Towards a Desktop–AR Prototyping Framework: Prototyping Cross-Reality Between Desktops and Augmented Reality

Robbe Cools*†
KU Leuven
Department of Computer Science

Matt Gottsacker*‡
University of Central Florida

Adalberto Simeone§
KU Leuven
Department of Computer Science

Gerd Bruder¶
University of Central Florida

Greg Welch‖
University of Central Florida

Steven Feiner**
Columbia University

Abstract

Augmented reality (AR) head-worn displays (HWDs) allow users to view and interact with virtual objects anchored in the 3D space around them. These devices extend users' digital interaction space compared to traditional desktop computing environments by both allowing users to interact with a larger virtual display and by affording new interactions (e.g., intuitive 3D manipulations) with virtual content. Yet, 2D desktop displays still have advantages over AR HWDs for common computing tasks and will continue to be used well into the future. Because of their not entirely overlapping set of affordances, AR HWDs and 2D desktops may be useful in a hybrid configuration; that is, users may benefit from being able to work on computing tasks in either environment (or simultaneously in both environments) while transitioning virtual content between them. In support of such computing environments, we propose a prototyping framework for bidirectional Cross-Reality interactions between a desktop and an AR HWD. We further implemented a proof-of-concept seamless Desktop–AR display space, and describe two concrete use cases for our framework. In future work we aim to further develop our proof-of-concept into the proposed framework.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality

1 Introduction

Augmented reality (AR) head-worn displays (HWDs) can add virtual content to users' physical surroundings. This property gives AR HWDs great potential in numerous computing contexts, many of which are yet unexplored. In this paper, we focus on using AR HWDs in stationary single-user personal computing environments. In this domain, researchers have explored using them for extending users' 2D screens [5, 11, 21] or replacing them altogether [8, 10] to give the user a larger digital workspace. These devices also show promise for computing tasks that require users to manipulate and study objects in 3D, such as in 3D modeling and design tasks [1, 7, 9, 26] or when annotating 3D objects for educational purposes [35].

Some researchers have even posited that AR HWDs will eventually replace desktop computing environments entirely. However, for many tasks, desktop computer displays currently have advantages over AR HWDs. The displays themselves are often better: they provide higher resolutions and better color representations. Further, AR HWDs are less ergonomic than desktop displays because they add weight to the user's head, which causes fatigue over time, and restrict the user's field of view. Desktop screens are also less expensive and more widely available than AR glasses. While some of these differences may become marginal as more investment is made to solve the grand challenges facing AR displays [3, 20], it is our position that the semi-fixed nature of desktop displays will continue to be useful for computing tasks long into the future. Rather than imagining a world in which one kind of display replaces another, it is perhaps more interesting to imagine ways in which different kinds of displays can interoperate so that people may use different displays (or even both displays simultaneously) to benefit from their complementary capabilities.

Such interactions are characterized as Cross-Reality (CR) because they involve the "transition between or concurrent usage of multiple systems on the reality–virtuality continuum [30]." The reality–virtuality (RV) continuum [25] describes systems that immerse users in virtual content to varying degrees along an axis from fully physical to fully immersive virtual reality (VR). To contextualize our CR area of interest in the RV continuum, desktop displays are fixed in the physical world, and AR displays are at a more virtual point along the continuum because they add virtual content to the physical space. We define the Desktop–AR CR space as encompassing systems that include both a desktop display and an AR HWD and that support interactions and transitions among the distinct display spaces provided by each device. The Desktop–AR space is not to be confused with Desktop VR (or Fish Tank VR [34]), where a virtual environment is displayed on a stereoscopic desktop display and coupled to the user's head position. Nor is it to be confused with Desktop AR (sans en dash), where a user can interact with AR content displayed using a tabletop 3D monitor (e.g., Schmandt's stereoscopic workstation [29]). In this paper, we focus on a single user using both a desktop display and an AR HWD simultaneously. Because the aim of this paper is to produce a tool that is immediately useful, we concentrate on optical see-through AR HWDs because presently they allow the user to view the desktop display in its actual resolution (compared to video see-through HWDs that can only view the physical environment at the same resolution as virtual objects).

Currently, prototyping Desktop–AR applications is difficult because it requires custom implementation of the system on AR and desktop applications separately. In this paper, we propose a unified prototyping framework that combines desktop displays and AR HWDs in their current forms to support Desktop–AR developers in prototyping these kinds of systems for different use cases. We discuss characteristics of the interactions afforded by this combination and present novel CR interaction and transition techniques that will be supported by our framework. Our proof-of-concept implementation allows users to select and manipulate content on and across a 2D desktop display and the 3D space extended from the desktop displayed through an AR HWD, using either the mouse or their hands as input devices for the displays.

