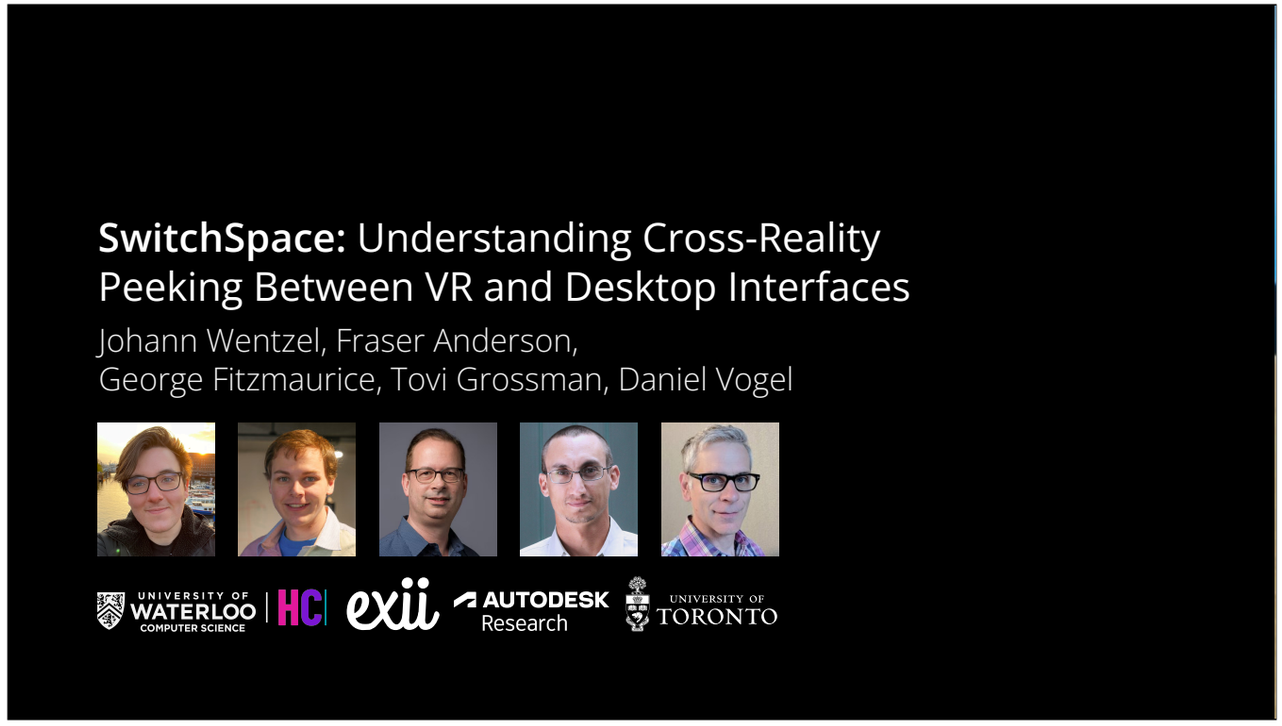


Hi everyone, and thanks for coming to my talk today. I'm Johann Wentzel, I'm a PhD candidate at the University of Waterloo, and I'm here presenting "SwitchSpace, Understanding cross-reality peeking between VR and desktop interfaces". I finished this project with the help of my coauthors: Fraser Anderson, George Fitzmaurice, Tovi Grossman, and Dan Vogel.

TLDR - *too long, didn't read*

Switching between VR and desktop can be cumbersome, especially in workflows that use both.

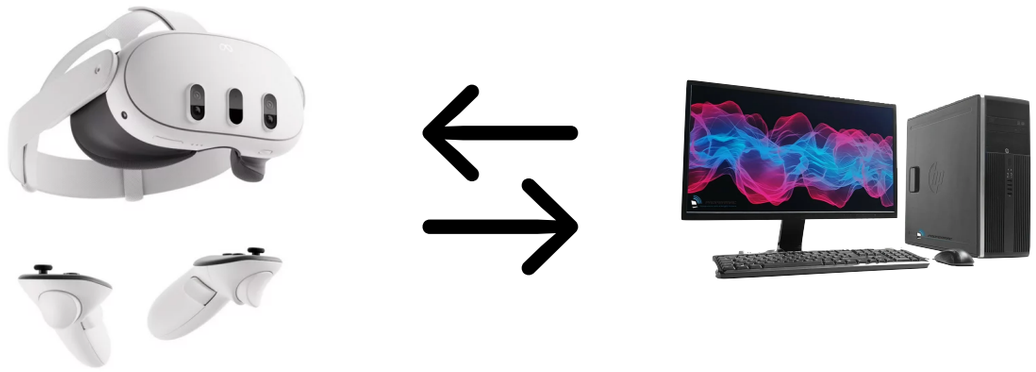
Quick, temporary "peeking" techniques can make these workflows faster, more precise, and more comfortable.

