SwitchSpace: Understanding Context-Aware Peeking Between VR and Desktop Interfaces

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

Fraser Anderson
Autodesk Research
Canada
fraser.anderson@autodesk.com

George Fitzmaurice
Autodesk Research
Canada
george.fitzmaurice@autodesk.com

Tovi Grossman
Department of Computer Science,
University of Toronto
Canada
tovi@dgp.toronto.edu

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

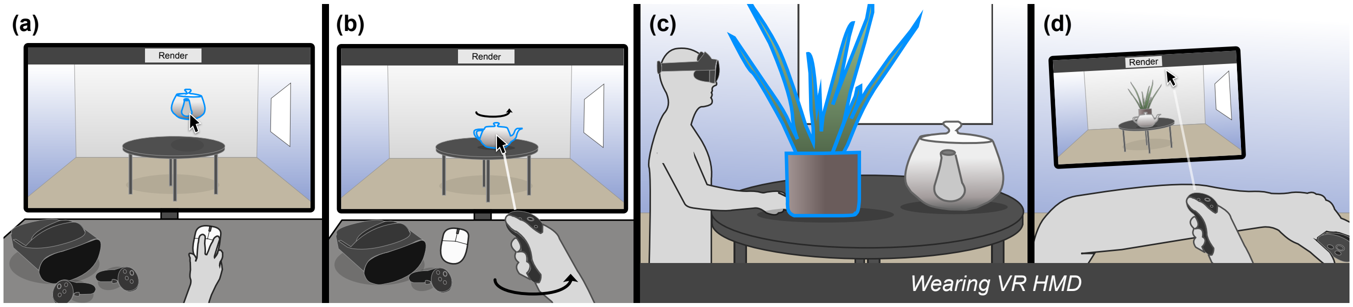

Figure 1: An envisioned desktop/VR 3D environment rendering application using "peeking" techniques. (a) The desktop interface, which supports interaction from the mouse and the VR controllers. The user brings the teapot into the scene using the mouse (*Desktop*). (b) The user switches to the VR controller 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).

ABSTRACT

Cross-reality tasks, like creating or consuming virtual reality (VR) content, often involve inconvenient or distracting switches between desktop and VR. An initial formative study explores cross-reality switching habits, finding most switches are momentary "peeks" between interfaces, with specific habits determined by current context. The results inform a design space for context-aware "peeking" techniques that allow users to view or interact with desktop from VR, and vice versa, without fully switching. We implemented a set of peeking techniques and evaluated them in two levels of a cross-reality task: one requiring only viewing, and another requiring input and viewing. Peeking techniques made task completion faster, with increased input accuracy and reduced perceived workload.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '24, May 11–16, 2024, Honolulu, HI, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642358

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; **Virtual reality**; **Mixed / augmented reality**.

KEYWORDS

transitional interfaces, controlled experiments, interaction techniques, Virtual Reality

ACM Reference Format:
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), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3613904.3642358

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 [13] or an articulated 2D display as a viewing window [11, 43]. Previous work also explores "cross-reality blending" approaches to unify the user's real and virtual environments, such as presenting real objects in VR [24, 31, 40, 53] or exiting VR more fluidly by fading in the real environment [26]. 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 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 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.

2 BACKGROUND AND RELATED WORK

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. [9] and Marriott et al. [30] for a wider review. We discuss specific related work for cross-reality interfaces, including real-virtual alignment, interaction space conversions, and transitional interfaces.

2.1 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 [14, 18, 27, 34, 45], aligning real and virtual worlds can enable more comfortable interaction with real-world objects [10, 40]. Hartmann et al. [24] 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 [47] 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 [2, 37] which can be exploited to provide touch feedback [12] or increase comfort [49].

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 [22, 31, 35].

Aligning the real and virtual *workspace* can also be beneficial. Zielasko et al. [53, 54] 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. [46] 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. [53] and Wagner Filho et al. [46] for improving comfort and functionality, and further explores real-virtual aligned input expanding on work like McGill et al. [31]. 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.

