston, Nishant Panchal, Philippe Kimura-Thollander, Jeff Hotnog, Douglas Cheong, Gabriel Reyes, and Gregory D. Abowd. "NotifiVR: Exploring Interruptions and Notifications in Virtual Reality." In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (Apr. 2018). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 1447–1456. ISSN: 1941-0506. DOI: 10.1109/TVCG.2018.2793698.

[52] Gizmo XYT. *Boy With Cerebral Palsy Has The Most Creative Use Of Xbox Controller*. May 2020. URL: https://www.youtube.com/watch?v=Tam8jB0vtPE (visited on 09/22/2024).

[53] Camille Godineau. *The new Xbox adaptive controller, another step towards digital inclusion?* en-US. Sept. 2018. URL: https://mastersofmedia.hum.uva.nl/blog/2018/09/23/the-new-xbox-adaptive-controller-another-step-towards-digital-inclusion/ (visited on 09/22/2024).

[54] Mayank Goel, Leah Findlater, and Jacob Wobbrock. "WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry." en. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Austin Texas USA: ACM, May

2012, pp. 2687–2696. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208662. URL: https://dl.acm.org/doi/10.1145/2207676.2208662 (visited on 09/01/2023).

[55] Google. *Tilt Brush by Google*. URL: https://www.tiltbrush.com/ (visited on 04/19/2024).

[56] Raphael Grasset, Andreas Duenser, and Mark Billinghurst. "Moving Between Contexts - A User Evaluation of a Transitional Interface." In: *18th International Conference on Artificial Reality and Telexistence 2008*. Yokohama Japan, Dec. 2008, pp. 137–143.

[57] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. "MultiFi: Multi Fidelity Interaction with Displays On and Around the Body." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 3933–3942. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702331. URL: https://doi.org/10.1145/2702123.2702331 (visited on 01/24/2023).

[58] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. "Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards." In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Mar. 2018, pp. 159–166. DOI: 10.1109/VR.2018.8446059.

[59] Sandra G. Hart. *NASA Task Load Index (TLX)*. NTRS Author Affiliations: NASA Ames Research Center NTRS Document ID: 20000021487 NTRS Research Center: Ames Research Center (ARC). Jan. 1986. URL: https://ntrs.nasa.gov/citations/20000021487 (visited on 09/05/2023).

[60] 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*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–12. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300577. URL: https://doi.org/10.1145/3290605.3300577 (visited on 01/24/2023).

[61] Steven Houben, Paolo Tell, and Jakob E. Bardram. "ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space." In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS '14. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 119–128. ISBN: 978-1-4503-2587-5. DOI: 10.1145/2669485.2669493. URL: https://dl.acm.org/doi/10.1145/2669485.2669493 (visited on 09/22/2024).

[62] I-Illusions. *Space Pirate Trainer on Steam*. en. URL: https://store.steampowered.com/app/418650/Space_Pirate_Trainer/ (visited on 04/19/2024).

[63] Robert J. K. Jacob. "Human-computer interaction: input devices." In: *ACM Comput. Surv.* 28.1 (Mar. 1996), pp. 177–179. ISSN: 0360-0300. DOI: 10.1145/234313.234387. URL: https://dl.acm.org/doi/10.1145/234313.234387 (visited on 09/22/2024).

[64] Dhruv Jain, Sasa Junuzovic, Eyal Ofek, Mike Sinclair, John Porter, Chris Yoon, Swetha Machanavajhala, and Meredith Ringel Morris. "A Taxonomy of Sounds in Virtual Reality." en. In: *Designing Interactive Systems Conference 2021*. Virtual Event USA: ACM, June 2021, pp. 160–170. ISBN: 978-1-4503-8476-6. DOI: 10.1145/3461778.3462106. URL: https://dl.acm.org/doi/10.1145/3461778.3462106 (visited on 05/07/2023).

[65] Nikhita Joshi, Parastoo Abtahi, Raj Sodhi, Nitzan Bartov, Jackson Rushing, Christopher Collins, Daniel Vogel, and Michael Glueck. "Transferable Microgestures Across Hand Posture and Location Constraints: Leveraging the Middle, Ring, and Pinky Fingers." en. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. San Francisco CA USA: ACM, Oct. 2023, pp. 1–17. ISBN: 9798400701320. DOI: 10.1145/3586183.3606713. URL: https://dl.acm.org/doi/10.1145/3586183.3606713 (visited on 10/27/2024).

[66] Yvonne Kammerer, Katharina Scheiter, and Wolfgang Beinhauer. "Looking my way through the menu: the impact of menu design and multimodal input on gaze-based menu selection." In: *Proceedings of the 2008 symposium on Eye tracking research & applications*. ETRA '08. New York, NY, USA: Association for Computing Machinery, Mar. 2008, pp. 213–220. ISBN: 978-1-59593-982-1. DOI: 10.1145/1344471.1344522. URL: https://doi.org/10.1145/1344471.1344522 (visited on 01/24/2023).