Mostly as a commentary on my own attention span for presentations, I'm going to summarize our findings here.

First, switching between VR and desktop happens often, but it can be cumbersome. Talking with people who use both, the act of switching between VR and desktop makes these workflows harder.
Second, as a remedy, techniques that let you quickly and temporarily "peek" between the two can make these worklows faster, more precise, and more comfortable.

Shocker of the Day

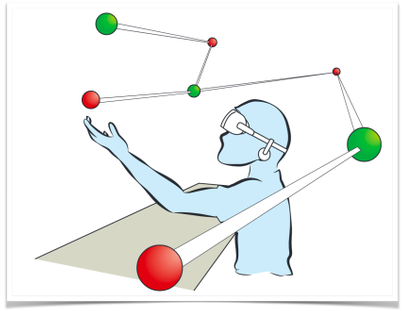
Manipulating hardware is hard!



It might not come as a surprise to you, but moving hardware around is hard! If I was a VR developer who needed to make a bunch of tweaks to Unity, for example, I’d have to completely take off the headset, put down the controllers, and then find my mouse and keyboard to keep going. Likewise going from desktop to VR can be just as difficult. This transition isn’t just about changing physical devices; it's about the mental shift required, which often disrupts the flow of work.

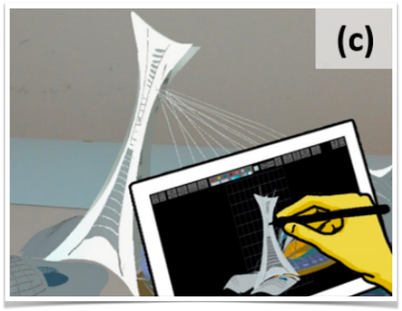
What's Been Explored

Real-Virtual Alignment



DeskVR
Zielasko et al., 2017

2D Interactions in 3D



SymbiosisSketch
Arora et al., 2018

2D-3D Transitional Interfaces

Toward the Design of Transitional Interfaces
Carvalho et al., 2010

So this kind of issue lies in the purview of three pretty related areas. First, there's real-virtual alignment, which brings to mind DeskVR, a system where the user's real and virtual desks were aligned. Next is the idea of using 2D interactions in 3D, which solutions like SymbiosisSketch have explored. Likewise, we've also got 2D-3D transitional interfaces, which Carvalho tested out using a monitor, a wiimote, and a CAVE VR system. What we aren't seeing is work specifically examining the usability implications of transitional interfaces between VR and desktop.

Formative Study

24 VR users (17 use for recreation, 7 for work)

Prefer VR to desktop for **immersion**, **spatial understanding**, and **future-proofing**

Prefer desktop to VR for **precision**, **convenience**, and **multitasking**.

Transitions occurred frequently in **rapid prototyping tasks**.

We started our exploration with a formative study of 24 VR users. We found that people preferred using VR over desktop when they wanted to feel more immersed or needed better 3D understanding of their content. Likewise, people preferred desktop to VR for added precision (like that of a mouse), as well as for convenience and the ability to complete multiple tasks at once.

Talking about transitions between VR and desktop, we saw that this often happened in rapid prototyping tasks, like writing code, previewing 3D models or environments, or other tasks like that.

Formative Study

1. Cross-reality transitions are **temporary**.

2. Temporary peeks **decouple input from output**.

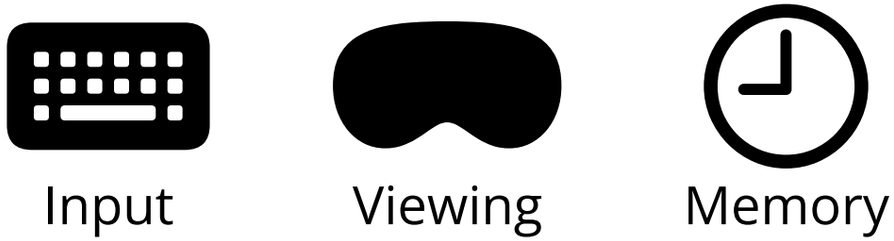
I [transition] if I need to tune the scene or the visualization parameters, so it's a constant VR/non-VR change"

We learned two primary things:

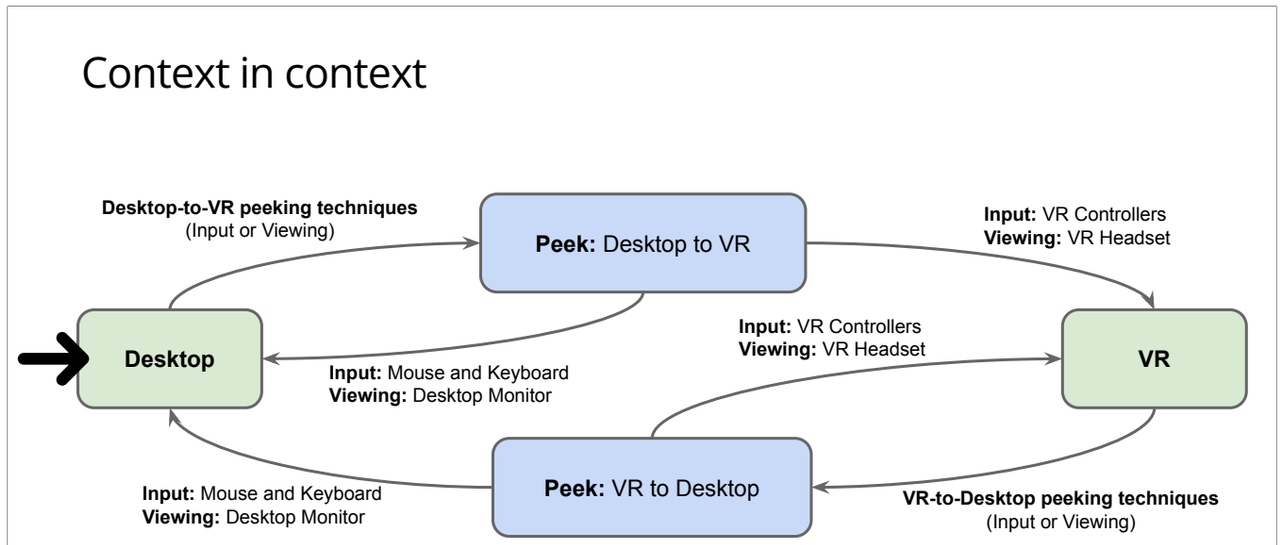
first, these transitions, while discomforting, were QUICK and TEMPORARY. People would switch from a primary modality (think VR for example), to a secondary modality (desktop in this case), then back to the primary. It's a quick peek from one to the other, then back to the first.

Next, these peeks are primarily in a single mode of use. It decouples input from output - people would primarily just put on or take off the headset, or grab the mouse to quickly make tweaks without putting on the headset.

