CHI 2025: CHI Conference on Human Factors in Computing Systems
April 26 - May 1, 2025
Yokohama, Japan

Conference Sponsors:
SIGCHI

RESEARCH-ARTICLE

MotionBlocks: Modular Geometric Motion Remapping for More Accessible Upper Body Movement in Virtual Reality

JOHANN WENTZEL, University of Waterloo, Waterloo, ON, Canada

ALESSANDRA LUZ, University of Waterloo, Waterloo, ON, Canada

MARTEZ E MOTT, Microsoft Research, Redmond, WA, United States

DANIEL VOGEL, University of Waterloo, Waterloo, ON, Canada

Open Access Support provided by:

University of Waterloo

Microsoft Research

CHI '25: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (April 2025)
https://doi.org/10.1145/3706598.3713837
ISBN: 9798400713941

MotionBlocks: Modular Geometric Motion Remapping for More Accessible Upper Body Movement in Virtual Reality

Johann Wentzel
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
jdwentze@uwaterloo.ca

Alessandra Luz
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
alessandra.luzlam@uwaterloo.ca

Martez E Mott
Microsoft Research
Redmond, Washington, USA
mamott@microsoft.com

Daniel Vogel
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
dvogel@uwaterloo.ca

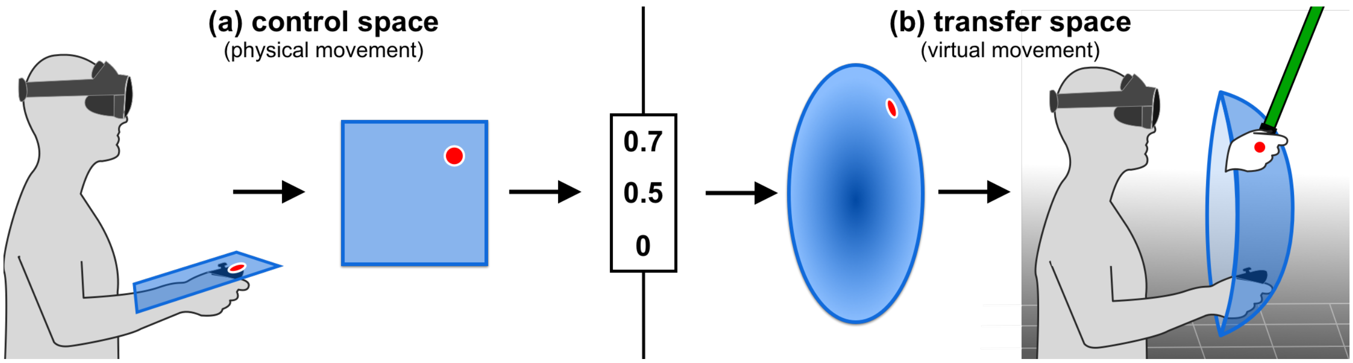

Figure 1: *MotionBlocks* is an approach for constructing complex 3D input using more accessible ranges of motion or simpler input devices. MotionBlocks uses geometric primitives to capture and reconstruct motion, bringing input from *control space* (the range of motion the user can accomplish) to *transfer 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.

Abstract

Movement-based spatial interaction in VR can present significant challenges for people with limited mobility, particularly due to the mismatch between the upper body motion a VR app requires and the user's capabilities. We describe *MotionBlocks*, an approach which enables 3D spatial input with smaller motions or simpler input devices using modular geometric motion remapping. A formative study identifies common accessibility issues within VR motion design, and informs a design language of VR motions that fall within simple geometric primitives. These 3D primitives enable collapsing spatial or non-spatial input into a normalized input vector, which is then expanded into a second 3D primitive representing larger, more complex 3D motions. An evaluation with people with mobility limitations found that using geometric primitives for highly customized upper body input remapping reduced physical workload, temporal workload, and perceived effort.

CCS Concepts

• **Human-centered computing** → **Gestural input**; **Empirical studies in accessibility**; **Virtual reality**.

Keywords

review, formative design study, spatial input, spatial computing, accessibility, controlled experiment, geometric primitives

ACM Reference Format:
Johann Wentzel, Alessandra Luz, Martez E Mott, and Daniel Vogel. 2025. MotionBlocks: Modular Geometric Motion Remapping for More Accessible Upper Body Movement in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3706598.3713837

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713837

1 Introduction

Virtual reality (VR) makes implicit assumptions about user abilities [47, 48] 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 [12, 13, 18, 19], 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 [17, 25, 49] 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 make 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**) *Which issues from the formative study do these more accessible input mappings resolve?*

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 subcategories of findings both within and outside those defined by previous work and 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 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 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 upper body 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.

2 Related Work

Our work draws inspiration from previous work focusing on eliciting accessibility challenges in computing platforms, previous VR accessibility solutions, as well as previous technical approaches for spatial input remapping.