2.2 2D Interactions Within 3D

Many cross-reality workflows make use of 2-dimensional input within 3-dimensional environments. For example, Kim et al. [25] explored cursor movement techniques for spatial augmented reality, finding that using a head-mounted cursor with a perspective-based targeting technique [33] was superior for long distances. Similarly, Zhou et al. [51] 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. [13] 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 [43] 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. [42] and Arora et al. [1] 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. [6], Zhu et al. [52] 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 [48] 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 [42, 48]. 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.

2.3 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. [32] 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. [39] 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. [21] 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. [7] 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. [29] 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 [15] 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. [38] 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. [26] evaluated several 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. [16] 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. [20] 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. [36] 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. [31] 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. [11] 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.

Table 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			

Transitional interfaces bridge the gap between disparate input and output mechanisms. George et al. [16] explore transitions between VR and reality, but do not focus on transitions between input devices or the role of context within their design space. Serrano et al. [39] provide guidelines for transitioning input devices for AR, while McGill et al. [31] and Knibbe et al. [26] 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. [11] 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.

2.4 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.

Table 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."*

3 FORMATIVE STUDY

We conducted a formative study with 24 frequent VR users (demographic details in Table 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 2) about their preferences and habits between VR and desktop use.

3.1 Results

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

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"*.

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].

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. [26], 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"*.

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 [26, 31, 52]. The *Transitioning* feedback provides additional design considerations for improving VR-desktop cross-device interactions.

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.

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 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.

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

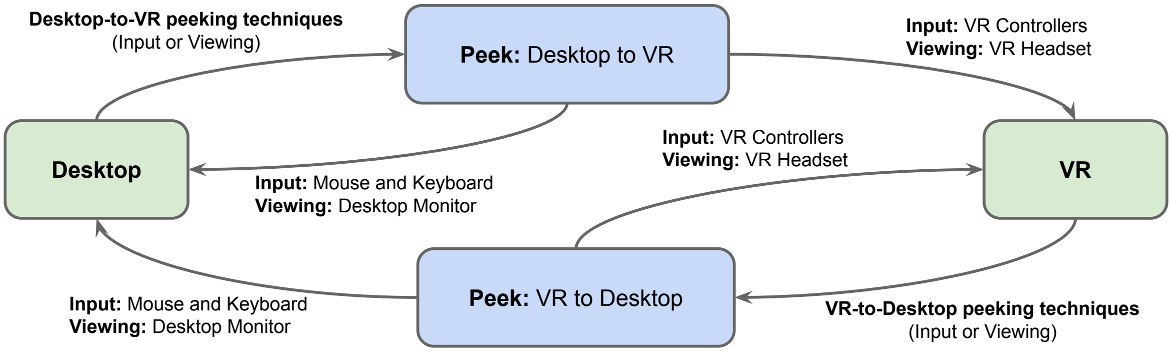

Figure 2: The state machine used in our description of a context-aware cross-reality interface, between primary states *Desktop* and *VR* (green). Changes in input or viewing device trigger state transitions. Changing from the standard input devices or the standard viewing device for a primary state brings the state machine to a secondary"peek" state (blue).

4.1 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 [50]. 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.

4.2 Context State Machine

We represent the design space as a collection of primary and secondary states (Figure 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.

4.3 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*).

4.3.1 *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 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.

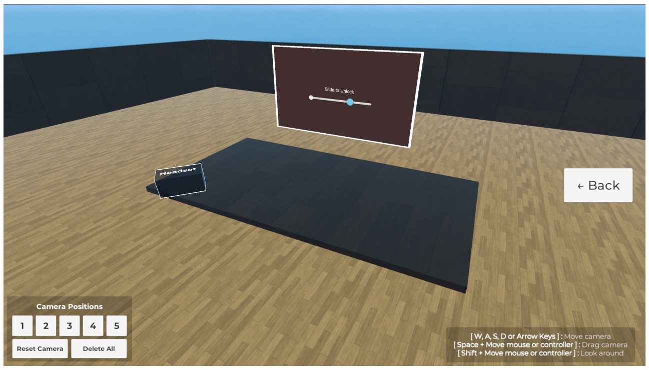
Slide to Unlock
Headset
← Back
Camera Positions
1
2
3
4
5
Reset Camera
Delete All
[W, A, S, D or Arrow Keys] : Move camera
[Space + Move mouse or controller] : Drag camera
[Shift + Move mouse or controller] : Look around

Figure 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).