[67] Mohamed Kari and Christian Holz. "HandyCast: Phone-based Bimanual Input for Virtual Reality in Mobile and Space-Constrained Settings via Pose-and-Touch Transfer." en. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Hamburg Germany: ACM, Apr. 2023, pp. 1–15. ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3580677. URL: https://dl.acm.org/doi/10.1145/3544548.3580677 (visited on 04/21/2024).

[68] Alexey Karpov and Andrey Ronzhin. "A Universal Assistive Technology with Multimodal Input and Multimedia Output Interfaces." en. In: *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access*. Ed. by Constantine Stephanidis and Margherita Antona. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 369–378. ISBN: 978-3-319-07437-5. DOI: 10.1007/978-3-319-07437-5_35.

[69] Simeon Keates and Peter Robinson. "The use of gestures in multimodal input." In: *Proceedings of the third international ACM conference on Assistive technologies*. Assets '98. New York, NY, USA: Association for Computing Machinery, Jan. 1998, pp. 35–42. ISBN: 978-1-58113-020-

1. DOI: 10.1145/274497.274505. URL: https://doi.org/10.1145/274497.274505 (visited on 07/03/2022).

[70] Simeon Keates and Shari Trewin. "Effect of age and Parkinson's disease on cursor positioning using a mouse." In: *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. Assets '05. New York, NY, USA: Association for Computing Machinery, Oct. 2005, pp. 68–75. ISBN: 978-1-59593-159-7. DOI: 10.1145/1090785.1090800. URL: https://dl.acm.org/doi/10.1145/1090785.1090800 (visited on 09/22/2024).

[71] Daekun Kim, Nikhita Joshi, and Daniel Vogel. "Perspective and Geometry Approaches to Mouse Cursor Control in Spatial Augmented Reality." en. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Hamburg Germany: ACM, Apr. 2023, pp. 1–19. ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3580849. URL: https://dl.acm.org/doi/10.1145/3544548.3580849 (visited on 09/04/2023).

[72] Jarrod Knibbe, Jonas Schjerlund, Mathias Petraeus, and Kasper Hornbæk. "The Dream is Collapsing: The Experience of Exiting VR." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1–13. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174057. URL: https://doi.org/10.1145/3173574.3174057 (visited on 01/24/2023).

[73] Yoshiki Kudo, Anthony Tang, Kazuyuki Fujita, Isamu Endo, Kazuki Takashima, and Yoshifumi Kitamura. "Towards Balancing VR Immersion and Bystander Awareness." en. In: *Proceedings of the ACM on Human-Computer Interaction* 5.ISS (Nov. 2021), pp. 1–22. ISSN: 2573-0142. DOI: 10.1145/3486950. URL: https://dl.acm.org/doi/10.1145/3486950 (visited on 09/10/2023).

[74] Rachel L. Franz, Sasa Junuzovic, and Martez Mott. "Nearmi: A Framework for Designing Point of Interest Techniques for VR Users with Limited Mobility." en. In: *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. Virtual Event USA: ACM, Oct. 2021, pp. 1–14. ISBN: 978-1-4503-8306-6. DOI: 10.1145/3441852.3471230. URL: https://dl.acm.org/doi/10.1145/3441852.3471230 (visited on 05/10/2023).

[75] Juyoung Lee, Sang Chul Ahn, and Jae-In Hwang. "A Walking-in-Place Method for Virtual Reality Using Position and Orientation Tracking." en. In: *Sensors* 18.9 (Sept. 2018). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 2832. ISSN: 1424-8220. DOI: 10.3390/s18092832. URL: https://www.mdpi.com/1424-8220/18/9/2832 (visited on 09/05/2023).

[76] Minkyung Lee, Mark Billinghurst, Woonhyuk Baek, Richard Green, and Woontack Woo. "A usability study of multimodal input in an augmented reality environment." en. In: *Virtual Reality* 17.4 (Nov. 2013), pp. 293–305. ISSN: 1359-4338, 1434-9957. DOI: 10.1007/s10055-013-0230-0. URL: http://link.springer.com/10.1007/s10055-013-0230-0 (visited on 07/14/2022).

[77] William E. Lorensen and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), pp. 163–169. ISSN: 0097-8930. DOI: 10.1145/37402.37422. URL: https://dl.acm.org/doi/10.1145/37402.37422 (visited on 10/14/2024).

[78] Feiyu Lu and Yan Xu. "Exploring Spatial UI Transition Mechanisms with Head-Worn Augmented Reality." en. In: *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–16. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3517723. URL: https://dl.acm.org/doi/10.1145/3491102.3517723 (visited on 09/06/2023).

[79] Jock Mackinlay and Stuart K Card. "A Semantic Analysis of the Design Space of Input Devices." en. In: (), p. 46.

[80] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. "SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1–14. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173928. URL: https://dl.acm.org/doi/10.1145/3173574.3173928 (visited on 09/22/2024).

[81] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H. Thomas. *Immersive Analytics*. en. Google-Books-ID: vaVyDwAAQBAJ. Springer, Oct. 2018. ISBN: 978-3-030-01388-2.