2.1 Understanding Accessibility Challenges

Researchers have evaluated accessibility in other computing platforms. Previous work in eliciting accessibility challenges includes surveys of individual games [51], diary studies [32], and most importantly, direct observation studies [2]. Anthony et al. [1] introduced a method for identifying accessibility barriers in YouTube videos, which allows researchers to systematically analyze diverse and rich data sources. Wentzel et al. [45] 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. [2], using a contextual inquiry formative study to elicit accessibility issues, which then inform the design of a more customizable technical solution.

2.2 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 1, 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. [50] 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. [37] explored barriers to VR accessibility for people with photosensitive epilepsy, finding various contributing factors across hardware, interfaces, applications, and individual sensitivity. Palaniappan et al. [33] used VR and joint force calculations to identify comfort areas surrounding the body to provide objective ergonomic insights. Similarly, Cook et al. [10] analyzed VR controller hardware ergonomics, finding that controller designs may be inaccessible for older hands. Gerling et al. [18, 19] 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. [30, 31] 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. [39] used a combination of surveys and interviews to create a collection of accessible freehand gestures for people with spinal muscular

Table 1: 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	Contextual Inquiry, User Study	✓	Direct	Variety of mobility limitations, detailed in Table 2 (N = 10). 8 also completed user study.
Creed et al. [12]	Sandpit		Indirect	9 disabled persons, 8 researchers, 11 academic experts, 14 stakeholders (N = 38, 4 with multiple roles)
Creed et al. [13]	(1) Sandpit		Indirect	15 unspecified, 7 researchers (N = 22)
	(2) Sandpit		Indirect	14 disabled persons, 9 researchers, 10 unspecified (N = 33)
Tian et al. [39]	Elicitation Study		Indirect	People with spinal muscular atrophy (N = 16)
Mott et al. [31]	Interviews (re: Hardware)	✓	Both	Variety of mobility limitations, 4 used VR (N = 16)
Gerling et al. [18]	(1) Survey		Indirect	All wheelchair users, 14 powered, 9 manual, 1 both (N = 25)
	(2) Usability Study	✓	Direct	All wheelchair users, 2 powered, 12 manual (N = 14)
	(3) Usability Study	✓	Direct	All wheelchair users, 3 manual, 1 manual + propulsion (N = 4)
Gerling and Spiel [19]	Theory-led Analysis		Indirect	Wheelchair-using participants from previous paper [18]
Yin et al. [50]	Survey		Indirect	One or more types of access needs, impairments, disabilities and/or long-term health conditions (N = 101)
Cook et al. [10]	Pilot Study		Direct	Older adults (unspecified)
Palaniappan et al. [33]	Usability Study	✓	Direct	Person with tetraplegia - C4/C5 spinal cord injury (N = 1)

atrophy. Creed et al. [12, 13] 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.

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 the generalizability of experiential qualitative results [14]. 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.

2.3 Previous Approaches to Improve VR Accessibility

There exist some previous approaches to improve motor accessibility for VR applications. Thiel and Steed [38] 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 [25] 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. [49] 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.

2.4 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 [34] or HOMER [6], 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. [40] 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 [29] used a simulated annealing approach to create dynamic ergonomic transfer functions based on known targets near the user's body. RNL amplification [44] is an alternative approach using a non-linear transfer function to make arms-reach VR input more ergonomic. Both techniques reduced fatigue, which

Table 2: 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

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.

2.5 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.

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 [4] focusing on direct observation of participant VR usage. Contextual inquiries can motivate VR design [42], and using unmodified VR applications establishes a similar context for observation as if they bought the hardware themselves.

3.1 Participants

We recruited 10 people with mobility limitations (7 identified as men, 3 as women, age 15–68, median age 25) to participate in the study. Table 2 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.

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 for assistance.

Participants used five VR applications (Table 3). 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. [39], 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 a category for bimanual interaction.

3.3 Protocol

Each session lasted approximately one hour, and participants were compensated with a $50 Amazon gift card. After providing informed consent, 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 total across 3 users, within 3 applications: *Space Pirate Trainer* (3), *Beat Saber* (3), and *Walkabout Mini Golf* (3). The skips were due to discomfort and nausea, further details are provided 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.[1]

[1]The full study script is provided as supplementary material.

Table 3: Applications used by study participants. Categorizations are from Tian et al [39].

Title	Description	Menu Interaction	Selection & Manipulation	Locomotion	Bimanual
TheBlu [46]	Exploration of an underwater scene, uses smooth joystick locomotion.	Near	Near	Near	None
Tilt Brush [20]	3D drawing, needs bimanual input and teleport locomotion.	Near	Near	Near, Far	Required
Walkabout Mini Golf [27]	Mini golf game, uses precise arm motion and teleport locomotion.	Far	Near	Near, Far	None
Space Pirate Trainer [23]	First-person shooter game, needs large body motions to dodge bullets.	Far	Far	Near	Optional
Beat Saber [3]	Rhythm game requiring arm swings, body motion to dodge hazards.	Far	Near	Near	Required

Table 4: 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**

3.4 Results (RQ1)

We analyzed session data using open and axial coding [11]. The first author read and open-coded the participant dialogue and facilitator notes, using inductive analysis [28] to identify common themes. Two authors independently coded the dataset and discussed disagreements to refine the codes. We divide the themes into three categories (Table 4) and discuss them in the subsections below.