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 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). The virtual controller's ray intersects with a collision plane aligned with the real monitor, at the coplanar 2D point P_c (Figure 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 4b). The cursor is placed at P_w, and scaled to maintain visual angular size (Figure 4c).

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

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 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 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 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 4.3.3). 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 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*.

4.3.3 Real-Virtual Alignment and Calibration. In addition to real-virtual alignment increasing spatial awareness and decreasing discomfort when transitioning [24, 26], 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.

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 6.3.

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

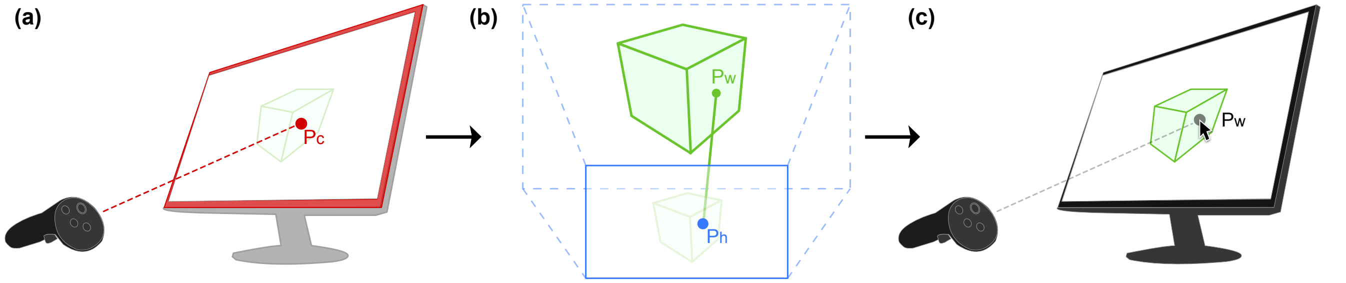

Figure 4: Our approach for using the VR controller to point through the real monitor, for *Desktop*-to-*VR* peeking. (a) In the virtual environment, the controller's raycast intersects a collision plane aligned with the real monitor, at point P_c. (b) P_c is converted to a screen-space point P_h, and 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.

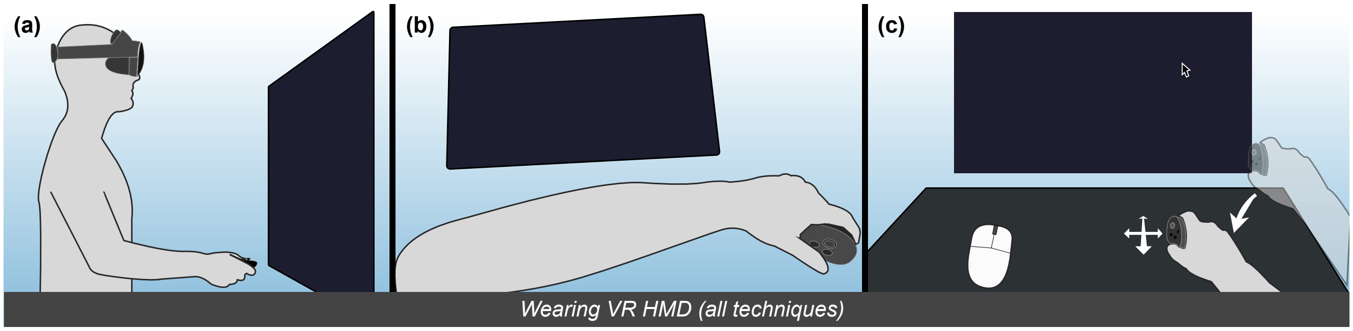

Figure 5: Interactive views of the desktop UI, for peeking from *VR* to *Desktop*: (a) a world-anchored view; (b) a body-anchored view; and (c) a view on the desk aligned with the real monitor, summoned by the mouse or a mouse-like movement of the *VR* controller.

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

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.

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.

5.3.1 Math Task. Participants were presented with a multiple-choice addition question with a missing variable X (Figure 6). The math question, the value of X, and the correct multiple-choice answer were decided randomly every trial. All 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.

5.3.2 Unlock Subtask. In half of the trials, math questions and answers were hidden behind an initial unlock subtask (Figure 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.

5.3.3 Task Placement. In VR, UI panels (for either math questions or answers) were placed in one of four positions (Figure 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.