Our goal is to discuss and refine the Desktop–AR prototyping

*Both authors contributed equally to this research.
†e-mail: robbe.cools@kuleuven.be
‡e-mail: gottsacker@knights.ucf.edu
§e-mail: adalberto.simeone@kuleuven.be
¶e-mail: bruder@ucf.edu
‖e-mail: welch@ucf.edu
**e-mail: feiner@cs.columbia.edu

framework with members of the CR community to gather feedback, and further guide development. In future work, the framework will be made available to the community as an open-source Unity plugin.

2 Related Work

In this section, we present related research on CR interactions and transitions. We then review related work on user interfaces that incorporate ideas applicable to the Desktop–AR design space.

CR systems can include either 1) multiple users at different points on the RV continuum or 2) a single user using multiple systems at different points on the RV continuum, either alternating or concurrently. Previous research on CR with users at different points on the continuum includes interaction between HWD and non-HWD users as well as interaction between HWD users with different levels of virtuality, such as VR and AR. The HWD-to-non-HWD interaction involves HWD-user situational awareness [12, 28], as well as allowing non-HWD users to interact with virtual content [18, 19, 32]. Research on HWD users with different levels of virtuality is centered around making content of one reality accessible to the other [6, 13].

Prototyping CR has been investigated in VRception [16], where a system was presented to create prototypes of CR applications across the entire RV continuum [25] spanning physical reality, AR, augmented virtuality, and VR. While VRception allows for simulating the Desktop–AR scenario, our prototyping framework supports prototyping on actual hardware and makes it possible for developers to incorporate peripheral devices such as a mouse.

In our prototyping framework, a single user uses both AR and desktop interfaces, and is thus interacting with content at different points on the continuum. Connecting interfaces with different interaction affordances introduces interaction asymmetry. There is much research on the general problem of reconciling these cross-device differences and providing meaningful and useful interactions across both interfaces. Indeed, a recent review by Brudy et al. sorted through 510 papers on the subject [4]. In this work, we focus specifically on bidirectional interactions between desktop and head-worn AR devices, which is characterized in the Cross-Device Taxonomy [4] as single-user, synchronous, spatially or logically distributed, semi-fixed, and personal.

In 1991, Feiner and Shamash [11] noted the asymmetric benefits of using 2D and 3D displays together to visualize and interact with virtual content and explored "hybrid user interfaces" that allowed users to move 2D windows between each display completely or partially (i.e., part of a window could be visible on a desktop display while the remainder was simultaneously visible on the AR HWD). This hybrid design effectively treated the displays as complementary.

Benko et al. [2] built "cross-dimensional gestural interaction techniques" to transition virtual content between a 2D display and a 3D AR HWD. Their system used an AR HWD, a tracked glove, and a projected 2D display, and it allowed users to pull and push virtual objects between the 2D display and the HWD's 3D space. More recently, Roo and Hachet [27] presented the OneReality system, an instrumented Mixed Reality (MR) environment that allowed users to transition virtual content among projected tabletop displays, handheld displays, and AR and VR HWDs. In XD-AR, Speicher et al. [31] presented a framework for cross-device interaction and transitions between handheld, projected, and head-worn AR displays. BISHARE [37] presented a design space for single-user interaction between head-worn and phone-based AR, They explored using the phone for spatial interactions and hand-tracked interactions with the phone display, similar to how our framework describes CR interaction between desktop and AR. Zhu et al. emphasized the different interaction strengths for each device (e.g., mobile phones are useful for efficient and precise input as well as high-resolution and full-color 2D content, while AR HWDs are good for displaying 3D imagery in the user's spatial environment) as well as differentiating between phone-centric and HWD-centric interactions [37].