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 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"*.

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*

Figure 2: P7 leaning out of their wheelchair to avoid hazards in *Beat Saber*.

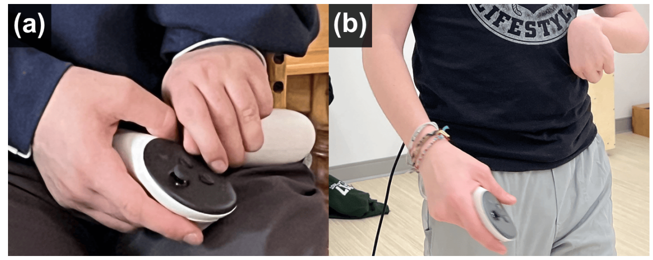

Figure 3: Examples of controller usage strategies for bimanual applications: (a) P10 uses his lap to keep his left controller steady for the bimanual interactions in *Tilt Brush*; (b) P5 plays *Beat Saber* with one arm, implicitly required to play the one-handed challenge mode.

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 [41], 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 to play *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, 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 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 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 two demos of *Walkabout Mini Golf* and one demo of *Beat Saber* 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.

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 reduced 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].

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. P10's limited hand dexterity and discomfort caused him to hold the controller with the buttons rotated away. Overall, controller grasp issues as well as dexterity challenges prompted participants to skip six total demos for *Space Pirate Trainer* (3), *Beat Saber* (2), and *Walkabout Mini Golf* (1).

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 wheelchairs 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.

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 [49], use in constrained spaces [40], and more accessible freehand gestures [39]. Still, little work proposes a general method for overcoming these motor issues in a highly customizable way.

Previous work argues for *ability-based design* [47, 48], a design methodology that places the burden of adaptation on the system rather than the user. Ability-based design emphasizes system-level adaptations, focusing specifically on what the user *can* do, rather than what they *cannot* do. Addressing motor input issues in an ability-based way requires the system to adapt to the user's range of motion, but individually adapting to every possible range of motion could be impractical for developers to implement. Previous work examining accessible input [45] suggests that general-purpose accessibility solutions should be designed for broader categories of input configuration instead of for every individual variation. As such, a concise categorization of body motion could enable a more feasible way to create versatile and customizable accessibility solutions that do not require individualized adaptations. With this in mind, our results motivate a categorization of the kinds of spatial motions that might be possible for a given user, as well as motions required by a given application.

4 MotionBlocks

The formative study identified that many VR arm and body movements are inaccessible. In the spirit of ability-based design [48], our goal is to design an adaptable system to match the input ability of an individual using a mapping between geometric categories of spatial input [45].

Inspired by previous work in 3D input remapping [24, 29, 34, 44], 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 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. [9, 15, 34, 36, 44]), 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 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.

4.1 Motion Primitives (RQ2)

We describe six motion primitives derived from VR body motions observed in our study (Figure 5), along with the dimensionality of each primitive's associated input space. We focus on upper body

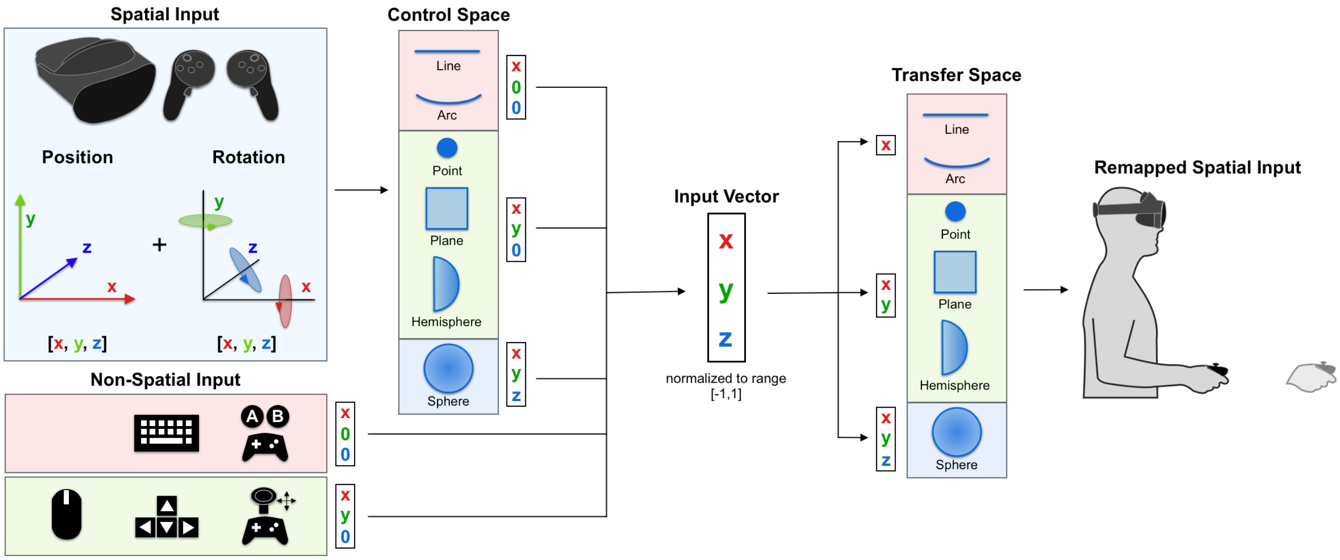

Figure 4: Our implementation of MotionBlocks for customizable VR input remapping. Physical motion along a control-space motion primitive is collapsed into a normalized input vector, which is then mapped to movement along a transfer-space motion primitive. The input vector normalization step also enables non-spatial 1-dimensional or 2-dimensional input devices to be remapped to transfer-space primitives.