5.3.4 NASA-TLX and Post-Questionnaire. After each experiment condition, participants answered the first half of the NASA-TLX

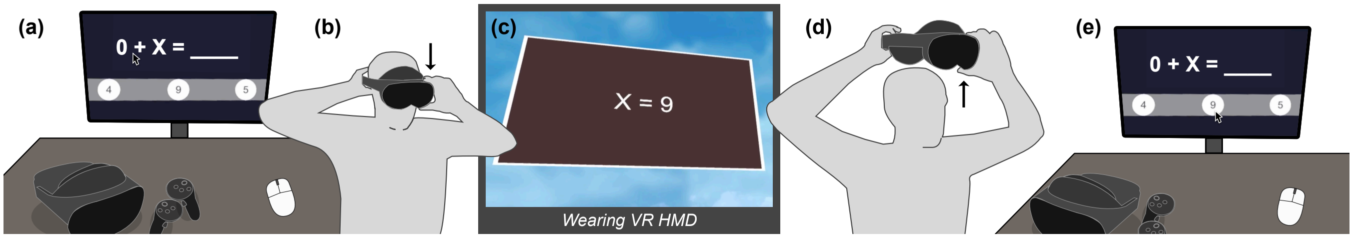

Figure 6: The view-only peek variation of the study task, in the baseline technique, and the desktop-vr direction. (a) The participant views the math problem on the desktop. (b) They don the headset. (c) They see the missing variable in *VR*. (d) They remove the headset. (e) They answer the math problem on the desktop.

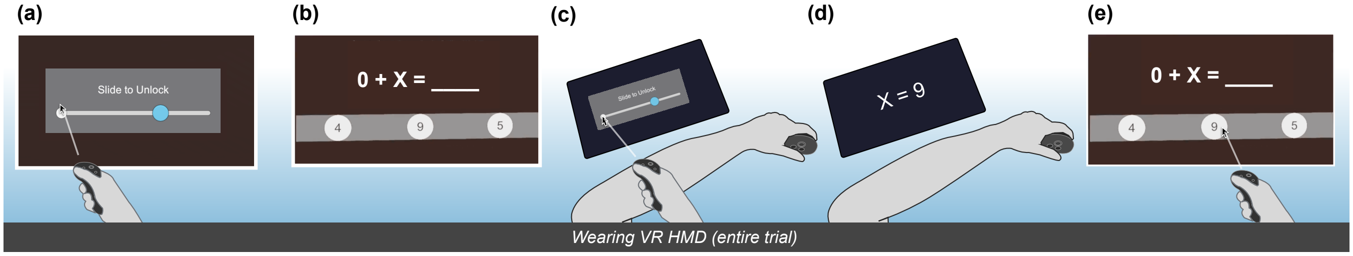

Figure 7: The input+view peek variation of the study task, in the switchspace technique, and the vr-desktop direction. (a) The participant unlocks the VR panel by completing the unlock subtask. (b) They see the math problem in *VR*. (c) They use a VR-to-Desktop peeking technique (this shows one of many 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*.

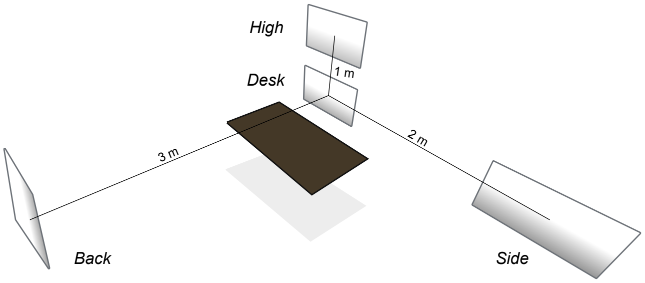

Figure 8: The positions of the task panels in the virtual environment: desk, high, side, and back.

questionnaire [23] for perceived workload. At the end of the experiment, participants completed a post-questionnaire (Table 3) for feedback about peeking techniques, their preferences, and overall experiences.

5.4 Design

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.

