Title
Interactive Cross-Dimensional Media for Collaboration and Guidance in Mixed Reality Environments

Author
Thoravi Kumaravel, Balasaravanan

Publication Date
2022

Peer reviewed|Thesis/dissertation

Interactive Cross-Dimensional Media for Collaboration and Guidance in Mixed Reality Environments

by

Balasaravanan Thoravi Kumaravel

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Björn Hartmann, Chair
Professor Eric Paulos
Professor Luisa Caldas

Summer 2022

Interactive Cross-Dimensional Media for Collaboration and Guidance in Mixed Reality Environments

Abstract

Interactive Cross-Dimensional Media for Collaboration and Guidance in Mixed Reality Environments

by

Balasaravanan Thoravi Kumaravel

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Björn Hartmann, Chair

Collaboration and multi-user interactions are key aspects of many software tasks. In traditional desktop interfaces, such elements are well supported through built-in collaboration functions or general-purpose techniques such as screen and video sharing. Collaboration and guidance may also be required in Mixed Reality environments, where users carry out spatial actions in a three-dimensional space. However, not all users may have access to the same Mixed Reality interface. All of them may not have access to the same information, the same visual representation, or the same interaction affordances. Such asymmetries make communication and collaboration between users harder. To address these issues, we introduce Interactive Cross-Dimensional Media. In these media, the visual representation of information streams can be changed between 2D and 3D. Different representations can be chosen automatically based on context or through associated interaction techniques that give users control over exploring spatial, temporal, and dimensional levels of detail. This ensures that users understand and interpret any information or interaction across different dimensions, interfaces, and spaces. We have deployed these techniques in four different contexts: (1) Mixed Reality telepresence for remote instruction of physical tasks, (2) asynchronous video-based instruction of virtual tasks, (3) live asymmetric guidance of virtual tasks, and (4) live interactive spectating of virtual tasks. Through user studies of these systems, we show that Mixed Reality environments that provide users with Interactive Cross-Dimensional Media interfaces improve performance and user experience in multi-user and collaboration settings.

Dedicated to my amma and appa,
Poongodi Kumaravel and Dr. Kumaravel Munusamy

Contents

List of Figures

List of Tables

Acknowledgments

To my advisor Björn Hartmann, I express my heartfelt gratitude. I joined your group as a student who was aware of the existence of HCI as a field. From then to now, your friendship, mentorship, support, and advice have shaped me into what I am now. Thank you for believing in me and constantly encouraging me to explore the domains I was passionate about. Thank you for actively supporting me in collaborating with other researchers in the industry. Whenever I faced an issue, even outside of my Ph.D., you always listened to and helped me with it. Whenever I faced an issue, even outside of my Ph.D., you always listened to and helped me with it. Whenever I came out of a meeting with you, I felt clearer and happier than when I entered. Over the years, I have learned a lot from you - scoping and evaluating research problems, writing papers, presenting research work, visual prototyping, constructive feedback, and the value of a good work-life balance. You have always prioritized my well-being and success, helped bring the best out of me, and made me feel comfortable and confident at every stage of my Ph.D. I feel fortunate and grateful that you are my advisor. In my career, I aspire to be the kind of friend and mentor to others, as you have been to me.

To my dissertation and quals committee members, thank you for your taking your time and offering valuable feedback. Our discussions actively shaped the conceptual contributions of this dissertation. Eric Paulos, thank you for all the conversation over the years and for always inspiring me to be creative and pursue bold ideas. The CS260B course you taught while I was new to HCI was very helpful and gave me a picture of the breadth of the field and the various research methodologies. Luisa Caldas, thank you for having me in your group meetings and showing me ways in which my research can make an impact in the field of architectural design. Finally, Marti Hearst, thank you for highlighting to me the potential of Interactive Cross-Dimensional Media in the domain of accessibility.

To Cuong Nguyen and Stephen DiVerdi from my Adobe internships, you played a crucial role in my developing interest and passion for collaborative interactions in Mixed Reality. I still remember the day I visited Adobe to try out VR painting and the feeling of being amazed by it. This dissertation would have looked very different if not for that and our continued collaborations and discussions.

To Fraser Anderson, Tovi Grossman, and George Fitzmaurice from my Autodesk internship, thank you for all the time, effort, and support you provided me. Our collaboration provided me with the confidence to build sophisticated systems and also taught me how to synthesize conceptual frameworks and design spaces. To Roya Shams-Zadeh-Amiri for being a super-user of Loki and helping with testing and carrying out the interactions with the system. To Justin Matejka for assistance with some of the figures in this dissertation.

To Andy Wilson, from my Microsoft Research internships, thank you for nurturing my visions, sharing your philosophies and ideologies about things - technical and non-technical, supporting me with your incredible engineering superpowers, and showing me the sheer joy of building systems that will benefit humanity. These have played a key role in my career plans post-Ph.D.

To my colleagues at *bcrew*, *Fort Paulos* and Berkeley Institute of Design - Andrew Head, Amy Pavel, Eldon Schoop, Elena Glassman, Forrest Huang, J.D. Zamfirescu, James Smith, Jeremy Warner, Richard Lin and Shm Almeda, Yash Pant, Cesar Torres, Christine Dierk, Molly Nicholas, Sarah Sternman and Kevin Tian - Thank you for making my initial transition to the field of HCI seem effortless, providing feedback on ideas, talks, research prototypes and papers, being always ready and willing to help, sharing diverse perspectives of the field of HCI, all the socials and board game parties, casual desk and hallway conversations, brainstorms. Special thanks to Christine Dierk, with whom I did my first HCI research project and learned valuable techniques, methodologies, and skillsets. To my mentees - Stephanie Daffara, Cyrus Vaccha, Daisy Chen, Haohua Lyu, Viren Khandal, and Erin Kraemer - Thank you for exploring new research ideas with me and teaching me to be a mentor.

To Paul Wright, who believed and admitted me to the AME lab - Thank you for always being supportive and encouraging. Thank you for taking meticulous care to ensure a smooth transition of labs for me when you retired. To the members of the AME lab where I worked during my first year - Paul Wright, Jim Evans, Bernard Kim, Dan Chapman, Marty Cowell, Michail Kiziroglou, Payton Goodrich, Zachary Gima and Zhiwei Wu - Thank you for your warm welcome to Berkeley, the countless research brainstorms, the socials, and for easing my transition to the United States.

To my advisors during my undergrad days - Arunachalam Narayanan, Sanjay Sarma, and Ravindran Balaraman - Thank you for empowering me to work on ambitious research ideas even as an undergraduate. Without them, I would not have had the interest and confidence to pursue a Ph.D. straight out of a bachelor's. To all the fantastic souls at the Center For Innovation (CFI), IIT Madras - Thank you for teaching me to pursue my dreams with persistence, showing me the joy of engineering, and teaching me to be a leader.

To my housemates over the years - Ajay Yadav, Prashanth Kotaru, and Ayush Gupta - Thank you for keeping me sane during the covid lockdown, always cheering me up, advice on important life matters, and for all the exciting times. To Vipul Gupta and other members of the Friday Ultimate Frisbees, it is because of you I learned to play Ultimate Frisbee, got my fitness going, and truly enjoyed the fun of a team sport. To Aiswarya Goutham Gouthaman - Thank you for helping with testing and logistics of a couple of works in this dissertation and for getting me to learn and enjoy Skiing.

To the *Bearpack* and *I am groot* friends - thank you for making my time at Berkeley an eventful one. To my friends from high school - Srikrishnan Sethuraman, Francis Prashanth, Siddharth Budaraju and the broader *Wolfpack*, and To the CFI *R14* team and my other friends from undergraduate - Thank you for always being there for me over the last few years, just a call away to discuss anything, anytime.

Finally, I owe immense gratitude to my parents - Poongodi Kumaravel and Kumaravel Munusamy. If not for all the painstaking effort and sacrifices that they had made, I would not be here. They are always available when I want to talk to them and have been the most extensive support at every stage of my life. I dedicate this dissertation to them.

Chapter 1

Introduction

Computing devices and their interfaces have evolved much since the invention of the first computer by Charles Babbage. We have come a long way from punch cards to today's touch screen UIs. Today, Mixed Reality (MR) environments are an emerging class of interfaces enabled by 3D displays and interactions. In these MR environments, a user is embodied within the computing environment and performs spatial actions. Therefore, such environments coexist in the physical space of users and can have varying proportions of virtual content in them, as defined by Milgram's Reality-Virtuality continuum [122]. The continuum spans from an entirely real world to an entirely virtual world. In this dissertation, we will focus on environments with some virtual content in them - Augmented Reality (AR), Augmented Virtuality, and Virtual Reality (VR).

Augmented Reality and Virtual Reality environments have been built and studied since the mid-1960s, such as Ivan Sutherland's Sword of Damocles [173, 174]. However, since only the mid-2010s, these systems are now on the cusp of commercial success. Starting with the release of Oculus' DK2 in 2014, they have grown beyond research prototypes in labs to popular mass consumer platforms today. As of this writing, with 15 million Oculus Quest 2 headsets sold, it has surpassed the number of Xbox consoles sold this year. The headsets are powered by a growing ecosystem of applications, games, and media. While these MR headsets have primarily been used for entertainment, prior and ongoing research show promise in their ability to enhance our everyday workspaces. For Augmented Reality, the more common forms today are tablet/mobile phone-based Augmented Reality. These are used for games, distance measurements, navigation, testing the appearance of products before purchase, creative content production, and many other applications. Wide-scale adoption of headset-based Augmented Reality devices such as Hololens is yet to occur. Today, the mass consumer market use cases do not justify their cost. However, these are used in workplaces for enterprise, education, medicine, and other applications.

This dissertation analyzes interactions in AR and VR environments under the unified umbrella of Mixed Reality (MR) environments. A major problem in these MR environments today is that they are focused on the usage and needs of a single user. Today they tend to isolate the user from their peers [67, 69]. It is a challenge to share, co-experience, and

collaborate in these environments, especially if these were not programmed to be collaborative or if peers use different types of interfaces to access a Mixed Reality environment. These *asymmetries* induce communication barriers between the users. For instance, most Mixed Reality systems can "mirror" the first-person view of the Mixed Reality user to an external display. However, this only provides a partial view of the larger Mixed Reality environment, and even so, it is hard to watch because of its shaky nature. It is also difficult for a bystander to interact with the Mixed Reality user solely by using a video feed. Section 4.1 discusses other types of asymmetry and collaboration challenges that can occur in Mixed Reality environments.

1.1 Purpose and Thesis

The purpose of this dissertation is to answer the question : *How do we enhance collaboration and guidance for users of Mixed Reality environments?*

And the dissertation's thesis is that "*Mixed Reality environments that provide their users with our novel Interactive Cross-Dimensional Media interfaces improve performance and user experience in multi-user interaction and collaboration.*"

1.2 Challenges

This dissertation explores the challenges that arise when multiple users want to interact with, collaborate with and offer guidance to one another in Mixed Reality environments.

xspaces and XD media: CSCW for Mixed Reality Environments

It is hard to analyze these challenges for collaborative work with the help of existing frameworks from the Computer-Supported Cooperative Work (CSCW) literature. Most of these frameworks are primarily targeted at conventional desktop computer interfaces that typically take the form of 2D UI on flat-screen displays. In contrast, Mixed Reality environments use a combination of 2D UI and 3D and spatial UI, all of which users use in an embodied manner. In this dissertation, we start with the simplest CSCW analysis tool, Johansen's Space-Time matrix [95] and adapt it to make it suitable for categorizing and thinking about collaborative interactions in Mixed Reality environments. To do that, in this dissertation we propose the concept of an *Extended Space* (*xspace*), and use that to define an *Extended Space - Time* matrix. With this matrix, we can categorize collaborative interactions between users of Mixed Reality interfaces as *symmetric* or *asymmetric*. We identify four potential sources that can lead to *asymmetry* between users.

To mitigate these asymmetries between users, we propose novel media and associated interaction techniques - *Interactive Cross-Dimensional Media*. These media have two key properties - P1: Variable dimensional representations and P2: Meaningful translation of

interactions. These properties give rise to three primary interactions - Perspective exploration, Temporal exploration, and Dimensional-detail exploration. Using the concept of *xspaces* and Interactive Cross-Dimensional Media, we propose a design methodology to help design collaborative interactions for Mixed Reality environments. As exemplars, we analyze four scenarios where there exist challenges in multi-user interactions in Mixed Reality environments, and we will show how the use of Interactive Cross-Dimensional Media can mitigate these.

Loki: Facilitating Remote Instruction of Physical Tasks

Teaching physical tasks used to require in-person interactions. This allows the learner and the instructor to take different perspectives and use the different shared objects in their environment. Today, doing this through video calls is not uncommon. However, teaching through these is hard because it restricts users' freedom to take different perspectives. Furthermore, neither of them has any way to offer and ask for contextual guidance anchored to a physical space. Mixed Reality technology is promising for such scenarios. While prior work has introduced specific configurations of MR-based instruction, we find that a broader design space exists for Mixed Reality systems that aim to facilitate remote instruction of physical tasks. Different configurations of this design space can allow for different stages of instruction identified by the learning science literature [33]. Each of these stages may have an *xspace* and a nature of *asymmetry* that is different from the other.

To address this, we present the Loki system that allows users to flexibly move between these different configurations of the identified design space and support the different stages of instruction. For each configuration, Loki offers an Interactive Cross-Dimensional Medium that mitigates the *asymmetry* between users who work in the *xspace* for that corresponding stage of instruction. The medium uses 2D videos, 3D spatial captures, and abstract 3D graphics along with its associated interaction techniques that allow users to take different perspectives, annotate the local and remote environments, and record and review their performance and that of their peers. We validate Loki's utility through walkthroughs of how Loki can help in different scenarios and a qualitative user study (n=8) with one such scenario. This showed that Loki enhanced the instruction of physical tasks.

TutoriVR: Video-Based Asynchronous Instruction for Design Tasks in Virtual Reality

Next, we move on to enhancing the instruction of virtual tasks, and a popular one today is the task of 3D designing using Virtual Reality. Today, advanced skills and techniques of VR design tasks are mostly learned using videos posted on online platforms like Youtube, Twitch etc. This is mostly an asynchronous form of instruction. Such videos are common because they are relatively easy to create. An instructor must hit a record button, demonstrate desired techniques, and stop the recording. This is then easy to upload and share on online

platforms. But just like what we saw for physical tasks, learners face many issues learning VR tasks from 2D videos.

We conducted a formative study (n=6) and found two broad issues. The first is depth judgment: it is hard for users to perceive depth from a 2D video because it flattens the depth information for the viewers. Hence, it can be hard to figure out the relative depth of different elements. However, the instructor who recorded the video had no issues with depth because they saw their environment in 3D due to the stereo view of their VR headset. This is one source of asymmetry between the instructor and the learner. The second set of issues relates to the understanding of the instructor's interactions in the VR environment which result from multiple sources:

- Deceiving/inadequate perspectives in videos
- Rich spatio-temporal nature of controller actions
- Controller actions not being visible either because they are occluded or because they are outside of the instructor's viewport

To address these, we present TutoriVR, which deploys embedded widgets within existing VR applications. These widgets display the video tutorials and supplement them with appropriate Interactive Cross-Dimensional Media that provide 3D and contextual aids to enhance the learner's understanding. It provides the learner with a stereo view of the scene and isolated 3D reconstructions of the instructor's actions in the video. These allow for perspective, temporal and dimensional-detail exploration of the virtual tasks and actions that the instructor in the video is performing. An exploratory user evaluation (n=10) showed that these Interactive Cross-Dimensional Media helped users better understand the VR scene and the instructor's actions recorded in the video-based asynchronous instruction.

TransceiVR: Video-Based Synchronous Collaboration for Virtual Reality Tasks

What if one wants to carry out such interactions synchronously with a VR user? This is a common need today. Virtual Reality (VR) users often need to work with other users who observe them outside of VR with the help of an external display. Today's de-facto solution for such interactions is the VR mirror. It shows an external user a first-person video feed of what the VR user sees. But, VR user does not see the external user and their actions.

On the other hand, the external user cannot see VR scene elements outside the VR user's view. These asymmetries make communication between the two difficult. We conducted formative interviews (n=5) with experts to understand the goals and challenges in such interactions. This revealed two challenges. First, the constantly moving first-person view was hard to watch and interact with. The second set of issues related to talking about and referring to the different types of VR scene elements — transient elements, VR controllers, gestural actions, and directional and attentional elements. To address these challenges, we

present TransceiVR, which uses a novel Cross-Dimensional Medium that allows external users to explore the VR scene spatially or temporally, annotate elements in the VR scene at correct depths, and discuss via a shared static virtual display. Similar to TutoriVR, it retrofits and functions over existing VR applications. An initial co-located user evaluation (n = 10 pairs) showed that the medium made these synchronous collaborative interactions more effective and successful with respect to task time, error rate, and perceived workload.

DreamStream: Immersive and Interactive Spectating for VR

While the systems above enhance guidance and collaboration for tasks carried out in VR, neither focuses on enriching the spectator's viewing experience. Today's VR experiences are compelling for the VR user because they feel immersed in them and have increased spatial awareness. They view the scene in 3D because VR is a stereo display, can carry out enhanced spatial interactions, and move around naturally in the virtual world as they would in real life. None of these hold for the spectators who watch them. If anything, watching video feeds of VR experiences is worse than regular video feeds. VR feeds are often first-person, shaky, and uncomfortable to watch. VR users move around a lot, and this can disorient the spectators. Compared to traditional 2D media and interfaces, VR feeds increase the disconnect between the spectators and a VR user. Even if spectators have access to a VR headset, they are forced to watch a 2D video today. To address this asymmetry, we present the DreamStream system that enables Cross-Dimensional Media interactions by performing a live 3D reconstruction of the VR scene, VR user, and their actions. The medium allows spectators to view the reconstructed scene using VR headsets and see everything in stereo 3D. They can navigate freely in the scene and take independent perspectives. They can also interact with the primary VR user they are spectating, using abstract 3D avatars and laser pointers. Our user evaluation (n=12) showed that DreamStream offered increased immersion and made communicating and interacting with the VR user easier.