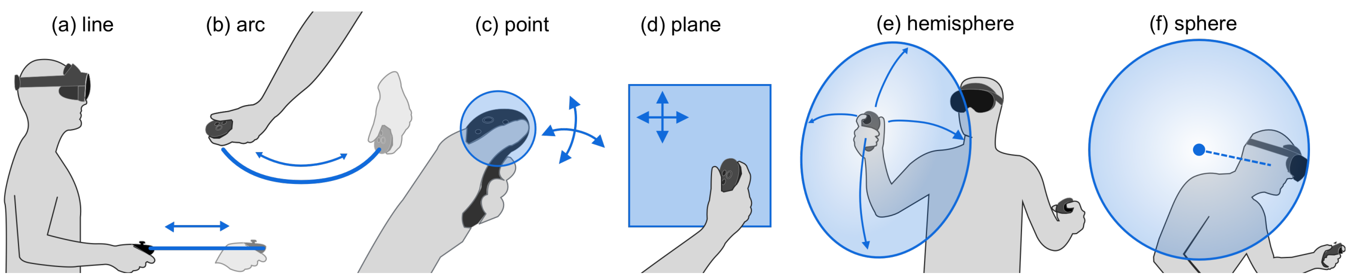

Figure 5: Motion primitives are geometric representations of potentially inaccessible movements in VR applications: (a) *Line* for 1D translation toward an object; (b) *Arc* for 1D rotation around a point with a given radius; (c) *Point* for 2D rotation without translation; (d) *Plane* for bounded 2D translation; (e) *Hemisphere* for 2D rotation around a point with a given radius; (f) *Sphere* for 3D translation.

movement (arms, head, torso) as it would provide the most immediate benefit for our selected applications, but the concept can be extended to support other types of body movements like legs.

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.

4.2 Input Vector

A key benefit of motion primitives is their ability to simplify complex 3D motor interactions, reducing their required input dimensions [7, 26] 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). Regardless of dimensionality, all motion primitives in our implementation allowed for modification in how control-space axes map to transfer-space axes. For example, a control-space *Line* primitive could be configured to translate hand motion in the body's local Y-axis into X-axis motion in its corresponding transfer-space primitive.

4.2.1 One-Handed Bimanual Input. A normalized input vector enables a key function for some accessible input configurations: mapping multiple virtual controllers to the physical motion and button inputs of only one hand. One control-space primitive can be mapped to multiple transfer-space primitives simultaneously. For example, the physical motion of one controller, captured by a single control-space primitive, can be mapped to the transfer-space primitives of both virtual controllers, meaning both virtual controllers move in response to only one moving physically. This mapping creates a symmetric one-handed bimanual interaction [49].

Our system also supports asymmetric one-handed bimanual interactions [49]. Because tracked motion (using a control-space primitive) and physical input (like joysticks or buttons) can provide separate input vectors, they can be mapped to separate transfer-space primitives. As a result, one physical device can provide input to multiple transfer-space primitives simultaneously. For example: the motion of the right physical controller can be mapped to a control-space primitive, and subsequently to a transfer-space primitive for the right virtual controller. At the same time, input from the right controller joystick can be mapped to a transfer-space primitive for the left virtual controller. As a result, the right controller's physical motion controls the motion of the right virtual controller, and its joystick controls the motion of the left virtual controller.

4.2.2 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.

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. To test the general approach, in our study, we implemented a Unity VR 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 see in VR how the motion primitives were placed relative to their body, as well as how their hand motion was being remapped. Participants could also move and resize primitives directly within the VR configuration application.

4.4 System

Our implementation of MotionBlocks depends on a custom-developed SteamVR input driver, similar to the approach taken by OpenVR Input Emulator[2]. This driver uses DLL injection of the SteamVR controller driver via MinHook[3] 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

[2]https://github.com/matzman666/OpenVR-InputEmulator
[3]https://github.com/TsudaKageyu/minhook

Table 5: Games chosen by each participant in the MotionBlocks study.

ID	Walkabout Mini Golf	Beat Saber	Tilt Brush	Space Pirate Trainer	TheBlu
P1	✓	✓			✓
P2		✓		✓	✓
P5		✓	✓	✓	
P6			✓	✓	✓
P7	✓	✓		✓	
P8		✓		✓	✓
P9	✓	✓		✓	
P10		✓		✓	✓
Total	3	7	2	7	5

between them. Our implementation optionally applied the 1€ filter [8] 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 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.

