

RealityLens: A User Interface for Blending Customized Physical World View with Virtual Reality

Chiu-Hsuan Wang
National Yang Ming Chiao Tung
University
Hsinchu, Taiwan
wanglolly.cs05@nycu.edu.tw

Bing-Yu Chen
National Taiwan University
Taipei, Taiwan
robin@ntu.edu.tw

Liwei Chan
National Yang Ming Chiao Tung
University
Hsinchu, Taiwan
liweichan@nycu.edu.tw



Figure 1: *RealityLens* allows users to customize a blended physical world view with reality lenses deployed to serve their different needs. For instance, *RealityLens* can be used to (a) enable communication with non-VR users when watching 360° video, (b) utilize the familiar physical interface such as keyboards and smartphones during VR remote meetings, and (c) avoid both static and moving obstacles when immersed in VR games.

ABSTRACT

Research has enabled virtual reality (VR) users to interact with the physical world by blending the physical world view into the virtual environment. However, current solutions are designed for specific use cases and hence are not capable of covering users' varying needs for accessing information about the physical world. This work presents *RealityLens*, a user interface that allows users to peek into the physical world in VR with the reality lenses they deployed for their needs. For this purpose, we first conducted a preliminary study with experienced VR users to identify users' needs for interacting with the physical world, which led to a set of features for customizing the scale, placement, and activation method of a reality lens. We evaluated the design in a user study (n=12) and collected the feedback of participants engaged in two VR applications while encountering a range of interventions from the physical world. The results show that users' VR presence tends

to be better preserved when interacting with the physical world with the support of the *RealityLens* interface.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**.

KEYWORDS

Virtual Reality, Blending Reality, Augmented Virtuality

ACM Reference Format:

Chiu-Hsuan Wang, Bing-Yu Chen, and Liwei Chan. 2022. RealityLens: A User Interface for Blending Customized Physical World View with Virtual Reality. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*, October 29–November 2, 2022, Bend, OR, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3526113.3545686>

1 INTRODUCTION

The virtual reality (VR) technique provides users with an immersive experience but at the cost of losing awareness of the physical world (PW). Current VR systems (e.g., Valve Chaperone and Oculus Guardian) provide static play area boundaries to prevent user collision. However, simple tasks such as picking up a cup or communicating with others remain difficult. While today's techniques such as HTC Vive's front-facing camera and Oculus's Passthrough mode allow users to access the PW view, it requires them to pause their VR experience and hence breaks their VR presence.

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.

UIST '22, October 29–November 2, 2022, Bend, OR, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9320-1/22/10...\$15.00

<https://doi.org/10.1145/3526113.3545686>

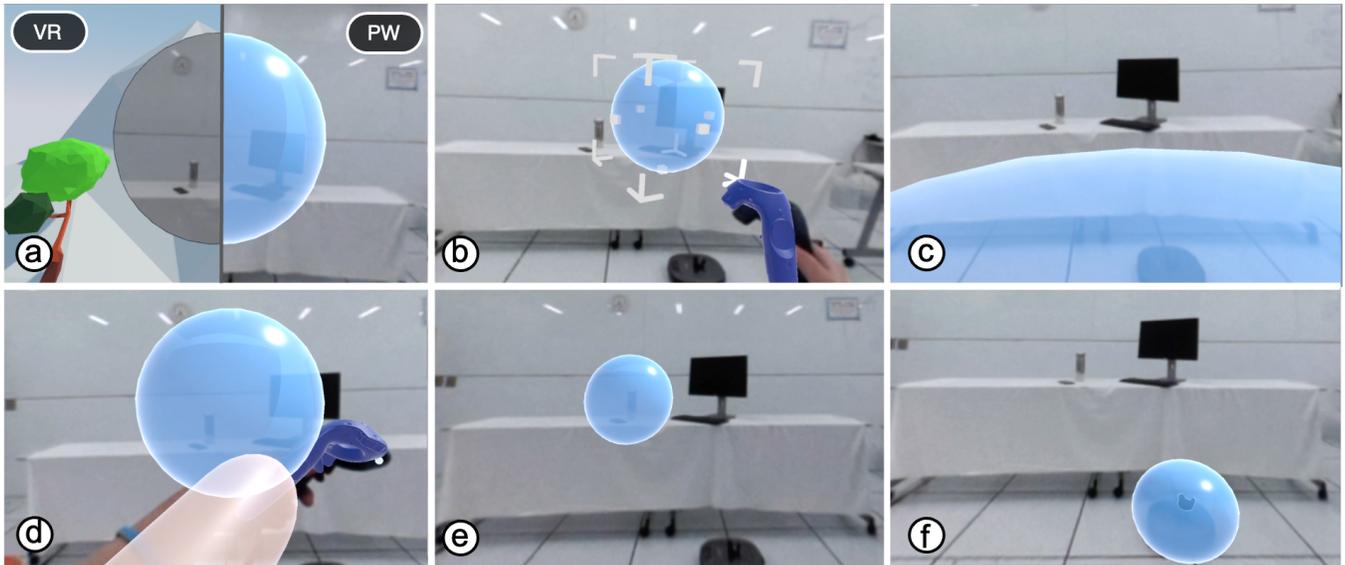


Figure 2: *RealityLens* allows users to create a reality lens. (a) In configuration mode, where users see the PW through the passthrough function, the reality lens is shaped as a virtual sphere. In VR mode, it turns into a reality lens. By (b) adjusting its size and placing it in different positions, such as (c) on the user’s field-of-view, (d) around the user’s hand, (e) in the environment for capturing the region, or (f) on a moving object like a robot vacuum, *RealityLens* serves to users’ diverse needs for interacting with the PW in VR.

Previous studies suggest blending situated PW views into the virtual environment (VE), which provides users with PW information without interrupting users’ VR experience [12, 20]. This technique has been applied to meet VR users’ needs that involve PW interaction, for instance, to enhance users’ PW awareness [23], prevent user collision [12], provide access to nearby objects [20], and enable communication with bystanders [7, 20, 31].

While previous works have shown a high diversity of users’ needs for interacting with the PW in VR, current solutions are limited to user interfaces designed for specific use cases. There is still little understanding of the users’ overall needs for PW interactions; therefore, a solution that supports users’ customized needs for blending the PW content into the VE is still lacking.

We propose *RealityLens*, a user interface that allows users to create a reality lens through which they are enabled to peep into PW in VR. Users can customize the lens’s size, placement, and activation method based on their needs for obtaining PW information during their VR experience, for instance, to communicate with a bystander (Figure 1a), utilize a familiar physical interface in VR (Figure 1b), and avoid PW collisions with both static and moving obstacles (Figure 1c).