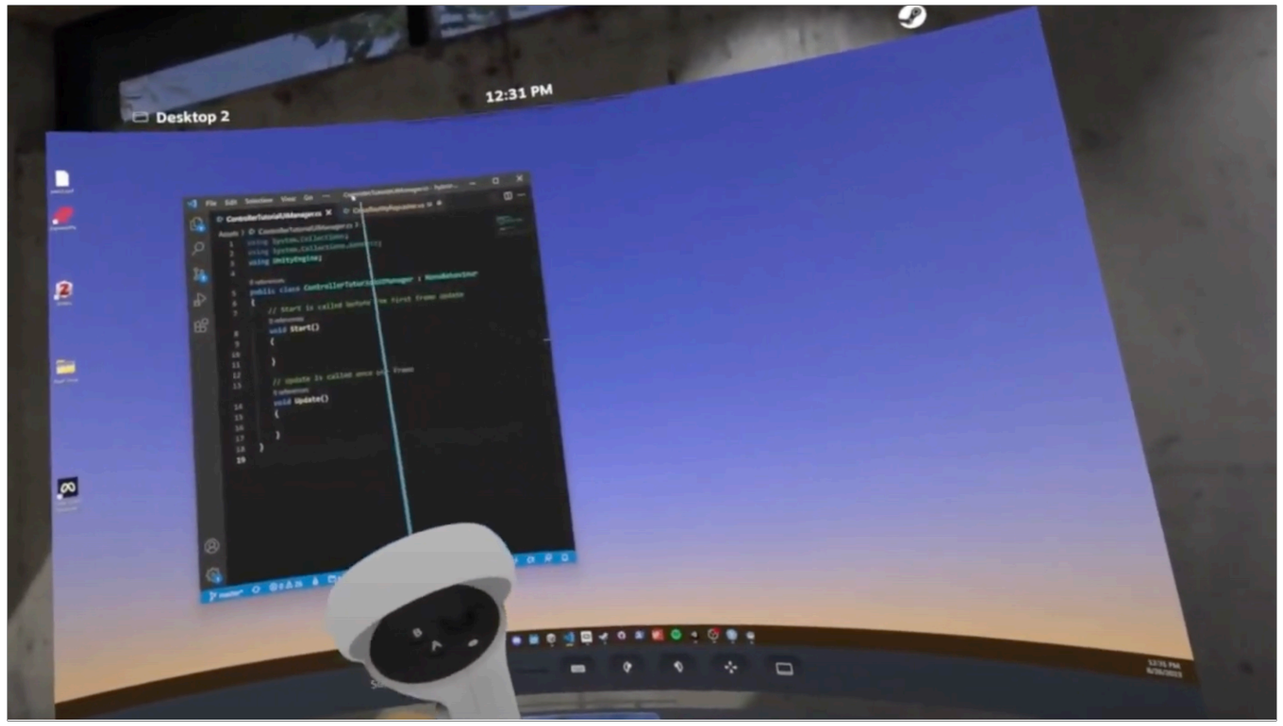
Defining Context



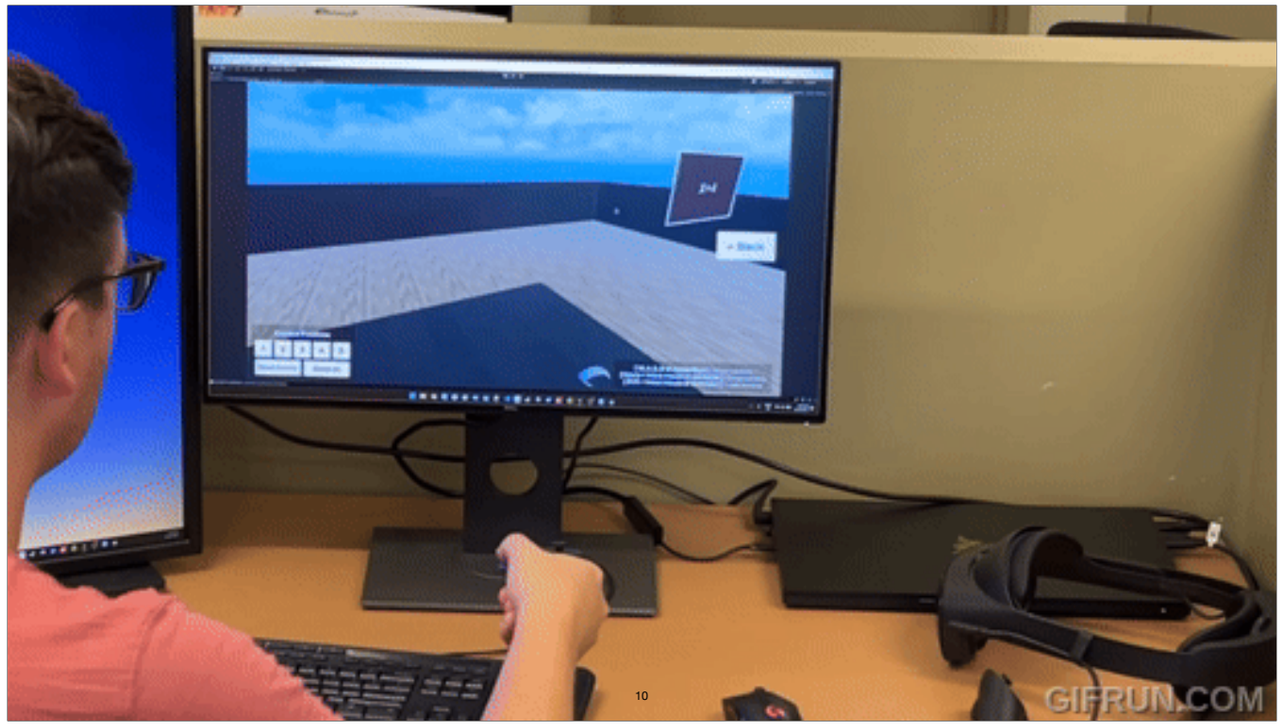
We took the findings from the formative study and summarized them into a definition of Context: which combines three elements. Input (the input device they’re using, like controllers or a mouse or a keyboard), viewing (how they’re viewing the content, be it via a monitor or a headset, and memory. Memory is a special case motivated by the finding about peeks being temporary switches followed by a return. Memory encaspulates the principle that the system needs to respond to not only their current arrangement of input and viewing devices, but what came before.



These principles motivated us to represent context as a state machine - people start in one the green states, like desktop or VR, then peek in either input or viewing devices, and either return back or fully switch. Say, for example, I’m in VR, with controllers. I can quickly peek to desktop by opening