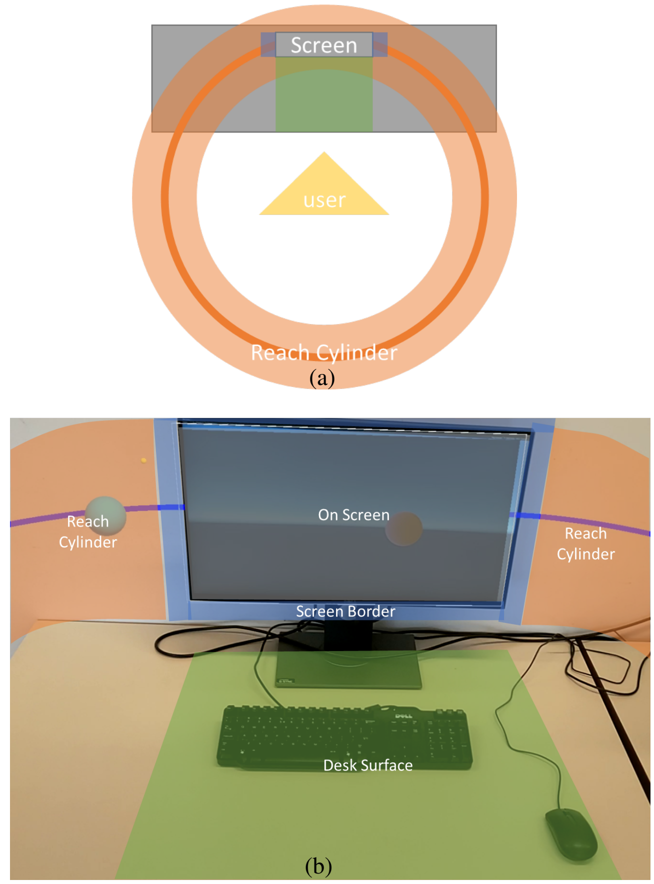

Figure 1: Subdivision of the display space: (a) top-down view of user and their computing space, (b) computing space from the perspective of user marked by yellow triangle in (a). *On Screen* is the 2D space displayed on the screen. *Screen Border* is an area around the screen where AR content can be anchored to the screen. *Reach Cylinder* is an area all around the user that is within reach from their seated position. *Desk Surface* is the tabletop space in front of the user.

Related to the desktop input side of the Desktop–AR CR space, Zhou et al. [36] created a "depth-adaptive cursor" that integrated a conventional 2D desktop mouse into 3D space viewed through a VR HWD. Additionally, Kim and Vogel [22] extended 2D mouse interaction into projected AR through a cursor that moves along the 3D geometry of virtual objects in the projected AR space.

Applying these CR concepts, Wang et al. [33] implemented a data visualization system for physicists that allows seamless usage of both screen and AR space. Their application can show 3D visualizations both on the screen and in AR. Similar to our work, they also allowed the mouse to be used in 3D space, though with a button to switch between spaces rather than moving the mouse over the screen edge.

3 Proposed Desktop–AR Prototyping framework

In this section, we propose a general-purpose CR framework that would allow developers to easily implement Desktop–AR CR interaction techniques. The framework would provide a seamless display space between desktop and AR, with support for mouse-based and hand-tracking input, as well as basic CR interactions and transitions. Developers can use this framework to prototype applications that make concurrent or alternating use of Desktop and AR.

To help define CR interactions in our framework, we divide the Desktop–AR display space, similar to Zhu et al. [37], into *On Screen (OS)* and *Spatial*. The spatial display space is further subdivided into *Screen Border (SB)*, *Reach Cylinder (RC)*, and *Desk Surface* (see Fig. 1). The *Reach Cylinder (RC)* is an approximation of where the user can reach in 3D space. To aid in this approximation, we can use "joint-centered kinespheres" [23] that model users' nearby reachable space. We discuss interactions and transitions across different subdivisions of the space, but with limited emphasis on the *Screen Border* space because we envision it being mainly used to add extra user interface elements to *On Screen* applications.

We first present our choice of input devices for the Desktop–AR framework: mouse and hand tracking (the mouse being the traditional input on the desktop screen and hand tracking being the traditional spatial input). Then we identify the ways in which these can be used to interact "across realities," by which we mean using the input device outside of its traditional display area or performing an interaction in one display area with effects in the other. Finally, we use these CR interactions to perform CR transitions, and describe methods by which content can be moved between display spaces. Our goal is to implement these proposed interactions and transitions into the prototyping framework, allowing developers to use them when prototyping Desktop–AR applications.

3.1 Input Modalities

Our framework considers two input modalities: mouse and hand tracking. First, the mouse serves as a traditional 2D input device, moving a cursor with two degrees of freedom (DOF). We chose to include this input device because it is the most common for desktop computing. Second, hand tracking allows for directly interacting with content in 3D space, and is enabled by the AR HWD. Current MR devices offer both hand tracking and motion controllers as options for spatial input, yet we chose not to include motion controllers in our framework. The main reason for focusing on hand tracking instead of motion controllers is the ease of switching between modalities, as hand tracking avoids the need to put down the motion controller when switching to the mouse. To maintain a seamless input modality in a different way, the desktop could be equipped with a touchscreen, so the user could use their hand to interact with content displayed on the AR HWD and on the desktop. In this paper, we focus on the more common personal computing case of a user using a keyboard and mouse primarily for their tasks.

Both input modalities, mouse and hand tracking, have strengths and weaknesses. The mouse is more precise, but typically limited to 2D movements, though extensions to 3D environments have been investigated [36]. Hand tracking is less precise, but allows for direct and intuitive 3D interaction. Because of these differences in precision and dimensionality, mouse and hand tracking are complementary input modalities.