Figure 2 illustrates the interaction flow to create a reality lens. We consider *RealityLens* a system-wide interface similar to the Oculus Guardian system. Users can invoke the interface to set up their reality lens before and during the VR application. To do so, users have to enter the configuration mode by activating the passthrough function (turning on the video see-through camera), which takes them to see the PW. With seeing the PW, they can add a reality lens to the PW and customize the lens based on their needs for acquiring PW information when returning back to VR. The lens

is shaped as a virtual sphere in the configuration mode. In the VR mode, it appears as a reality lens (Figure 2a).

Users can adjust the size of the lens (Figure 2b) and place it in different positions for various interactions with the PW, such as attaching to the users’ field of view (FoV) (Figure 2c), on the user’s body (Figure 2d), in the environment (Figure 2e), or to a specific object (Figure 2f). Furthermore, the interface allows users to decide on one of the three activation methods to control the visibility of a reality lens in VR, which includes always available, triggered by users, and triggered by custom events (see more details in the *RealityLens* section). These features help users customize the blending of the PW content in VR based on their needs.

Based on the prototype interface implemented, we conducted a user study to evaluate the effectiveness of the proposed interface and explore how *RealityLens* affects user experience. The result shows that *RealityLens* provides users with a continuous VR experience and hence tends to improve their VR presence when interacting with the PW. Moreover, we discuss *RealityLens*’ effectiveness in different applications and users’ strategies for deploying the reality lens. At the end of the paper, we provide the design insights on a user interface for customizing the PW content into VR.

1.1 Contribution

This work presents a user interface that supports users’ customized needs for acquiring PW information when immersed in VR. We conducted a study to collect users’ needs and derive design guidelines, based on which the *RealityLens* interface was implemented. Our study provides design insights into users’ strategies in deploying reality lenses for their custom needs and how these reality lenses affect users’ VR experience.

2 RELATED WORK

This section reviews how recent research bridges VR and PW, and then focuses on the studies that blend PW into VE.

2.1 Bridging Virtual Reality and the Physical World

VR allows users to immerse themselves in the virtual world but isolates them from the PW. However, since VR users are physically located in the PW, low awareness of the PW in VR causes various problems such as the VR users' safety, the difficulty of accessing PW objects and the communication gap between the VR users and bystanders. As such, a number of studies have attempted to bridge VR and PW from different aspects.

Hagan et al. [24] conducted an online survey to investigate interactions between VR users and bystanders in the wild. Their findings suggest that in addition to interrupting the VR experience, bystanders also play an important role in moderating the VR user's experience. Previous works have also demonstrated novel systems that aim to improve the bystanders' awareness of the VR user's presence [4], facilitate shared use of the physical space [35], and enrich collaboration between VR users and bystanders [10, 11, 14, 29, 32, 33].

Other studies attempt to merge digital information external to VR into VR, such as displaying smartphone notifications [30]. Along the same lines, prior work has explored how notification forms [9, 36], positions [13], and moments [5] on display in VR affect the user experience.

2.2 Blending the Physical World into Virtual Environment

Augmented Virtuality (AV) from the reality-virtuality continuum [22] refers to an approach that augments the virtual view with elements from physical reality. This approach integrates the PW view or elements into VR, allowing users to receive the PW information when immersed in VR.

Recent studies have explored AV methods that combine the PW and the VE to benefit the user experience. For instance, laying the virtual objects on the physical objects provides VR users with passive haptic feedback that enhances their VR presence [17, 27, 28]. By generating the virtual world based on the PW, one can be immersed in VR while traveling in the PW [6, 34]. Remixed Reality [18], on the other hand, explores the space and time manipulation of the PW based on its 3D reconstruction in VR. With the reality-virtuality continuum [22], OneReality [25] presents a design space concerning the blending of different levels of users' presence.

Other studies have tried to blend the PW into VE to enable VR users' PW interaction. McGill et al. [20] identified usability challenges in VR and addressed these challenges with a prototype that allows to blend the PW objects into VR. Since then, several works also explored designs to address different usages, such as avoiding PW object collision [15], allowing users to access nearby objects like drinks or the keyboard [1, 3], notifying users of bystanders' existence [16, 21, 23], and further enabling communication with them [31]. Recently, George et al. [8] conducted a survey which identifies factors that affect the transition in the reality-virtuality continuum and presented a design space for cross-reality interaction.

More recent research has presented novel techniques to support users' PW interactions in VR. ModularHMD [7] presents a novel head-mounted display (HMD) that utilizes the peripheral regions to enable ad-hoc interaction with PW objects and bystanders. RealityCheck [12] demonstrates a system that blends the 3D reconstruction of the entire PW with VE. One of the techniques they have explored, "flashlight into reality," allows users to select the blended region in the PW using the controller as a flashlight. These works blend the PW view in different forms and provide users with various interaction techniques for acquiring PW information.

While most studies tried to evaluate the effectiveness of different blending techniques or understand users' perception about them, *RealityLens* aims to provide a user interface that assists users in placing and deciding the passthrough lens behavior. To the best of our knowledge, *RealityLens* is the first design that allows users to customize the blended region's behavior to cover their needs for interacting with PW in VR.

3 REQUIREMENTS ANALYSIS

We conducted a preliminary study to identify VR users' needs for interacting with the PW. This study aims to (1) investigate the users' experience of obtaining PW information when immersed in VR, and (2) understand design considerations for blending the customized PW content with the VE.

This study comprised two phases: an interview and a body-storming phase. In the interview phase, a semi-structured interview was conducted to understand participants' experience interacting with the PW in VR. In the body-storming phase, participants were asked to experience and provide feedback on a prototype application featuring basic functions of the interface.

3.1 Prototype Application

We developed a VR prototype application to help participants experience the user interface for blending the PW content into the VE. In this prototype, participants entered the configuration mode, where they would see the PW in view, and were allowed to create fixed-size reality lenses (50 cm diameter) in the environment with the controller. All the lenses they created were world-reference, and participants could create or delete one at the controller's position by pressing the controller's trackpad button. After setting up reality lenses they needed, participants exited the configuration mode and started experiencing a 360° video viewing application. Through the reality lenses they set up earlier, participants can acquire PW information while watching the 360° video in VR. This prototype application was dedicated to obtaining users' feedback on the interface design.

3.2 Procedure

The study occurred in a seminar room at the university. Participants were seated in front of a 2 x 1 m table with a laptop. Besides the table, the room was decorated with furniture (including chairs, tables, shelves, and a whiteboard) to resemble a workspace. After briefing the study, we started participants with the interview phase of the study, where we asked them about their experience interacting with the PW while immersed in VR including their motivation and approach. This phase lasted approximately 15 minutes.

In the body-storming phase, participants completed tasks that required them to interact with the PW using the prototype application. The tasks were derived from previous research including typing on their computer [20], checking the clock on the wall outside of the play area [12], and communicating with a bystander [7, 31]. They were also asked to picture how they would utilize the prototype in the scenarios they mentioned in the interview phase. Participants were encouraged to think aloud during the activity. In the end, another interview was conducted to gain their feedback and opinions about the interface design after completing the tasks. The body-storming phase lasted 20 minutes, and the second interview took 10 minutes.