[82] Victoria McArthur, Steven J. Castellucci, and I. Scott MacKenzie. "An empirical comparison of "wiimote" gun attachments for pointing tasks." In: *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. EICS '09. New York, NY, USA: Association for Computing Machinery, July 2009, pp. 203–208. ISBN: 978-1-60558-600-7. DOI: 10.1145/1570433.1570471. URL: https://dl.acm.org/doi/10.1145/1570433.1570471 (visited on 09/22/2024).

[83] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. "Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice." In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (Nov. 2019), 72:1–72:23. DOI: 10.1145/3359174. URL: https://dl.acm.org/doi/10.1145/3359174 (visited on 04/18/2024).

[84] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. "A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays." en. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Seoul Republic of Korea: ACM, Apr. 2015, pp. 2143–2152. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702382. URL: https://dl.acm.org/doi/10.1145/2702123.2702382 (visited on 09/06/2023).

[85] Mighty Coconut. *Walkabout Mini Golf™*. URL: https://www.mightycoconut.com/minigolf (visited on 04/19/2024).

[86] Matthew B. Miles, A. Michael Huberman, and Johnny Saldana. *Qualitative Data Analysis: A Methods Sourcebook*. en. Google-Books-ID: powXBAAAQBAJ. SAGE Publications, Apr. 2013. ISBN: 978-1-4833-2379-4.

[87] Alexandre Millette and Michael J. McGuffin. "DualCAD: Integrating Augmented Reality with a Desktop GUI and Smartphone Interaction." In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. Sept. 2016, pp. 21–26. DOI: 10.1109/ISMAR-Adjunct.2016.0030.

[88] Mizino: In Over My Head. *Gaming with one hand: Computer Games*. Jan. 2019. URL: https://www.youtube.com/watch?v=TIzGGcHJ1Ec (visited on 09/22/2024).

[89] Roberto A. Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. "Erg-O: Ergonomic Optimization of Immersive Virtual Environments." In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 759–771. ISBN: 978-1-4503-4981-9. DOI: 10.1145/3126594.3126605. URL: https://dl.acm.org/doi/10.1145/3126594.3126605 (visited on 10/31/2023).

[90] Martez E. Mott and Jacob O. Wobbrock. "Cluster Touch: Improving Touch Accuracy on Smartphones for People with Motor and Situational Impairments." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300257. URL: https://dl.acm.org/doi/10.1145/3290605.3300257 (visited on 09/22/2024).

[91] Martez Mott, Edward Cutrell, Mar Gonzalez Franco, Christian Holz, Eyal Ofek, Richard Stoakley, and Meredith Ringel Morris. "Accessible by Design: An Opportunity for Virtual Reality." en. In: *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. Beijing, China: IEEE, Oct. 2019, pp. 451–454. ISBN: 978-1-72814-765-9. DOI: 10.1109/ISMAR-Adjunct.2019.00122. URL: https://ieeexplore.ieee.org/document/8951960/ (visited on 07/18/2022).

[92] Martez Mott, John Tang, Shaun Kane, Edward Cutrell, and Meredith Ringel Morris. ““I just went into it assuming that I wouldn’t be able to have the full experience”: Understanding the Accessibility of Virtual Reality for People with Limited Mobility.” en. In: *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. Virtual Event Greece: ACM, Oct. 2020, pp. 1–13. ISBN: 978-1-4503-7103-2. DOI: 10.1145/3373625.3416998. URL: https://dl.acm.org/doi/10.1145/3373625.3416998 (visited on 07/10/2022).

[93] Christian Müller and Garrett Weinberg. “Multimodal Input in the Car, Today and Tomorrow.” In: *IEEE MultiMedia* 18.1 (Jan. 2011). Conference Name: IEEE MultiMedia, pp. 98–103. ISSN: 1941-0166. DOI: 10.1109/MMUL.2011.14.

[94] Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin. “Perspective cursor: perspective-based interaction for multi-display environments.” en. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Montréal Québec Canada: ACM, Apr. 2006, pp. 289–298. ISBN: 978-1-59593-372-0. DOI: 10.1145/1124772.1124817. URL: https://dl.acm.org/doi/10.1145/1124772.1124817 (visited on 09/14/2023).

[95] Maia Naftali and Leah Findlater. “Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments.” In: *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. ASSETS ’14. New York, NY, USA: Association for Computing Machinery, Oct. 2014, pp. 209–216. ISBN: 978-1-4503-2720-6. DOI: 10.1145/2661334.2661372. URL: https://dl.acm.org/doi/10.1145/2661334.2661372 (visited on 04/19/2024).

[96] Joseph O’Hagan and Julie R. Williamson. “Reality aware VR headsets.” en. In: *Proceedings of the 9TH ACM International Symposium on Pervasive Displays*. Manchester United Kingdom: ACM, June 2020, pp. 9–17. ISBN: 978-1-4503-7986-1. DOI: 10.1145/3393712.3395334. URL: https://dl.acm.org/doi/10.1145/3393712.3395334 (visited on 09/10/2023).