3.2 Desktop–AR CR Interactions

We consider the following cases of CR interaction in our framework:

- using an input device outside of its traditional display space
- using an input device in one display space with effects in the other
- using an input device with effects in both display spaces
- bimanual interaction in the same or different display spaces

Tables summarizing Desktop–AR CR Interaction can be found in Appendix A and Appendix B.

Using an input device outside its traditional display space. we consider the following interaction scenarios: using the mouse in 3D space, or using hand interactions on the screen. As a basic mouse interaction, extra user interface (UI) elements added to the *screen border* can be easily accessed. For more distant interactions, we envision the mouse to remain a 2D input device, thus limiting its reach to the surface of a cylinder in 3D space rather than adding a third axis of movement. This approach allows the mouse to move off the screen onto the *Screen Border* and along the *Reach Cylinder* surface. However, we expect areas farther along the *Reach Cylinder* to be more difficult to access with the mouse in a traditional desktop setup. For example, the area opposite the desk and behind the user would be difficult to access as the mouse needs to remain on the desk. Users may position content nearby but outside of the HWD field of view (FOV) (or outside of their human FOV once AR HWD FOV is wide enough), so it is necessary to help users maintain awareness of out-of-view objects, as in [14, 15, 17]. As a possible solution to this limited reach and view, extra functionality could allow the *Reach Cylinder* to be rotated, moving along all the content attached to it.

Hand interaction could be enabled on the screen via raycasting, as the hand cannot pass the physical screen for direct interaction with the objects. Alternatively, the *On Screen* objects could be augmented with handles that stick out of the screen to allow hand manipulation. Hand tracking *On Screen* would provide less precision than mouse input, but it may be useful if users are working primarily with 3D object manipulations.

Users may also perform an **interaction in one display space with effects in the other**. For example, users could use the mouse for fine manipulation of an object displayed *On Screen* that is reflected in other display spaces such as on the *Desk Surface* to allow multiple simultaneous perspectives. As another example, users may use their tracked hands to perform coarse manipulations (e.g, 90° rotations) of an object on the *Desk Surface* that are reflected *On Screen*.

In the Desktop–AR CR space, it is also possible to **use an input device with effects in both display spaces**. This CR interaction builds on the previous one and adds that the user may manipulate an object that is duplicated in two different display spaces. This case uses the same interaction techniques as the previous one but offers additional views on the virtual object being manipulated.

The Desktop–AR CR space also affords **bimanual interaction in the same or different display spaces**. That is, users may use a tracked hand and mouse input simultaneously to interact with virtual objects. When the user's mouse and tracked hand are in the same display space (e.g., both *On Screen*), the mouse can be used for fine-grained manipulations and the hand can be used for coarser direct manipulations. When the user's mouse and tracked hand are in different display spaces, the virtual object could be mirrored in each display space to give the user different perspectives on and interaction affordances for the object. Alternatively, in the distributed input scenario, the user could use the separate input display spaces to cause a CR transition.

3.3 Desktop–AR CR Transitions

We use *CR transition* to refer to transitioning content between 2D and 3D space in either direction. We envision the main method for CR transitions to be based on spatial positioning; that is, content positioned in the space behind the physical display is rendered in 2D and transitioned to 3D as it moves out of this area at the sides or front of the physical display. The Desktop–AR display space is much larger than the limited area of the physical display, so moving objects may require covering a greater virtual distance than users may prefer. Thus, this CR task benefits from novel transition techniques.

We propose the following two novel techniques to more efficiently transition objects: bimanual and batched. First, the bimanual technique uses the mouse and hand tracking at the same time. The user first makes a gesture with the hand not using the mouse to indicate a position in 3D space. Then, objects on the screen that are selected with the mouse transition to this position. In contrast, the user may make a gesture in 3D space to mark objects that are then transitioned to the mouse's on-screen position when clicked. Second, the batched technique allows the user to select multiple objects, either *On Screen* or in 3D space, and then switch modality only once, after which the selected objects can be transitioned sequentially, either by clicking in screen space or making a gesture in 3D space.

Another design aspect of the transition is that objects that transition can either be moved or duplicated. In traditional systems, changes made to an object would not be reflected in a copy. In our framework, however, one may wish to manipulate an object on the

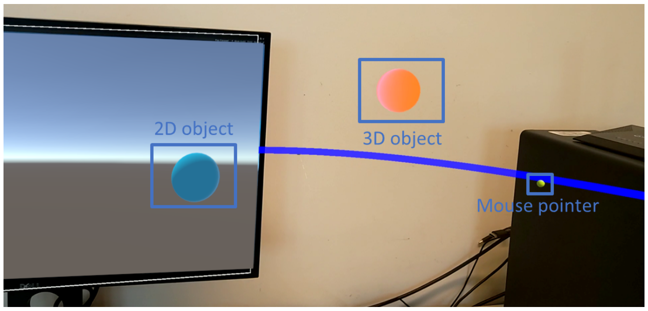

Figure 2: MR capture of proof-of-concept system.

screen, but visualize the changes in AR. In this case, a transitioned object needs to remain synchronized with its *On Screen* counterpart. When multiple such distributed visualizations exist, it could become difficult to know which ones belong together; thus, some indication of their connection is required. For example, a line could visually connect corresponding *On Screen* and spatial content.