3.3 Participants and Apparatus

We recruited six participants (5 male, 1 female) from the local university who had VR experiences and were unrelated to this research. Five had at least half a year of VR research experience ($M = 1$ year), and three were VR gamers with a VR environment installed in their homes. In the body-storming phase, an HTC Vive Pro HMD running on a laptop (i7, GTX1080, 16G RAM) was utilized for the VR experience.

3.4 Summary of the User Experiences

Here we report reasons why users need to interact with the PW in VR. The most common reason mentioned by all participants was accessing nearby physical objects such as smartphones, keyboards, and VR controllers. Participants also reported occasions that required them to interact with bystanders such as answering their family members or asking for help from their colleague (P3, P5, P6). For room-scale VR applications, participants mentioned that they had experienced accidentally bumps (P1-P3, P6), which made them pay more attention to the PW when moving in VR. To prevent collisions, they had even asked someone to stay beside them. These reasons show the diversity of users' needs, which aligns with the results in previous research [8].

Except for motivations to interact with the PW, we further investigated existing approaches they had used for obtaining PW information in VR. All participants reported that they would like to avoid putting on and taking off the HMD. However, in scenarios that require a high-resolution view and fine interaction, such as replying to messages on a smartphone or using a computer, participants had to take off the HMD (P1, P3-P6). While the VR passthrough function offers a workaround to see the PW through the HMD, participants said that the resolution of current passthrough function is insufficient (P3, P4).

For scenarios that do not require fine interaction, such as finding the VR controllers or seeing the surroundings when moving around, participants would utilize the limited gap beneath the HMD to find their target (P4-P6). However, they also admitted that this solution is limited and uncomfortable in the long term. Some participants tried reaching the nearby objects using their memory about the surroundings and/or bodily perception without visual assistance (P1-P3, P6), but this could result in more collisions, accidentally knocking over nearby objects. In sum, participants considered currently no suitable solutions available to overcome these issues.

4 DESIGN CONSIDERATIONS

This section presents design considerations of the reality lens based on the results of the body-storming activities.

4.1 Size of a Reality Lens

Since users' quest for PW content in VR is diverse, the interface should provide the flexibility to adjust the size of a reality lens for different usages. For instance, participants mentioned that the default lens size is sufficient for checking the clock but too small to view the full keyboard (P1, P2, P5, P6). Even to the same target, the proper size for a lens differs in different contexts. For example, when blending bystanders into the VE, participants mentioned the proper size of a lens based on the type of communication, such as including the upper body when body language is important (P3), and including only the bystander's face for a short conversation (P1, P6).

4.2 Preserving VR Presence

All participants stated that a reality lens should only be visible in VR when needed; otherwise, it can occlude the VE and reduce users' VR presence. During the study, participants considered the PW view on the keyboard effective but also worried it was annoying on the clock. P3 recommended that "It (the reality lens) should be turned on only when I want to check it; otherwise, it occludes VR content for the rest of the time."

Regarding activation method, participants suggested triggering by themselves was the easiest method, but some use cases should be triggered by system detection. For instance, if a lens is used to prevent physical collisions, the system can detect and trigger the lens when the user gets too close to the obstacle associated with the lens (P1, P3, P4). Participants also mentioned cases they prefer to rely on system detection. P5 said, "The reality lens on the phone can turn on itself when a notification comes in, so I don't need to check it once in a while."

4.3 Prepared and Accidental Events

The interface should afford use cases that can be prepared before entering a VR experience, as well as accidental events during a VR experience. For PW content that is frequently used or predicted by users viable during a VR experience, such as a smartphone or physical keyboard, users can set up a reality lens on them before entering VR. However, other use cases are impossible to capture beforehand such as answering incoming bystanders. Therefore, the interface should offer users the flexibility to handle both use cases.

Among these cases, participants mentioned that moving objects appearing in the play area should be carefully addressed (P4-P6). Since most users will confirm the play area's safety before entering VR, moving objects are more likely to appear after users are immersed in VR, such as pets wandering about the play area. Since moving objects increase the risk of physical collision, it requires users' greater attention.

5 REALITYLENS

Based on the design considerations, we propose *RealityLens*, a user interface that supports deploying and customizing a reality lens through which users are able to see the PW in VR. This allows the

users to obtain relevant PW information for their needs while in VR. This section presents the design space of *RealityLens*, followed by an implementation of the user interface.

5.1 Design Space

We present two design factors: **placement** and **availability**. The **placement** factor describes where a reality lens is placed and the **availability** factor specifies when and how a reality lens should be activated during a VR experience. These two factors define the behavior of a reality lens.

5.1.1 Placement. The placement factor determines the position control of a lens, and it has two modes: the *user-centric* and *environment-centric* mode, according to whether the lens is added in relation to the user and the environment, respectively. The lens is set to the *user-centric* mode when it is placed on the user's body. Otherwise, the lens is placed in the PW and set to the *environment-centric* mode.

User-centric (view-anchored, body-anchored): A *user-centric* reality lens is controlled with the user's body movement. Given where it attaches to on the body, a *user-centric* reality lens further behaves as a *view-anchored* or *body-anchored* lens. A *view-anchored* lens is when the lens is attached to the user's FoV, which makes the lens always available in view [7]. A *body-anchored* lens is when the lens is placed on the user's body besides the FoV, such as on the user's hand [20]. Compared to a *view-anchored* lens, a *body-anchored* lens provides users with a greater degree of freedom of control since it is decoupled from users' FoV, allowing users independent position control of the lens with the body movement (e.g., placing the lens on the wrist to use as a watch).

Since users can leverage body movement to quickly adjust the lens position for accessing PW content, A *user-centric* reality lens can be considered as a universal reality lens suitable for addressing accidental usage during the VR experience.

Environment-centric (world-anchored, object-anchored): The *environment-centric* reality lens is placed in the PW environment. Compared to a *user-centric* lens, a *environment-centric* lens is a target-specific lens that supports viewing a specific region or object. It is suitable for prepared usage to acquire PW information.

A *environment-centric* lens is further characterized as a *world-anchored* or *object-anchored* lens based on the target on which the association is made. A *world-anchored* lens is attached to a PW region to capture objects within this area, such as a table near the user, or a waiting spot for the bystanders [35]. By contrast, an *object-anchored* lens is when the lens is attached to an object or a person, such as a nearby keyboard [20] or a bystander using the same environment [20, 31, 35]. Regarding position control, a *world-anchored* lens is fixated at a location the lens is associated with, while an *object-anchored* lens follows an object it attaches to. *Object-anchored* lenses, therefore, are suitable to capture objects that may change their location during a VR experience.