[97] Sharon Oviatt. “Ten myths of multimodal interaction.” In: *Commun. ACM* 42.11 (Nov. 1999), pp. 74–81. ISSN: 0001-0782. DOI: 10.1145/319382.319398. URL: https://dl.acm.org/doi/10.1145/319382.319398 (visited on 09/22/2024).

[98] Sharon Oviatt. *MULTIMODAL INTERFACES*. en. Pages: 439-458 Publication Title: The Human-Computer Interaction Handbook. CRC Press, Sept. 2007. ISBN: 978-0-429-16397-5. DOI: 10.1201/9781410615862-32. URL: https://www.taylorfrancis.com/chapters/edit/10.1201/9781410615862-32/multimodal-interfaces-sharon-oviatt (visited on 07/03/2022).

[99] PARTYCHAT. *Dad Builds Adaptive Controller for Disabled Daughter | Super Punch*. Feb. 2020. URL: https://www.youtube.com/watch?v=IvK1uJ0kzS8 (visited on 09/22/2024).

[100] Shanmugam Muruga Palaniappan, Ting Zhang, and Bradley S. Duerstock. "Identifying Comfort Areas in 3D Space for Persons with Upper Extremity Mobility Impairments Using Virtual Reality." en. In: *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. Pittsburgh PA USA: ACM, Oct. 2019, pp. 495–499. ISBN: 978-1-4503-6676-2. DOI: 10.1145/3308561.3353810. URL: https://dl.acm.org/doi/10.1145/3308561.3353810 (visited on 05/07/2023).

[101] Duc-Minh Pham and Wolfgang Stuerzlinger. "HawKEY: Efficient and Versatile Text Entry for Virtual Reality." In: *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*. VRST '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 1–11. ISBN: 978-1-4503-7001-1. DOI: 10.1145/3359996.3364265. URL: https://dl.acm.org/doi/10.1145/3359996.3364265 (visited on 09/05/2023).

[102] Fabian Pointecker, Judith Friedl, Daniel Schwajda, Hans-Christian Jetter, and Christoph Anthes. "Bridging the Gap Across Realities: Visual Transitions Between Virtual and Augmented Reality." en. In: *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Singapore, Singapore: IEEE, Oct. 2022, pp. 827–836. ISBN: 978-1-66545-325-7. DOI: 10.1109/ISMAR55827.2022.00101. URL: https://ieeexplore.ieee.org/document/9995285/ (visited on 11/11/2023).

[103] John R. Porter. "Understanding and addressing real-world accessibility issues in mainstream video games." In: *SIGACCESS Access. Comput.* 108 (Jan. 2014), pp. 42–45. ISSN: 1558-2337. DOI: 10.1145/2591357.2591364. URL: https://dl.acm.org/doi/10.1145/2591357.2591364 (visited on 09/22/2024).

[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. New York, NY, USA: Association for Computing Machinery, Nov. 1996, pp. 79–80. ISBN: 978-0-89791-798-8. DOI: 10.1145/237091.237102. URL: https://doi.org/10.1145/237091.237102 (visited on 01/24/2023).

[105] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. arXiv:1612.00593 [cs]. Apr. 2017. DOI: 10.48550/arXiv.1612.00593. URL: http://arxiv.org/abs/1612.00593 (visited on 09/11/2024).

[106] Leah M. Reeves et al. "Guidelines for multimodal user interface design." In: *Commun. ACM* 47.1 (Jan. 2004), pp. 57–59. ISSN: 0001-0782. DOI: 10.1145/962081.962106. URL: https://dl.acm.org/doi/10.1145/962081.962106 (visited on 09/22/2024).

[107] David Rozado, Jason Niu, and Andreas Duenser. "FaceSwitch - Low-Cost Accessibility Software for Computer Control Combining Gaze Interaction and Face Gestures." In: *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*.

OzCHI '15. New York, NY, USA: Association for Computing Machinery, Dec. 2015, pp. 197–201. ISBN: 978-1-4503-3673-4. DOI: 10.1145/2838739.2838809. URL: http://doi.org/10.1145/2838739.2838809 (visited on 07/03/2022).

[108] Natalie Ruiz, Ronnie Taib, and Fang Chen. "Examining the redundancy of multimodal input." In: *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*. OZCHI '06. New York, NY, USA: Association for Computing Machinery, Nov. 2006, pp. 389–392. ISBN: 978-1-59593-545-8. DOI: 10.1145/1228175.1228254. URL: https://dl.acm.org/doi/10.1145/1228175.1228254 (visited on 09/22/2024).

[109] Luis Salvador-Ullauri, Patricia Acosta-Vargas, Mario Gonzalez, and Sergio Luján-Mora. "Combined Method for Evaluating Accessibility in Serious Games." en. In: *Applied Sciences* 10.18 (Jan. 2020). Number: 18 Publisher: Multidisciplinary Digital Publishing Institute, p. 6324. ISSN: 2076-3417. DOI: 10.3390/app10186324. URL: https://www.mdpi.com/2076-3417/10/18/6324 (visited on 03/16/2023).