1.3 Contributions

This dissertation explores guidance and collaborative interactions in Mixed Reality environments. With Mixed Reality emerging as a popular interface among users, we aim to better understand and address challenges in such interactions. To achieve this, the dissertation makes the following conceptual contributions:

- The concept of Extended Spaces (*xspaces*) and an adapted CSCW matrix that introduces the notion of users being in symmetric or asymmetric *xspaces* (Chapter 4). The matrix allows for better classification and understanding of collaborative work using Mixed Reality interfaces.
- The concept of *Interactive Cross-Dimensional Media (XD media)* (Chapter 4), novel media and associated interaction techniques that allow users to carry out spatial, tem-

poral, and dimensional-detail explorations of information of interest. A design methodology leverages the notion of *xspaces* and *XD media* to design systems that enhance user communication and collaboration by mitigating their asymmetry.

The dissertation then shows how these concepts can aid collaboration and guidance in MR through four interactive systems and their evaluations:

- *Loki* (Chapter 5), a system that uses a novel Cross-Dimensional Medium and associated interaction techniques that allows users to switch to different configurations of our novel design space. The design space explores real-time bi-directional Mixed Reality based remote instruction of physical tasks. Our qualitative evaluation showed that by supporting a range of visual representations and various mechanisms for data capture and rendering, Loki's Cross-Dimensional Medium serves as a rich communication medium that leverages spatial data, video, annotations, and playback to help connect users as they teach and learn physical tasks.

- *TutoriVR* (Chapter 6), a system that uses VR-embedded widgets, which are application interfaces that are observable and interactable directly inside existing commercial VR applications. The widgets leverage Interactive Cross-Dimensional Media to mitigate the *xspace* asymmetry between the learner and the instructor. An exploratory user evaluation (n=10) showed that these Interactive Cross-Dimensional Media helped users better understand the VR scene and the instructor's actions recorded in the video-based asynchronous instruction.

- *TransceiVR* (Chapter 7), a system that facilitates synchronous interaction between a non-VR external user and a VR user. It provides the external user with a touch-tablet interface with a Cross-Dimensional Medium that allows them to perform perspective and temporal exploration, annotate the VR scene, carry out discussions using a shared static virtual display, and trigger highlights on controllers. A user evaluation of the system (n = 10 pairs) showed an improved efficiency in communication in such interaction scenarios where users use such interfaces.

- *DreamStream* (Chapter 8), a system that uses a novel streaming approach to carry out interventions at the VR platform level, uses 3D capture systems, leverages depth buffers to stream 3D views at scale via conventional encoders such as H.264. The system implements a Cross-Dimensional Medium with immersive and 2D interfaces, which allows the spectators of a VR activity to cohabit the virtual space of the VR player and offers interactions similar to a multi-user system. A user evaluation (n=12) showed that DreamStream's VR viewer interface offered increased immersion and made it easier to communicate and synchronously interact with the VR player. It also made them feel co-present with the VR player.

1.4 An overview of the dissertation

Chapter 2 provides a review of background knowledge on four areas that are relevant to the thesis — the computer-supported collaborative work matrix, a space of software implementation approaches, designing systems for spectating VR activities, and the instruction of physical tasks.

Chapter 3 provides a review of prior works that are relevant to the dissertation, specifically on collaboration and tutorial systems for conventional user interfaces; systems that aid in the instruction of physical, virtual and Mixed Reality tasks; and research on asymmetric collaboration and spectating activity.

Chapter 4 introduces the two main conceptual advances of this dissertation: Extended Spaces and Cross-Dimensional Media.

Chapters 5–8 describe four implementations of Cross-Dimensional Media to support different collaboration and guidance tasks, as described above.

In Chapter 9, we restate the contributions. Across this dissertation, we discuss how we can analyze challenges in collaboration and guidance in Mixed Reality environments. We show how Interactive Cross-Dimensional Media can enhance the quality of multi-user interactions for general spectating, guidance, and collaboration. We reflect on the strengths and challenges of Interactive Cross-Dimensional Media across the four contexts we have explored. We conclude the dissertation with an outline of possible extensions.

1.5 Statement of Multiple Authorship and Prior Publications

This thesis is based on the following previously published papers: TutoriVR at CHI 2019 [186], Loki at UIST 2019 [184], TransceiVR at UIST 2020 [185], DreamStream at CHI 2022 [183], and the journal article titled "Interactive Mixed-Dimensional Media for Cross-Dimensional Collaboration in Mixed Reality Environments" at Frontiers Virtual Reality published in 2022 [182]. I am the primary author of each publication and led all of the corresponding projects. But, this research could not have been completed without my advisor Björn Hartmann and my coauthors, who have generously contributed, and provided valuable advice and ideas throughout these works. Cuong Nguyen and Stephen DiVerdi for TutoriVR and TransceiVR; Fraser Anderson, Tovi Grossman, and George Fitzmaurice for Loki and Andy Wilson for DreamStream.

Chapter 2

Background

Mixed Reality interfaces have a dual nature. They can be viewed through the lens of traditional desktop interfaces. An alternative lens is to treat them as 3D spaces in which users perform spatial tasks in an embodied manner. Sections 2.1, 2.2 and 2.3 discuss relevant background knowledge for the former and Section 2.4 the latter.

2.1 Computer Supported Cooperative Work Matrix

A key focus of this dissertation is collaborative interactions in Mixed Reality computing environments. A long-running and popular approach to analyzing computer interfaces for meetings and collaboration is to use the canonical CSCW matrix proposed by Johansen [95]. The canonical CSCW matrix has two axes. On one axis, we have *Time / Synchronicity*. It is binary valued depending on whether the user interaction happens simultaneously. If it is, then it is said to be a synchronous interaction. Otherwise, it is an asynchronous interaction. On the other axis, we have *Space*, which refers to the *Physical Space*. This is also binary valued depending on whether users are in the same physical space. If they are, then it is a co-located interaction. Otherwise, it is a remote interaction. For this reason, this matrix is also called the *Time-Space matrix* (see Figure 2.1).

When developing an interface to facilitate collaboration between a small group of two or more users, we categorize them using the different quadrants of the CSCW matrix. For example, consider two users who operate in the same space but at strictly different times - this would be considered a colocated-asynchronous interaction. In this case, the interface we build to support the users would need to persist information across time — e.g., through team rooms or public displays. Similarly, suppose users are in different physical spaces. In that case, we want to ensure that our interface transmits the relevant information to the different spaces — e.g., through screen sharing or email.

While the CSCW matrix is a popular way to classify and understand cooperative systems, it is not without limitations. The matrix falls short in categorizing numerous modern systems, and to address such limitations, various extensions have been proposed to the CSCW

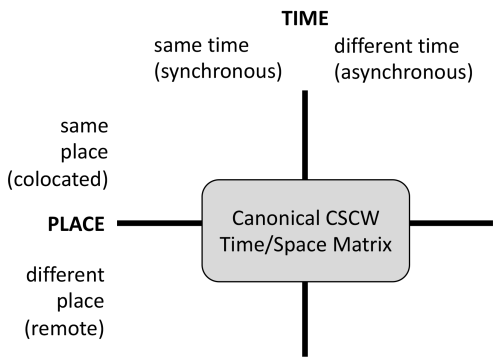


Figure 2.1: Canonical CSCW Time/Space Matrix.

matrix. Dix. et Al's book, *Human-computer Interaction*[41], refines the matrix by adding more granularity to the time axis - *Concurrent synchronized*, *Serial*, *Mixed* and *Unsynchronized*. Andriessen [4] proposed an extension of the matrix by taking into consideration five types of collaboration technology - *Communication systems*, *Information Sharing systems*, *Cooperation systems*, *Coordination systems* and *Social encounter systems*. This extends the 2x2 matrix into a taxonomy that takes the form of table with the dimensions 5x3, where the five rows are the types of technology, and the three columns are the regions in the conventional matrix - same-place/same-time, different-place/same-time, different-place/different-time. Grudin [66] proposed an extension of the matrix, where asynchronous interactions are further divided as *different but predictable time*, and *different but unpredictable time*. Remote interactions are also further divided as *different but predictable place*, and *different but unpredictable place*. This added an extra row and column to the matrix resulting in a 3x3 matrix. Furthermore, it was also known that using the matrix is primarily for small group cooperative work. To address this, Lee and Pain [109] propose the *Model of Coordinated Action* that utilizes seven dimensions, adding five more to Johansen's CSCW matrix. The seven dimensions are: *Synchronicity*, *Physical distribution*, *Scale*, *Number of Communities of practice*, *Nascence*, *Planned permanence* and *Turnover*.

The work and systems in this dissertation will focus on the cooperative work between small groups of users, some of whom will use Mixed Reality interfaces. We will start with Johansen's conventional CSCW matrix, and in chapter 4, we will see why the matrix is insufficient for studying collaborative interactions in Mixed Reality. We will address it by proposing a modified matrix, the *Extended Space - Time matrix*, in which we redefine the notion of *Physical Space* to that of an *Extended Space* which better captures the nature of tasks in Mixed Reality. While challenges still exist due to the binary nature of the time axis, this has been addressed by the aforementioned works and is not the focus of this dissertation.

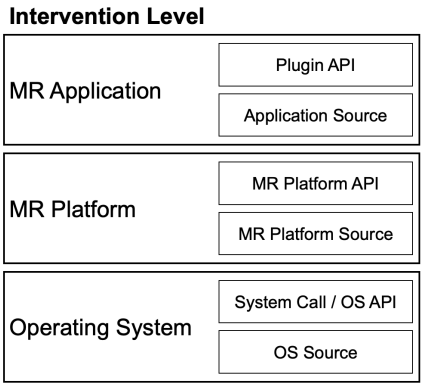


Figure 2.2: Implementation approaches can make interventions at different levels of the MR software stack.

2.2 Space of Software Implementation Approaches

User interfaces for collaborative interactions in Mixed Reality applications can operate and intervene at different levels. This means there exists a broad space of implementation approaches that require the interface to have different levels of programmatic access to the desired MR Application that needs to allow for collaboration and guidance (see Figure 2.2):

1. MR Application source code access: For applications where a MR developer has source-level access, support for asymmetric communication can be included in the application, either by rewriting the application or, more easily, by including a module that encapsulates collaboration functionality prior to compilation. For instance, one can incorporate the external user in the VR environment by placing additional virtual cameras controlled by and rendered to the external user. Prior work such as ShareVR [69], WebTransceiVR [113] and FaceDisplay [68] use this approach.

2. MR application plug-in access: MR application developers may also choose to expose plug-in access, which can allow third-party developers to implement similar features without needing source code access. While in conventional desktop interfaces, application-specific plug-in APIs are common (e.g., in Adobe Creative Suite, CAD software), we are unaware of any MR application that currently allows this.

3. MR Platform API access: MR platforms are the most common way in which applications are deployed today. For instance, the SteamVR platform has over 4000 MR titles, including BeatSaber, Tiltbrush, Blocks, SkyrimVR, and Half-life Alex [168]. For closed source applications that run on top of such a MR platform, this layer's API can extract relevant runtime information and inject graphics into the running application. The TutoriVR system (Chapter 6) and TransceiVR system (Chapter 7) discussed in this dissertation utilize this approach. Major platforms currently allow API-level access to video feeds of both eyes, poses of MR hardware (headsets, controllers, trackers), and the ability to inject graphical

overlays independently to each of the eyes. Such access makes it possible to render custom 3D elements in the virtual scene of a closed source application. Some projects also have modified and recompiled the platform API/DLL to add additional functionality. The DreamStream system (Chapter 8) in this dissertation and RealityCheck [79]) utilize this approach wherein they modify the OpenVR library used by the MR applications.

4. Operating System (OS) level: An application might also operate at the operating system level; however, calls are likely too low-level. As with MR plug-ins, we are unaware of projects that operate directly at this level. TransceiVR is built on the platform API level of access, which strikes a balance between universal applicability (it supports any application running on the platform API) and richness of application state that can be sensed and modified (video feeds and pose data, but no scene graph or application semantics).

From formative studies that were carried out as a part of this dissertation work, we found that it is common for users to have collaborative conversations about MR software that they or their organization had written and had source-code-level access to; as well as applications that were only available as closed source binaries (such as Blocks[59], or Quill[150]). To be broadly useful, Cross-Dimensional Media interactions in this dissertation were designed and implemented so that they can operate at the *Platform API* access level for MR applications. Thus, it can retrofit any existing single-user MR applications with novel interactions that facilitate collaboration and guidance.

2.3 Designing Systems for Spectating VR Activities

Multi-user apps and games can be networked and shared with spectators in various ways. Typically there is a player who performs some activity, observed by spectators. Today, there are a few broad approaches in which the player's activity could be shared in VR:

- *Multiplayer approach*: The player and the spectators each have a local copy of the application assets. For spectators, the player experience is recreated by transmitting the application data and states such as player and object poses and events. Most multiplayer games today use this approach.
- *Streaming approach*: A video stream of the player's VR activity is streamed to spectators typically via an online service such as Twitch, MixCast, or Liv. The video stream may be generated from the rendered first-person view of the player or a separate third-person camera.
- *Cloud approach*: Games may now be hosted in the cloud, and therefore may support spectators in the same manner as locally hosted multiplayer games, but with the advantage that all assets are stored on the cloud and rendered video frames are streamed to each player and spectator.

Given these broad approaches to sharing activities in VR, the following design considerations come to light:

Where is the spectator view rendered?

Multiplayer approach - Rendering at everyone's PC

The *multiplayer approach* recreates the player's experience for spectators using the same graphical assets found on the player's system. The view for the player and all the spectators is rendered on each computer. Such a local rendering has the advantage that it can proceed asynchronously with the player and, therefore, can meet VR hardware's high frame rate and low latency requirements. In addition, these natively support giving the player and spectators their own view in the VR application.

However, this approach can be very demanding for the spectator, especially for occasional use: they must have a copy of the application installed on their system, and their system must be powerful enough to run the entire application in VR. Finally, the application must be designed and developed to support multi-user features.

Streaming Approach - Rendering at the player's PC

The *streaming approach* on other hand, is a popular activity sharing approach today. Here, a 2D video feed of the VR user's activity is encoded and streamed to multiple spectators simultaneously using desktop sharing techniques and standardized video codecs. The spectator is typically not in an immersive environment and need not have the VR application installed. This is possible because the infrastructure facilitating video streaming is widely available across every device and scales for many users. Additionally, for most computing systems today, screen-capturing ability is supported at the platform level, as outlined in the previous subsection. This means that, in contrast to other approaches, any existing VR application can be streamed as a 2D video.

While streaming approaches benefit from the generic nature of 2D video capture, they lack support for multiple views. Spectators have no control over their view of the scene since only a single video stream is encoded and transmitted to all the spectators. Hence they all take the same perspective of the VR scene. They typically see the player's first-person view, which can be difficult to watch due to camera movement, especially in VR.

Cloud approach - Rendering at cloud server

With the *cloud approach*, the application is stored and run on a remote server. The spectator and the player need not have the application installed on their local computers. This approach is, therefore, closer to supporting the "drop-in" access of streaming. For multiplayer games on the cloud, every user can independently choose and control their view of the player's scene. The server in the cloud is ideally powerful enough to simultaneously render, encode and transmit the different video streams corresponding to viewports of different spectators. However, just like with the *Multiplayer approach*, such a multi-spectator support needs to be built into the application during development and does not work for every application. This is currently deployed for non-VR games by services such as Google

Stadia, Amazon Luna, and Xbox Cloud Gaming. An important consideration is the latency between a user's input action (often made using a controller) and the updated video stream that results from the input action, i.e., the time between a spectator or player invoking a move action and the updated stream corresponding to the move action. Latency to cloud systems is typically on the order of 50-100ms. This is high for Mixed Reality applications because frames need to be rendered at around 60-90fps, which means we need a latency between 11-16ms. Compensating for this difference is actual system latency, and perceived latency by a user is still an active area of research.

How is the view rendered to the spectator?

The VR player performs activities in an immersive 3D environment. However, spectators' views can be rendered in different ways. For example, they might share the same immersive environment and view 2D, stereo, or 360-degree video.

Each of these modalities preserves detail at different levels. 2D video is the most common but degrades the sense of 3D depth and the spectator's overall spatial and directional awareness [185]. Stereo video enhances the perceived 3D depth and is used today in 3D TVs and 3D cinema. This requires some kind of hardware to provide different video streams to the user's eyes. Today, the 2D and stereo feed of any VR application can be captured at a platform level [185]. Theoretically, it is feasible to have the spectators observe the activity in VR via a stereo video feed that fills the left and right eye displays with corresponding video views. However, when viewed in VR, such stereo feeds may cause motion sickness for spectators depending on when there is significant camera motion. The TutoriVR (chapter 6) system in this dissertation will use a stereo feed, but it renders it like a 3D TV within VR. Furthermore, it applies "*blacktaping*" techniques to reduce the user discomfort caused by window violation [5].

360-degree video enhances spatial awareness of a scene by allowing the user to change their viewpoint and is becoming increasingly common in online video sharing platforms [49]. Stereo 360-degree video is an increasingly popular option in VR but provides limited means of changing the viewer's position. Additionally, access to such 360 video feeds is not present in existing applications and needs to be supported explicitly by developers.

Finally, a complete 3D environment is rendered locally, like a game, and ideally retains the perceived depth through stereo rendering. It allows spectators to look around the VR player's environment and move within it.

What interactivity is offered to the spectator?

A vital aspect of any activity between the player and the spectator is the feeling that they are together - a sense of co-presence.

Interaction within *multiplayer approach* and the *cloud approach* is limited by what the developer has anticipated and built into the game. For example, spectators may adopt different perspectives on the scene and perceive different parts of the action. To facilitate

co-presence in such scenarios, existing systems typically adopt an avatar-based representation of each user in the space. These avatars can be abstract or real-life replicas of the users and can vary in degrees of freedom depending on the fidelity of the pose that is tracked for each user. These are commonly seen in today's social VR applications such as Job Simulator, Mozilla Hubs, and Horizon Workrooms.

With the *streaming approach*, the interaction between the player and spectators depends on what the streaming platform supports. Today, that is typically text chat and is rendered as an overlay in the VR scene or shown on a web page. To allow for interaction at scale, text chat has been enhanced with techniques such as displaying viewer counts, consolidated emoji reactions, polls, etc., being used. For a smaller group of users, audio chat and screen control sharing and annotations also become feasible. With 360-degree videos, techniques that convey the viewport of different users have been used in prior research systems [131].