5.1 Participants

We re-recruited 8 participants from the formative study (all but P3 and P4). Their mobility limitations can be found in Table 2. Sessions took place at a local disability outreach foundation, or in the participants' homes. The protocol was approved by an ethics review board.

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.

5.2.1 Task. Following informed consent and a brief re-introduction to the VR hardware, facilitators and participants started a 10-minute collaborative configuration and familiarization process. The participant donned the headset and entered a simplified virtual environment where they worked with the facilitator to identify and adjust the control-space primitive that best aligned with their natural range of motion. Once the configuration was complete, they were given additional time to familiarize themselves with how their movements were translated into virtual inputs in VR environment.

Participants freely chose and used three of the five applications (Table 5) for at least 10 minutes each, with facilitators enabling and disabling the remapping system as necessary for UI navigation. Participants would also provide feedback about both the control-space and transfer-space primitives, with the facilitators able to make real-time adjustments to remapping configurations if necessary. Participants could adjust or swap between transfer-space primitives if a given configuration was insufficient or uncomfortable. To facilitate direct comparison, participants completed rounds of gameplay both with the remapping system applied and in a baseline condition where no remapping was available. If a participant experienced discomfort during trials with MotionBlocks enabled, the facilitator could instantly turn off the remapping. Fortunately, this did not occur in our study; participants kept MotionBlocks remapping enabled for the vast majority of the time where it was available.

At the end of the study, participants completed the first half of the NASA-TLX questionnaire [21] 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.

5.2.2 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 3): *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 *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

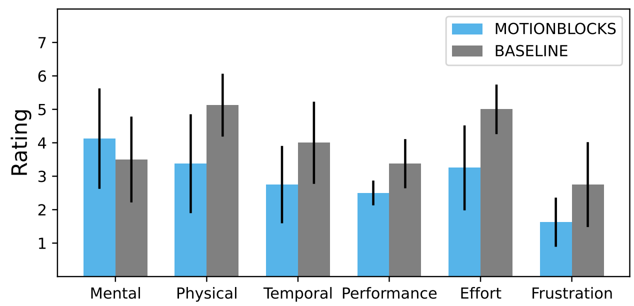

Figure 6: Results from the NASA-TLX questions for MOTIONBLOCKS and BASELINE (no input remapping). Error bars represent 95% CI.

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.

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.

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).

5.3.1 NASA-TLX. Configurable input remapping reduced perceived workload across several categories (Figure 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*.

5.3.2 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 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 7b). Other participants who could provide smaller 3-dimensional motions chose to use *Sphere* primitives to amplify their comfortable motions (Figure 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, the low fidelity might have caused issues matching participants' physical or intended virtual motion ranges. As a result, participants mostly preferred providing two-dimensional or three-dimensional input which MotionBlocks then upscaled. 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].

5.3.3 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 [44] 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.

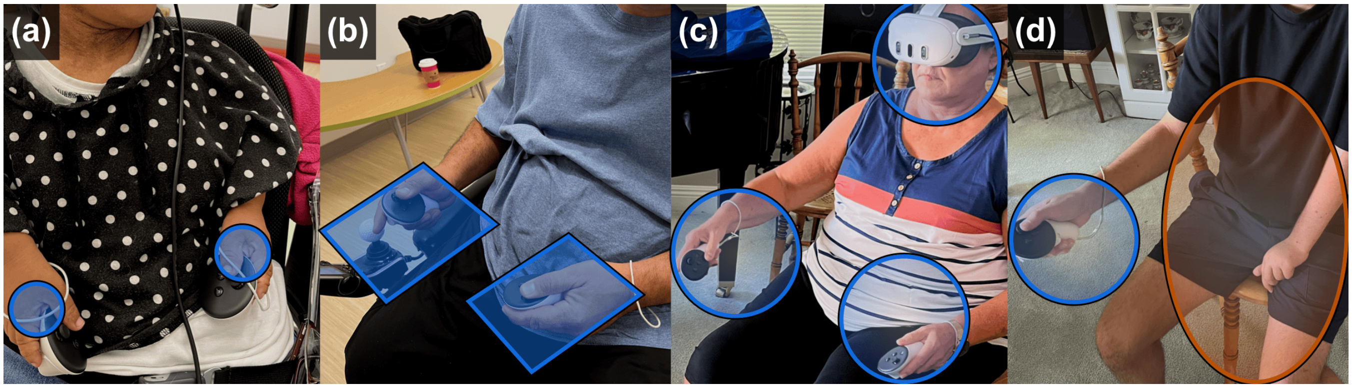

Figure 7: Examples of control-space motion primitive configurations: (a) P1 using *Point* primitives tracking 2D wrist rotation; (b) P6 using *Plane* primitives tracking 2D translation forward and sideways across their lap; (c) P9 using *Sphere* primitives tracking smaller, more comfortable 3D movements 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.

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"*.

5.3.4 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 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 [49] 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"*.

5.3.5 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*.

5.3.6 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"*.

5.4 Discussion

This goal of this study was to answer **RQ3** by evaluating MotionBlocks as a method to address the kinds of motor capability difficulties observed in the earlier formative study. 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.