[110] Majed Samad, Elia Gatti, Anne Hermes, Hrvoje Benko, and Cesare 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*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–13. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300550. URL: https://dl.acm.org/doi/10.1145/3290605.3300550 (visited on 09/05/2023).

[111] Anthony Savidis and Constantine Stephanidis. "Unified user interface design: designing universally accessible interactions." In: *Interacting with Computers* 16.2 (Apr. 2004). _eprint: https://academic.oup.com/iwc/article-pdf/16/2/243/2147562/iwc16-0243.pdf, pp. 243–270. ISSN: 0953-5438. DOI: 10.1016/j.intcom.2003.12.003. URL: https://doi.org/10.1016/j.intcom.2003.12.003.

[112] Jan-Henrik Schröder, Daniel Schacht, Niklas Peper, Anita Marie Hamurculu, and Hans-Christian Jetter. "Collaborating Across Realities: Analytical Lenses for Understanding Dyadic Collaboration in Transitional Interfaces." In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–16. ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3580879. URL: https://doi.org/10.1145/3544548.3580879 (visited on 09/13/2023).

[113] I. Scott Mackenzie and Stan Riddersma. "Effects of output display and control—display gain on human performance in interactive systems." In: *Behaviour & Information Technology* 13.5 (Sept. 1994). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01449299408914613, pp. 328–337. ISSN: 0144-929X. DOI: 10.1080/01449299408914613.

URL: `https://doi.org/10.1080/01449299408914613` (visited on 09/12/2024).

[114] Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. "Gluey: Developing a Head-Worn Display Interface to Unify the Interaction Experience in Distributed Display Environments." In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 161–171. ISBN: 978-1-4503-3652-9. DOI: `10.1145/2785830.2785838`. URL: `https://doi.org/10.1145/2785830.2785838` (visited on 01/24/2023).

[115] Shot Callers Esports. *RockyNoHands: The Gamer Who Can Beat You With His Mouth | Twitch Documentary Preview | Stream On*. Sept. 2017. URL: `https://www.youtube.com/watch?v=ZMvikz2cA-8` (visited on 09/22/2024).

[116] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. "Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences." en. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Seoul Republic of Korea: ACM, Apr. 2015, pp. 3307–3316. ISBN: 978-1-4503-3145-6. DOI: `10.1145/2702123.2702389`. URL: `https://dl.acm.org/doi/10.1145/2702123.2702389` (visited on 09/10/2023).

[117] Alice Smith, John Dunaway, Patrick Demasco, and Denise Peischl. "Multimodal input for computer access and augmentative communication." In: *Proceedings of the second annual ACM conference on Assistive technologies*. Assets '96. New York, NY, USA: Association for Computing Machinery, Apr. 1996, pp. 80–85. ISBN: 978-0-89791-776-6. DOI: `10.1145/228347.228361`. URL: `https://doi.org/10.1145/228347.228361` (visited on 07/03/2022).

[118] R. William Soukoreff and I. Scott MacKenzie. "Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI." en. In: *International Journal of Human-Computer Studies* 61.6 (Dec. 2004), pp. 751–789. ISSN: 10715819. DOI: `10.1016/j.ijhcs.2004.09.001`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1071581904001016` (visited on 09/08/2023).

[119] Laura South, Caglar Yildirim, Amy Pavel, and Michelle Borkin. *Barriers to Photosensitive Accessibility in Virtual Reality*. en. Jan. 2024. DOI: `10.31219/osf.io/ywqba`. URL: `https://osf.io/ywqba` (visited on 04/20/2024).

[120] Microsoft Support. *How to use Xbox Controller Assist | Xbox Support*. URL: `https://support.xbox.com/en-CA/help/account-profile/accessibility/copilot` (visited on 09/22/2024).

[121] Hemant Bhaskar Surale, Aakar Gupta, Mark Hancock, and Daniel Vogel. “TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–13. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300243. URL: http://doi.org/10.1145/3290605.3300243 (visited on 07/18/2022).

[122] Felix J. Thiel and Anthony Steed. “"Lend Me a Hand" – Extending the Reach of Seated VR Players in Unmodified Games Through Remote Co-Piloting.” en. In: *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. Lisbon, Portugal: IEEE, Mar. 2021, pp. 214–219. ISBN: 978-1-66544-057-8. DOI: 10.1109/VRW52623.2021.00047. URL: https://ieeexplore.ieee.org/document/9419332/ (visited on 03/23/2024).

[123] Jean-Philippe Thiran, Ferran Marqués, and Hervé Bourlard. *Multimodal Signal Processing: Theory and Applications for Human-Computer Interaction*. en. Google-Books-ID: mNSduMPXZy4C. Academic Press, Nov. 2009. ISBN: 978-0-08-088869-9.