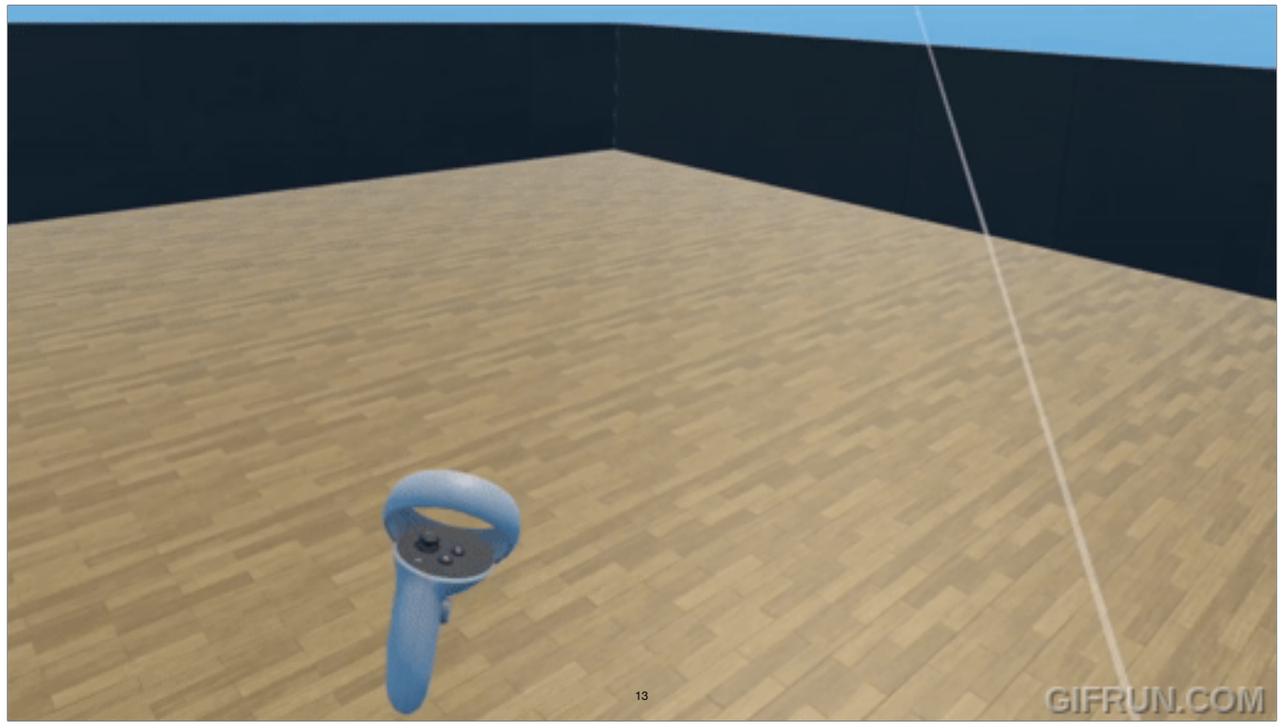
Say, for example, I'm in VR, with controllers. I can quickly peek to desktop by opening the SteamVR menu that lets me see my Windows desktop. This is an example of a *peeking technique*, that lets me quickly get to information on my desktop without having to fully switch. Based on this, we decided to implement several techniques that extend this idea. Like…



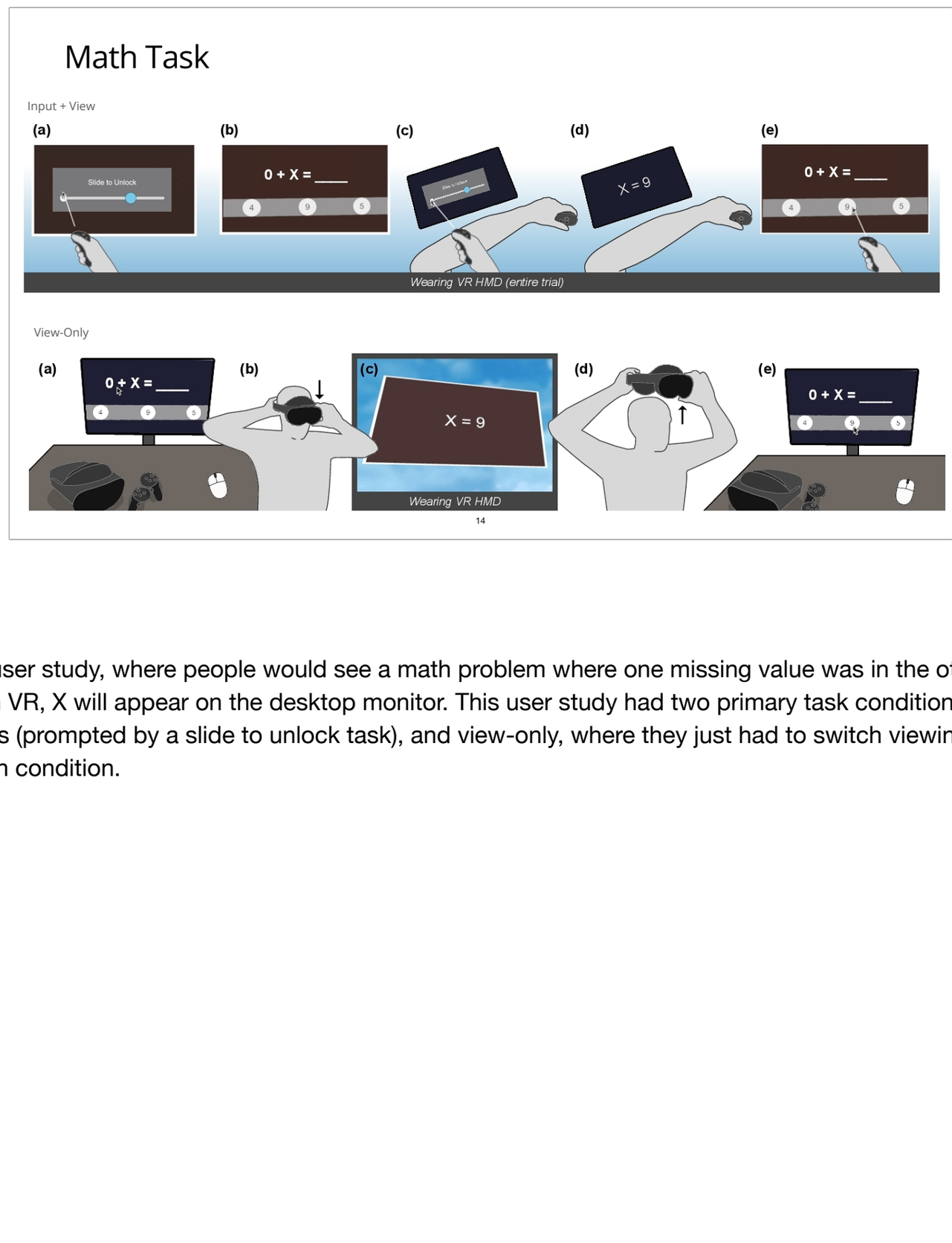
If I want to work in the VR scene but not don the headset, I can view and manipulate the headset’s camera angle from desktop and point my controllers “into” the monitor for interaction.