Table 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?"*

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* 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*

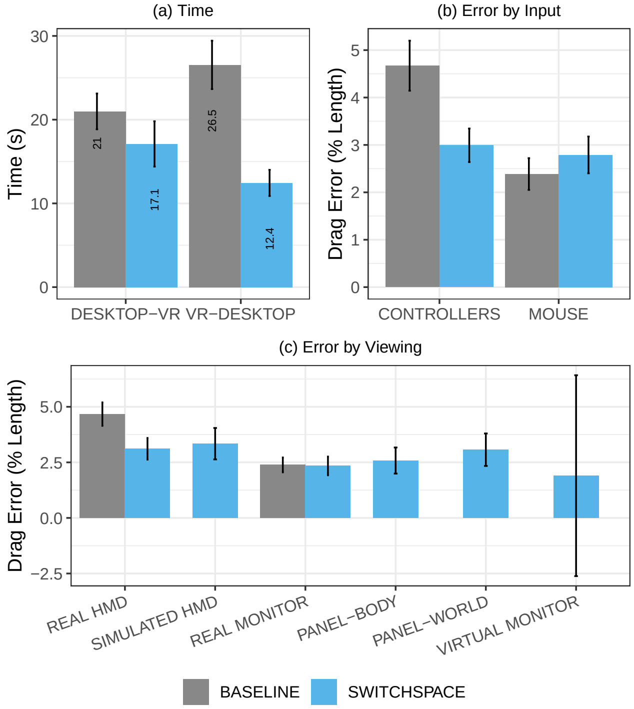

Figure 9: Results: (a) *Time* by TECHNIQUE and DIRECTION; (b) *Drag Error* by TECHNIQUE and INPUT; and (c) *Drag Error* by TECHNIQUE and VIEWING. Error bars represent 95% CI. BASELINE could only use REAL HMD and REAL MONITOR. VIRTUAL MONITOR was rarely chosen, resulting in high variance.

Trial is the number of changes in the participant's *input* technique, *viewing* technique, or overall *context* (as in Figure 2) in a single trial. Our implementation senses state transitions as described in Sections 4.1–4.2, 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.

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.

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 [41].

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 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).

5.5.3 Drag Error. When using peeking techniques to complete the unlock task, the controllers were as accurate as the mouse (Figure 9b), and the VR headset was as accurate as the desktop display (Figure 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.

5.5.4 Transitions. Participants changed *context* states more often when peeking techniques were available (Figure 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 ($F_{1,461.6} = 124.7, p <$

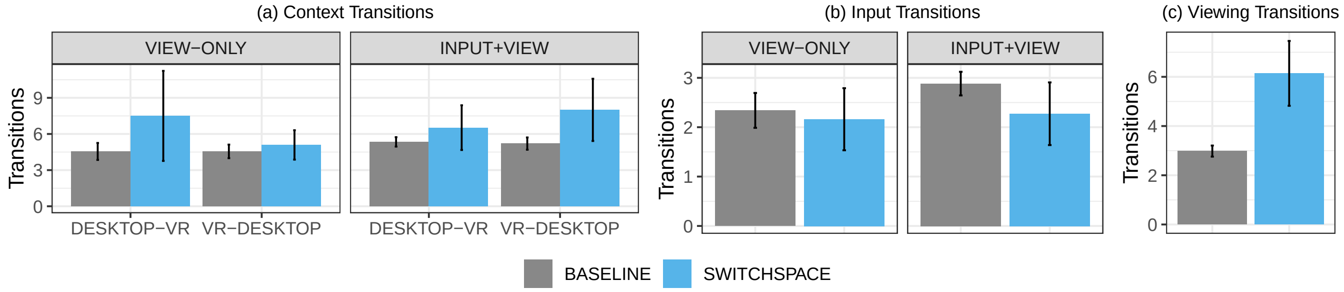

Figure 10: Results: (a) *Context Transitions* by technique, direction, and peek; (b) *Input Transitions* by technique and peek; and (c) *Viewing Transitions* by technique. Error bars represent 95% CI.

.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 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 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 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).

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% ± 10.2% for side, 14.1% ± 7.9% for high, 12.5% ± 8.8% for desk, and 15.6% ± 8.1% for back.

5.5.6 NASA-TLX. Peeking techniques reduced perceived workload across several categories, but varied based on task direction (Figure 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 ($Z = 2.57, p < .05$).

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.