5.1.2 Availability. The availability of a reality lens specifies how the lens is triggered to appear during a VR experience. We design a reality lens to have one of the three modes on availability: *always-on*, *user-triggered*, and *event-triggered*. A *always-on* lens is constantly present in the VE such that the PW content in it is always visible

to the user [12, 20]. However, since persistent PW content in VR can easily affect one's VR experience, a reality lens in this mode should be carefully handled and is more suitable for capturing a small region or PW content in need of frequent access.

In comparison, a *user-triggered* lens is not visible during the VR experience unless users turn it on [7]. A reality lens in this mode gives the control of the lens's presence to the users and is suitable for capturing PW content less often in usage.

Lastly, a *event-triggered* lens remains invisible in VR until a specific event in association meets. Depending on available events supported by the system, a *event-triggered* lens can have a wide utility; for instance, it can appear when the user gets close to the lens [12, 35] and when the object the lens attaches to is moving. Furthermore, when connecting to Internet of Things (IoT) devices, a *event-triggered* lens can be triggered by device events such as being triggered when a doorbell rings. *Event-triggered* lenses, thus, meet user needs related to automation.

5.2 User Interface

Based on the design space, we implemented the *RealityLens* interface to assist users in customizing reality lenses for blending their desired PW content into VR. We utilized HTC Vive Pro HMD for its built-in front-facing stereo camera view. The project was implemented in Unity 2019.3.24f with HTC Vive SRWorks SDK¹ run on a VR-ready laptop (i7, GTX1080, 16G RAM).

Before putting on the HMD, users first attach markers, which are implemented with HTC Vive trackers, to PW objects they want to capture during the later VR experience. These markers are used to track the objects' position in the PW, particularly for objects subject to location changes. We envision future markers in the form of stickers for easily tagging the environment. Then, they can wear the HMD to set up the reality lens.

Users set up a reality lens by entering the configuration mode, which can be activated by double-clicking the controller's menu button. In the configuration mode, users first see the PW in view, implemented through the HMD's passthrough function. They create a reality lens by pressing the controller's trackpad button. As shown in Figure 2a, the reality lens is rendered as a virtual sphere in the configuration mode. Users can adjust the lens' size through the frame elements (Figure 2b) and manipulate its position by dragging it with the controller. Returning to the VR mode, the reality lens is rendered as a lens in VR.

We provide four placement modes: *view-anchored*, *body-anchored*, *world-anchored*, and *object-anchored*. Users determine the placement mode of a lens by the placement point. By placing it on the user's head, the reality lens attaches to the user's view and becomes *view-anchored* (Figure 2c). Placing the lens on the body beyond the user's view, a skin color region on the part of the body appears, indicating the operation being *body-anchored* and the region to host the lens (Figure 2d). Alternatively, if the user places the lens on an object augmented with a marker, the lens is set to the *object-anchored* mode (Figure 2f). Otherwise, the reality lens is placed in the environment and set to the *world-anchored* mode (Figure 2e).

¹SRWorks SDK: <https://developer-express.vive.com/resources/vive-sense/srworks-sdk/>

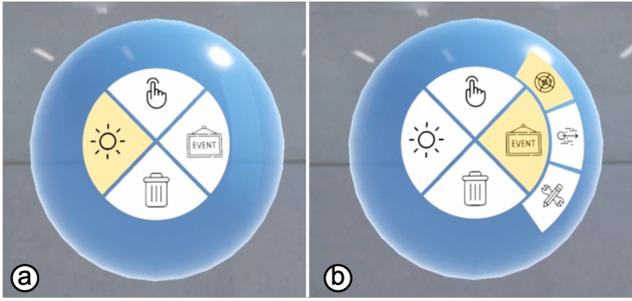


Figure 3: The context menu to configure availability of a reality lens, including (a) four items: *always-on* (left), *user-triggered* (top), *event-triggered* (right), and *deletion* (bottom). (b) The second layer menu under the *event-triggered* mode includes three methods: *Proximity* (top), *Movement* (middle), and *Custom* (bottom).

To configure the availability factor of a reality lens, users touch the lens and press the controller’s trigger button to bring up a context menu (Figure 3), which includes four modes to select from (Figure 3a): *always-on* (left), *user-triggered* (top), *event-triggered* (right), and *deletion* (bottom).

The default mode is *always-on*; that is, the reality lens is constantly visible in VR. With the *user-triggered* mode, the reality lens is not visible until the user long presses the menu button. On releasing the button, the *user-triggered* reality lens returns to be invisible.

Under the *event-triggered* mode, we further offer three triggering methods, which can be selected from a second layer menu once the mode is chosen (Figure 3b). The first method is *Proximity* (at the top of the second-layer menu), which turns the lens visible when the user encounters a safety range set for the lens (within 0.7 m, as suggested by SteamVR). The second method, *Movement*, turns the lens visible when it is under translation or rotation changes; the lens disappears after it becomes stationary (more than 10 seconds without movement). The third method, *Custom*, allows the reality lens to be triggered by a device’s customized event when the lens is placed on it; for example, a lens can be triggered by a phone call when placing on a smartphone. Note that this activation method was simulated using wizard-of-oz (WoZ) in our study. In the future, it can be implemented using cross-device communication.

6 USER EVALUATION

We conducted a user study to understand how *RealityLens* assists VR users in interacting with the PW and to explore how users utilize *RealityLens* to acquire PW information. We were also interested in how *RealityLens* affects user experience and presence in VR.

6.1 Study Design

The study structure followed a within-subjects design with one independent variable, the system. There were two system conditions, the *RealityLens* condition, in which participants experienced and completed user tasks with the help of the proposed interface, and the *Baseline* condition, which offered only the basic passthrough function. In the *RealityLens* condition, participants were instructed to set up reality lenses before a task and could adjust them during

the task. In the *Baseline* condition, participants were provided with the passthrough function, allowing them to acquire the PW in view. The VR application would be paused when the passthrough function was activated and resumed when the participant turned off the function to return to the VR application.

6.2 Apparatus

The same implementation mentioned in the User Interface section was used. Additionally, since the *RealityLens* interface requires the users’ body posture for the implementation of the *body-anchored* feature, we had the participants use hand controllers or wear HTC Vive trackers on their wrists to track their hands. The arm posture was estimated using the hand positions and the HMD position with the Inverse-Kinematic (IK) technique. Since the wrist trackers may affect user experience, we also had participants wear them in the *Baseline* condition.

6.3 Procedure

Our study took place in a university lab containing a 3.5 x 3 m VR playground surrounded by furniture (including doors, chairs, stools, and tables) to resemble a living space with VR installation. In addition, a VR PC, a bottle, and a smartphone were placed on one of the desks (1 x 0.5 m) to simulate the workspace desktop. Participants were first introduced to the study purpose as well as the *RealityLens* interface. A hands-on training session was then provided for them to learn and practice the interface. The training session ended when the participants could engage with the interface without the experimenter’s help. The introduction and training together took around 40 minutes.