2.4 Instruction of Physical Tasks

When we treat Mixed Reality interfaces as 3D spaces in which users perform spatial actions in an embodied manner, a specific form of multi-user interaction often occurs: Guidance and instruction of physical procedural skills. The question of how to appropriately teach such skills has received significant attention in the learning sciences and specialized domains where such skills are essential to job performance, for example, in surgery [153, 155] and athletics [94, 121]. Several learning theories are particularly applicable when designing systems to support skill acquisition.

Fitts and Posner's three-stage model of motor skill acquisition [51] describes a process that begins with a cognitive stage (where movements are actively observed, reasoned, and talked about); an associative phase, where some aspects of movement are controlled consciously and some aspects are automated; and the autonomous phase, where movements become fluid, accurate, and largely automatic. Focus shifts from acquiring gross, general movements to finer details through this process.

Collins et al.'s model of cognitive apprenticeship [33] highlights the changing role of the teacher in moving a learner from novice to expert performance. It starts with the teacher modeling the desired action. At the same time, the learner observes, then offers feedback through coaching and scaffolding as the learner performs it, and finally, a phase for the learner for review and reflection.

Kolb's experiential learning cycle [104] also highlights the importance of action and reflection. It distinguishes four stages: having a concrete experience, reflecting on it, followed by abstract conceptualization (drawing conclusions), and active experimentation. Taken together, these theories suggest that any system targeted at teaching physical tasks should be dynamic and fulfill several requirements: allow the learner to observe the teacher (for *modeling*) and vice versa (for *coaching*); allowing the teacher to provide effective feedback during task performance (live) or after a task (through a recording); enabling abstraction and conceptualization for both teacher and learner (e.g., through annotations and other

ways of going beyond direct observation); and supporting a learner's reflection after task performance, e.g., by jointly reviewing its recording.

Different systems in this dissertation aim to support one or more of these requirements. For example, two works - TutoriVR and Loki - focus on learning about such physical actions for carrying out tasks. While TutoriVR aims to exclusively support the *modeling* phase of doing virtual tasks, Loki supports all of these requirements of learning physical tasks by allowing the learner and instructor to select appropriate views and representations of each other's performance spaces.

Chapter 3

Related Work

This dissertation focuses on studying and enhancing multi-user interactions in Mixed Reality environments. These interactions can take different forms, and in this dissertation, we focus on three common types - instruction, task-oriented collaboration, and sharing activities for spectating. As before, this chapter will discuss these interactions from the lens of treating Mixed Reality Interfaces as traditional desktop interfaces and through the lens of 3D spaces in which users carry out spatial tasks. Besides this, we will also review prior work within the domain of Mixed Reality itself.

3.1 Systems that Support Tasks Performed in Conventional Computer Interfaces

Tutorial Systems

Providing effective instructions for conventional desktop applications has been an active research area within HCI. Previous work focuses on tutorials for desktop applications such as design software [60, 30, 147, 64], sketching [50] and 3D modeling [39, 43]. A major theme in these works is to generate high-quality tutorials based on the raw video recordings of a user workflow. This approach can reduce the authoring overhead of the tutorial creator and encourage more sharing of community-generated tutorials. This dissertation will aim to extend this concept to Mixed Reality applications. An important concern in these works is at what application level the system should record the workflow. The recording can either be *application-specific* (e.g., Blender edit operations [39], Photoshop commands [30, 60], Paint.NET document edits and UI events [65]) or *application-agnostic* (e.g., Collecting platform level data such as mouse and keyboard inputs [8, 127, 130], OS-level activity tracing [126]). The systems in this dissertation will focus on the latter type of recording, in which we aim to retrofit systems over existing applications.

Prior work has also explored mixed-modal guidance, where different media types are employed to guide a user's actions. Recent examples of this include MixT [30], where a mix

of video and text content is used to convey information; ToolClips [64], where the contextual video is used to guide actions in a software tool; ElectroTutor [190], which uses a combination of textual instruction, interactive questions, and signals to generate tutorials on building physical computing systems; and Torta [126] that uses a combination of screencast videos along with underlying OS activity traces for generating mixed-media GUI and command-line app tutorials. Our work employs Cross-Dimensional Media, focusing primarily on the visual modality. However, in contrast to prior work, it uses visual representations that span different dimensions of the visual modality.

Synchronous Instruction and Collaboration

One of the earliest instance of synchronous remote instruction and collaboration was Doug Engelbart's "Mother of all demos" [46] in 1968. It showcases elements such as video conferencing, version control and collaborative document editing etc,. working in tandem in a single system. Early work in collaboration over computer software have focused on concurrency issues that can occur when users work on a shared computer task. Greenberg and Marwood [62] analyzed how these issues can affect collaboration tasks over computer software. They suggest how the different techniques to solve these can have varying impacts on how users perceive and interact with the task. Prior work has studied such collaboration issues in specific domains such as document editing [170], code editing [40], engineering design [36, 158] and has developed tools to solve these issues.

General-purpose techniques to access and share 2D application windows or entire desktops have been pioneered by Virtual Network Computing (VNC) tools. These sharing features are most often implemented using video streaming codecs so that other users on the call need not download or install the shared application [154]. This paradigm has been extended to support simultaneous screen-share and collaboration by a group of users [72, 43]. Along with sharing such video streams, most commercial video conferencing tools today include the ability for a remote party to control single-user software on someone else's computer using screen sharing and input event injection. It also allows them to share a whiteboard and have multiple users perform annotations over the shared screen. Cross-Dimensional Media in this dissertation borrows from these interactions but extends them to suit the needs of Mixed Reality Environments.

Retrofitting

A common way to share instruction and collaboration for conventional computer interfaces is by retrofitting them with other software. Such retrofitting is not new for traditional computer software and frequently uses a combination of available platform APIs underneath a closed source application (e.g., using UI toolkit overloading [44] or accessibility APIs) and reverse engineering approaches to extract information where APIs are unavailable. The value of retrofitting and reverse engineering has been well established in the HCI research community. As Chen et. al [28] write:

"*Mission critical applications and legacy systems may be difficult to revise and rebuild, and yet it is sometimes desirable to retrofit their user interfaces with new collaborative features without modifying and recompiling the original code.*"

Computer vision-based reverse engineering approaches have been used to enhance desktop software with new interaction techniques [42], automate GUI tasks [42], and extract reusable data from rendered information visualizations [156]. Furthermore, they have been used to improve the usability of video tutorials [147].

3.2 Systems that Support Physical Tasks

Instruction of Physical tasks

Many systems designed to support the teaching of physical tasks are automated systems, providing automated feedback through heuristics or measures computed from comparisons to template actions [61, 177, 187]. These systems often focus on experiential learning, allowing the trainees to perform an action and then receive feedback from their actions which they can reflect on. However, this feedback is often coarse and may sometimes be inaccurate. This approach may not be appropriate for teaching physical tasks, which often requires the learner to observe a demonstration by the instructor, perform it alongside the instructor, and then get feedback [33, 84, 103, 186]. Additionally, providing learners with some control over the availability and modality of feedback is beneficial for learning [94, 161].

Current systems for teaching physical tasks primarily focus only on the psychomotor phase of learning [3, 81, 165]. However, this is just a part of learning physical tasks, another part being the elements of the environment and associated interactions with them. We are unaware of any systems that fully support cognitive apprenticeship [33] discussed in Section 2.4. To address this gap, the Loki System (Chapter 5) in this dissertation provides a bi-directional medium where both the learner and instructor can transition between each other's physical environments to enable modeling and coaching and reflection in a meaningful manner.

Computer-mediated Telepresence and Collaboration for physical tasks

Early work in this domain relied on video-conferencing tools. Works from Ishii et. al such as Videodraw [176], TeamWorkStation [90], ClearBoard [91] use a combination of monitors, cameras and translucent screens to facilitate a shared collaborative drawing board across two or more users. Works from Gaver [56], and Hollan and Stornetta's "Beyond being there" [83] suggest that while a purely audio-video medium conveys a limited set of information, it need not mean that collaboration over it is worse than in-person interactions in which more information is available. It could even offer beneficial interactions that are not possible with in-person interactions such as the ability to perform digital annotations. To this end, prior

work identified the value of using Mixed Reality environments for collaboration over physical tasks [16] and has proposed extensions such as spatial annotations and tracked objects [3, 22, 16]. The remote user usually has access to a 2D video feed [108], and more recently, 360 video and 3D point cloud feeds [97, 179, 184], or a virtual replica [138] of the local user operating in their physical environment. In addition, they guide them through an audio channel with the aid of other peripherals such as laser pointers [1, 70] or augmented reality devices [87, 105].

Prior works have demonstrated the value of having tools and interaction techniques such as annotations [70, 141], sharing hand gestures [2, 88, 101] and trade-offs between different viewpoints [55, 166]. Even earlier works [90, 92] identified that for seamless remote collaboration, it is not sufficient to have only 2D annotations. However, it is also important to have access to physical and digital tools, awareness of gaze and gesture, and a way to manage the digital and physical workspaces. The Cross-Dimensional Media in Loki (Chapter 5) leverages these techniques and tools to facilitate remote instruction of physical tasks.

Mixed Reality interactions provide an interesting parallel to these scenarios, where the elements—MR user, the MR environment, the external user, and the MR user's video feed — are correspondingly analogous to a local user, their physical environment, remote user, and the 2D video feed they view. This analogy points us towards potentially useful interaction techniques introduced by these works, such as environment annotations, live sharing video streams and content, and enabling independent spatial exploration of the remote user's environment, which may be adapted to work in Mixed Reality environments. However, there are fundamental differences between real and virtual environments. Unlike the real world, the elements in a virtual world commonly have dynamic and digital behavior, e.g., objects and menus can spawn and disappear anywhere, and the scale of objects and the world can change. These differences break the interaction techniques and introduce interesting problems in Mixed Reality environments. In this dissertation, features and design of systems that focus on tasks in virtual spaces - TutoriVR (Chapter 6), TransceiVR (Chapter 7) and DreamStream (Chapter 8) are inspired by this prior work, but extend it to address unique Mixed Reality problems.

Interactive Cross-Dimensional Media builds on these prior works and findings wherein it supports these rich interactions, as well as provides us with a unified framework to broadly think about systems that involve users, each of whom perceives and interacts with the same data through different visual representations and interaction affordances. In this dissertation, we build systems that augment existing mixed reality systems. Specifically, these systems use data similar to prior work: depth maps, color video feeds, and pose tracked actions. We develop Interactive Cross-Dimensional Media that use these data to facilitate collaboration in Mixed Reality environments. We propose a novel design methodology (Section 4.4) that allows for designing and understanding user interactions that meaningfully translate across the different representations of Mixed Reality environments.

3.3 Systems for Mixed Reality Tasks

Tutorial systems

Early research in MR explores the potential of immersive displays in spatial tasks such as modeling [157] and sketching [191]. More recent work focuses on examining the mechanics of sketching in mid-air, focusing on understanding the quality of the stroke [6, 194] or improving it via correction techniques [9] and advanced workflows [7]. Providing effective learning materials is also vital to help develop such skills. Works in this dissertation complement this research by characterizing users' needs in learning and working with mixed reality applications by introducing systems that leverage Interactive Cross-Dimensional Media for facilitating tutorials, collaboration, and richer spectating for tasks in Mixed Reality.

Tutorial systems designed for such mixed reality environments primarily focus on harnessing spatial tracking and feedback to help users learn motor tasks [198, 193, 165, 3, 123]. Video is a lightweight format for capturing these workflows. However, motor skill is only one aspect of larger tasks. Therefore, systems in this dissertation use motor activity videos and supplement them with additional information regarding the 3D virtual scene, tool usage, task demonstration, and content progression [50].

Commercial MR applications like BeatSaber, Gravity Sketch, Medium, and Quill provide dedicated 2D video tutorials online, and MR uses standard video widgets such as play, pause, and seek. Alternatively, a MR application may provide APIs for users to capture all the geometric information and the application states of a design workflow [137]. A 3D reconstruction of this capture can help users freely explore the workflow from different perspectives. Ponto et al. [148] use tracked HMD data to produce stable videos while preserving the recorder's original intent. Greenhalgh et al.'s [63] work discusses different Spatio-temporal mechanisms that allow recording and playback of multi-user VR sessions. These works assume source-code access to the entire application to facilitate the playback of tasks in Mixed Reality spaces. TutoriVR, a tutorial system for Mixed Reality tasks introduced in this dissertation (Chapter 6), also includes a video player. However, it is more tightly coupled and is responsive to the user's workflow. It can display video interfaces and enable interactive tutorial features directly inside the Mixed Reality environment in an application-agnostic manner and retrofits over existing applications without source-code access.

Multi-user Mixed Reality interfaces

Research on collaboration in digital 3D virtual environments goes back many years [47, 32]. One of the earliest proposals for a multi-user VR system is DIVE [26], in which users are part of a shared 3D virtual environment in which they can share the screens of their 2D desktop applications. *Populated Web* [13] proposes a multi-user 3D web browsing experience in which web pages are laid out in a shared 3D virtual space. Using a 2D desktop interface or VR headset, users can spatially navigate web content and represent themselves as avatars, interacting with each other through text, voice, and video. Though the work focuses on web

browsing, the system's core interactions are strikingly similar to those of today's social 3D virtual environments such as Horizon Workrooms, Mozilla Hubs, and VRChat.

Immersive three-dimensional environments in which users are embodied supposedly allow them to perform tasks that are inherently 3D in nature more easily. Examples of such tasks include 3D modeling and design, planning surgeries, performing architectural layouts, and watching immersive media content such as movies and games. Developing Mixed Reality systems requires understanding the unique goals and challenges posed by the specific task and its target users. The recent emergence of consumer-grade VR devices allows researchers to understand better and design collaborative VR systems for these tasks. Xia et. al [197] notes challenges in collaborative scene editing through VR interface. Their SpaceTime system uses interaction techniques to allow for parallel manipulation and conflict resolution of 3D designs. Nguyen et. al's CollaVR [131] system allows for efficient synchronous collaborative review of 360-degree media content. Prior works also highlight other essential factors to facilitate collaboration in virtual environments. These include enabling co-presence in virtual environments, allowing independent exploration in 3D space, and access to relevant 3D information across various interfaces [12, 11, 71, 189]. These works validate the need for collaboration in Mixed Reality systems and promise increased productivity and social engagement. In addition, they demonstrate how effective collaboration can be supported if it is designed into the core of each application.

In the context of Mixed Reality applications, very few works have explored retrofitting approaches discussed earlier for conventional computer softwares. Recent works by Zhao et. al [201] and Hartmann et. al [79] do this, highlighting the value of retrofitting for application-independent compositing of information into Mixed Reality environments. Works in this dissertation, specifically TutoriVR (Chapter 6), TransceiVR (Chapter 7) and DreamStream (Chapter 8) achieve multi-user interactions in existing VR applications through providing collaborative interfaces that retrofit at the platform level outlined in Section 2.2.

Systems in this dissertation adopt such interactions from these prior work. However, they show how such functionality can be retrofitted and may also be possible without requiring every user to use Mixed Reality interfaces, i.e., they can work in asymmetric interaction scenarios.

Asymmetric Collaboration for Mixed Reality tasks

Some applications employ asymmetric interactions, where one player operates using Mixed Reality interfaces and one or more other users operate without it but may have other accessories such as cheat sheets or controllers, with which they can indicate and perform actions in the application environment. Tasks carried out in these applications may be cooperative, in which the users work together, e.g. *Black Hat Cooperative* [15], *Eye in the Sky* [48] or competitive, where they work against one another e.g. *Panoptic* [142], and *Nemesis perspective* [129].

Beyond these applications, primarily games, recent research [54, 69, 68, 195] has identified the isolating nature of existing MR applications and proposed techniques to address

this problem. ShareVR [69] uses spatial augmented reality (SAR) projections [14], mobile displays, and tracked spaces to display the virtual world onto the physical world and vice-versa. FaceDisplay [68] has three touch displays that are directly mounted on the VR headset, which act as viewports and a means for the external user to interact with the VR user and play a VR game. TeleSight [54] uses an instrumented robotic head that mimics the VR user's HMD pose to achieve a similar interaction. Sometimes these systems require an instrumented space with a rig of depth cameras and projectors. Von Willich et al.'s work renders different representations of a passerby into a custom VR app and compares them [195]. RealityCheck [79] composites 3D renderings of the user's physical surroundings into a MR application and projects the VR environment into physical space in an application-agnostic manner. In general, these works try to reduce the asymmetry by mapping the virtual environment onto a real environment and vice versa. DreamStream (Chapter 8) builds on this approach by using depth camera captures and having spectators use MR interfaces while also having a 2D UI interface as an alternative. In contrast, TransceiVR (Chapter 7) starts from the perspective of enabling external non-MR users to communicate with MR users without requiring additional tracking or projection infrastructure.

These works also provide interesting insights into asymmetric MR interactions but differ from the scope of works in this dissertation in that they assume these interactions are predominantly spatial. Additionally, they focus on custom-written applications where collaboration is key to the user experience. Systems in this dissertation focus on improving the collaborative interactions and spectating in existing MR applications that were not necessarily designed for multi-user experience and interaction.

Spectating Mixed Reality Activity

Spectating live video game streams is an increasingly popular online activity. Typically, a live screen capture of the host player's gameplay is encoded and streamed at scale to multiple spectators. Through real-time chat and other mechanisms, spectators can influence how players go about their games [163, 167], and host players must consider how their in-game activities impact the spectator experience. This is not easy because spectators themselves may have different goals. Cheung and Huang [29] identify nine different spectator personas involving a varying degree of interaction with the host player. We may consider spectating as a form of collaboration, where the goal is entertainment for all, and as such can face many of the same challenges.

Providing an excellent spectating experience for Mixed Reality activity is even more challenging for several reasons. Foremost is that while the player operates in an immersive environment, spectators view a 2D video stream. The 2D video is limited in conveying the player's scale and richness of the 3D experience. Secondly, the choice of the viewport of the video can dramatically impact VR streams. Often VR streamers share their first-person view (FPV), but this can be shaky and difficult to watch. With a third-person view (TPV), spectators may miss some critical parts of the action. This problem is related to the challenges in watching FPV videos of users doing physical tasks [55]. JackIn [97] stitches

frames from the FPV video stream and maps them to spatially registered frames immersively surrounding a spectator, thereby decoupling the head motions of the spectator from that of the camera. In contrast to approaches that mitigate the effects of FPV videos, an online survey by Emmerich et. al [45] suggests that most viewers prefer the first-person version when spectating VR gameplay as 2D video feeds.