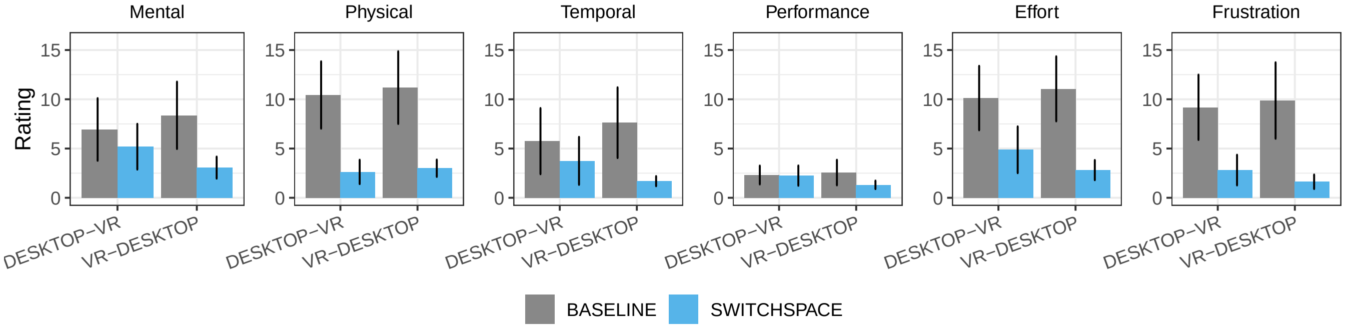

Figure 11: Results from the NASA-TLX questions by TECHNIQUE and DIRECTION. Error bars represent 95% CI.

This caused the large confidence interval in the VIRTUAL MONITOR *Drag Error* result (Figure 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].

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.

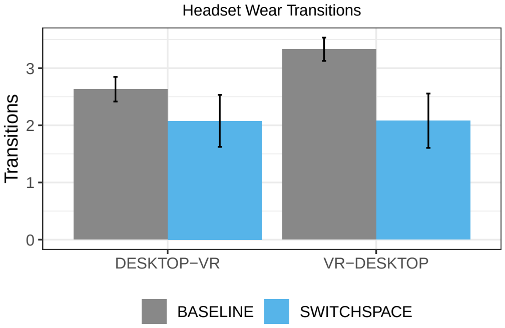

Figure 12: *Headset Wear Transitions* by TECHNIQUE and DIRECTION. Error bars represent 95% CI.

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 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 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 9c. Participants forgetting the value of X mid-trial in the BASELINE technique suggests that in addition to disorientation or discomfort [26], 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.

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.

6.1 Design Recommendations

Our results suggest general design guidelines for cross-reality workflows, building on earlier work in cross-reality blending [20, 26, 31] but with specific focus on transitioning between desktop and VR.

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 [24, 31] or appropriate real-virtual aligned models [40], 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.

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.

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 [11]. 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.

6.2 Limitations

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 [3–5, 7], 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.

6.2.2 Other Usage Contexts. Our work did not consider all variations of cross-reality interaction, like responding to bystanders [14, 27] or different approaches to reality blending [34]. 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.

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. [7]. Future work should evaluate more complex tasks with a larger set of peeking techniques.

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 [20]. 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.

6.3 Future Work

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

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 [7], 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 [8] or walking-in-place [28, 44]) 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.

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.

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. [16] or Do et al. [14].

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 [19]. Some work addresses situational impairments in augmented reality using gaze-based interfaces [17], but little work examines how context can be used to overcome additional situational impairments for VR interfaces. With regard to our study, using a primary interface for a task that requires the temporary use of another interface can be considered a type of situational impairment. Future work could design for such situational impairments more explicitly.

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.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. 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)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3173574.3173759

[2] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2016. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 1968–1979. https://doi.org/10.1145/2858036.2858226

[3] Mayra Donaji Barrera Machuca and Wolfgang Stuerzlinger. 2019. The Effect of Stereo Display Deficiencies on Virtual Hand Pointing. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. https://doi.org/10.1145/3290605.3300437

[4] Anil Ufuk Batmaz, Mayra Donaji Barrera Machuca, Duc Minh Pham, and Wolfgang Stuerzlinger. 2019. 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)*. 585–592. https://doi.org/10.1109/VR.2019.8797975 ISSN: 2642-5254.