[124] Jingze Tian, Yingna Wang, Keye Yu, Liyi Xu, Junan Xie, Franklin Mingzhe Li, Yafeng Niu, and Mingming Fan. *Designing Upper-Body Gesture Interaction with and for People with Spinal Muscular Atrophy in VR*. arXiv:2403.16107 [cs]. Mar. 2024. URL: http://arxiv.org/abs/2403.16107 (visited on 04/12/2024).

[125] Shari Trewin, Simeon Keates, and Karyn Moffatt. “Developing steady clicks: a method of cursor assistance for people with motor impairments.” In: *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. Assets ’06. New York, NY, USA: Association for Computing Machinery, Oct. 2006, pp. 26–33. ISBN: 978-1-59593-290-7. DOI: 10.1145/1168987.1168993. URL: https://dl.acm.org/doi/10.1145/1168987.1168993 (visited on 09/22/2024).

[126] Michael Tsang, George W. Fitzmzurice, Gordon Kurtenbach, Azam Khan, and Bill Buxton. “Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display.” In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH ’03. New York, NY, USA: Association for Computing Machinery, July 2003, p. 698. ISBN: 978-1-58113-709-5. DOI: 10.1145/1201775.882329. URL: https://doi.org/10.1145/1201775.882329 (visited on 01/24/2023).

[127] Wen-Jie Tseng, Samuel Huron, Eric Lecolinet, and Jan Gugenheimer. *FingerMapper: Mapping Finger Motions onto Virtual Arms to Enable Safe Virtual Reality Interaction in Confined Spaces*. arXiv:2302.11865 [cs]. Feb. 2023. DOI: 10.1145/3544548.3580736. URL: http://arxiv.org/abs/2302.11865 (visited on 03/07/2023).

[128] Matthew Turk. "Review Article: Multimodal interaction: A review." In: *Pattern Recogn. Lett.* 36 (Jan. 2014), pp. 189–195. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2013.07.003. URL: https://doi.org/10.1016/j.patrec.2013.07.003 (visited on 09/22/2024).

[129] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks. "Walking > walking-in-place > flying, in virtual environments." In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., July 1999, pp. 359–364. ISBN: 978-0-201-48560-8. DOI: 10.1145/311535.311589. URL: https://doi.org/10.1145/311535.311589 (visited on 01/24/2023).

[130] Daniel Vogel and Géry Casiez. "Conté: multimodal input inspired by an artist's crayon." In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. UIST '11. New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 357–366. ISBN: 978-1-4503-0716-1. DOI: 10.1145/2047196.2047242. URL: https://dl.acm.org/doi/10.1145/2047196.2047242 (visited on 09/22/2024).

[131] Julius Von Willich, Markus Funk, Florian Müller, Karola Marky, Jan Riemann, and Max Mühlhäuser. "You Invaded my Tracking Space! Using Augmented Virtuality for Spotting Passersby in Room-Scale Virtual Reality." en. In: *Proceedings of the 2019 on Designing Interactive Systems Conference*. San Diego CA USA: ACM, June 2019, pp. 487–496. ISBN: 978-1-4503-5850-7. DOI: 10.1145/3322276.3322334. URL: https://dl.acm.org/doi/10.1145/3322276.3322334 (visited on 09/10/2023).

[132] J. A. Wagner Filho, C.m.d.s. Freitas, and L. Nedel. "VirtualDesk: A Comfortable and Efficient Immersive Information Visualization Approach." en. In: *Computer Graphics Forum* 37.3 (2018). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13430, pp. 415–426. ISSN: 1467-8659. DOI: 10.1111/cgf.13430. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13430 (visited on 09/06/2023).

[133] Chiu-Hsuan Wang, Bing-Yu Chen, and Liwei Chan. "RealityLens: A User Interface for Blending Customized Physical World View into Virtual Reality." In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST '22. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 1–11. ISBN: 978-1-4503-9320-1. DOI: 10.1145/3526113.3545686. URL: https://dl.acm.org/doi/10.1145/3526113.3545686 (visited on 11/11/2023).

[134] Chiu-Hsuan Wang, Chia-En Tsai, Seraphina Yong, and Liwei Chan. "Slice of Light: Transparent and Integrative Transition Among Realities in a Multi-HMD-User Environment." en. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Virtual Event USA: ACM, Oct. 2020, pp. 805–817. ISBN: 978-1-4503-7514-6. DOI:

10.1145/3379337.3415868. URL: https://dl.acm.org/doi/10.1145/3379337.3415868 (visited on 04/06/2023).

[135] Peng Wang, Peng Wu, Jun Wang, Hung-Lin Chi, and Xiangyu Wang. "A Critical Review of the Use of Virtual Reality in Construction Engineering Education and Training." In: *International Journal of Environmental Research and Public Health* 15 (June 2018), p. 1204. DOI: 10.3390/ijerph15061204.

[136] Kirby Weerink. *Developing virtual reality in forensic mental healthcare : a contextual inquiry*. en. info:eu-repo/semantics/masterThesis. Publisher: University of Twente. Dec. 2017. URL: https://essay.utwente.nl/74243/ (visited on 04/23/2024).