4 Proof-of-Concept Desktop–AR Workspace

As a first step towards validating our prototyping framework, we created a proof-of-concept Desktop–AR workspace in Unity Engine 2021.2.16f. The system consists of two application instances running simultaneously, one on a desktop computer with a physical flat-panel display and the other on a Microsoft HoloLens 2. Both are instances of the same project and are networked to synchronize the virtual environment (VE) state, but differ in the way they visualize the VE. A virtual orthogonal camera is positioned in front of the physical display, and records the VE with the same viewport size. The resulting image is rendered onto the desktop display's application. The HoloLens 2 application renders the same VE in world space, blocking the part already covered by the physical display.

The system supports two interaction types: mouse and hand interaction. The mouse is implemented as a cursor in 3D space, moving on a plane that is flat on the display but curved towards the user along a cylindrical surface in 3D space (Fig. 2). This allows the 2-DOF mouse to work in the 3D *Reach Cylinder*. Curving the plane avoids steep viewing angles onto the workspace as the cursor moves farther to the side, similar to how some large computer monitors are curved. Objects near the cursor can be moved by holding the left mouse button. As a second interaction technique, we implemented MRTK[1] hand tracking, which allows manipulation of content away from the mouse curve freely into 3D space. The MRTK "far interaction" technique can be used to manipulate objects on the screen or at a distance.

As shown in the demo video[2], the proof-of-concept system supports transitioning content between 2D and 3D space based on its position. Content positioned behind the display in 3D space is rendered on the 2D display. This 2D content transitions to 3D when moved outside of the screen space, and vice versa. While this transition method is intuitive, it might not be the most efficient as it requires large mouse movements to cover the distance between spaces. Additionally, it does not support more complex interactions such as copying objects into 3D space rather than moving them. It also would be useful to further explore the rendered position of virtual content. For example, the AR HWD could render all content in front of or behind the desktop display, with only content positioned in the plane of the desktop display being rendered on the desktop display. This would allow traditional 2D content to be rendered with the full resolution and stability of the desktop screen, while 3D content and closer or farther 3D windows would be rendered on the AR HWD. We plan to investigate the *asynchronous* Desktop–AR CR space as well (e.g., user switching between a VST AR HWD and a desktop display: when the user puts on the HWD, the content transitions from 2D to 3D, and vice versa when they take off the HWD). As future work, we will refine the proof of concept with the functionality described in Sect. 3.

5 Use Cases

We explore the computing context of a user working in three dimensions, such as when creating a virtual world or doing computer-assisted design (CAD). These tasks involve creating, shaping, and manipulating 3D virtual objects. Desktop displays are useful for detailed viewing of 3D objects and for fine-grained edits and manipulations. AR HWDs are also useful in this case because they allow users to view 3D objects in actual 3D space. This feature allows users to more naturally view and manipulate the objects from multiple perspectives by directly moving the object in front of them or moving their head around it. This can improve users' spatial ability and help them more reliably perceive and mentally represent the objects in three dimensions [24]. For these reasons, an important CR interaction for the virtual world builder or CAD worker is easily transitioning objects to and from the desktop and AR HWD. Using the AR HWDs hand tracking, they should be able to grab objects from the surface of the desktop display and bring them onto the *Desk Surface* space in front of the monitor where they can manipulate them with their hands, similar to Benko et al. [2]. At the same time, the objects could be duplicated on the original desktop display so that the user may make finer adjustments with the mouse.

The virtual object the user is designing may consist of numerous nested objects. Thus, the user might want to work on a specific piece of the object while maintaining awareness of how that part relates to the whole. The user could work on the part *On Screen* and in the *Desk Surface* space while the whole virtual object could be displayed through the AR HWD in the *Reach Cylinder* to the side of the desktop display. The whole object could have an augmentation highlighting the part being worked on and the current perspective from which the user is viewing the part. The user could do detailed work on the part *On Screen* and in the *Desk Surface* and perform more macro-level manipulations (e.g., scaling the object to fit in with surrounding objects) on the whole model in nearby AR space.