Today, Liv and MixCast are popular general-purpose solutions for streaming VR activity. They offer stabilized first-person views, increased field of view, and third-person views (TPVs). Liv and MixCast also allow compositing 2D video or an abstract avatar of the player placed between the set background and foreground layers of a VR feed. The output of these systems is ultimately a 2D video, in which the viewer experience is very different from that of the player. These systems do not allow for spatial interaction between the player and viewers, nor do viewers have independent control of their viewport. Some features are built into certain games to facilitate enhanced spectator experience, e.g., JobSimulator. To increase spectator immersion in VR, vReal allowed spectators to be part of the game's virtual space. All these tools require developer support. From our interviews with expert streamers (Section 8.5), we learned that while vReal was thought to be promising, it had failed to gain traction with developers and was shut down.

Conventional 2D *streaming approaches* (Section 2.3) can work with any existing MR application. However, supporting interactions typical of multi-user VR applications will likely require deep integration with custom SDKs during development, particularly as they manipulate elements of the application's 3D scene. In contrast, systems in this dissertation performs interventions at the *Platform level* (Section 2.2).

Capturing and transmitting virtual spaces' 3D geometry and texture information is more complex and less standardized. RealityCheck allows co-located viewers to spectate the VR experience mapped onto a physical environment. It also composites a Kinect capture of the surrounding physical world into the VR scene. To do this, RealityCheck intercepts the graphical rendering pipeline of the VR application. DreamStream system in this dissertation (Chapter 8) builds directly on this approach, modifying it to suit the needs of interactive and immersive spectating.

Niederauer et. al [133] proposes a non-invasive technique to access and reconstruct the 3D geometry of a scene rendered in OpenGL. This is an inspiring idea, but their technique does not work well with today's VR applications. Firstly, it assumes that geometry culling is turned off, which allows it to capture the entire architectural model of the surrounding scene. However, VR applications today cull parts of the scene outside the player's field of view. Secondly, it makes assumptions on how geometry information is stored and requires user inputs to generate the scene geometry correctly. DreamStream system (Chapter 8) in this dissertation achieves such 3D reconstruction by intercepting the graphic rendering pipeline itself and retrieving the depth buffer that was used. DreamStream does not fail entirely, even if culling is performed. This is because DreamStream does not capture the entire scene at a single moment but continuously captures the parts of the scene in view. For the same reason, DreamStream does assume the static nature of the scene nor about the structure of the geometry buffer.

In summary, DreamStream facilitates multi-user interactions typical of *multiplayer approach* described in the previous subsection. In addition, it leverages the scalable content distribution pipelines typical of today's video *streaming approaches*. Furthermore, since this is done at a platform level, it works with many of today's existing MR applications.

Chapter 4

Rethinking Mixed Reality Collaboration with *xspaces* and Interactive Cross-Dimensional Media

In this chapter, we will revisit the canonical CSCW matrix discussed in section 2.1 and see its limitations in categorizing collaborative interactions in Mixed Reality. We will then propose an extension to the matrix by introducing the notion of an *Extended Space*. Users will have barriers to efficient collaboration when they are *asymmetric* in the way they see and interact

Term	**Definition**
Extended Collaborative Space (*xspace*)	The Mixed Reality space perceived by a user, containing all objects, both physical and virtual, that are relevant to the collaboration task at hand.
Interactive Cross-Dimensional Medium	A medium that can offer different visual representations and associated interactions across and within display types.
Display Type	Output hardware a user employs to view a Mixed Reality environment. E.g., VR headset, AR headset, 2D screen
Visual Representation	The way in which a data stream from a physical or virtual environment is captured and presented. E.g., 3D geometry, 3D point cloud, Stereo video, 2D video, 2D image
Output Dimensionality	The number of dimensions inherent in a given spatial representation. E.g., 3D for point clouds, 2D for video.
Input Dimensionality	The degrees of freedom (DOF) available for providing input. E.g., 2D on a touch screen vs. 3D using a VR controller

Table 4.1: Key terms and definitions used throughout this paper

with the *Extended Space*. Subsequently, in this chapter, we will introduce Interactive Cross-Dimensional media, which has properties and interactions that help mitigate these barriers. To serve as a quick reference, the different terms and definitions helpful for this chapter are summarized in Table 4.1.

4.1 The Canonical CSCW Matrix Cannot Describe All Mixed Reality Collaborations

The Canonical CSCW matrix uses the *physical space* for categorizing different user interactions. This leaves out critical details that are required for carrying out collaborative interactions in Mixed Reality environments. This is because, in these environments, information may be anchored to 3D virtual or physical spaces accessible only to specific users, even if they are physically co-located. A key aspect of Mixed Reality systems is that they often blur the distinction between physical and digital spaces used for work. Through studying prior works and existing Mixed Reality interactions, we realized that the canonical CSCW matrix is insufficient to capture the nuances of the interactions between the different collaborating users in Mixed Reality environments. To illustrate this, we list a few counterexamples (CEs) where categorizing interactions into the canonical CSCW matrix is not particularly effective and can be deceiving.

CE1 *Co-located spectator viewing mirror video feed of a VR user*: The two users are still co-located and synchronous. However, the VR user cannot see the external world. While the spectator, as well as the VR user, can see the virtual world, they do not have the same *Output and Input Dimensionality* (Defined in Table 4.1). i.e., the spectator can only see the first-person view feed of the VR user and can only do so through a 2D video. Furthermore, they have no interaction with the video. This is illustrated in Figure 4.1 (1).

CE2 *Co-located spectator of an AR user*: Though the interaction between the two users can be considered co-located and synchronous, it is important to note that the spectator can not see the virtual elements seen by the AR user. This is illustrated in Figure 4.1 (2).

CE3 *Co-located VR users using different VR apps*: Here, the collaborating users are co-located and synchronous. However they operate in different virtual worlds and sometimes may even be completely unaware of the context of actions of another user.

CE4 *Social VR apps*: Perhaps the most common of scenarios is the realm of social VR apps. Here, the collaborating users might be in remotely located physical spaces but are in the same co-located virtual space. Moreover, they may all see the virtual world in 3D or participate through a 2D web interface. This is illustrated in Figure 4.1 (3).

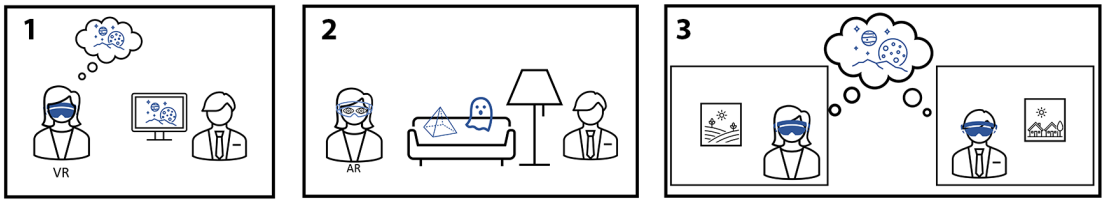


Figure 4.1: Three example scenarios that have no meaningful place in the canonical CSCW matrix - (1) A user is watching a video mirror of a co-located VR user. (2) an AR user and another user who does not wear a headset and (3) two remotely located VR users who are part of the same virtual scene.

CE5 *Mixed Reality Telepresence*: Collaborative interactions in Mixed Reality environments are a well-studied topic in the literature. Here, a multitude of interactions may not fit the conventional CSCW matrix. For instance, with Loki (Chapter 5), in the case of a remote guidance task carried out by two users, A help-seeker user (learner) wears an AR headset, whereas a remote expert wears a VR headset. Though they are located remotely, both view the same physical environment of the help-seeker in 3D. While the help-seeker views it directly through the lenses of the AR headset, the remote expert views a live 3D point cloud reconstruction of the same.

Prior work [78, 79, 69, 68, 185, 186, 184, 27, 189, 107] as well as all the systems in this dissertation have built multi-user systems and interactions that fit one or more the above-listed examples, allow for collaboration and communication between users in Mixed Reality environments. However, these systems do not have a meaningful place in the canonical CSCW matrix.

Adapting the CSCW Matrix for Mixed Reality environments

Can the CSCW matrix be adapted to categorize better and analyze Mixed Reality environments? A naive solution would be adding more dimensions to the CSCW matrix. For instance, *Virtual space* can be added as an additional dimension. However, a significant issue exists that, with such a 3D matrix representation, the common collaboration scenarios would often straddle across multiple Octants. Furthermore, the nature of these scenarios also varies based on the differences in the *visual representation* and modality with which users perceive these spaces, as well as differences in their *Interaction affordances*. So a full accounting may lead to a five-dimensional matrix, which is unwieldy to think about, and reason using as a tool. To avoid such complex formulations, we propose a simpler modification to the matrix, making it better suited to deal with interactions that occur in Mixed Reality spaces (see Figure 4.3, Right). We conflate these different parameters to carve out a hypothetical space that we term as *extended collaborative space* (abbreviated as *xspace*). In

the subsequent section, we describe this and show that simply replacing the *Physical Space* axis with *xspace* makes it easier to categorize and analyze collaborative interactions in Mixed Reality environments.

4.2 Extended Collaborative Space (*xspace*)

We define *xspace* as the Mixed Reality space in the Milgram continuum that is perceived by a user, containing associated information of all objects, both physical and virtual, that are relevant to the collaboration task. It is important to note that the *xspace* varies with the collaborative task. For example, in an Augmented Reality space where the virtual elements can interact with the physical world (e.g., a virtual pet jumping on a couch), all the virtual elements, as well as the physical elements that it interacts with, such as the couch, constitute its *xspace*. On the other hand, in a VR activity where the user works only with virtual objects, the *xspace* consists of the virtual environment only. In the same VR activity, however, if the VR user has to interact with some elements of their physical surroundings too (e.g. FaceDisplay [68], Mutual Human Actuation [27]), then those elements are also part of the VR user's *xspace*.

Visual representations of *xspaces*

We define *visual representation* as how a data stream from a physical or virtual environment is captured and presented, e.g., 3D geometry, 3D point cloud, Stereo video, 2D video, and 2D image. Some of these representations are illustrated in Figure 4.2. The collaborating users may each experience and interact with the different components of the *xspace* through different modalities that act as a lens to the *xspace*. These modalities may facilitate different *visual representations* that may have different *output dimensionalities* (Table 4.1) associated with them. For instance, in a Mixed Reality telepresence system, one may see another user's *xspace*, directly through their own eyes. However, if the other user is located in a physically remote place, then they may see the *xspace* through a live 2D video (Figure 4.2e), a live 3D reconstruction (Figure 4.2b). They may also see it through a combination of both, such as in Hologlyphs in Loki (Chapter 5). In contrast, the local user may see the same *xspace* by viewing the physical world directly through their eyes (or video see-through) and possibly any other virtual objects on top of them rendered through an AR device. These *modalities* for perceiving *xspaces* can be broad in terms of other sensory characterizations such as haptic, auditory, and visual. However, in this dissertation, we will focus only on the *Visual modality* which is the dominant one in current Mixed Reality environments. Specifically, we explore the different *visual representations* of the visual modality, e.g., reconstructed 3D point clouds, 3D interactions, stereo video, and regular 2D video.

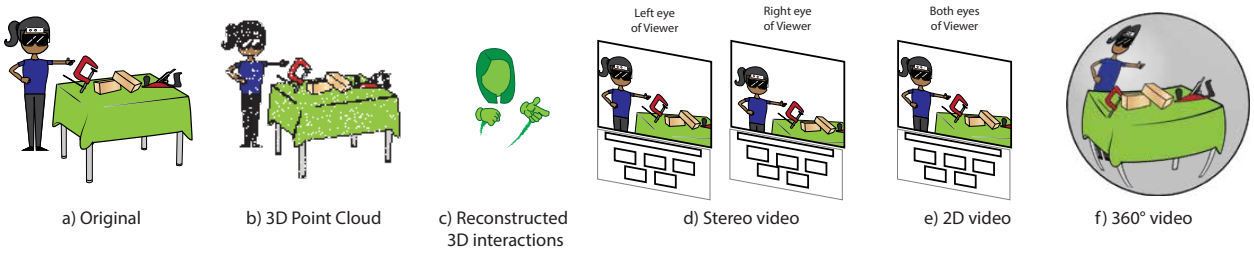


Figure 4.2: Examples of a few possible visual representations of *xspaces* that are used in prior work

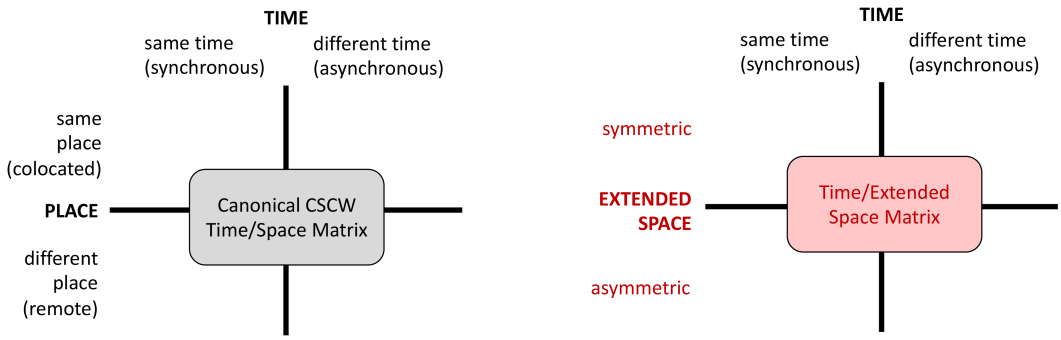


Figure 4.3: Canonical CSCW Matrix (left) and our matrix for extended collaborative spaces in Mixed Reality environments (right).

Extended Space - Time Matrix

In the canonical CSCW matrix, we classified the users on the *Physical Space* axis as either co-located or remote. In the Extended Space-Time matrix, we will classify them based on how they see and interact with the Extended Space. This can be symmetric or asymmetric (Figure 4.3). However, what makes users working in an *xspace* be asymmetric?

Asymmetry in *xspaces* occurs because one user can not see and/or interact with certain parts of a Mixed Reality environment in the same manner as another user. Sometimes, a user has no access to these parts. For instance, in CE2 above, the spectator cannot access the virtual elements. However, in many other cases, the *visual representation* (Defined in Table 4.1) in which users view the spaces are different. For example, this is seen in CE1, where the external artist sees the virtual environment through a 2D video feed, whereas the VR user sees and operates in the same environment in 3D.

In such scenarios, some users can carry out 3D operations in a Mixed Reality environment, while others carry out 2D UI-based operations on a screen. Some can view the information in

System & Purpose	Access to Physical Space		Access to Virtual Space		Visual Fidelity		Interaction Affordance	
	Without System	With System	Without System	With System	Without System	With System	Without System	With System
TutoriVR Asynchronous Instruction by one VR instructor to another VR learner	NA	NA	Asymmetric. VR tutor has full access to their virtual space. The VR learner sees only what gets recorded in the video.	Mostly same as without system. But the VR learner can also see all the controller interactions that may not be present in the video frame.	VR tutor sees their environment in VR and perceives 3D. The VR learner sees only a 2D video.	Better than without system. The VR learner can see the video as well as controller interactions in 3D.	Asymmetric. The VR learner has no affordance beyond navigating the video.	Better than without system. The VR learner can take different perspective and access additional information about controller interactions
RealityCheck Collaboration between a co-located VR user and non-VR user for a task in the physical world	Asymmetric. VR user cannot see the physical space	VR user can see the physical space in 3D	NA	NA	Asymmetric - Because the VR user does not see the physical space.	Better than Asymmetric - The VR user can now see the real world in 3D through 3D point clouds.	Asymmetric. The VR user cannot interact with the physical world because they cannot see it.	Symmetric. The VR user and the non-VR user can interact with the physical world in the same capacity
TransceiVR Collaboration between a VR user and non-VR user for a VR task	NA	NA	Asymmetric. External user only sees what the VR user sees	Better than without system. External user can see along directions beyond what the VR user sees.	Asymmetric	Asymmetric	Asymmetric. The external user has no affordance.	Better than without system. External users can now spawn direction arrows, annotations, shared screens and trigger controller highlights in the scene.
DreamStream Synchronous spectating of a VR player by another spectator	NA	NA	Asymmetric. Spectator only sees what the VR user sees	Better than without system. External user can independently explore the VR scene	Asymmetric	Better than Asymmetric - The spectator can now see the VR Scene in 3D through 3D point clouds.	Asymmetric. The external user has no affordance.	Better than without system. External users can now move around the scene, and point at objects using laser pointers
SpaceTime Collaboration between two VR users for 3D modeling	NA	NA	Symmetric	Symmetric	Symmetric	Symmetric	Asymmetric. Only one user can control and manipulate a virtual object at a time.	Symmetric. Both users can work with same set of the virtual objects, at the same time, in a similar manner.

Figure 4.4: Sources of *xspace* asymmetry addressed by systems in prior works and in this dissertation.

3D, while others only through 2D. Thus, in these asymmetric *xspaces*, there is an uneasy co-existence of different combinations of 2D and 3D *input and output dimensionalities* (Defined in Table 4.1) for the collaborating users.

In summary, it is important to note that asymmetry in *xspaces* is a multi-faceted concept with multiple factors. Users can belong to asymmetrical *xspaces* due to any of four possible sources:

- Users have unequal access to task-relevant components of the physical parts of the *xspace*
- Users have unequal access to task-relevant components of the virtual parts of the *xspace*
- Difference in output dimensionality with which different components of the *xspace* are perceived by the users
- Difference in Interaction affordance / input dimensionality of different components of the *xspace* for the users.

Empowered by this notion of *xspace*, we can now categorize prior works based on the baseline Mixed Reality scenarios in which they try to address collaboration issues. This is shown in Figure 4.5. Typically these prior works try and bridge the asymmetry across the *xspaces* that users operate in. A few selected ones are tabulated in Figure 4.4.