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**).

6.1 Design Recommendations

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 relative 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.

Thoughtful consideration of remapping preferences is also critical for increasing the broader applicability of geometric motion remapping. Although our work originally targets accessibility use cases, thoughtful consideration of user preferences enables MotionBlocks remapping to address a variety of situational impairments, like working in constrained spaces [40], being seated at a desk [52], or having to switch between using VR and desktop input devices [43].

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 increased 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.

6.2 Limitations

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.

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.

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.

6.2.4 Evaluating Accuracy. Participants in the final study aimed to find motion primitive configurations that were "good enough" to use a VR application. Even when the dimensions of control and transfer space primitives were very different, participants did not comment on a loss of accuracy. However, the mismatch between physical movement and virtual movement created by motion remapping likely affects accuracy in some way. Especially when high precision is critical, future work should explore the effect of primitive shape mismatches on accuracy.

6.3 Future Work

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

6.3.1 Native Low-Level Support. The implementation used in our study relied on a DLL injection into the current SteamVR Input driver. This could lead to instability or reduced functionality as the SteamVR Input API evolves and the undocumented aspects we leverage in the DLL change. In the future, SteamVR could adopt MotionBlocks remapping as a supported feature, and integrate it formally into the SteamVR Input API (and associated DLL). Adopting MotionBlocks natively, in addition to the accessibility benefits, can also make VR usage more feasible for a variety of use cases. Because MotionBlocks remapping reduces the amount of motion necessary for VR, it could reduce fatigue in extended sessions or encourage more ergonomic movements. Likewise, MotionBlocks remapping could make VR usage more feasible in confined spaces like desks, plane seats, or smaller rooms.

6.3.2 Integrated User Interface. Our current implementation uses a separate Unity application to choose primitives, configure their size and position, and activate them in the DLL to be used in a native application. This configuration interface was not designed for end users. In our study, the facilitator operated the interface and the participant visualized the size and position of primitives in VR. To show them in VR, we had to exit their desired game, fully switching to our Unity configuration application. Future work could design an accessible configuration process for users to perform without such a switch, enabling faster re-configuration.

An explicit visualization of the control and transfer primitives in the context of physical controller positions, and within any consumer VR application, would be ideal. This could address the break in proprioceptive cues when physical and virtual controllers are separated, increase body ownership [44], and mitigate issues like accidentally configuring overlapping left and right-hand primitives (experienced by some participants in our study). One technical approach for inserting a motion primitive interface into VR applications is to use more DLL injection. Hartmann et al. [22] demonstrate a method to capture and modify the rendering pipeline in existing 3D games. This could be a way to overlay a configuration and visualization interface onto existing VR applications. Of course, with the cooperation of vendors, such a user interface could be officially integrated into future VR applications or VR operating systems.

6.3.3 Onboarding Experience. In our study, we observed some participants initially struggle to understand how their actual movements translated to the motions of the virtual controllers. There was a learning curve, and facilitator guidance was needed before it felt comfortable and more familiar. In a commercial deployment, an automated onboarding process could be created. It should be short, conducted in situ, and scaffolded to maximize user confidence. VR headset manufacturers have created gamified onboarding experiences for general VR usage[4] and basic controller functions[5]. We imagine a similar experience to introduce and familiarize new users with accessible motion remapping.

6.3.4 Automatic Primitive Activation. Our current implementation requires manual selection and activation of motion primitive mappings. However, future work could investigate methods to use contextual cues, like body position and proximity to virtual objects [22, 43], or even automated recognition of the application. For example, analysis of graphics, system processes, or other features could detect when an application that supports motion remapping is launched, and then load related motion primitive configurations created by the user or a larger community. Motion primitives could then automatically activate or deactivate to match the application state (e.g. in *Beat Saber*, one mapping for playing a song and another mapping for navigating in-game menus).

6.3.5 Extended Customization. To support individuals with a greater variety of dexterity, customization can provide even finer control. Consider someone whose dexterity changes over their range of motion. One control-space primitive would not leverage both the range of motion and different levels of dexterity. Extended customization could introduce methods to blend different control-space primitives according to the degree of arm extension, creating different levels of input amplification in areas around the body. Moreover, there might be situations or mobility limitations that could be addressed with more individual and customized motion primitives. For example, sampling points from the user's movement and using those to construct a "manifold" control primitive representing that individual's range of motion. A similar process could be used to generate manifold transfer primitives from sampled positions of ideal virtual controller positions during an application task (such as playing Beat Saber). Once a bijective mapping is created between these two manifolds, the user would have a highly customized mapping transforming their movement ability to what is required by the application.

6.3.6 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 approach could reduce setup time or inspire more accessible configurations.

6.3.7 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 [16] or Iterative Closest Point [5] to match user movement point clouds to appropriate motion primitives intelligently, or even use a neural network approach like PointNet [35] to classify point clouds into their most appropriate primitive.

[4] *First Encounters* by Meta: https://www.meta.com/experiences/first-encounters/6236169136472090/
[5] *First Steps* by Meta: https://www.meta.com/experiences/first-steps/1863547050392688/

6.3.8 Missing Input Axes. Because motion primitives might differ in input complexity between control 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 infer missing values dynamically 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.