6 Conclusion

In this paper, we propose a Desktop–AR CR prototyping framework for combined mouse and hand inputs across a Desktop–AR display space. To describe this framework, we discussed the input modalities involved and outlined CR interactions and CR transitions that the framework would support. We then presented a proof-of-concept implementation of such a Desktop–AR display space, and explored two potential use cases for it. At this workshop, we hope to discuss the proposed system with members of the CR research community and gather feedback on the kinds of interactions and transitions that should be integrated into the framework concept. As future work, we will develop the proposed prototyping framework and make it available for use by the community.

Acknowledgments

This research is supported by *Internal Funds KU Leuven* (C14/20/078); the National Science Foundation under Collaborative Award Numbers 1800961, 1800947, and 1800922 (Dr. Ephraim P. Glinert, IIS) to the University of Central Florida, University of Florida, and Stanford University respectively, and Award Number 2037101 to Columbia University; the Office of Naval Research under Award Number N00014-21-1-2578 (Dr. Peter Squire, Code 34); and the AdventHealth Endowed Chair in Healthcare Simulation (Prof. Welch).

[1]`https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05`

[2]`https://youtu.be/amBV7uX0r9c`

References

[1] R. Arora, R. Habib Kazi, T. Grossman, G. Fitzmaurice, and K. Singh. SymbiosisSketch: Combining 2D & 3D sketching for designing detailed 3D objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2018.

[2] H. Benko, E. Ishak, and S. Feiner. Cross-dimensional gestural interaction techniques for hybrid immersive environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pp. 209–216, Mar. 2005. ISSN: 2375-5334. doi: 10.1109/VR.2005.1492776

[3] M. Billinghurst. Grand Challenges for Augmented Reality. *Frontiers in Virtual Reality*, 2, 2021.

[4] F. Brudy, C. Holz, R. Rädle, C. J. Wu, S. Houben, C. N. Klokmose, and N. Marquardt. Cross-device taxonomy: Survey, opportunities and challenges of interactions spanning across multiple devices. In *Conference on Human Factors in Computing Systems - Proceedings*, p. 28. Association for Computing Machinery, May 2019. doi: 10.1145/3290605.3300792

[5] A. Butz, T. Hollerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping users and computers in a collaborative 3D augmented reality. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pp. 35–44. IEEE, 1999.

[6] R. Cools, J. Han, and A. L. Simeone. SelectVisAR: Selective Visualisation of Virtual Environments in Augmented Reality. In *DIS 2021 - Proceedings of the 2021 ACM Designing Interactive Systems Conference: Nowhere and Everywhere*, pp. 275–282. Association for Computing Machinery, Inc, June 2021. doi: 10.1145/3461778.3462096

[7] B. R. De Araùjo, G. Casiez, and J. A. Jorge. Mockup builder: Direct 3D modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012*, pp. 173–180. 2012.

[8] S. Di Verdi, D. Nurmi, and T. Hollerer. ARWin—a desktop augmented reality window manager. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 298–299. IEEE, 2003.

[9] J. J. Dudley, H. Schuff, and P. O. Kristensson. Bare-handed 3D drawing in augmented reality. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 241–252, 2018.

[10] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2D windows for 3D augmented reality. In *Proceedings of the 6th annual ACM symposium on User interface software and technology - UIST '93*, pp. 145–155. ACM Press, Atlanta, Georgia, United States, 1993. doi: 10.1145/168642.168657

[11] S. Feiner and A. Shamash. Hybrid user interfaces: breeding virtually bigger interfaces for physically smaller computers. In *Proceedings of the 4th annual ACM symposium on User interface software and technology - UIST '91*, pp. 9–17. ACM Press, Hilton Head, South Carolina, United States, 1991. doi: 10.1145/120782.120783

[12] M. Gottsacker, N. Norouzi, K. Kim, G. Bruder, and G. Welch. Diegetic Representations for Seamless Cross-Reality Interruptions. In *ISMAR*, 2021.

[13] J. G. Grandi, H. G. Debarba, and A. Maciel. Characterizing asymmetric collaborative interactions in virtual and augmented realities. In *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, pp. 127–135. Institute of Electrical and Electronics Engineers Inc., Mar. 2019. doi: 10.1109/VR.2019.8798080

[14] U. Gruenefeld, A. E. Ali, S. Boll, and W. Heuten. Beyond Halo and Wedge: visualizing out-of-view objects on head-mounted virtual and augmented reality devices. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, pp. 1–11. Association for Computing Machinery, New York, NY, USA, Sept. 2018. doi: 10.1145/3229434.3229438

[15] U. Gruenefeld, A. E. Ali, W. Heuten, and S. Boll. Visualizing out-of-view objects in head-mounted augmented reality. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, pp. 1–7. Association for Computing Machinery, New York, NY, USA, Sept. 2017. doi: 10.1145/3098279.3122124

[16] U. Gruenefeld, J. Auda, F. Mathis, S. Schneegass, M. Khamis, J. Gugenheimer, and S. Mayer. VRception: Rapid Prototyping of Cross-Reality Systems in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–15. ACM, New York, NY, USA, Apr. 2022. doi: 10.1145/3491102.3501821