[5] Anil Ufuk Batmaz, Rumeysa Turkmen, Mine Sarac, Mayra Donaji Barrera Machuca, and Wolfgang Stuerzlinger. 2023. 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)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3611659.3615686

[6] H. Benko, E.W. Ishak, and S. Feiner. 2005. Cross-dimensional gestural interaction techniques for hybrid immersive environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.* IEEE, Bonn, Germany, 209–327. https://doi.org/10.1109/VR.2005.1492776

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

[8] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. 2016. Point & Teleport Locomotion Technique for Virtual Reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. ACM, Austin Texas USA, 205–216. https://doi.org/10.1145/2967934.2968105

[9] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. 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)*. Association for Computing Machinery, New York, NY, USA, 1–28. https://doi.org/10.1145/3290605.3300792

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

[11] Felipe G. Carvalho, Daniela G. Trevisan, and Alberto Raposo. 2012. Toward the design of transitional interfaces: an exploratory study on a semi-immersive hybrid user interface. *Virtual Reality* 16, 4 (Nov. 2012), 271–288. https://doi.org/10.1007/s10055-011-0205-y

[12] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. 2017. 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)*. Association for Computing Machinery, New York, NY, USA, 3718–3728. https://doi.org/10.1145/3025453.3025753

[13] K. Coninx, F. Van Reeth, and E. Flerackers. 1997. A hybrid 2D/3D user interface for immersive object modeling. In *Proceedings Computer Graphics International.* 47–55. https://doi.org/10.1109/CGI.1997.601270

[14] Youngwook Do, Frederik Brudy, George W. Fitzmaurice, and Fraser Anderson. 2023. Vice VRsa: Balancing Bystander's and VR user's Privacy through Awareness Cues Inside and Outside VR. https://openreview.net/forum?id=gItvr7Xl66

[15] Andreas Fender and Jörg Müller. 2019. 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)*. Association for Computing Machinery, New York, NY, USA, 303–314. https://doi.org/10.1145/3343055.3359715

[16] Ceenu George, An Ngo Tien, and Heinrich Hussmann. 2020. 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)*. 412–424. https://doi.org/10.1109/ISMAR50242.2020.00067 ISSN: 1554-7868.

[17] Yalda Ghasemi and Heejin Jeong. 2022. Using Gaze-based Interaction to Alleviate Situational Mobility Impairment in Extended Reality. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 66, 1 (Sept. 2022), 435–439. https://doi.org/10.1177/1071181322661224 Publisher: SAGE Publications Inc.

[18] Sarthak Ghosh, Lauren Winston, Nishant Panchal, Philippe Kimura-Thollander, Jeff Hotnog, Douglas Cheong, Gabriel Reyes, and Gregory D. Abowd. 2018. NotifiVR: Exploring Interruptions and Notifications in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (April 2018), 1447–1456. https://doi.org/10.1109/TVCG.2018.2793698 Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[19] Mayank Goel, Leah Findlater, and Jacob Wobbrock. 2012. WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 2687–2696. https://doi.org/10.1145/2207676.2208662

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

[21] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. 2015. 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)*. Association for Computing Machinery, New York, NY, USA, 3933–3942. https://doi.org/10.1145/2702123.2702331

[22] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. 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)*. 159–166. https://doi.org/10.1109/VR.2018.8446059

[23] 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).

[24] 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

[25] Daekun Kim, Nikhita Joshi, and Daniel Vogel. 2023. Perspective and Geometry Approaches to Mouse Cursor Control in Spatial Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–19. https://doi.org/10.1145/3544548.3580849

[26] Jarrod Knibbe, Jonas Schjerlund, Mathias Petraeus, and Kasper Hornbæk. 2018. The Dream is Collapsing: The Experience of Exiting VR. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association

for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174057

[27] Yoshiki Kudo, Anthony Tang, Kazuyuki Fujita, Isamu Endo, Kazuki Takashima, and Yoshifumi Kitamura. 2021. Towards Balancing VR Immersion and Bystander Awareness. *Proceedings of the ACM on Human-Computer Interaction* 5, ISS (Nov. 2021), 1–22. https://doi.org/10.1145/3486950

[28] Juyoung Lee, Sang Chul Ahn, and Jae-In Hwang. 2018. A Walking-in-Place Method for Virtual Reality Using Position and Orientation Tracking. *Sensors* 18, 9 (Sept. 2018), 2832. https://doi.org/10.3390/s18092832 Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[29] Feiyu Lu and Yan Xu. 2022. Exploring Spatial UI Transition Mechanisms with Head-Worn Augmented Reality. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–16. https://doi.org/10.1145/3491102.3517723

[30] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H. Thomas. 2018. *Immersive Analytics*. Springer. Google-Books-ID: vaVyDwAAQBAJ.