After the training session, the participants were instructed to experience and complete the tasks using one of the system conditions. After completing all tasks, participants had to answer the questionnaire to evaluate their VR presence under the assigned system. After a 10-minute break, the same process was repeated on the other system condition. The system condition order was counterbalanced. The *Baseline* condition took 15 minutes and the *RealityLens* session took 25 minutes accommodating setting up and adjusting the reality lens.

In the end, participants answered the system usability questionnaire, and a post-study semi-structured interview was conducted to understand how participants utilized *RealityLens* as well as their overall feedback on the system. Overall, the study took less than 2 hours per participant.

6.4 Task

Participants experienced two VR applications for each system condition, Typing in VR (Figure 4a) and a VR Survival Shooting Game (Figure 4b). These two applications aimed to understand how participants utilize and perceive the system in stationary and dynamic activities. In both applications, participants were involved in the main task that requires their engagement in VR while handling additional event tasks we imposed on them.

In addition to the main task, participants had to complete the event tasks that would appear every 40 seconds during their VR experience. For each VR application, we prepared four different event tasks that required participants to interact with the PW. Each



Figure 4: Participants were tasked to engage in the two VR applications: (a) Typing in VR and (b) VR Survival Shooting Game while responding to VR-triggered and environment-triggered events. (c) VR-triggered events are delivered within the VR system in form of notifications. Environment-triggered events are external to the VR system.

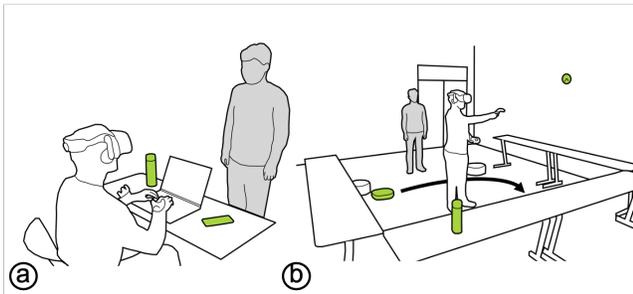


Figure 5: Scene setup for the two applications. (a) Participants sit at the desk and use the physical keyboard for typing, with a smartphone and a bottle placed on the desktop beside the participants. (b) Participants play a survival shooting game in the play area, with a bottle placed on the area boundary, a clock on the wall outside the area, and a robot vacuum wandering within the area during the experience.

event task appeared twice during the VR experience in random order. The main task ended when the participants completed all event tasks.

The four event tasks, comprising VR-triggered and environment-triggered events, were introduced to the participants before the task started. VR-triggered events are delivered within the VR system in form of notifications to simulate VR users' intention to interact with the PW, such as drinking water or checking clock time (Figure 4c). Environment-triggered events are external to the VR system, such as picking up a call on a smartphone from the PW.

6.4.1 Typing in VR. This application simulates a VR workspace and requires participants to complete typing tasks in VR using a physical keyboard. Participants were seated in front of a desk with a computer, a smartphone, and a bottle beside them (Figure 5a). The main task was to enter phrases that appeared in the VR scene (Figure 4a) using the physical keyboard. We used the MacKenzie 500 phrase set [19] as the target typing phrases. The order of the phrases was randomized and displayed in the upper block of the panel; participants repeated the same phrases in the lower block.

The four event tasks in this application were: Drink water, Check phone messages, Hang up the phone call, and Answer bystanders' greetings (see Figure 5a for the setup). These four event tasks were intended to simulate common PW interactions when using VR at desktop. The first two tasks (Drink water and Check phone messages) were VR-triggered events; participants picked up the bottle or smartphone beside them pretending to perform the action, and returned the item to complete the task. The other two tasks (Hang up the phone call and Answer bystanders' greetings) were environment-triggered events. Participants hung up the phone when the smartphone rang and replied to the bystander, played by the experimenter, when the bystander approached and greeted them.

6.4.2 VR Survival Shooting Game. This application provided an immersive game experience at room scale. Participants experienced the game in the play area (Figure 5b). They had to clear away enemies by shooting with the controller while avoiding attacks from the enemies. Since the enemies appeared from all directions, their sense of the PW can become disoriented. We further placed bonus points around the play area to prompt participants to walk, such that the need to care for physical collision is heightened.

The four event tasks in this application included: Drink water, Check clock time, Answer bystanders' greetings, and Avoid collisions with the robot vacuum (see Figure 5b for the setup). The first two tasks (Drink water and Check clock time) were VR-triggered events. To complete the task, participants had to pick up the bottle and check the clock outside the play area. The other two tasks (Answer bystanders' greetings and Avoid collision with the robot vacuum) were environment-triggered events. Participants had to answer the bystander standing beside the door for greeting his or her visit. Furthermore, they had to avoid colliding with the robot vacuum moving around the play area that simulated the case of pets or children. These event tasks simulated typical PW interactions during an immersive room-scale VR experience.

6.5 Measurement

This user study aimed to understand user strategies when interacting with *RealityLens* and explored how it affects user experience. We recorded how participants set up their reality lenses during the tasks and measured their presence in VR using IPQ questionnaires [26] on a 7-point Likert scale. Because the participants' deployment of the reality lens was uncontrolled, we did not compare task performance between conditions (such as the typing speed and game score) and only ensured all tasks were completed. At the end of the study, participants answered a questionnaire on system usability (*RealityLens* is simple to use / easy to learn) on a 7-point Likert scale, and their feedback was collected during the interview.

6.6 Participants

We recruited 12 participants (6 females, 6 males) with an average age of 25.9 years (SD = 4.03) by posting a recruiting message on social media groups of local universities. Six participants had at least 6 months of VR experience in development or gaming, and six had experienced VR only 5 times or less.

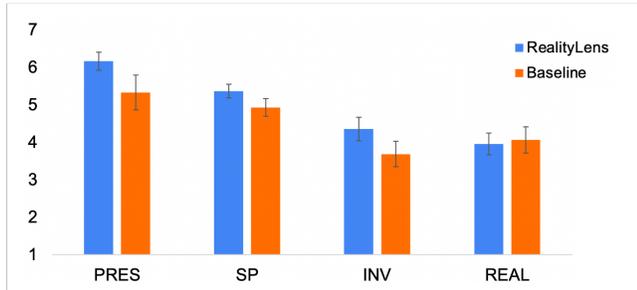


Figure 6: Averaged user ratings and standard error regarding general presence (PRES), spatial presence (SP), involvement (INV), and realism (REAL) during their experience across the study tasks.

6.7 Result

All participants completed the task successfully. This section presents the questionnaire results accompanied by the interview responses organized by category.

6.7.1 VR Presence. Figure 6 shows the results of the IPQ questionnaire. The averaged ratings on presence (PRES) were ($M=6.17$, $SD=0.83$) for *RealityLens* and ($M=5.33$, $SD=1.61$) for *Baseline*. A paired-samples *t*-test showed that the *RealityLens* condition tended to deliver a higher general presence than the *Baseline* condition ($t(11)=2.057$, $p=0.064$).