[17] U. Gruenefeld, D. Ennenga, A. E. Ali, W. Heuten, and S. Boll. EyeSee360: designing a visualization technique for out-of-view objects in head-mounted augmented reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, pp. 109–118. Association for Computing Machinery, New York, NY, USA, Oct. 2017. doi: 10.1145/3131277.3132175

[18] J. Gugenheimer, E. Stemasov, J. Frommel, and E. Rukzio. ShareVR: Enabling co-located experiences for virtual reality between HMD and Non-HMD users. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2017-May, pp. 4021–4033. Association for Computing Machinery, May 2017. doi: 10.1145/3025453.3025683

[19] J. Gugenheimer, E. Stemasov, H. Sareen, and E. Rukzio. FaceDisplay: Towards asymmetric multi-user interaction for nomadic virtual reality. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, pp. 1–13. Association for Computing Machinery, New York, New York, USA, Apr. 2018. doi: 10.1145/3173574.3173628

[20] Y. Itoh, T. Langlotz, J. Sutton, and A. Plopski. Towards indistinguishable augmented reality: A survey on optical see-through head-mounted displays. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021.

[21] B. R. Jones, H. Benko, E. Ofek, and A. D. Wilson. IllumiRoom: Peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 869–878, 2013.

[22] D. Kim and D. Vogel. Everywhere Cursor: Extending Desktop Mouse Interaction into Spatial Augmented Reality: Extended Abstract. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pp. 1–7. ACM, New Orleans LA USA, Apr. 2022. doi: 10.1145/3491101.3519796

[23] P. Lubos, G. Bruder, O. Ariza, and F. Steinicke. Touching the sphere: Leveraging joint-centered kinespheres for spatial user interaction. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, pp. 13–22, 2016.

[24] J. Martín-Gutiérrez, J. L. Saorín, M. Contero, M. Alcañiz, D. C. Pérez-López, and M. Ortega. Design and validation of an augmented book for spatial abilities development in engineering students. *Computers & Graphics*, 34(1):77–91, 2010.

[25] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented Reality: A class of displays on the reality-virtuality continuum. Technical report, 1994. Publication Title: 282 / SPIE Volume: 2351.

[26] P. Reipschläger and R. Dachselt. DesignAR: Immersive 3D-modeling combining augmented reality with interactive displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pp. 29–41, 2019.

[27] J. S. Roo and M. Hachet. One Reality: Augmenting How the Physical World is Experienced by combining Multiple Mixed Reality Modalities. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pp. 787–795. Association for Computing Machinery, New York, NY, USA, Oct. 2017. doi: 10.1145/3126594.3126638

[28] R. Rzayev, S. Mayer, C. Krauter, and N. Henze. Notification in VR: The effect of notification placement, task, and environment. In *CHI PLAY 2019 - Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pp. 199–211. Association for Computing Machinery, Inc, New York, NY, USA, Oct. 2019. doi: 10.1145/3311350.3347190

[29] C. Schmandt. Input and display registration in a stereoscopic workstation. *Displays*, 5(2):89–92, 1984.

[30] A. Simeone, M. Khamis, A. Esteves, F. Daiber, M. Kljun, K. Pucihar, P. Isokoski, and J. Gugenheimer. *International Workshop on Cross-Reality (XR) Interaction*. Nov. 2020. Pages: 114. doi: 10.1145/3380867.3424551

[31] M. Speicher, B. D. Hall, A. Yu, B. Zhang, H. Zhang, J. Nebeling, and M. Nebeling. XD-AR: Challenges and opportunities in cross-device augmented reality application development. *Proceedings of the ACM on Human-Computer Interaction*, 2(EICS):1–24, 2018.

[32] C.-H. Wang, S. Yong, H.-Y. Chen, Y.-S. Ye, and L. Chan. HMD Light: Sharing In-VR Experience via Head-Mounted Projector for

Asymmetric Interaction. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pp. 472–486. ACM, New York, NY, USA, Oct. 2020. doi: 10.1145/3379337.3415847

[33] X. Wang, L. Besançon, D. Rousseau, M. Sereno, M. Ammi, and T. Isenberg. Towards an Understanding of Augmented Reality Extensions for Existing 3D Data Analysis Tools. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, Apr. 2020. doi: 10.1145/3313831.3376657

[34] C. Ware, K. Arthur, and K. S. Booth. Fish tank virtual reality. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pp. 37–42, 1993.

[35] K. Yu, U. Eck, F. Pankratz, M. Lazarovici, D. Wilhelm, and N. Navab. Duplicated reality for co-located augmented reality collaboration. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2190–2200, 2022.