We thus replace the notion of a co-located or remote physical space with that of an asymmetric or symmetric extended collaborative space (*xspace*). Analogous to the canonical

matrix, if we identify that users interact with *xspace* asymmetrically, then there should exist communication barriers between them arising from one or more of the sources listed above that lead to the asymmetry. The specific sources and collaboration goals inform the type of tools and systems needed to bridge that gap.

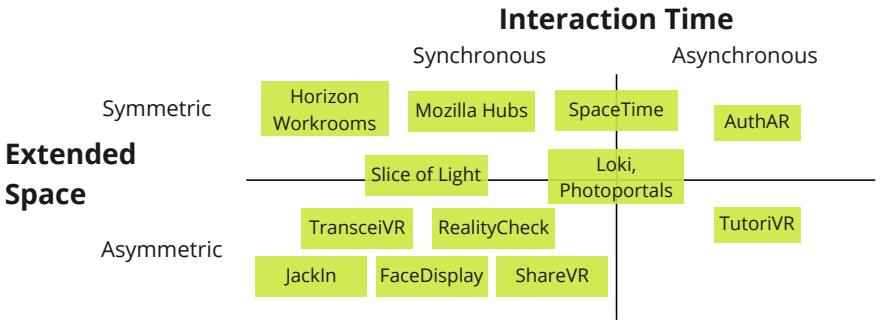


Figure 4.5: Baseline scenarios that systems in prior work and this dissertation seek to address collaboration issues.

4.3 Interactive Cross-Dimensional Media (XD media)

To help bridge the asymmetry between users who operate across asymmetrical *xspaces*, in this dissertation, we introduce and propose *Interactive Cross-Dimensional Media* (abbreviated as *XD media* and pronounced as Cross-D media). We formally define it as:

"*A medium that has components of varying visual dimensional representation. Different representations can be chosen automatically based on context or through associated interaction techniques that give users control over exploring spatial, temporal, and dimensional levels of detail. It ensures that any information or interaction makes sense across different dimensions, interfaces, and spaces.*"

The definition associates two key properties to a cross-dimensional medium:

P1: Variable dimensional representations

P2: Meaningful translation of interactions

Property 1: Variable dimensional representations

The first property of a cross-dimensional medium is that it should have *visual representations* of different *Output Dimensionality* for the task-relevant data of an *xspace*. Figure 4.2 shows a subset of such *visual representations* that are possible. The medium's specific representations may vary based on the context, devices, and interfaces that the different users use. This

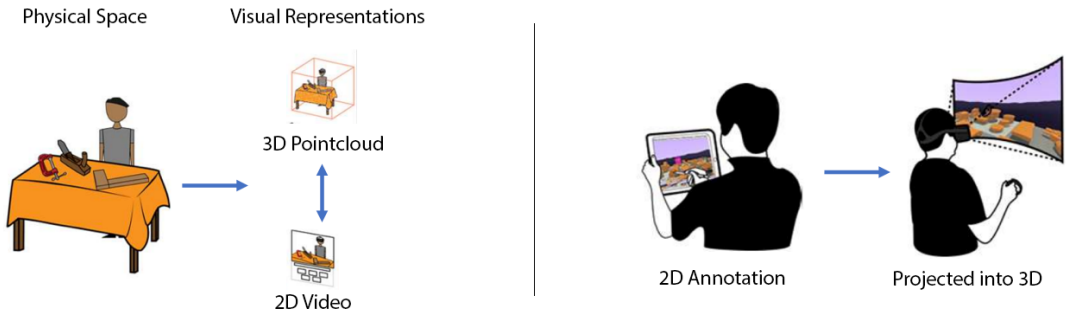


Figure 4.6: Illustration of the Two properties of XD Media - Left: Property 1 - XD Media has different visual representations of the same data. Right: Property 2 - XD media Meaningfully translates interactions across representations

would allow for presenting an *xspace* in *visual representations* that are best suited for the interface and device each user operates with. As mentioned before, the visual representation can be a 2D video, a 3D point cloud, an abstract 3D graphic, a stereo video, 360 video, etc. Some of these are illustrated in Figure 4.2. It may also be presented as a collage of the above forms, providing users with alternative information sources. The works in this dissertation have used different combinations of such representations. For instance, the Loki system that we will see in Chapter 5, uses a combination of live 2D videos, 3D point clouds, and abstract 3D avatars to present instructional information for the instruction of physical tasks (Figure 5.4).

Property 2: Meaningful translation of interactions

The second property of a XD medium is that it should meaningfully translate interactions across the different representations it supports. It should allow changing between 2D and 3D while providing input to an *xspace*. For example, in the TransceiVR system that we will see in Chapter 7, annotations made over a 2D video feed can either be projected to the 3D space (Figure 4.6 Right and Figure 7.3) in a meaningful manner or can be shared as such in a 2D format if required. Depending on the context, either one of the interactions might be helpful. While 3D projected annotations can be of use for referring to objects in the VR environment, sometimes annotations projected directly in the 3D environment may not make sense. For instance, in Figure 7.8, the annotations made over a video frame are used for shared discussions. This is analogous to the interactions that happen with desktop screen-sharing tools today.

Figure 4.7: Analogous data captured in virtual and physical worlds. (L) In VR, VR scene image, corresponding depth texture, pose of VR HMD (camera), and the VR user's input pose and actions are captured. (R) In physical world, besides the scene image, depth data, tracked pose of the depth camera as well as their actions (through skeletal tracking) are captured

Common XD Media Interaction Techniques

These two properties give rise to three specific interactions we have used across the research systems in this dissertation.

1. **Perspective exploration:** Allows a user to get different perspectives of the *xspace*.
2. **Temporal exploration:** Allows a user to navigate across and interact with the historical trace of the *xspace*.
3. **Dimensional-detail exploration:** Allows a user to explore and enhance different dimensions of the underlying *xspace*. This is similar to the level-of-detail (LOD) effects in computer graphics [128], in which one provides different representations of a single artifact, each with a different level of complexity. In this work, we use a similar analogy to the context of the dimensional representation.

Data Requirements

While the physical space and time aspects of collaborative interactions have been studied in the CSCW literature, interactions between users spanning different *xspaces*, using different modalities, have been studied to a lesser extent. Our work demonstrates that systems that deploy Interactive Cross-Dimensional Media techniques allow for effective collaboration between users of Mixed Reality environments. These users might span multiple sectors of the new Extended Space-Time matrix, i.e., across different *xspaces* and synchronicity. Our

proposed Interactive Cross-Dimensional Media techniques allow for efficient communication of information and collaboration across these sectors.

To achieve this, in this dissertation's research systems, we will make assumptions about what kind of data is available about an *xspace* to characterize it. Interactive Cross-Dimensional media that we have used in our works involve capturing four specific types of data: (1) RGB video frames, (2) Corresponding depth data, (3) Tracked Camera, and (4) Position tracked input actions by the user in the *xspace*. These are illustrated in Figure 4.7.

4.4 Using *xspaces* and XD Media: A proposed design methodology

Broadly, we will use the following higher-level methodology for designing collaborative interactions for Mixed Reality environments. First, we seek to identify the fundamental factors that cause issues in collaboration, and we do that through a series of steps that leverage our extended CSCW matrix.

Step 1: Identify the extended space (*xspace*) for the collaboration task. The *xspace* is defined as the Mixed Reality space perceived by a user, containing those physical and virtual objects relevant to the collaboration task at hand. Sometimes, it may seem that a task may seem to have more than one *xspace*. However, with our experience using this methodology, such a task can usually be broken down into smaller sub-tasks that each have a single *xspace*.

Step 2: Identify how each user perceives and interacts with the *xspace* with the interface they use in the baseline scenario. Then, use that to determine which quadrant of the *xspace*-time matrix they fall into.

Step 3: If they fall into either of the asymmetric quadrants, identify the potential sources of asymmetry as one or more of the four factors defined above (i.e., (i) Access to task-relevant physical space, (ii) Access to task-relevant virtual space, (iii) Difference in *output dimensionality* with which the *xspace* is perceived by the users, (iv) Difference in Interaction affordance / *input dimensionality* for the users.)

Step 4: For the specific task, gauge the impact of each asymmetry on collaboration efficiency between the users.

Once these asymmetries are identified and prioritized, the next step is to design and develop interfaces to mitigate the critical asymmetries of concern. This can be any interface, but in our work, we propose Interactive Cross-Dimensional media as a candidate for solving collaboration issues due to such asymmetry in *xspaces*. These media have a set of standard interaction techniques and patterns that facilitate perspective, temporal and dimensional-detail exploration. Moreover, these techniques are well suited to broadly mitigate the various asymmetries we identified through our extended space CSCW matrix.

In the following chapters, we will see four different systems that leverage the *xspace*-time matrix framework to analyze collaboration issues when using different Mixed Reality environments. We show that the use of *Interactive Cross-Dimensional Media* mitigates the adverse effects of asymmetry and aids collaboration and communication in settings where complete symmetry is impossible or not desirable.

Chapter 5

Loki

5.1 Preamble

Remotely instructing and guiding users in physical tasks has offered promise across various domains. While it has been the subject of many research projects, current approaches are often limited in the communication bandwidth (lacking context, spatial information) or interactivity (unidirectional, asynchronous) between the instructor and the learner. Systems that use Mixed Reality for this purpose are promising. However, they typically do not support all the different approaches to learning that are required for the Instruction of Physical tasks (Discussed in section 2.4). Most prior systems focus on only one of the approaches.

In this chapter, we will develop systems to facilitate instruction for physical tasks by leveraging the framework and methodology outlined in Section 4.3. However, we will see that the different instructional approaches will require users to operate in different *xspaces* and use different interfaces. To better understand this, in this chapter, we identify a design space specific to using Mixed Reality systems for the remote instruction of physical tasks.

We will then present Loki, a novel system that allows users to operate flexibly in various regions of this identified design space. To do this, Loki leverages an Interactive Cross-Dimensional Medium that consists of videos, 3D spatial captures, and abstract 3D graphics, along with its associated interaction techniques that will allow users to explore and annotate the local and remote environments and record and review their own performance as well as their peers'. Furthermore, Loki is designed to enable quick and easy transitions between the different configurations of the identified design space and reduces the *asymmetry* between the users. We validate its utility through various scenarios and a qualitative user study.

5.2 Introduction

The ability to remotely guide and instruct users in physical tasks has excellent value since it can connect novices with instructors and improve how people learn new skills and trades [61,

81]. In addition, the ability to re-skill and develop workers effectively is vital as the nature of work changes and workforces become more dynamic [85]. Within the HCI community, researchers have leveraged novel interfaces such as AR, VR, and other modalities to guide or teach physical skills and activities [3, 80, 165, 177, 187]. This research has driven commercial offerings which aim to direct users and provide guidance on job sites and during maintenance tasks [192, 120, 52]. However, these approaches typically rely on asynchronous learning, tutorial generation, or presenting contextually relevant information such as guidance cues.

Additionally, these approaches often rely on a single modality of capture and visual representation (e.g., 2D video, 3D mesh) to teach or guide the remote participant. While these methods can be effective, the spatial nature of physical tasks is often lost or reduced, as is the ability to interact with an instructor in a bi-directional manner. Rich interaction with an instructor can result in tailored guidance and close the loop by supporting demonstrations by the instructor [34, 104, 162, 196].

In recent years, telepresence technology has advanced rapidly with commoditization of real-time spatial capture devices [82], more prevalent availability of VR and AR interfaces, and novel interactions for Mixed Reality (MR) interfaces [105, 140]. These technologies have the potential to augment current training techniques and bridge the gap between instructor and learner by leveraging contextual cues and spatial information, allowing recording and playback of scenes, and enabling spatial annotations. However, it is not evident how to leverage these novel technologies in combination to exploit their unique value.

While prior work has introduced specific configurations of MR-based instruction, we find that there exists a broader design space for Mixed Reality systems aimed at facilitating remote instruction of physical tasks. Different configurations in this design space will allow for instruction approaches identified by the learning science literature (Section 2.4). Furthermore, each of these configurations may have an *xspace* that is different from the other, thereby introducing different levels of *asymmetry* between users.

To flexibly support the different stages of instruction, we highlight that it is essential for a single system to allow for moving between these different configurations. To support this, we develop and introduce Loki (Figure 5.1), a system for physical task training that supports operations across the different dimensions of the identified design space. In this work, we will refer to them as 'transitions'. Loki supports these transitions between the various modalities and data enabled by Mixed Reality. These transitions are essential to facilitate learning throughout the skill acquisition process since the learner's needs can change even within a single session of learning a physical skill.

Loki is comprised of two symmetric spaces. Each space supports a single user and contains an immersive Mixed Reality display utilizing pass-through AR to enable transitions between virtual and augmented reality. The physical environment of each user is spatially captured and streamed in real-time to the remote user alongside video, audio, and annotation data. Both users can navigate between their local and remote environments in real-time and interact synchronously with live and recorded data. This flexibility allows for novel workflows that bring the instructor and learner closer together, allowing for richer collaboration and improved training opportunities. We discuss and illustrate the value of Loki's mode

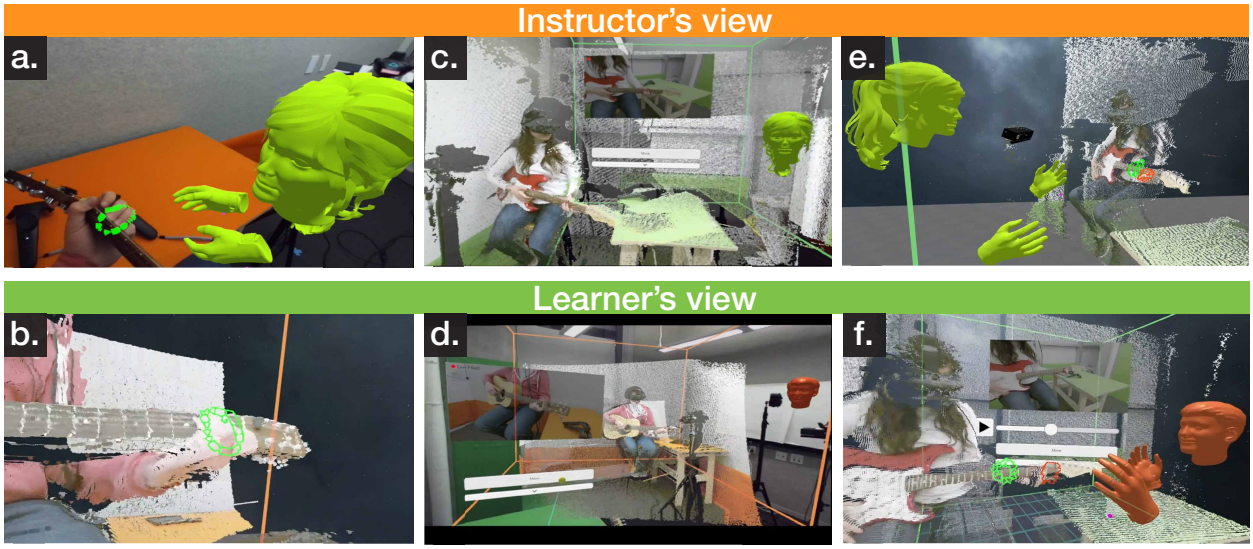


Figure 5.1: An instructor (orange) teaching a learner (green) how to play a chord on a guitar in Mixed Reality using Loki. The learner, who is in VR (d), observes the instructor who is in AR (a) demonstrating the chord. Next, the learner uses spatial annotations to ask a question about the performance. Then, both enter AR, and the learner begins to practice while the instructor provides occasional coaching (b, e). Lastly, the learner's performance is recorded, and both instructor and learner review the recorded performance in VR and discuss the errors (c, f).

transitions, and the corresponding system features through scenarios performed using the working Loki system. The scenarios illustrate that these additional affordances can allow users to learn a variety of physical tasks in a flexible manner. We also discuss a qualitative user evaluation of Loki in which users learned a 3D foam carving task remotely.

In summary, the primary contributions of this chapter are:

- A design space that explores real-time bi-directional Mixed Reality based remote training of physical tasks.
- An Interactive Cross-Dimensional Medium and associated interaction techniques allow users to navigate and utilize the breadth of visual representations and presentation modalities within this space and enable effective learning workflows.

5.3 Identifying the *xspace(s)* for the Instruction task

As discussed in Section 4.3, to be able to design collaborative interactions for Mixed Reality environments, we will begin with the first step of identifying the *xspace* for the task.

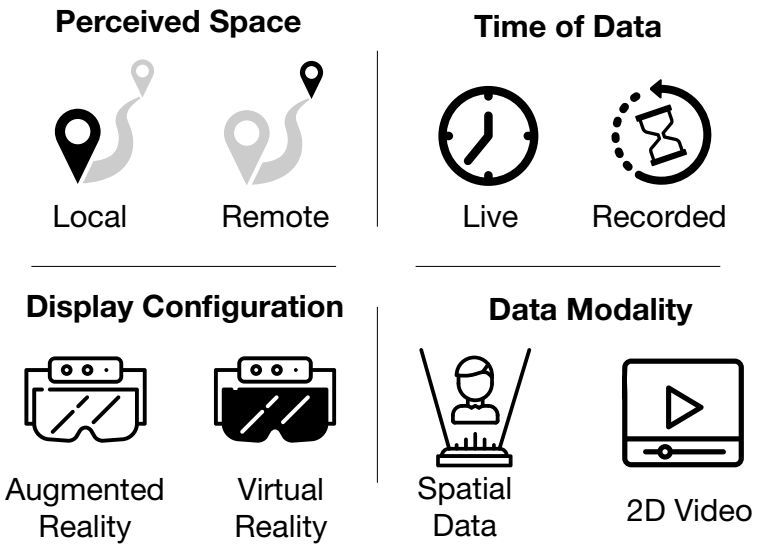


Figure 5.2: The dimensions of the design space for remote teaching of physical tasks.

While the broader collaborative task here is the instruction of physical tasks, we know from our earlier discussion in Section 2.4 that process of instruction for physical tasks can take different approaches depending upon the desired target learning outcome. Moreover, when the learner and the instructor are remotely located, each of these different approaches may focus on a different space, resulting in a different *xspace*.