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"*.

Acknowledgments

This work was made possible by NSERC Discovery Grant 2018-05187, Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces".

References

[1] Lisa Anthony, YooJin Kim, and Leah Findlater. 2013. 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)*. Association for Computing Machinery, New York, NY, USA, 1223–1232. https://doi.org/10.1145/2470654.2466158

[2] Rakesh Babu, Rahul Singh, The University of North Carolina at Greensboro, USA, Jai Ganesh, and Infosys Technologies Limited. 2010. Understanding Blind Users' Web Accessibility and Usability Problems. *AIS Transactions on Human-Computer Interaction* 2, 3 (Sept. 2010), 73–94. https://doi.org/10.17705/1thci.00015

[3] Beat Games. [n. d.]. Beat Saber - VR rhythm game. https://beatsaber.com/

[4] Erik Berndt, Dominic Furniss, and Ann Blandford. 2015. Learning Contextual Inquiry and Distributed Cognition: a case study on technology use in anaesthesia. *Cognition, Technology & Work* 17, 3 (Aug. 2015), 431–449. https://doi.org/10.1007/s10111-014-0314-y

[5] P.J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (Feb. 1992), 239–256. https://doi.org/10.1109/34.121791 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[6] Doug A. Bowman and Larry F. Hodges. 1997. 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)*. Association for Computing Machinery, New York, NY, USA, 35–ff. https://doi.org/10.1145/253284.253301

[7] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. 1991. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems* 9, 2 (April 1991), 99–122. https://doi.org/10.1145/123078.128726

[8] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639

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

[10] David M. Cook, Derani Dissanayake, and Kulwinder Kaur. 2019. Virtual Reality and Older Hands: Dexterity and accessibility in hand-held VR Control. In *Proceedings of the 5th International ACM In-Cooperation HCI and UX Conference on - CHIuXiD'19*. ACM Press, Jakarta, Surabaya, Bali, Indonesia, 147–151. https://doi.org/10.1145/3328243.3328262

[11] Juliet Corbin and Anselm Strauss. 2014. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications. Google-Books-ID: hZ6kBQAAQBAJ.

[12] Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarcar, and Ian Williams. 2023. Inclusive AR/VR: accessibility barriers for immersive technologies. *Universal Access in the Information Society* 23, 1 (Feb. 2023), 59–73. https://doi.org/10.1007/s10209-023-00969-0

[13] Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarcar, and Ian Williams. 2023. Inclusive Augmented and Virtual Reality: A Research Agenda. *International Journal of Human–Computer Interaction* (Aug. 2023), 1–20. https://doi.org/10.1080/10447318.2023.2247614

[14] Kevin Doherty and Gavin Doherty. 2018. The construal of experience in HCI: Understanding self-reports. *International Journal of Human-Computer Studies* 110 (Feb. 2018), 63–74. https://doi.org/10.1016/j.ijhcs.2017.10.006

[15] Tiare Feuchtner and Jörg Müller. 2018. 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)*. Association for Computing Machinery, New York, NY, USA, 31–43. https://doi.org/10.1145/3242587.3242594

[16] Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (June 1981), 381–395. https://doi.org/10.1145/358669.358692

[17] Rachel L. Franz, Jinghan Yu, and Jacob O. Wobbrock. 2023. Comparing Locomotion Techniques in Virtual Reality for People with Upper-Body Motor Impairments. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York NY USA, 1–15. https://doi.org/10.1145/3597638.3608394

[18] Kathrin Gerling, Patrick Dickinson, Kieran Hicks, Liam Mason, Adalberto L. Simeone, and Katta Spiel. 2020. Virtual Reality Games for People Using Wheelchairs. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–11. https://doi.org/10.1145/3313831.3376265

[19] Kathrin Gerling and Katta Spiel. 2021. 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)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445196

[20] Google. [n. d.]. Tilt Brush by Google. https://www.tiltbrush.com/

[21] Sandra G. Hart. 1986. NASA Task Load Index (TLX). https://ntrs.nasa.gov/citations/20000021487 NTRS Author Affiliations: NASA Ames Research Center NTRS Document ID: 20000021487 NTRS Research Center: Ames Research Center (ARC).

[22] Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D. Wilson. 2019. RealityCheck: Blending Virtual Environments with Situated Physical Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300577

[23] I-Illusions. [n. d.]. Space Pirate Trainer on Steam. https://store.steampowered.com/app/418650/Space_Pirate_Trainer/

[24] Mohamed Kari and Christian Holz. 2023. HandyCast: Phone-based Bimanual Input for Virtual Reality in Mobile and Space-Constrained Settings via Pose-and-Touch Transfer. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–15. https://doi.org/10.1145/3544548.3580677

[25] Rachel L. Franz, Sasa Junuzovic, and Martez Mott. 2021. Nearmi: A Framework for Designing Point of Interest Techniques for VR Users with Limited Mobility. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Virtual Event USA, 1–14. https://doi.org/10.1145/3441852.3471230

[26] Jock Mackinlay and Stuart K Card. [n. d.]. A Semantic Analysis of the Design Space of Input Devices. ([n. d.]), 46.