[31] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. 2015. A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 2143–2152. https://doi.org/10.1145/2702123.2702382

[32] Alexandre Millette and Michael J. McGuffin. 2016. DualCAD: Integrating Augmented Reality with a Desktop GUI and Smartphone Interaction. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 21–26. https://doi.org/10.1109/ISMAR-Adjunct.2016.0030

[33] Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin. 2006. Perspective cursor: perspective-based interaction for multi-display environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Montréal Québec Canada, 289–298. https://doi.org/10.1145/1124772.1124817

[34] Joseph O'Hagan and Julie R. Williamson. 2020. Reality aware VR headsets. In *Proceedings of the 9TH ACM International Symposium on Pervasive Displays*. ACM, Manchester United Kingdom, 9–17. https://doi.org/10.1145/3393712.3395334

[35] Duc-Minh Pham and Wolfgang Stuerzlinger. 2019. HawKEY: Efficient and Versatile Text Entry for Virtual Reality. In *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology (VRST '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3359996.3364265

[36] Fabian Pointecker, Judith Friedl, Daniel Schwajda, Hans-Christian Jetter, and Christoph Anthes. 2022. Bridging the Gap Across Realities: Visual Transitions Between Virtual and Augmented Reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, Singapore, Singapore, 827–836. https://doi.org/10.1109/ISMAR55827.2022.00101

[37] Majed Samad, Elia Gatti, Anne Hermes, Hrvoje Benko, and Cesare Parise. 2019. 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)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300550

[38] Jan-Henrik Schröder, Daniel Schacht, Niklas Peper, Anita Marie Hamurculu, and Hans-Christian Jetter. 2023. 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)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3544548.3580879

[39] Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. 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)*. Association for Computing Machinery, New York, NY, USA, 161–171. https://doi.org/10.1145/2785830.2785838

[40] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 3307–3316. https://doi.org/10.1145/2702123.2702389

[41] R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies* 61, 6 (Dec. 2004), 751–789. https://doi.org/10.1016/j.ijhcs.2004.09.001

[42] Hemant Bhaskar Surale, Aakar Gupta, Mark Hancock, and Daniel Vogel. 2019. TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300243

[43] Michael Tsang, George W. Fitzmzurice, Gordon Kurtenbach, Azam Khan, and Bill Buxton. 2003. Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. Association for Computing Machinery, New York, NY, USA, 698. https://doi.org/10.1145/1201775.882329

[44] 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

[45] Julius Von Willich, Markus Funk, Florian Müller, Karola Marky, Jan Riemann, and Max Mühlhäuser. 2019. You Invaded my Tracking Space! Using Augmented Virtuality for Spotting Passersby in Room-Scale Virtual Reality. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. ACM, San Diego CA USA, 487–496. https://doi.org/10.1145/3322276.3322334

[46] J. A. Wagner Filho, C.m.d.s. Freitas, and L. Nedel. 2018. VirtualDesk: A Comfortable and Efficient Immersive Information Visualization Approach. *Computer Graphics Forum* 37, 3 (2018), 415–426. https://doi.org/10.1111/cgf.13430 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13430.

[47] Chiu-Hsuan Wang, Bing-Yu Chen, and Liwei Chan. 2022. 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)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3526113.3545686

[48] Chiu-Hsuan Wang, Chia-En Tsai, Seraphina Yong, and Liwei Chan. 2020. Slice of Light: Transparent and Integrative Transition Among Realities in a Multi-HMD-User Environment. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 805–817. https://doi.org/10.1145/3379337.3415868

[49] 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

[50] 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

[51] Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–17. https://doi.org/10.1145/3491102.3501884

[52] Fengyuan Zhu and Tovi Grossman. 2020. 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)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376233

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

[54] 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