For instance, when the instructor adopts the *Modelling* approach, they will demonstrate to the learner by performing the physical task themselves in their own space and have the learner observe them. Here the *xspace* is the instructor's body, the instructor's physical space, and any virtual elements that may be anchored to the instructor's physical space (such as annotations). In this case, while the instructor can see the physical parts of their *xspace* directly or using pass-through AR, there is no way for the learner to do this since they are remote. Therefore, this asymmetry has to be addressed by providing the learner with some appropriate form of *visual representation* that could adequately communicate the three-dimensional nature of the physical task.

In a *Coaching* approach, these are swapped wherein an instructor will watch the learner perform the task and provide hints and feedback. Here the *xspace* will constitute the learner's body, learner's physical space, and the virtual elements attached to it.

In a *Reflection/Review* approach, where an instructor/learner may want to view a playback of either of their tasks, it is not possible to view directly or through passthrough AR since the relevant data they seek is in the past.

We see that the different approaches need different *xspaces* and time of data, and due to the physically remote nature of the instructor and learner, a broader design space exists here.

5.4 Designing Remote Teaching of Physical Tasks

In the instruction of physical tasks, the performance of a task may occur in different physical spaces, and one may need to see either live information or a recorded one. In addition to these two fundamental dimensions, we also need to consider how AR, VR, and spatial capture technology can record, augment and facilitate the remote teaching of physical skills. To understand this, we present a design space with four dimensions of perceived space, time, the display configuration, and data modality to form a design space (Figure 5.2) which can be used to categorize the functionalities of a system.

While there are other elements of this space that one could explore (e.g., haptics, handheld AR, embedded sensors), however, this design space focuses primarily on head-mounted Mixed Reality displays, which seem to have the largest potential for this domain, though this space could be expanded upon in future work.

Perceived Space

The perceived space dimension in the context of this chapter is closely related to the *xspace* for a given physical task. Perceived space here consists of all the physical spaces of the *xspace* that the user can see and interact with, either directly or through some visual representation. In the case of a bi-directional interface, each user has the potential to see and interact with their own local space, which is the environment that they are physically within and the objects within that space. The user would primarily interact in this space to execute the task or action in their own environment with their own objects or tools.

The user may also see the other participant's remote space, which is the environment and objects of the other user. They see this using different *visual representations* / Data Modality discussed later in this section. With these representations of remote space, the user can observe, inspect and comment on the remote user's actions, body movements, and interactions with tools or objects.

Additionally, a user may see and interact with both their own local space as well as the remote space. With this configuration, a user can see and interact with the remote user as they perform a task within their own local environment, facilitating a 'work-along' scenario. A user may also choose to see no environment and only render the audio and an avatar of the remote user. This configuration allows users to have a focused conversation about the task, free from other environmental distractions.

Time of Data

The *Time of Data* dimension refers to "when" the data, that the users are interacting with, was captured. Note that this is different from the "Time" axis in the Space-Time matrix. The *Time of Data* could be live, in which case the users see a real-time view of their local or remote space. In such a scenario, it boils down to a *synchronous* interaction between the two users. On the other hand, it could be recorded data of another user and their space

being replayed for observation and review. Here the *Time of Data* is in the past, and thus this could allow for an *asynchronous* interaction between them.

Besides these two types of interaction, it could also be a recorded data stream of either user's actions and environments. Both users may navigate it together and have synchronous communication regarding it through voice, gesture, gaze, annotations, and other tools. In this case, it is a synchronous interaction in which the *Time of Data* is the past.

Display Configuration

The display configuration dimension refers to how the user can see and interact with a given space, which can take many forms depending on the technology available. In this chapter, we explore augmented reality to observe and interact with user's space. This gives the user a direct view of the environment, the ability to interact with it naturally, and the ability to augment and annotate the local space. Additionally, augmented reality enables the user to situate the *visual representations* of the remote person (and their space) in their own space and interact with them as if they are present. However, having an AR view of their current space may be distracting when focusing on the remote environment or reviewing recorded data. For these circumstances, Loki offers the ability to enter virtual reality, where only the desired data is rendered. This allows users to eliminate distractions from their current environment and focus on the data of interest for modeling and reflection.

Furthermore, such a choice allows for different ways in which a user can choose to carry out perspective exploration interactions (Section 4.3) of the Cross-Dimensional Medium. In this chapter, we focus on head-mounted displays, which enable switching between VR and AR. Other display configurations, such as projected or hand-held video-see-through AR, are also possible but outside the scope of the investigation of the current chapter.

Data Modality

The data modality dimension refers to the collected data type and the *visual representation* used to convey the information. Using spatial data capture, the user can see and interact with the local or remote space using a 3D reconstruction of that environment. This spatial information allows the user to obtain 3D information not readily available through other forms (e.g., video), navigate to novel viewpoints to avoid occlusion issues, and add annotations in 3D space. Additionally, the user could see a video of the space. The video can provide a high resolution, easily understood mechanism to comprehend the environment and activities. In the local space, multiple videos can provide varied viewpoints to enable third-person views of the user's own actions and environment. The choice of modality allows a user to carry out the *dimensional-detail* interactions (Section 4.3) of the Cross-Dimensional Medium. Many other technologies can be leveraged to provide novel lenses to view and interact with the spaces (e.g., embedded sensors, infrared or other non-visual imaging, recording audio). In this chapter and across this dissertation, we limit our exploration to the visual ones.

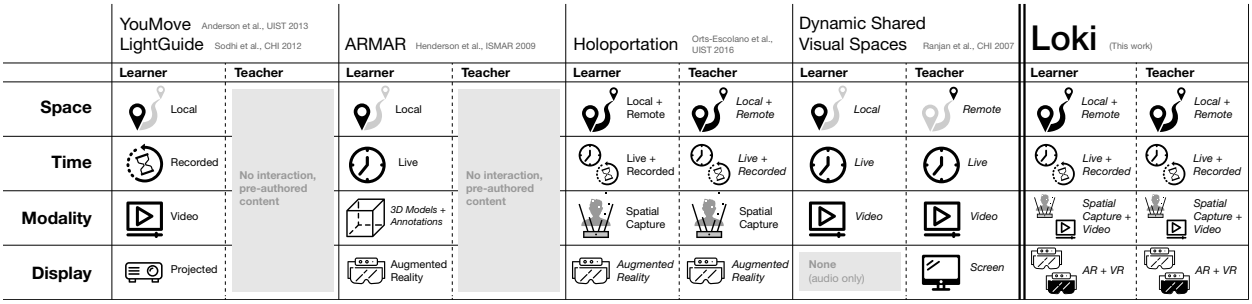


Figure 5.3: Prior work in the area as they fall within the design space outlined above.

Summary

This design space can be used to characterize key design decisions in prior work and elucidate important gaps (Figure 5.3). Notably, most prior work makes a single set of choices along these dimensions and enforces that the learner has one type of display while the instructor has another. For example, remote assistance systems [151] often focus exclusively on real-time video information sent to the instructor, with only an audio channel for the learner. Some recent Mixed Reality projects offer access to both live and recorded spatial data [112, 140] but do not offer symmetric affordances for local and remote participants.

A key insight from our background review of the learning literature in Section 2.4 was that teaching physical tasks involves several distinct phases – observation of the teacher and the learner and vice versa; real-time feedback and reflection on past performance; and shifting foci from gross movements to detailed, subtle actions. Within a single training session, the ability to switch fluidly between these configurations could be very beneficial as the learning environment can be tailored to the optimal communication mechanism for that stage of learning. As a result, the interfaces for both the teacher and the learner must be flexible enough to support these multiple modes of interaction.

5.5 The Loki System

Loki enables remote instruction of physical tasks using bi-directional, Mixed Reality telepresence (Figure 5.4). The system comprises two spaces, each equipped with multiple cameras that capture RGB and depth data from the respective spaces. These cameras are tracked, repositioning them in the space to allow the remote participant to capture optimal viewing angles. This also enables focus and context interactions [10], providing higher resolution and/or overall context as needed. In addition, the user wears a Mixed Reality display (HTC Vive + Zed Mini), allowing them to transition between AR and VR and adapt the interface to their needs, performing input via two 6-DOF Vive controllers.

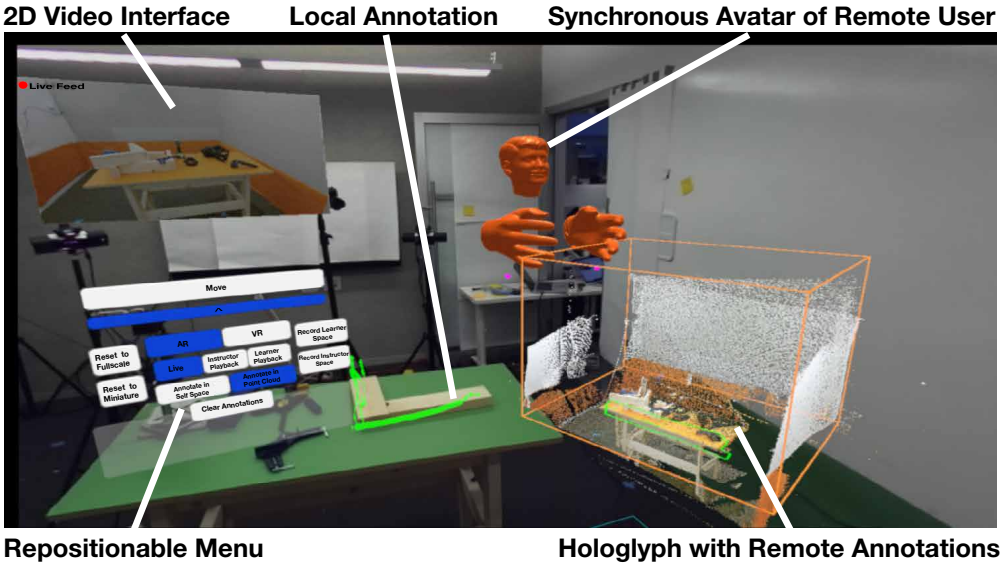


Figure 5.4: Loki overview, showing the view from the learner's perspective while they are in AR, viewing the remote location live through the hologlyph and video. (Note: Menu UI text emphasized for figure clarity)

Hologlyph + Avatars: Loki's Interactive Cross-Dimensional Medium

For most remote instruction approaches, there exists some form of asymmetry between the instructor and the learner in how they perceive and interact with the *xspace* pertaining to that approach. To reduce this asymmetry, Loki uses an Interactive Cross-Dimensional Medium termed as the Hologlyph. The spatial data is captured by depth cameras (Kinects) and rendered within a 3D widget (a hologlyph) that the viewer can manipulate. In addition to the point cloud of the captured environment, the hologlyph also contains any annotations anchored to that space. A color-coded bounding box outlines the bounds of the hologlyph region. If multiple Kinects are capturing the scene, data from all of them is calibrated and consolidated within a single hologlyph. The hologlyph can be scaled, positioned, and rotated within the space using the hand controllers.

It also contains the 2D color video streams captured by the Kinects and is presented in a floating window above the menu (Figure 5.4). The window is repositionable within the environment, allowing a user to position it in a convenient spot. The user can switch between the available camera feeds by raycasting at the video with the controller and pressing the trigger. Additionally, when teleporting around the hologlyph, if a user teleports to one of the snap-teleport spots associated with a camera, the camera feed will update with the relevant video feed. Finally, while viewing recorded data, the video widget has a scrubbing thumb

to allow the user to navigate back and forth through time. Scrubbing this timeline updates the playback time for both the video and the data within the hologlyph.

Besides the hologlyph, Loki offers a real-time rendering of the remote user's relative position from which they observe the local user's space to facilitate collaboration. This is done by rendering a 3D model of a set of head and hands that correspond to the remote user's head and hand poses as tracked by the HTC Vive. The avatar allows the local user to have a sense of what the remote user is seeing. The finer hand movements and gestures need to be observed using the Hologlpyh.

The videos, point clouds, and the synchronous avatar together allow for the property of variable visual representations for Loki's XD medium.

Interface Components

To navigate between the different visual representations and interact with them, Loki supports a variety of interface primitives.

Synchronous Audio – A real-time audio connection allows the remote and local users to speak and communicate verbally. It also relays ambient audio of the room (e.g., the sounds of power tools, musical instruments, etc.). Currently, the audio is just a single channel and is not spatially mapped.

Menu – To control the various modes and features available within Loki, a simple 2D menu is available, which is interacted with via raycasting using the controllers (Figure 5.4). The menu is spatially linked to the video and can be repositioned or collapsed when not in use to reduce visual obstruction and complexity of the scene. In addition, it provides shortcuts to quickly reset viewpoints and scales, rendering the Hologlyph in full-scale or in a miniature view [169]. It also contains buttons to selectively enable recording and playback of the local and remote spaces. Finally, it allows users to switch between AR and VR, depending on which mode might be more relevant to their current task.

Teleportation – To navigate the Hologlyph, the user can transform it using the controllers, walk around in their own space to change their viewpoint, or they can use teleportation. Pressing the center of the trackpad activates a standard projectile-based teleportation ray that allows the user to navigate to any point within the Hologlyph. The 3D point cloud in the Hologlyph would then enlarge to a 1:1 scale placing the user in the desired teleport location. Users can then adjust their orientation by manipulating the Hologlyph with standard "grab, drag and scale" interactions in VR.

Loki additionally supports snap-teleport points, which are points where the Kinect cameras are present. Green circles visualize these snap-teleports, and when the user teleports near these points, the teleport location snaps to that of the snap-teleport. The video in the video player switches to that of the corresponding Kinect in that location. This increases the spatial context when choosing an appropriate video feed in multi-camera settings.

Together, the menu and the ability to teleport and scale allow for the three interactions that we defined earlier - Perspective exploration, Temporal exploration, and Dimensional-detail exploration

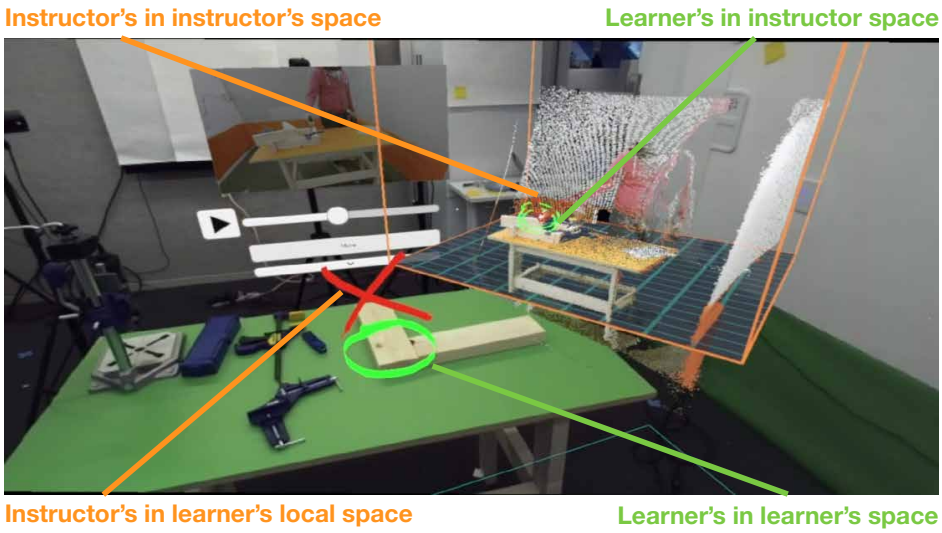


Figure 5.5: Annotations in the local (solid lines) and remote (outlined) spaces, as made by the instructor (orange) and the learner (green).

Multi-Space Annotations – Loki features a novel bi-directional and context-specific 3D annotation system (Figure 5.5). Within Loki there are two types of annotations: Annotations by the user in the context of the local space are rendered as solid lines, and those in the context of hologlyph are rendered as outlined lines. For instance, annotations by the local user in the remote space, achieved by annotating within the hologlyph, appear outlined to the local user and in solid line to the remote user in their respective environment. These annotation types are available to both users, with the color (orange or green) denoting the author of the annotation. The type of annotation (local or remote) is determined by context, with annotations created within the hologlyph defaulting to remote annotations and those outside the hologlyph defaulting to local annotations. However, this behavior can be overridden using the menu. **This ability of annotations, and avatars to meaningfully translate across corresponding virtual and physical counterparts is a crucial property of the Cross-Dimensional Medium here. This is shown in Figure 5.6.**

These annotations can facilitate communication and feedback in a learning scenario. For instance, the learner can use them to indicate a particular region of interest in their workspace (e.g., where on a workbench they intend to place items), or they can help instructors give guidance or feedback on a learner's actions, similar to telestration [22], but within a 3D space. Other forms of annotations, beyond free-hand curves, may be interesting to explore in the future [143].

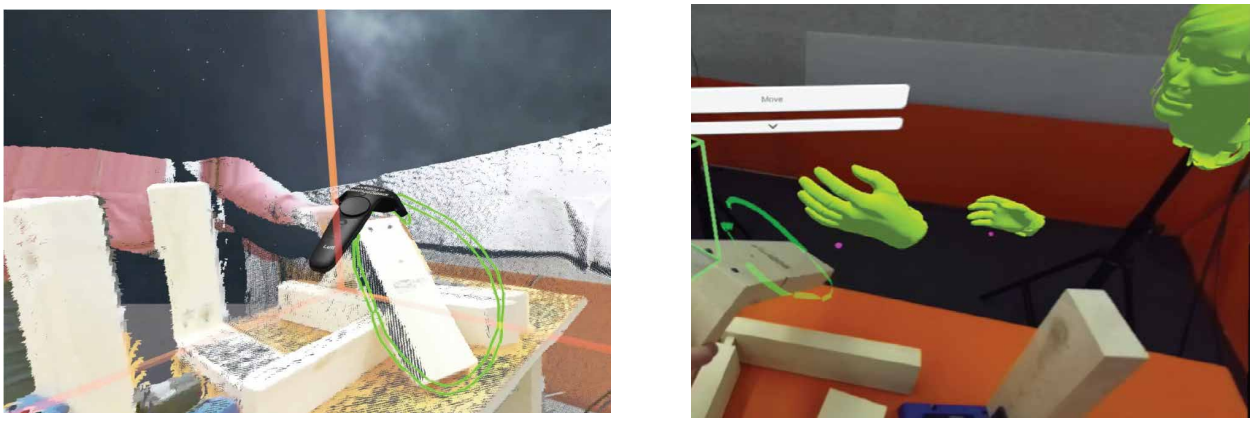


Figure 5.6: Left: A learner co-habits the 3D point cloud representation of a remote instructor and performs annotations; Right: The annotation as well as the learner rendered as a 3D avatar (in AR) in the corresponding position in the instructor's space

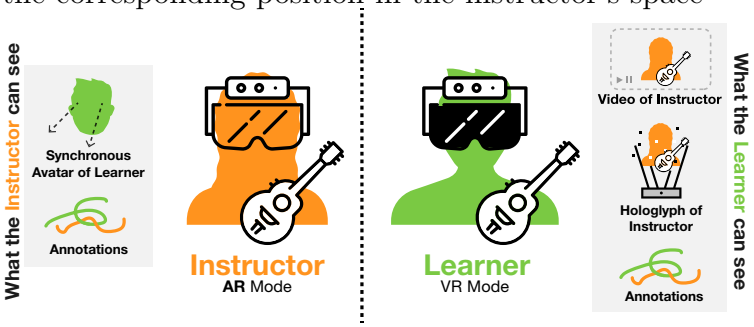


Figure 5.7: Observation lets learners focus on the instructor modeling the desired behavior.