[27] Mighty Coconut. [n. d.]. Walkabout Mini Golf™. https://www.mightycoconut.com/minigolf

[28] Matthew B. Miles, A. Michael Huberman, and Johnny Saldana. 2013. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications. Google-Books-ID: p0wXBAAAQBAJ.

[29] Roberto A. Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: Ergonomic Optimization of Immersive Virtual Environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 759–771. https://doi.org/10.1145/3126594.3126605

[30] Martez Mott, Edward Cutrell, Mar Gonzalez Franco, Christian Holz, Eyal Ofek, Richard Stoakley, and Meredith Ringel Morris. 2019. Accessible by Design: An Opportunity for Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, Beijing, China, 451–454. https://doi.org/10.1109/ISMAR-Adjunct.2019.00122

[31] Martez Mott, John Tang, Shaun Kane, Edward Cutrell, and Meredith Ringel Morris. 2020. "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. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Virtual Event Greece, 1–13. https://doi.org/10.1145/3373625.3416998

[32] Maia Naftali and Leah Findlater. 2014. 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)*. Association for Computing Machinery, New York, NY, USA, 209–216. https://doi.org/10.1145/2661334.2661372

[33] Shanmugam Muruga Palaniappan, Ting Zhang, and Bradley S. Duerstock. 2019. Identifying Comfort Areas in 3D Space for Persons with Upper Extremity Mobility Impairments Using Virtual Reality. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Pittsburgh PA USA, 495–499. https://doi.org/10.1145/3308561.3353810

[34] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. 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)*. Association for Computing Machinery, New York, NY, USA, 79–80. https://doi.org/10.1145/237091.237102

[35] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. https://doi.org/10.48550/arXiv.1612.00593 arXiv:1612.00593 [cs].

[36] I. Scott Mackenzie and Stan Riddersma. 1994. Effects of output display and control—display gain on human performance in interactive systems. *Behaviour & Information Technology* 13, 5 (Sept. 1994), 328–337. https://doi.org/10.1080/01449299408914613 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01449299408914613.

[37] Laura South, Caglar Yildirim, Amy Pavel, and Michelle Borkin. 2024. Barriers to Photosensitive Accessibility in Virtual Reality. https://doi.org/10.31219/osf.io/ywqba

[38] Felix J. Thiel and Anthony Steed. 2021. "Lend Me a Hand" – Extending the Reach of Seated VR Players in Unmodified Games Through Remote Co-Piloting. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, Lisbon, Portugal, 214–219. https://doi.org/10.1109/VRW52623.2021.00047

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

[40] Wen-Jie Tseng, Samuel Huron, Eric Lecolinet, and Jan Gugenheimer. 2023. FingerMapper: Mapping Finger Motions onto Virtual Arms to Enable Safe Virtual Reality Interaction in Confined Spaces. https://doi.org/10.1145/3544548.3580736 arXiv:2302.11865 [cs].

[41] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks. 1999. Walking > walking-in-place > flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 359–364. https://doi.org/10.1145/311535.311589

[42] Kirby Weerink. 2017. Developing virtual reality in forensic mental healthcare : a contextual inquiry. https://essay.utwente.nl/74243/ Publisher: University of Twente.

[43] 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 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3613904.3642358

[44] Johann Wentzel, Greg d'Eon, and Daniel Vogel. 2020. Improving Virtual Reality Ergonomics Through Reach-Bounded Non-Linear Input Amplification. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. https://doi.org/10.1145/3313831.3376687

[45] Johann Wentzel, Sasa Junuzovic, James Devine, John Porter, and Martez Mott. 2022. Understanding How People with Limited Mobility Use Multi-Modal Input. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–17. https://doi.org/10.1145/3491102.3517458

[46] Wevr. [n. d.]. theBlu. https://wevr.com/theblu

[47] Jacob O. Wobbrock, Krzysztof Z. Gajos, Shaun K. Kane, and Gregg C. Vanderheiden. 2018. Ability-based design. *Commun. ACM* 61, 6 (May 2018), 62–71. https://doi.org/10.1145/3148051

[48] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. *ACM Transactions on Accessible Computing* 3, 3 (April 2011), 1–27. https://doi.org/10.1145/1952383.1952384

[49] Momona Yamagami, Sasa Junuzovic, Mar Gonzalez-Franco, Eyal Ofek, Edward Cutrell, John R. Porter, Andrew D. Wilson, and Martez E. Mott. 2022. Two-In-One: A Design Space for Mapping Unimanual Input into Bimanual Interactions in VR for Users with Limited Movement. *ACM Transactions on Accessible Computing* 15, 3 (July 2022), 23:1–23:25. https://doi.org/10.1145/3510463

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

[51] Bei Yuan, Eelke Folmer, and Frederick C. Harris. 2011. Game accessibility: a survey. *Universal Access in the Information Society* 10, 1 (March 2011), 81–100. https://doi.org/10.1007/s10209-010-0189-5

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