If I want to use a mouse cursor specifically, if I’m wearing the headset, I can grab the mouse, and a cursor will appear in front of my face which I can move with the mouse.

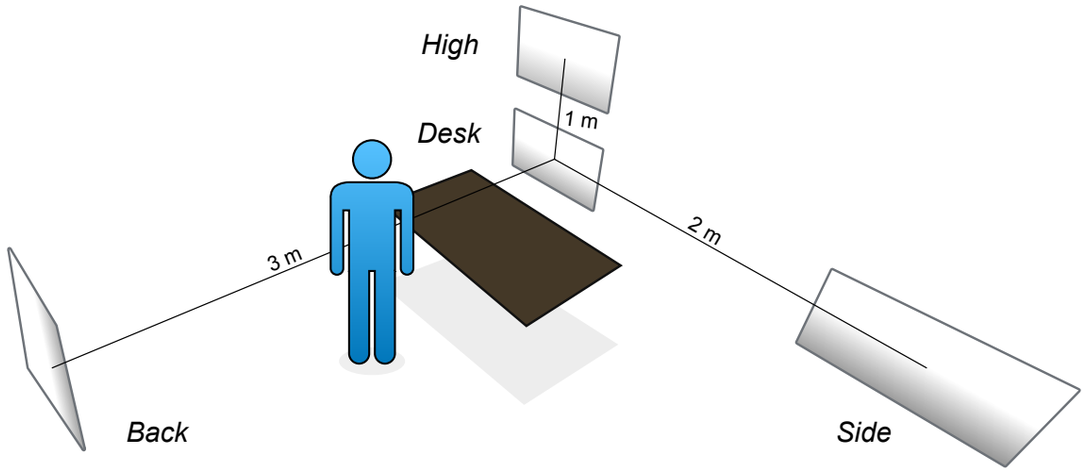
Likewise, if I’m wearing the headset and don’t want to grab the mouse, I can just place my controller on my desk sideways, and I can use it to control that cursor. This also works outside of VR.



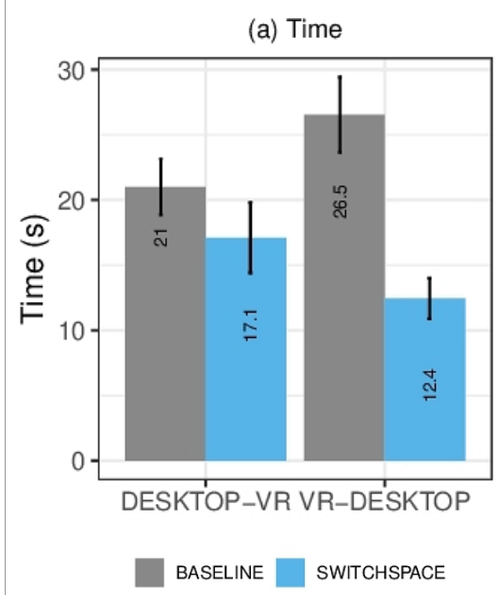
There are also several ways to peek at the desktop from VR. In addition to summoning the desktop view by tilting my hand like this, we also implemented the SteamVR panel view where it’s floating in the world in front of you, as well as co-located with the physical monitor.



These techniques were put to use in a user study, where people would see a math problem where one missing value was in the other modality. So if I have a math problem like 5 + X = blank, and I start in VR, X will appear on the desktop monitor. This user study had two primary task conditions: input plus view, where users had to switch in both input and viewing devices (prompted by a slide to unlock task), and view-only, where they just had to switch viewing devices. Peeking techniques were either enabled or disabled depending on condition.