Sample Configurations

While the flexibility of Loki offers many different configurations that might be useful depending on the scenario, we believe that a few common configurations will provide substantial value and enable effective learning workflows.

Observation – this configuration is intended to support the instructor in modeling the desired behavior and might be most beneficial within the cognitive phase of learning. The *xspace* here is the body of the instructor and their physical space, the body of the learner, and any annotations anchored to the physical space.

With Loki, the instructor and learner adopt the following usage to reduce their symmetry: The instructor is in AR with a view of their environment, and the learner is in VR, focusing on the hologlyph and the video (Figure 5.7). The instructor would perform the task they intend to teach, potentially annotating the points of interest. The learner could navigate

Factor	Level	**Relevant Physical Space** Work bench of the remote Instructor	**Relevant Virtual Space** Virtual Annotations and avatars anchored to the Instructor's workbench
Visual Representation	Same		With Loki - Virtual annotations and avatars are synced across the point clouds and the corresponding physical space
	Limited Representation - Same Output Dimensionality	With Loki - Live 3D point clouds and 3D avatars of users	
	Limited Representation - Reduced Output Dimensionality		
	Video Only	Without Loki	
	Not Visible		Without Loki
Interaction Affordance	Same		With Loki - Interface for accessing and interacting with virtual elements is symmetrical for both users
	Limited Representation - Same Input Dimensionality	With Loki - Free 3D exploration in the space, but cannot manipulate the physical objects	
	Limited Representation - Reduced Input Dimensionality		
	No Input	Without Loki	Without Loki

Figure 5.8: Role of Interactive Cross-Dimensional Media in facilitating *Observation* of physical tasks

between videos and around the hologlyph to obtain novel viewpoints and can annotate the instructor's environment as they ask questions. As the instructor can see the avatar representing the learner's viewpoint, they can ensure that the learner focuses on the correct elements. The instructor can also use the avatar to understand the viewpoint that the learner would like to see, and maybe move a Kinect to that location to give the learner a clear video feed from there. This helps bridge the asymmetry between the users with respect to the visual representation and interaction affordances of *xspaces* and is summarized in Figure 5.8.

Instructor Guidance – this configuration is intended to support coaching and scaffolding while the learner has a concrete experience [104] and could be most useful in the cognitive and associative phases of learning. Technically, it is similar to 'Observation'. However, the roles are reversed with the learner in AR and the instructor in VR (Figure 5.9). Within this mode, the instructor can scaffold the learner as they perform the task in their own environment, and provide proactive cues, guidance, or feedback on the performance using annotations, voice, and gestures.

Work Along – this configuration is intended to support active experimentation and occasional feedback through coaching and is likely most useful within the associative and automatic phases of learning. Here the *xspace* is the *union* of the *xspaces* of the *Observation* and *Instructor Guidance* that we saw above.

Thus, within this configuration, both instructor and learner are in AR, with the hologlyph and the video positioned, so they do not interfere with the primary task (Figure 5.10). Depending on the spatial layout of the physical rooms and the type of details required,

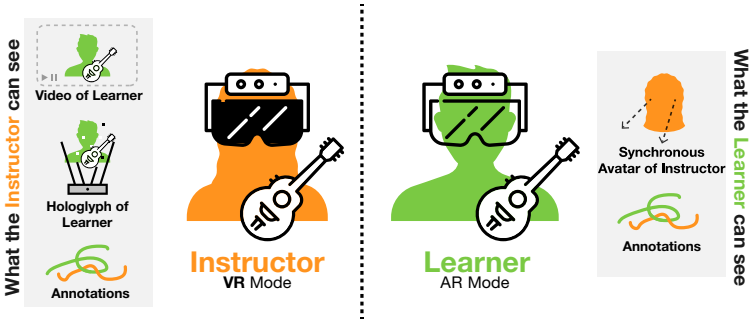


Figure 5.9: Instructor guidance enables the instructor to provide coaching and scaffolding and for the learner to have a concrete experience [104].

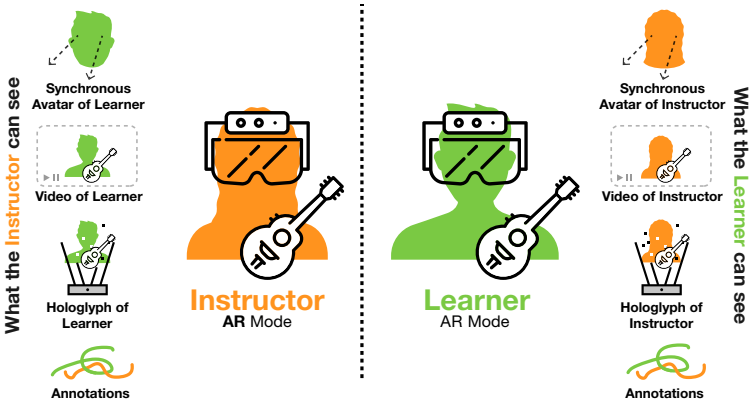


Figure 5.10: Work along enables feedback through coaching and allows the learner to experiment with the task actively.

the hologlyph may be a small world-in-miniature sitting on a workbench, or it could be a full 1:1 scale rendering. This configuration allows for constant, low-touch collaboration while performing independent work. The instructor can occasionally check on the learner's progress and interrupt them if necessary to provide guidance, or the learner can interrupt the instructor if they have a question or need assistance.

Collaborative review – this configuration is intended to support an opportunity for reflection and abstract conceptualization and is likely useful throughout all stages of learning. This involves accessing, interacting, and discussing synchronously with the recording of the task performance of either user by either user. Essentially, it is a synchronous (live) collaboration over recorded data. The *xspace* here is the recorded physical space, live body pose, and annotations of both users, in the context of the recording.

Within this configuration, both the instructor and learner are in VR, viewing a common

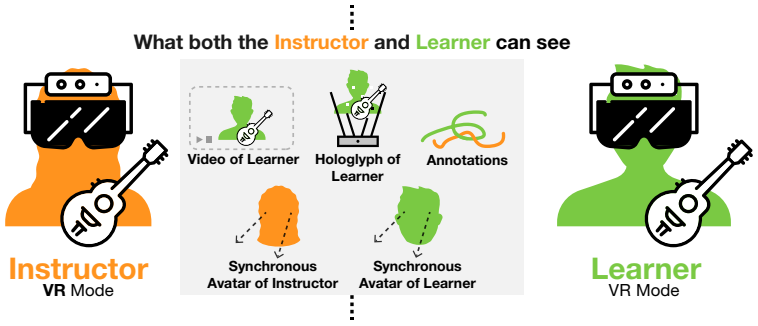


Figure 5.11: The collaborative review provides an opportunity for reflection and allows for abstract conceptualization through a shared VR experience.

hologlyph and video stream of the recorded data (Figure 5.11). This may be a recording of either the instructor or the learner. In this view, both users can see each other's avatar, speak, and collaboratively annotate the hologlyph. For example, the instructor could annotate key areas of interest (e.g., errors the learner made or parts of their own actions), and the learner can indicate locations where they have questions or where more details are needed for understanding the task. Though the users are remotely located, in this scenario, there exists no asymmetry between the users because without Loki such an interaction is not possible by either user.

Implementation

Loki consists of two symmetric hardware systems that each leverage Kinect depth cameras for spatial capture and an HTC Vive [188] and ZED Mini [200] for the Mixed Reality displays (Figure 5.12). The PCs that capture the Kinect data are Windows laptops with an Intel i5 chip. The PCs that render the MR content are gaming PCs with Intel i7 chip, nVidia GTX1080 graphics card, and Unity 2018.3.

Rendering Mixed Reality: Loki can transition between AR and VR using the HTC Vive headset and the ZED Mini. The ZED Mini is a stereo pass-through camera (Having FOV - 85^{o}(H) and 54^{o}(V)) designed for AR applications, capable of depth mapping and lighting estimation. These features allow it to process real-time object occlusions between the virtual and real worlds. To maintain a consistent coordinate system between VR and AR, the ZED Mini's native inside-out tracking is disabled and the Vive's tracking is used instead. In addition, as the user switches from AR to VR, the camera feeds from the ZED Mini are disabled and replaced with the VR camera's render. As rendering large point clouds in AR can be taxing on the computer, Loki uses custom shaders and reduces the update frequency of the point cloud from its native 30fps to 10fps. This allows the AR experience to remain high quality and responsive while giving the user enough context about the remote environment.

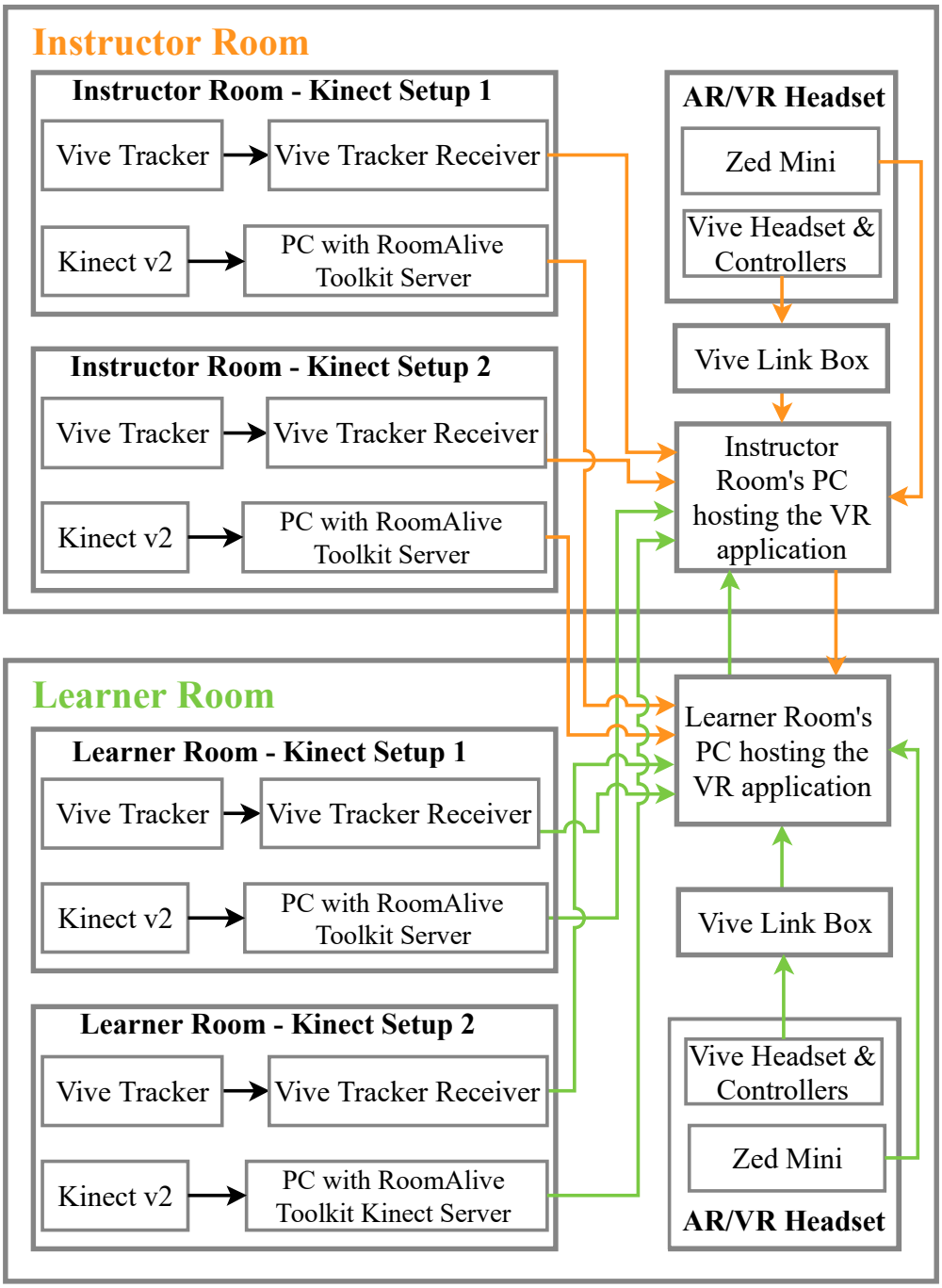


Figure 5.12: Overview of the hardware configuration of Loki across the two spaces.

Additionally, when switching to VR, the framerate is increased, providing a better experience when the user's attention is likely focused on the hologlyph.

Communication: The two PCs communicate with each other through a custom TCP/IP Unity plugin. The plugin serializes, sends and receives, and deserializes the custom data frame Loki uses to synchronize the experience across both PCs at approximately 66Hz. This data frame includes the users' hands and head positions, their current modes and controller states, the tracked positions of the Kinects, and other lightweight metadata needed to synchronize the systems. The Kinect data (RGB-D data) is transmitted to the remote PC using RoomAlive Toolkit's KinectV2 Server [96] program through a router. The RoomAlive toolkit handles the data capture, compression, and decompression of the Kinect data. In addition, live audio between the two rooms was transmitted through an IP telecom system.

Aligning Point Clouds: The HTC Vive Headset, controllers, and trackers are tracked by referencing HTC's IR emitters mounted in each room. The trackers are mounted to the Kinect cameras and track their positions, then used to dynamically auto-calibrate the multiple Kinect feeds at runtime. While the original RoomAlive Toolkit renders a mesh of the scene, the mesh tends to distort the finer details in the scene. This distortion is problematic for teaching some physical tasks where these finer details could play an important role. Therefore, we render the raw-point clouds using the Kinect's RGB-D data frames assembled using the toolkit and rendered using a custom shader. We then use a custom auto-calibration script that uses the position data from the trackers mounted on the Kinects to assemble and calibrate each Kinect's individual point cloud. While there is some offset between the point cloud captured by each Kinect, this error does not play a significant role for many tasks where precision matters Loki can be run with a single Kinect to eliminate this offset.

Telepresence and Spatial Synchronization: Spatial consistency is essential to maintain coherent telepresence between the two environments. The user's avatar and annotations made within one hologlyph must be accurately mapped to the augmented reality environment of the other user. To perform this mapping, Loki first computes the relative transform of the desired object to one of the Kinects that render these point clouds in the coordinate system of the hologlyph. Once that relative transform is computed, Loki renders a virtual copy of the object of interest in the real environment of the remote partner by computing its position with respect to the position of the corresponding real Kinect in the remote room in its own Vive coordinate system. We know the position of the real Kinect in the remote room through the Vive tracker attached to it. Once we have the basic pose, we can compute and render the scale of the rendered object per the use case scenario. To transfer annotation from real space to a corresponding hologlyph, we use the same transform pipeline but in the reverse direction. It is again important to note that, since we use the dynamically Vive-tracked position of Kinects as references between the two coordinate systems, we could change the position of Kinects and this pipeline ensures that the spatial transforms would still function, enabling a robust telepresence experience.

Shared Playback Space: As the Kinect data and video is bandwidth intensive, it is only stored on one of the PCs. During shared playback, the data needs to be synchronized across both PCs. When a user initiates playback of the remote user's recording, the data stream is

serialized in a binary file on a shared network drive. Once it is copied successfully and the file stream is closed, a 'ready for playback' flag is updated to synchronize both programs. The time to synchronize the binary file data varies depending on the length of the recording but is always under 10s for the durations tested (around 30-60 seconds of recorded content). The file is then opened in a read-only mode and copied into a local buffer while also deserializing and processing the stream to an appropriate data structure to support playback operations such as quickly seeking to an instance.

The synchronized coordination of network read-only streams for playback ensures that both users operate with the same set of file streams when they are in a playback. Following the initial synchronization, a shared immersive telepresence experience is facilitated through the sharing of playback metadata like the time of playback and the video player state as well as the spatially synchronized rendering of virtual avatars and annotations in the coordinate space of the respective playback rooms.

5.6 Evaluation

Scenario Evaluation

To validate Loki's Cross-Dimensional Medium and the utility of its interactions, we implemented and assessed several instructional scenarios that span the various instructional approaches. These scenarios were carried out and tested with Loki by the authors. The screenshots from those tests are included as figures in the respective scenario subsections.

Teaching Guitar: To instruct a learner on how to play a certain chord (Figure 5.1), an instructor positions one Kinect near the neck of the guitar so the learner can view a high-resolution video and depth map of the fretboard. The other Kinect is placed in such a manner so as to capture the context and body pose in which the guitar is being held and used. Next, they enter an *observation* configuration where the instructor demonstrates the proper fingering for the chord that they want the learner to hold and strums the strings as they play the chord (Figure 5.1a, d). The learner, in VR, carefully watches the video and point cloud and annotates to ask a question. Next, both users enter AR in a *work-along* configuration, each with a live fullscale point cloud in front of them (Figure 5.1b, e). The instructor can watch the learner perform and offers feedback. For instance, when the instructor hears a muffled note, they quickly inspect the point cloud in real-time and verbally coaches the learner on which finger needs to be moved. Later, the learner is still playing incorrectly, so the instructor and learner enter a *collaborative review* of the learner's performance, where the instructor scrubs to a particular point in time where the finger looks like it's touching the string, and they highlight the error for the learner (Figure 5.1c, f).