The averaged ratings on spatial presence (SP) were ($M=5.37$, $SD=0.63$) for *RealityLens* and ($M=4.93$, $SD=0.82$) for *Baseline*. For involvement (INV), participants rated ($M=4.35$, $SD=1.07$) for *RealityLens* and ($M=3.68$, $SD=1.18$) for *Baseline*. Realism (REAL) ratings showed means of ($M=3.96$, $SD=0.99$) for *RealityLens* and ($M=4.06$, $SD=1.2$) for *Baseline*. No significant difference was found.

Participants mentioned that *RealityLens* provided a more continuous VR experience, so they felt more immersed in the VR. P3 said, "I felt I was still in the virtual world even though I was trying to access the physical world's objects during the experience." In contrast, participants had to pause their VR experience and switch to the PW in the *Baseline* condition. P5 said, "It (switching to the PW) interrupted my experience and made me feel disturbed." Nevertheless, some participants stated that *RealityLens* interfered with their VR experience. P4 stated, "When it (the reality lens) appeared in my view, I would feel distracted from the VR experience," but they also admitted that this distraction enhanced their awareness of the PW.

6.7.2 RealityLens Interface. Regarding the interface usability, participants agreed that the *RealityLens* interface was easy to learn ($M = 5.41$, $SD = 1.08$) and simple to use ($M = 4.75$, $SD = 1.54$). When asked about their willingness to use *RealityLens* for their future VR experience, participants agreed that *RealityLens* helped them easily obtain the PW information and made the whole process more fluid; hence, they were willing to use it in the future ($M=6$, $SD=0.73$).

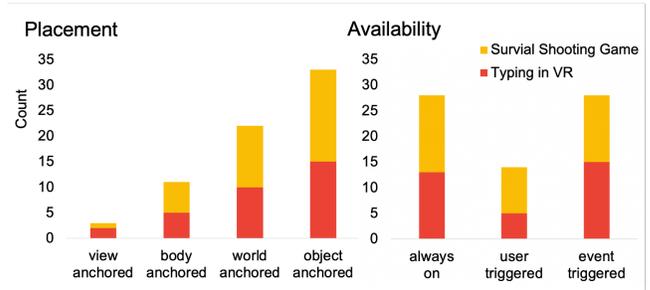


Figure 7: Usage count of each type of RealityLens across the study tasks.

6.8 Discussion

This section discusses user strategies for deploying the reality lens interface, as well as the effectiveness of the interface in different VR activities.

6.8.1 RealityLens Placement. As shown in Figure 7, most participants utilized the *environment-centric* reality lens to place on the target objects since it was considered the most intuitive solution. The *object-anchored* lens was primarily applied to interactive or moving objects such as the smartphone and the robot vacuum. Participants found this type of reality lens especially suitable for monitoring objects in the play area; hence, all participants applied this type of reality lens for the robot vacuum event. The *world-anchored* reality lens was mostly applied to static objects or was used to capture a certain PW region. For instance, some participants set up multiple small *world-anchored* reality lenses on the desk corner to prevent collision without capturing too much content (P1, P10). Alternatively, a reality lens that was placed on the desktop to capture the keyboard could also be used to check phone message (P2).

While the *user-centric* reality lens was less frequently utilized by the participants, it was found that it can be used for both accidental and prepared events. The *view-anchored* reality lens was viewed as the extension of the gap beneath the HMD and was therefore used by experienced VR users to provide them with a "more comfortable gap" (P1, P4). The *body-anchored* lens was considered a convenient and user-controllable lens. Participants placed it on the wrist of the non-dominant hand for the shooting game and raised their hand to access it when needed (P2, P7, P10–P12). They mentioned that this type of reality lens was a strong solution for most event tasks, and they would use it for both VR-triggered and environment-triggered tasks.

While participants considered the *user-centric* lens a universal solution, they mentioned that the *environment-centric* lens was also necessary for monitoring target objects or preventing user collision. Moreover, P11 mentioned that the *environment-centric* lens is suitable for objects that are frequently accessed during the experience, such as the smartphone in the typing task. This way, users did not need to move their bodies in order to access the PW content each time but simply glance over the *environment-centric* lens for easy access.

The result showed that both the task content and participants' strategies affected the participants' behavior patterns. For instance,

all participants set up the *environment-centric* reality lens on the tasks related to the PW collision, such as on the robot vacuum. For the other tasks, we found participants' behavior patterns could be roughly categorized into two types: some participants placed many *environment-centric* lenses on the targets (P1, P3–P6, P8, P9), while others utilized only one *user-centric* lens for most of the events (P2, P7, P10–P12). This further affected the usage count of lens types where the first type gave a higher number of reality lenses.

During the study, participants learned progressively about their preference for reality lens types. While we informed the participants about the event tasks' content before the application started, it was not until the events happened that they knew how the reality lens they set up earlier would perform and how they would perceive it during the tasks. Therefore, most participants were willing to try different types of reality lenses and adjust them to fit their needs during the application.

6.8.2 RealityLens Availability. Participants considered the *always-on* availability convenient since it did not require the user to turn it on or to figure out why it appeared when immersed in the VR application. Although convenient, participants also mentioned that this type of reality lens should only be applied to small targets; otherwise, it could occlude the VR content and break their presence in VR.

Also, because a *user-triggered* lens only becomes visible when triggered by users, participants reported that the *user-triggered* lens can be forgetful when they are *environment-centric* (i.e., placed in the environment). This concern becomes worse when users undergo an attentive main task since recalling the lenses in the environment further increases their cognitive load (P3, P5, P8). Correspondingly, other participants remarked setting *user-triggered* lenses to be *user-centric* (i.e., placed on the body) has the benefit of using the body as centralized storage, avoiding the retrieval problem found in managing multiple *environment-centric* lenses.

The *event-triggered* reality lens was another convenient way for participants to monitor a target during the tasks. Participants utilized this type of availability to monitor e.g., the smartphone's phone call and the moving robot vacuum. In addition to providing the PW content, this type of reality lens was viewed as another type of notification to remind the users of the event.

While the *user-triggered* lens was utilized less frequently in the study, we found that participants often set the *user-triggered* lenses to be proximity *event-triggered* to facilitate their access to the reality lens (P2–P4, P8, P9, P12). Participants stated that it was convenient to let the system trigger the reality lens since they just needed to approach the target during the task. This result aligns with previous research [20], which found that giving users full control may not be the most suitable method while allowing system control may be preferable.

6.8.3 RealityLens Effectiveness. *RealityLens* bridges PW and VR and provides participants with a fluid workflow. All participants stated that *RealityLens* helped them to access the PW objects without interrupting their work in VR. Some participants even mentioned that, since *RealityLens* blends the PW into VR well, they considered themselves situated in one reality (P1, P7).