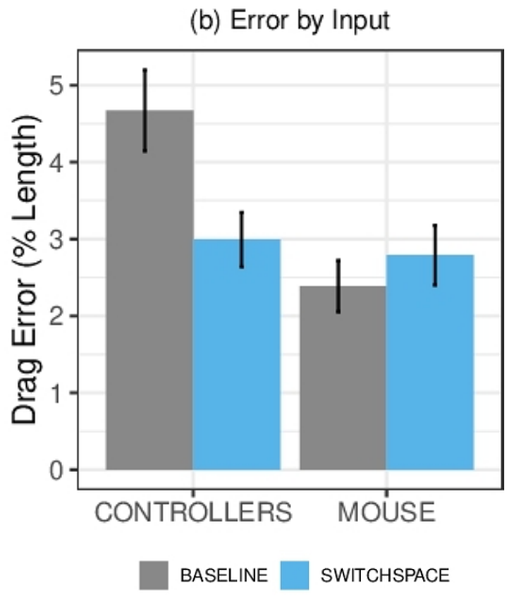


We placed panels in VR in four positions. At the position of the virtual monitor, 1 metre above it, 2 metres to its right, or 3 metres behind it, prompting users to turn around and move in the space.



Using peeking techniques made the cross-reality math task **faster**.

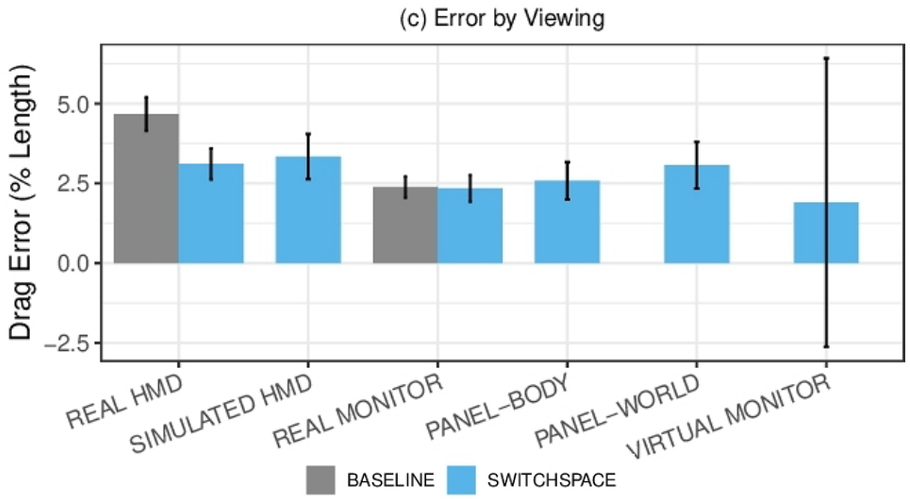
As for results. We can see that using peeking techniques, as denoted by the SwitchSpace bars, made completing the cross-reality task faster. This is especially pronounced in the bars on the right for the condition where people switched from VR to desktop.



Peeking techniques made the cross-reality math task **more precise**.

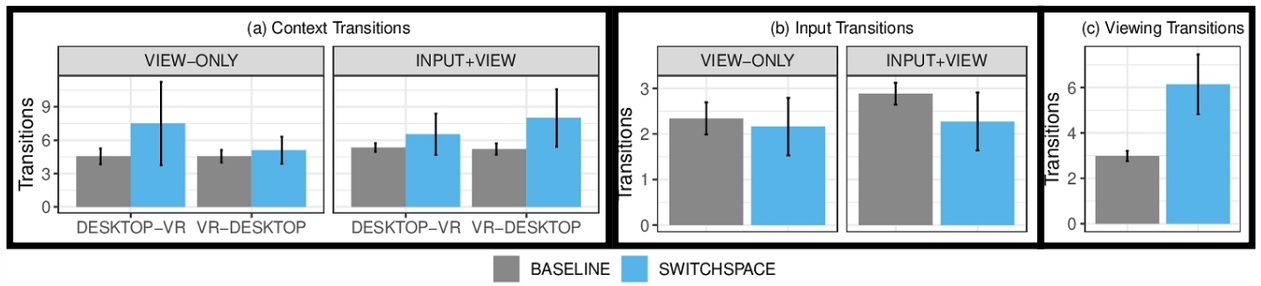
17

Likewise, using peeking techniques made the cross-reality task more precise. The ability to use peeking techniques allowed the accuracy of the controllers and the mouse to be somewhat equal in terms of their accuracy in the dragging task.