[137] Johann Wentzel. "Bring-Your-Own Input: Context-Aware Multi-Modal Input for More Accessible Virtual Reality." en. In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. Hamburg Germany: ACM, Apr. 2023, pp. 1–5. ISBN: 978-1-4503-9422-2. DOI: 10.1145/3544549.3577056. URL: https://dl.acm.org/doi/10.1145/3544549.3577056 (visited on 08/24/2023).

[138] Johann Wentzel, Fraser Anderson, George Fitzmaurice, Tovi Grossman, and Daniel Vogel. "SwitchSpace: Understanding Context-Aware Peeking Between VR and Desktop Interfaces." In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 1–16. ISBN: 9798400703300. DOI: 10.1145/3613904.3642358. URL: https://dl.acm.org/doi/10.1145/3613904.3642358 (visited on 09/11/2024).

[139] Johann Wentzel, Sasa Junuzovic, James Devine, John Porter, and Martez Mott. "Understanding How People with Limited Mobility Use Multi-Modal Input." en. In: *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–17. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3517458. URL: https://dl.acm.org/doi/10.1145/3491102.3517458 (visited on 06/27/2022).

[140] 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* (2024). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 1–15. ISSN: 1941-0506. DOI: 10.1109/TVCG.2024.3420236. URL: https://ieeexplore.ieee.org/document/10577478 (visited on 10/14/2024).

[141] Johann Wentzel, Greg d'Eon, and Daniel Vogel. "Improving Virtual Reality Ergonomics Through Reach-Bounded Non-Linear Input Amplification." en. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, Apr. 2020, pp. 1–12. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376687.

URL: https://dl.acm.org/doi/10.1145/3313831.3376687 (visited on 08/24/2023).

[142] Wevr. *theBlu*. en. URL: https://wevr.com/theblu (visited on 04/19/2024).

[143] Jacob O. Wobbrock, Krzysztof Z. Gajos, Shaun K. Kane, and Gregg C. Vanderheiden. "Ability-based design." en. In: *Communications of the ACM* 61.6 (May 2018), pp. 62–71. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3148051. URL: https://dl.acm.org/doi/10.1145/3148051 (visited on 07/03/2022).

[144] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. "Ability-Based Design: Concept, Principles and Examples." en. In: *ACM Transactions on Accessible Computing* 3.3 (Apr. 2011), pp. 1–27. ISSN: 1936-7228, 1936-7236. DOI: 10.1145/1952383.1952384. URL: https://dl.acm.org/doi/10.1145/1952383.1952384 (visited on 07/03/2022).

[145] *Xbox Adaptive Controller | Xbox*. en. URL: https://www.xbox.com/en-CA/accessories/controllers/xbox-adaptive-controller (visited on 05/23/2023).

[146] Momona Yamagami, Sasa Junuzovic, Mar Gonzalez-Franco, Eyal Ofek, Edward Cutrell, John R. Porter, Andrew D. Wilson, and Martez E. Mott. "Two-In-One: A Design Space for Mapping Unimanual Input into Bimanual Interactions in VR for Users with Limited Movement." In: *ACM Transactions on Accessible Computing* 15.3 (July 2022), 23:1–23:25. ISSN: 1936-7228. DOI: 10.1145/3510463. URL: http://doi.org/10.1145/3510463 (visited on 09/14/2022).

[147] Lulu Yin, John Dudley, Vanja Garaj, and Per Ola Kristensson. "Inclusive Immersion: Barriers to Enjoyment and Emerging Design Principles for More Accessible Virtual and Augmented Reality." en. In: ().

[148] Bei Yuan, Eelke Folmer, and Frederick C. Harris. "Game accessibility: a survey." en. In: *Universal Access in the Information Society* 10.1 (Mar. 2011), pp. 81–100. ISSN: 1615-5297. DOI: 10.1007/s10209-010-0189-5. URL: https://doi.org/10.1007/s10209-010-0189-5 (visited on 04/20/2024).

[149] Zack Collie. *Quadriplegic's Adaptive Gaming Set Up! (W/ Gameplay)*. Dec. 2020. URL: https://www.youtube.com/watch?v=4UtDzZTuLCk (visited on 09/22/2024).

[150] Thorsten O. Zander, Matti Gaertner, Christian Kothe, and Roman Vilimek. "Combining Eye Gaze Input With a Brain–Computer Interface for Touchless Human–Computer Interaction." In: *International Journal of Human–Computer Interaction* 27.1 (Dec. 2010). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2011.535752, pp. 38–51. ISSN: 1044-7318. DOI: 10.1080/10447318.2011.535752. URL: https://doi.org/10.1080/10447318.2011.535752 (visited on 09/22/2024).

[151] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. “SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision.” en. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, May 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300341. URL: https://dl.acm.org/doi/10.1145/3290605.3300341 (visited on 07/18/2022).

[152] Qian Zhou, George Fitzmaurice, and Fraser Anderson. “In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality.” en. In: *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–17. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3501884. URL: https://dl.acm.org/doi/10.1145/3491102.3501884 (visited on 07/19/2022).