Although *RealityLens* only blends specified PW content into the VE, participants mentioned an increased sense of safety similar

to the results of the previous study [12]. Participants found that *RealityLens* helped them locate the PW and increase their awareness about the PW. It enhances the safety without breaking their presence in VR (P4, P10). P4 stated, "Since the bottle is placed on the desktop, I get to know the desk position from the reality lens applied to the bottle. This works better than the chaperone system and increases the sense of safety in VR."

Participants reported that although *RealityLens* provided them with a more immersive experience, it also increased their effort to care for both the VR and PW content. Those who were more used to VR preferred *RealityLens* for a more continuous experience, since their skill allowed them to handle both VR and PW events concurrently. However, other participants, especially those with less VR experience, had to pay more attention to the VR experience itself. They complained taking care of both VR and PW events was demanding, so they preferred the *Baseline* condition which separated the two "realities."

We found that users' VR experience may further affect their presence in VR. Experienced participants seemed to relate their presence to the continuity in experience; hence, they had lower VR presence in the *Baseline* condition ($M = 6.17$, $SD = 0.75$ in *RealityLens*; $M = 4.6$, $SD = 1.86$ in *Baseline*). By contrast, novice participants tended to think that the *Baseline* condition allowed them to fully immerse in VR when they could not handle events in both "realities." Therefore, the *Baseline* condition prevented distractions and increased their VR presence ($M = 6.17$, $SD = 0.98$ in *RealityLens*; $M = 6$, $SD = 1.09$ in *Baseline*).

7 DESIGN INSIGHTS

This section presents design insights for a user interface that enables users to blend PW content (i.e., through a reality lens) into their VR experience.

The interface should provide the flexibility to address users' diverse needs, account for their personal habits, and allow them to learn and explore ways of deploying reality lenses in the field. We also found the preferred deployment between individuals differs. Furthermore, the interface should offer users the chance to evaluate their deployment at the beginning of the experience. Since the reality lens' pitfalls and opportunities are evaluated most effectively in the field, the interface should allow users to adjust the deployment anytime during the experience.

While letting users control the presence of reality lenses allows them to call out the lenses only when they need to access the PW content, it is less desirable when the users have an attentive task in hand. To lower their burden, expanding the ways of system-triggered lenses may be helpful.

Moreover, our interface requires the users to plan the deployment of the interface. Incorporating some degree of automation in the deployment is worth exploring in future work, for instance, suggesting users the places to deploy reality lenses by scanning the environment via the HMD with image recognition.

While blending the PW content into VR better preserves user presence in VR, compared to using the passthrough function; the latter might be preferred when the workload of handling VR and PW events in parallel is high. However, when VR activities are not intensive, the use of blending offers users a more fluid and

continuous experience. Hence, the passthrough function and blending method are suitable for different scenarios and should both be available for users.

8 LIMITATIONS AND FUTURE WORK

8.1 Extension of the study

Our study focused on user experience and strategies using *RealityLens*. Since we did not control how participants deployed the reality lens and had them set it up independently, we did not evaluate how *RealityLens* affected user performance on the VR tasks, which is worth exploring in future work. Also, we used a custom usability questionnaire to clarify that user strategies and feedback were not contaminated by apparent usability issues. To gain a comprehensive understanding of the usability of the interface, we advise future work conducting a formal questionnaire such as using the System Usability Scale [2].

8.2 Extension of RealityLens features

Our *RealityLens* interface relies on manual manipulation; this manual work can be reduced with some degree of automation. For instance, with object segmentation using computer vision techniques, the size and even the shape of a reality lens can be made fitting to the target object. In addition, the custom *event-triggered* feature in our study was simulated using WoZ. Future work can extend *RealityLens* to include a wider range of events (e.g., IoT systems) external to the VR system, enriching interactions between VR and the PW. Finally, *RealityLens* can leverage advanced blending techniques such as to enhance user immersion by painting the reality lens coherent to the VE [12].

8.3 Support for Non-VR Users' Participation

One of the goals of *RealityLens* is to help VR users communicate with bystanders. Future work can extend the support to the bystanders' active participation, for instance, allowing them to create a reality lens for interacting with VR users. This direction of research can also be found in previous studies [4, 35].

9 CONCLUSION

This work presented *RealityLens*, a user interface that allows users to peep into the PW in VR with reality lenses for their custom needs. We conducted a preliminary study to understand VR users' needs to interact with the PW and derived design considerations from the results. We developed the *RealityLens* design space, and implemented the *RealityLens* interface based on it. The proposed interface is evaluated in a study with users using the reality lens to help them handle a range of interventions from the PW. Based on the results, we presented design insights on a user interface for customizing the PW content into VR.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science and Technology of Taiwan (MOST109-2628-E-009 -010 -MY3, 110-2634-F-002-051, 111-2221-E-002-145-MY3, 111-2218-E-002-028), National Taiwan University, as well as the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of

Education (MOE). We thank Kuei-Hua Ai and Yu-Chiao Lo for their assistance in developing user study applications and Meng-Ting Shih for helping with the video shooting. Finally, we would also like to thank the reviewers for spending their time and effort providing us with insightful comments.

REFERENCES

- [1] Sidney Bovet, Aidan Kehoe, Katie Crowley, Noirin Curran, Mario Gutierrez, Mathieu Meisser, Damien O Sullivan, and Thomas Rouvinez. 2018. Using Traditional Keyboards in VR: SteamVR Developer Kit and Pilot Game User Study. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*. 1–9. <https://doi.org/10.1109/GEM.2018.8516449>
- [2] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [3] 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. <https://doi.org/10.48550/ARXIV.1502.04744>
- [4] Liwei Chan and Kouta Minamizawa. 2017. FrontFace: Facilitating Communication between HMD Users and Outsiders Using Front-Facing-Screen HMDs. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (Vienna, Austria) (MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, Article 22, 5 pages. <https://doi.org/10.1145/3098279.3098548>
- [5] Kuan-Wen Chen, Yung-Ju Chang, and Liwei Chan. 2022. Predicting Opportune Moments to Deliver Notifications in Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 186, 18 pages. <https://doi.org/10.1145/3491102.3517529>
- [6] Lung-Pan Cheng, Eyal Ofek, Christian Holz, and Andrew D. Wilson. 2019. VRoamer: Generating On-The-Fly VR Experiences While Walking inside Large, Unknown Real-World Building Environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 359–366. <https://doi.org/10.1109/VR.2019.8798074>
- [7] Isamu Endo, Kazuki Takashima, Maakito Inoue, Kazuyuki Fujita, Kiyoshi Kiyokawa, and Yoshifumi Kitamura. 2021. ModularHMD: A Reconfigurable Mobile Head-Mounted Display Enabling Ad-Hoc Peripheral Interactions with the Real World. In *The 34th Annual ACM Symposium on User Interface Software and Technology (Virtual Event, USA) (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 100–117. <https://doi.org/10.1145/3472749.3474738>
- [8] 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>
- [9] 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 (2018), 1447–1456. <https://doi.org/10.1109/TVCG.2018.2793698>
- [10] Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. 2017. ShareVR: Enabling Co-Located Experiences for Virtual Reality between HMD and Non-HMD Users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 4021–4033. <https://doi.org/10.1145/3025453.3025683>
- [11] Jan Gugenheimer, Evgeny Stemasov, Harpreet Sareen, and Enrico Rukzio. 2018. FaceDisplay: Towards Asymmetric Multi-User Interaction for Nomadic Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3173628>
- [12] 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* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300577>
- [13] Ching-Yu Hsieh, Yi-Shyuan Chiang, Hung-Yu Chiu, and Yung-Ju Chang. 2020. Bridging the Virtual and Real Worlds: A Preliminary Study of Messaging Notifications in Virtual Reality. Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3313831.3376228>
- [14] Hikaru Ibayashi, Yuta Sugiura, Daisuke Sakamoto, Natsuki Miyata, Mitsunori Tada, Takashi Okuma, Takeshi Kurata, Masaaki Mochimaru, and Takeo Igarashi. 2015. Dollhouse VR: A Multi-View, Multi-User Collaborative Design Workspace with VR Technology. In *SIGGRAPH Asia 2015 Emerging Technologies* (Kobe, Japan) (SA '15). Association for Computing Machinery, New York, NY, USA, Article 8, 2 pages. <https://doi.org/10.1145/2818466.2818480>