[36] Q. Zhou, G. Fitzmaurice, and F. Anderson. In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–17. ACM, New Orleans LA USA, Apr. 2022. doi: 10.1145/3491102.3501884

[37] F. Zhu and T. Grossman. BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14. ACM, Honolulu HI USA, Apr. 2020. doi: 10.1145/3313831.3376233

A TABLE DESKTOP–AR CR INTERACTIONS PART 1

Table 1: Desktop–AR CR Interactions part 1. This table lists examples illustrating why a developer or prototyper may want to implement certain CR interactions made possible by supporting mouse and tracked-hand input modalities both separately (cases 1–6), and simultaneously for bimanual interactions (cases 7–15, shown in the subsequent table). The *Input Modality* columns describe the space where each input modality is interacting; OS stands for *On Screen*, RC for *Reach Cylinder*, DS for *Desk Surface*. Interaction scenarios we consider unlikely are marked as such.

Case	Input Modality		Display space of primary object		
	Mouse	Hand	On-Screen [OS]	Reach Cylinder [RC]	Desk Surface [DS]
1	OS	-	Traditional 2D interaction; fine manipulations of 3D objects.	User is primarily working in 2D screen but wants to interact with a 2D window that was extended into RC space (e.g., [10, 11]).	User is primarily working in 2D screen but wants to interact with a 3D object on the DS using the mouse, as in [22].
2	RC	-	User is interacting with secondary object in RC that affects the primary object (e.g., a window with view controls is placed in RC)	User performs fine-grained manipulations on 3D object.	(Similar to 2[OS]) User is interacting with secondary object in RC that affects primary object on DS.
3	DS	-	User is interacting with secondary object in OS space that affects the primary object in DS space.	User is interacting with secondary object in RC space that affects the primary object in DS space.	User is performing fine-grained manipulations of object in DS space.
4	-	OS	User is performing direct 3D manipulation of 3D object displayed in high resolution.	User is directly manipulating something on screen that affects an object in RC space.	User is directly manipulating something displayed on screen that affects an object in DS space.
5	-	RC	User is rotating a large object in RC space while a part of that object is shown and updated live in OS space.	User is directly manipulating an object in RC space.	User is rotating a large object in RC space while a part of that object is shown and updated live in DS space.
6	-	DS	User is directly manipulating an object in DS space that is mirrored in high resolution in OS space.	User is directly manipulating an object in DS space that is part of a larger object displayed in RC space. Manipulations that affect the DS object affect the whole RC object.	User is directly manipulating an object in DS space.

B TABLE DESKTOP–AR CR INTERACTIONS PART 2

Table 2: Desktop–AR CR Interactions part 2. This table lists examples illustrating why a developer or prototyper may want to implement certain CR interactions made possible by supporting mouse and tracked-hand input modalities both separately (cases 1–6, shown in previous table), and simultaneously for bimanual interactions (cases 7–15). The *Input Modality* columns describe the space where each input modality is interacting; OS stands for *On Screen*, RC for *Reach Cylinder*, DS for *Desk Surface*. Interaction scenarios we consider unlikely are marked as such.

Case	Input Modality		Display space of primary object		
	Mouse	Hand	On-Screen [OS]	Reach Cylinder [RC]	Desk Surface [DS]
7	OS	OS	User uses the mouse to finely rotate an object while using their hand to scale the object.	(unlikely)	(unlikely)
8	OS	RC	Quickly move an object from OS space to RC space.	Quickly move an object from RC space to OS space.	(unlikely)
9	OS	DS	User performs fine adjustments in OS space while performing coarse manipulations (e.g., 90° rotations) in DS space.	(unlikely)	User directly manipulates object in DS space while adjusting fine parameters in OS space.
10	RC	OS	User makes a gesture to quickly move object from OS space into RC space.	User clicks object in RC space to quickly move it into OS space.	(unlikely)
11	RC	DS	(unlikely)	User performs fine translations of an object with the mouse in RC space while scaling or rotating the object using a virtual control in DS space.	Users makes a gesture in DS space to quickly move object into RC space.
12	RC	RC	(unlikely)	User performs fine manipulations of object with the mouse and coarse manipulations with their hand.	(unlikely)
13	DS	OS	User performs fine adjustments in DS space while performing coarse adjustments in OS space.	(unlikely)	Users performs fine manipulations in DS space while performing coarse adjustments in OS space.
14	DS	RC	(unlikely)	User selects an object in RC space and makes a gesture to quickly move it to DS space.	User clicks on objects in DS space to quickly move it to the position the user is pointing in RC space.
15	DS	DS	Perform fine and coarse manipulations on object and see them displayed in high resolution in OS space.	(unlikely)	Perform fine and coarse/direct manipulations on object in DS space.