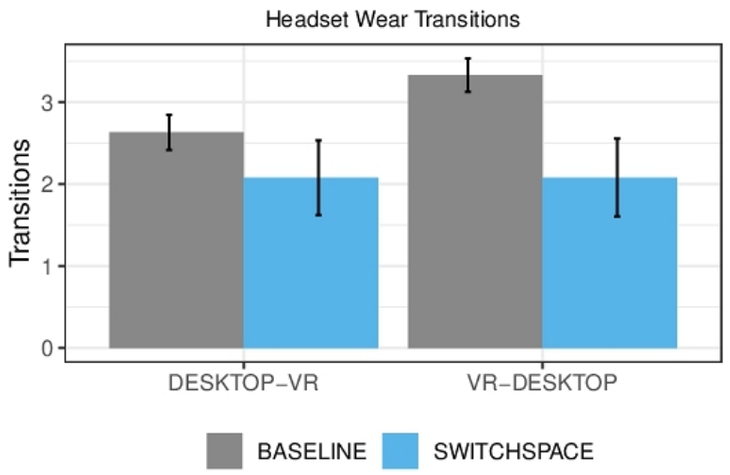
Using peeking techniques made
the cross-reality math task **more precise**.

Going on with this theme of precision, we can also break things down by the viewing technique that was used. The grey bars, showing cases where users couldn’t use peeking techniques and had to fully switch, had a much more pronounced difference between the headset and the real monitor. This is much less when peeking techniques were enabled. Also, that huge error bar on the virtual monitor technique is mostly a commentary on its popularity in this task specifically. It wasn’t summoned very often meaning the standard deviation of the error was super high.



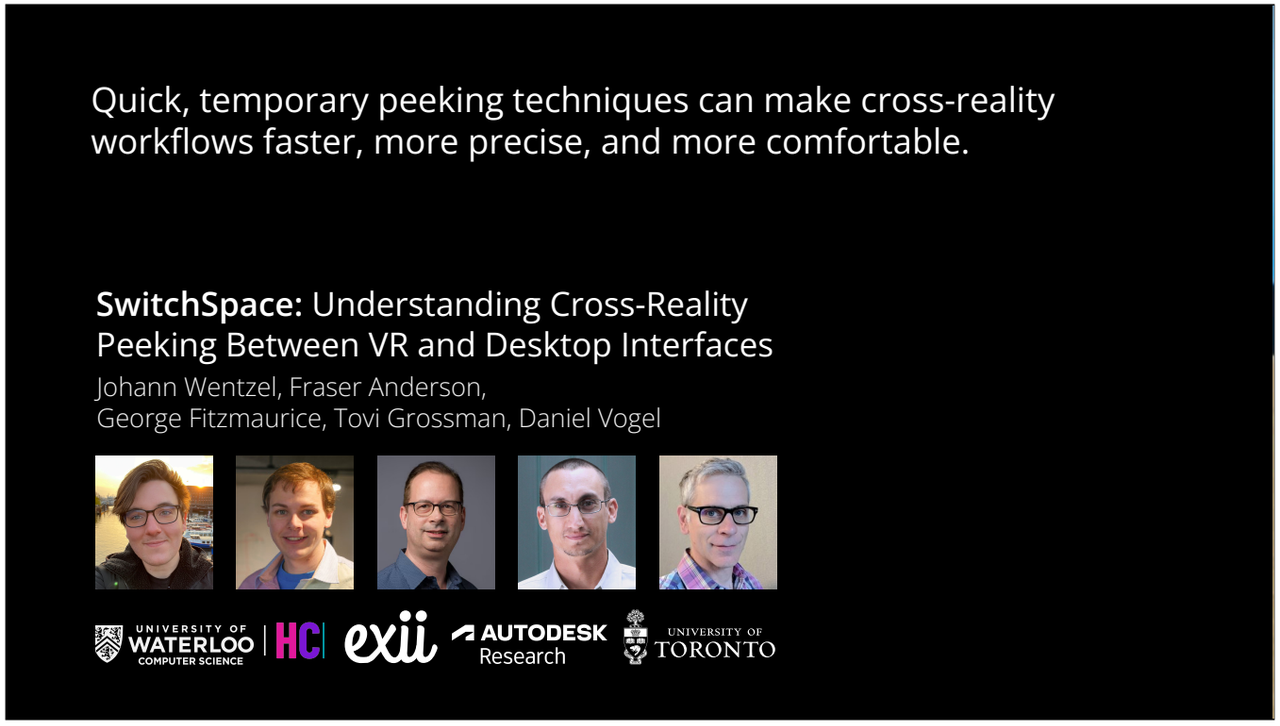
The ability to use peeking techniques caused people to **switch states more**.

So now what's interesting is looking at the results for transitions. The ability to use peeking techniques allowed people to switch between states, and look between the states, more often. For example, the graph on the left looks at context transitions. In both view-only and input+view conditions, people switch states more often with peeking techniques available. Likewise, people switched input devices more often as well. Finally, people switched in viewing state more often with peeking techniques.



The ability to use peeking techniques caused people to **wear or remove the headset less**.

And generally, in both desktop-to-VR and vr-to-desktop conditions, the ability to use peeking techniques caused people to wear or remove the headset less.



That brings us to the end of the presentation, and now I'd love to take any questions you may have. Otherwise, feel free to reach out to me or any of my coauthors!