- [15] Kohei Kanamori, Nobuchika Sakata, Tomu Tominaga, Yoshinori Hijikata, Kensuke Harada, and Kiyoshi Kiyokawa. 2018. Obstacle Avoidance Method in Real Space for Virtual Reality Immersion. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 80–89. <https://doi.org/10.1109/ISMAR.2018.00033>
- [16] Yoshiki Kudo, Anthony Tang, Kazuyuki Fujita, Isamu Endo, Kazuki Takashima, and Yoshifumi Kitamura. 2021. Towards Balancing VR Immersion and Bystander Awareness. *Proc. ACM Hum.-Comput. Interact.* 5, ISS, Article 484 (nov 2021), 22 pages. <https://doi.org/10.1145/3486950>
- [17] Chuan-en Lin, Ta Ying Cheng, and Xiaojuan Ma. 2020. *Architect: Building Interactive Virtual Experiences from Physical Affordances by Bringing Human-in-the-Loop*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376614>
- [18] David Lindlbauer and Andy D. Wilson. 2018. Remixed Reality: Manipulating Space and Time in Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (Montreal QC, Canada) (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3173703>
- [19] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (Ft. Lauderdale, Florida, USA) (CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 754–755. <https://doi.org/10.1145/765891.765971>
- [20] 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 (Seoul, Republic of Korea) (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2143–2152. <https://doi.org/10.1145/2702123.2702382>
- [21] Daniel Medeiros, Rafael dos Anjos, Nadia Pantidi, Kun Huang, Mauricio Sousa, Craig Anslow, and Joaquim Jorge. 2021. Promoting Reality Awareness in Virtual Reality through Proxemics. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 21–30. <https://doi.org/10.1109/VR50410.2021.00022>
- [22] Paul Milgram and Fumio Kishino. 1994. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Information Systems* vol. E77-D, no. 12 (12 1994), 1321–1329.
- [23] Joseph O'Hagan and Julie R. Williamson. 2020. Reality Aware VR Headsets. In *Proceedings of the 9th ACM International Symposium on Pervasive Displays (Manchester, United Kingdom) (PerDis '20)*. Association for Computing Machinery, New York, NY, USA, 9–17. <https://doi.org/10.1145/3393712.3395334>
- [24] Joseph O'Hagan, Julie R. Williamson, Mark McGill, and Mohamed Khamis. 2021. Safety, Power Imbalances, Ethics and Proxy Sex: Surveying In-The-Wild Interactions Between VR Users and Bystanders. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 211–220. <https://doi.org/10.1109/ISMAR52148.2021.00036>
- [25] Joan Sol Roo and Martin Hachet. 2017. 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 (Québec City, QC, Canada) (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 787–795. <https://doi.org/10.1145/3126594.3126638>
- [26] Thomas Schubert, Frank Friedmann, and Holger Regenbrecht. 2001. The Experience of Presence: Factor Analytic Insights. *Presence: Teleoperators and Virtual Environments* 10, 3 (06 2001), 266–281. <https://doi.org/10.1162/105474601300343603> arXiv:<https://direct.mit.edu/pvar/article-pdf/10/3/266/1623697/105474601300343603.pdf>
- [27] Lior Shapira and Daniel Freedman. 2016. Reality Skins: Creating Immersive and Tactile Virtual Environments. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 115–124. <https://doi.org/10.1109/ISMAR.2016.23>
- [28] 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 (Seoul, Republic of Korea) (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3307–3316. <https://doi.org/10.1145/2702123.2702389>
- [29] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Bjoern Hartmann. 2020. *TransceiVR: Bridging Asymmetrical Communication Between VR Users and External Collaborators*. Association for Computing Machinery, New York, NY, USA, 182–195. <https://doi.org/10.1145/3379337.3415827>
- [30] triple A code GmbH. 2020. *Holoswitch on Steam*. <https://store.steampowered.com/app/1183720/Holoswitch/>
- [31] 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 (San Diego, CA, USA) (DIS '19)*. Association for Computing Machinery, New York, NY, USA, 487–496. <https://doi.org/10.1145/3322276.3322334>
- [32] 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*. Association for Computing Machinery, New York, NY, USA, 805–817. <https://doi.org/10.1145/3379337.3415868>
- [33] Chiu-Hsuan Wang, Seraphina Yong, Hsin-Yu Chen, Yuan-Syun Ye, and Liwei Chan. 2020. *HMD Light: Sharing In-VR Experience via Head-Mounted Projector for Asymmetric Interaction*. Association for Computing Machinery, New York, NY, USA, 472–486. <https://doi.org/10.1145/3379337.3415847>
- [34] Jackie (Junrui) Yang, Christian Holz, Eyal Ofek, and Andrew D. Wilson. 2019. DreamWalker: Substituting Real-World Walking Experiences with a Virtual Reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (New Orleans, LA, USA) (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1093–1107. <https://doi.org/10.1145/3332165.3347875>
- [35] Keng-Ta Yang, Chiu-Hsuan Wang, and Liwei Chan. 2018. ShareSpace: Facilitating Shared Use of the Physical Space by Both VR Head-Mounted Display and External Users. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (Berlin, Germany) (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 499–509. <https://doi.org/10.1145/3242587.3242630>
- [36] André Zenner, Marco Speicher, Sören Klingner, Donald Degraen, Florian Daiber, and Antonio Krüger. 2018. Immersive Notification Framework: Adaptive & Plausible Notifications in Virtual Reality. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (Montreal QC, Canada) (CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. <https://doi.org/10.1145/3170427.3188505>