[153] Fengyuan Zhu and Tovi Grossman. “BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–14. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376233. URL: https://doi.org/10.1145/3313831.3376233 (visited on 01/24/2023).

[154] Daniel Zielasko, Marcel Krüger, Benjamin Weyers, and Torsten W. Kuhlen. “Menus on the Desk? System Control in DeskVR.” In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. ISSN: 2642-5254. Mar. 2019, pp. 1287–1288. DOI: 10.1109/VR.2019.8797900.

[155] Daniel Zielasko, Benjamin Weyers, Martin Bellgardt, Sebastian Pick, Alexander Meibner, Tom Vierjahn, and Torsten W. Kuhlen. “Remain Seated: Towards Fully-Immersive Desktop VR.” In: *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*. Mar. 2017, pp. 1–6. DOI: 10.1109/WEVR.2017.7957707.

APPENDICES

A

MULTI-MODAL INPUT: FULL SURVEY DEMOGRAPHICS

Abbreviation	Description
DH	Difficulty in holding
DT	Difficulty typing
F	Rapid fatigue
FN	Finger and wrist flexor paralysis
GP	Difficulty gripping
HP	Hand paralysis
JH	Joint hypermobility
LH	Left hand impairment
LS	Low strength
MD	Difficulty controlling movement direction
ML	Difficulty controlling movement distance
OH	Limited to one hand
PC	Poor coordination
QP	Quadriplegia
ROM	Limited general range of motion
RT	Slow reaction time
SM	Slow movements
SN	Lack of sensation
SP	Spasms
TR	Tremor
V	Vision issues
WK	Extreme progressive weakness
WL	Inability to walk

Table A.1: List of abbreviations and descriptions used in Table A.2.

Table A.2: Full demographics of survey participants. Includes the age at which they started using computers ("Computer Age"), acquired their mobility limitation ("Limitation Age"), and their self-reported computer expertise ("Computer Skill"). Mobility limitations are in shorthand, full descriptions are in Table A.1.

ID	Gender	Age	Computer Age	Limitation Age	Computer Skill	Mobility Limitations
1	W	33	10	0	5	MD, GP, DH, ML
2	M	27	6	1	5	SM, LS, GP, PC, DH, F
3	M	54	14	46	5	WL
4	M	36	8	16	4	LS, SP, GP, DH, TR
5	M	19	5	15	4	SP, GP, SN, FN, DH
6	M	32	8	1	4	LS, GP, DH
7	M	34	3	0	3	SM, MD, SP, PC, ML
8	W	25	7	2	4	SM, LS, MD, GP, PC, DH, ML, F
9	M	33	5	1	4	SM, LS, MD, WK, GP, DH, ML, F
10	M	27	5	1	5	MD, LH, GP, DH, ML
11	M	40	8	6	5	LS, DH, F
12	M	30	8	2	4	SM, SP, GP, DH
13	M	40	10	15	4	SM, LS, MD, GP, PC, DH, ML
14	M	24	6	0	5	SM, LS, GP, DH
15	M	27	2	0	5	GP
16	M	42	13	35	3	SM, LS, GP, DH, ML
17	W	53	15	42	3	PC, F, V
18	M	32	10	7	4	SM, LS, GP, DH, F
19	M	24	10	16	4	SM, LS, MD, GP, DH, ML, F
20	W	36	5	3	4	RT
21	M	62	14	46	5	SM, LS, MD, GP, PC, DH, ML, F
22	NB	30	5	20	4	GP, F, JH
23	M	30	10	26	5	LS, MD, HP, GP, SN, DH, ML
24	W	36	4	0	4	SM, LS, GP, PC, DH, F
25	NB	40	6	16	4	SM, LS, MD, SP, GP, PC, DH, ML, F, TR
26	M	43	10	16	5	SM, LS, MD, SP, GP, SN, DH, ML
27	M	35	7	1	4	SM, LS, GP, DH, F
28	W	33	8	21	4	SM, LS, MD, SP, GP, SN, PC, DH, ML, F
29	M	44	10	3	5	SM, LS, GP, DH, OH, F
30	N/A	29	6	21	5	SP, GP, SN, DH, F, TR
31	M	32	6	23	5	MD, SP, GP, DH, ML, PC, TR
32	W	48	14	24	3	MD, GP, SN, DH, ML, DT
33	M	32	11	25	4	LS, ML
34	M	61	12	56	5	SM, LS, GP, SN, DH, PC
35	M	29	8	19	4	QP, GP, DH
36	M	26	5	2	5	SM, LS, GP, DH, F
37	M	38	6	1	4	SM, LS, GP, DH
38	M	50	25	45	3	SM, MD, SN, ML, F, PC
39	M	62	25	25	4	QP
40	W	51	13	50	2	LS, MD, SN, DH, F, PC
41	M	8	7	0.2	3	SM, LS, MD, GP, DH, ML, ROM, PC
42	M	45	14	38	3	GP, SN, DH
43	M	18	10	16	4	SM, LS, MD, GP, SN, DH, ML, F, PC