Coaching Baseball: In coaching a person's baseball swinging action, the learner and instructor enter the *instructor guidance* configuration where the instructor observes the learner and comments on their performance (Figure 5.13a, b). After a few recorded demonstrations from the learner, the instructor and learner enter a *collaborative review* to comment on the

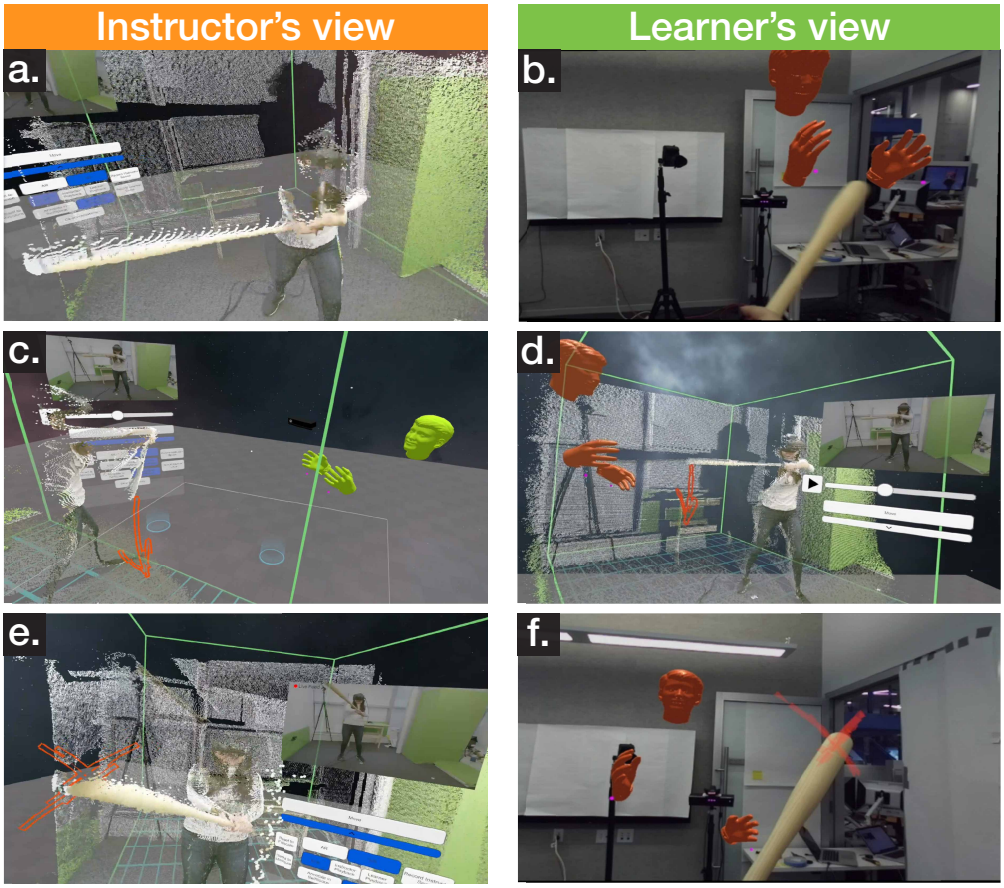


Figure 5.13: Overview of instructor coaching a learner through learning to swing a baseball bat. The instructor observes the learner's initial swings through VR (a, b), then records their performance for them to reflect on (c, d). After coaching, the instructor guides the swing through a target placed in the learner's AR space (e, f).

learner's swing action and indicates through annotations that the action needs to be lowered (Figure 5.13c, d). Both users then switch to live data, and the instructor annotates an easier target for the learner to aim for and offers real-time corrections to overcome their repeated error (Figure 5.13e, f).

Sculpting: In mentoring a learner on clay sculpting, the instructor and learner enter into an *observation* configuration where the learner gets an overview of the task and the instructor begins by forming the initial shape (Figure 5.14a , b). Following the introduction, they then switch to the *work-along* configuration, and the learner places the instructor's point cloud and video off to the side as they both work on their own (Figure 5.14c, d). Since the learner was focused on their own sculpture, they missed a critical step from the instructor. Rather than re-perform the step and spoil the instructor's sculpture, both instructor and learner

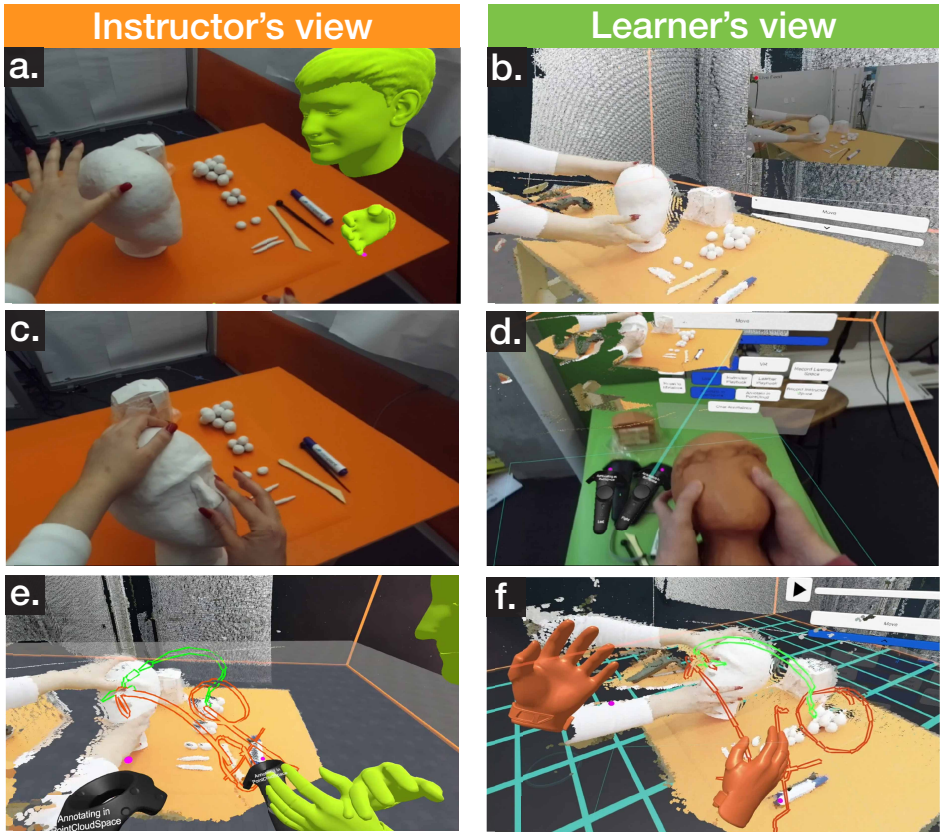


Figure 5.14: An instructor teaches a novice the sculpting techniques for modeling a head. The instructor models the behavior for the learner, who is viewing the demonstration in VR (a, b). Both users *work alongside* each other, with light coaching and feedback (c, d). After the learner misses a critical step, the instructor enters a *collaborative review* to highlight the essential elements (e, f).

enter a *collaborative review* of the instructor's performance in VR, where the instructor reviews the steps they took and annotates them to highlight important actions and tools used (Figure 5.14e, f).

Workshop Learning through Peers: Two members of a woodworking community use Loki while working to communicate conveniently. They have varied skill sets and often share tips with each other. Primarily working in the *work-along* configuration, both users focus on their task and place the remote peer in a miniature scale to the side of their workbench, occasionally observing their remote peer (Figure 5.15a, b). When one user encounters an issue, such as uncertainty in how two wooden pieces should be joined, they ask their remote peer for assistance. The remote peer (now the instructor) then enters the instructor guidance to observe the environment and context of the user through video and spatial capture. They

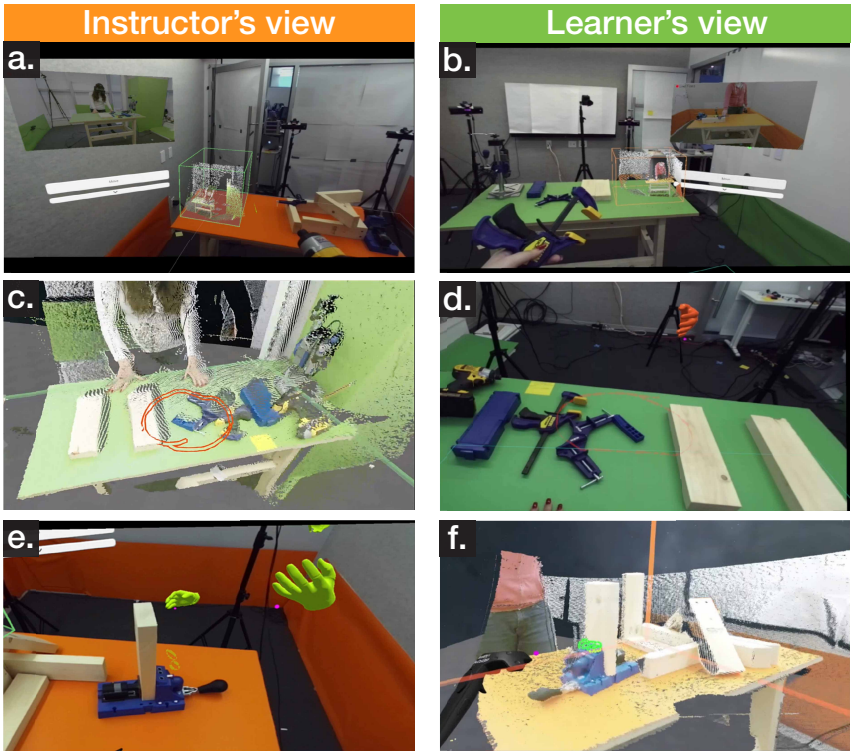


Figure 5.15: Overview of two peers working independently, providing on-demand mentorship. One peer encounters an issue joining two pieces of wood and asks his peer for help, who becomes the instructor. The instructor visits the peer in VR to examine their environment and understand the problem (a, b). The instructor then enters a *demonstration* mode, where they switch to AR and demonstrate possible solutions for joining the wood (c, d). The instructor then provides guidance directly in the learner's space to coach them on how to use the suggested tools (e, f).

see several options for the joint, such as metal brackets, pocket holes, or more complex joinery (Figure 5.15c, d). Both then transition to *observation* where the instructor demonstrates various types of points and coaches the user on how to use a pocket hole jig to drill holes in their boards (Figure 5.15e, f).

Exploratory User Study

We evaluated Loki by an evaluation where participants learned foam carving to create a 3D foam pyramid shown in Figure 5.16.

Procedure: The study required participants to learn a hot wire 3D foam carving task over a 30-minute session from a remotely located instructor (an author of this work) using Loki. Prior to the session, all users were given a 5-minute safety training on the usage of the

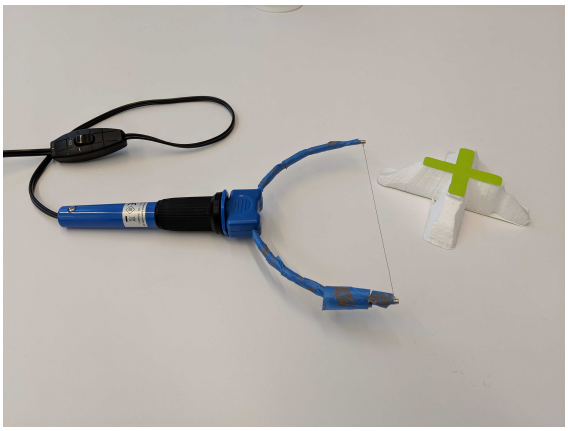


Figure 5.16: The end goal of the foam carving task in the user study

foam cutter, as well as a 15-minute training session on the Loki system itself. We recruited 8 participants (2 male, 6 female, age range 22-34 years), and they were compensated with a 50CAD gift card.

Measurement: After the session, the users completed a questionnaire regarding their ability to understand the different elements of the instruction and the utility of the different features and modes of Loki. The ratings were based on a 5-point Likert scale. Following the questionnaire, we then conducted semi-structured interviews to better understand the value of the different features of Loki to the participants.

Results and Discussion: In the post-task questionnaire, all users agreed or strongly agreed that the entire system helped them understand what was being taught. Using this same metric, 7 of 8 users found value in collaborative review, 6 in point clouds, 4 in videos, 5 in instructors' annotations, and 5 in the ability to switch between AR-VR modes. Depending on their personal learning styles and comfort, there were variations in how participants preferred to use our system. Users reported using the system differently depending on the task at hand. Most users first observed in VR, then transitioned to AR to perform the task:

P3: "*if I was ... actually building myself and like I definitely need to be in AR mode. But I think VR, it was nice if I was simply watching and didn't want the table and other things to be in the way.*"

Participants varied in how they used the different visual representations provided by Loki. Some liked to keep the 3D point cloud small and kept it on the side as reference material, while others preferred it in a 1:1 scale directly opposite or beside them. They reasoned about the tradeoffs between the point clouds and the video:

P2: "*the point cloud was good because if I miss something in real time, I could just turn around and see a slightly different perspective...and if you're in a video, you don't want to switch between perspectives, toggle between several videos just to find the right one.*"

They also mentioned how those tradeoffs affect the usage of other features like annotations

and collaborative playback review:

P5: "*point cloud has benefit, you get more 3D perception...you can annotate it in context of the 3D scene.*"

We found that participants appreciated Loki's unified workflow to allow for exploration of the different visual representations such as videos, 3D models, and point clouds, as well as the different interaction affordances such as annotations, collaborative review, and playback. Most of them also felt that these interactions helped them engage better with their partners in the one-on-one learning setting of the study and made the learning process enjoyable. P1 stated that engagement helps the learning process in an indirect manner:

"*It feels more like you're connected to them...I think that it makes the learning process more enjoyable, which would probably help me learn.*"

Another participant talked about how this engagement and telepresence gave rise to social dynamics of movements in local versus remote spaces:

P7: "*If you invite someone to your house, you feel more comfortable because it's your space and, but if you visit your friend's house, you feel less comfortable because that's another's house... [similarly] when [instructor] visits my space, I feel very comfortable. But when I visit [instructor]'s reconstructed space...I feel like I wanted to keep a social distance and to move in a certain distance that does not make him uncomfortable... Even though I know both are virtual spaces, but I feel different.*"

The study found that participants successfully used Loki in nuanced ways that exercised the different modes to communicate with an instructor within a single learning session. At the end of the study, participants came up with interesting use case scenarios for Loki, such as learning activities like cooking, swinging a bat, arts and crafts, origami folds, musical instruments such as flute, and discussing sitting postures with physicians. For these different use cases, participants described how the different features of Loki's Cross-Dimensional Medium could be used to accomplish the wide variety of learning outcomes in these tasks.

5.7 Limitations

Through our explorations, we have found utility and value in a system that can capture and relay spatial data. However, as different scenarios were examined and developed, several limitations were uncovered, and some exciting avenues for future work emerged.

Technological limitations

At the time of this work, there are some current technological limitations with the system that we anticipate will be resolved in the near future. The Mixed Reality hardware itself is somewhat limiting: the headset can be cumbersome, with cables occasionally interfering with the primary task, and the latency of the AR headset and the reduced field of view was restricted to some users. Additionally, the controller interface occupies the users' hands and interferes with their task performance. In the future, we anticipate that headsets will become

wireless and less intrusive, and voice and gesture interactions will become more robust and reliable, allowing the system to be used hands-free. The avatars are currently passive with no gaze and finger movements. This is primarily due to commercially available VR hardware. Gaze tracking is not yet common in VR headsets. While systems (Kinect, Leap) can track body/hand pose, these approaches usually fail when users interact with physical objects, as in most physical tasks Loki addresses. Once tracking is reliable, it would be a valuable extension to the avatars. Additionally, in our implementation, users occasionally experience interference between the Vive headset and the Kinect, which causes a temporary loss of tracking. Hardware that uses light in different bands could alleviate this issue.

High bandwidth - Works only on Local network

Loki's current implementation works only over a local network. This is because it uses lossless compression for the image as well as the depth frame. This means that the resulting texture size is high and are not optimized for transmission over the network. We address this technological limitation with the DreamStream system discussed in Chapter 8.

Annotations not anchored to objects

There is a rich space to explore with annotations within this context. Currently, the utility of live annotations is somewhat limited, and they are primarily useful for static objects, as moving objects become misaligned with their static annotations. One area of interest would be examining annotations that snap to content and stay attached even as that content moves through space and time. Additionally, the authoring of temporal annotations seems like a rich space to explore. Adapting some of the techniques proposed in prior dynamic illustration work [99, 98, 143] could allow for very rich annotations, or even annotations that the instructor and learner could interact with (e.g., creating a virtual baseball that moves along a trajectory and varies its speed).

Supports only two spaces

Currently, Loki is bi-directional and only supports connecting two remote spaces. However, there are several use cases [140] where a one-to-many or many-to-many connection may be useful, such as a distributed peer learning scenario where several people are connected in a spatially aware group chat or a scenario where one instructor is teaching a distributed cohort of learners. While there is some apparent value in these scenarios, managing these spaces and providing intuitive and effective ways of interacting with and managing these spaces remains an open research question.

Relies only on visual data

Lastly, Loki explored the use of spatial data and 2D video to capture and relay the people, objects, and environmental context between remote users. While this is a rich set of data, many other channels may be useful, especially when conveying skills that may contain a lot of embedded or tacit knowledge. For example, sensors to detect force or torque profiles, actuators to enable haptic experiences, or novel methods of abstracting or presenting the captured data may prove valuable in capturing and relaying skill-related information.

5.8 Summary

In this work, we have introduced a broader design space for exploring the domain of MR-based live instruction. We then presented Loki, a system that supports this flexible exploration for remote teaching of physical skills. By supporting a range of modalities and various mechanisms for data capture and rendering, Loki's Cross-Dimensional Medium serves as a rich communication medium that leverages spatial data, video, annotations, and playback that helps connect people as they teach and learn real-world tasks. We showed the value of these different features by describing a variety of scenarios we carried out, from teaching guitar to aiding in sculpting and peer learning. We then described a qualitative user evaluation that showed that users could use Loki and found the different features and modes of Loki valuable. We see that Loki achieved three things:

- It enhanced a user's understanding of a remote physical space through different Cross-Dimensional Media interactions.
- It provided a means to spatially interact with another user, which was aided by the medium's property of meaningful translation across representations.
- It enhanced a user's perception of "being at" the remote space by allowing a user to be embodied in the 3D point cloud of the remote space and see it through VR, making them feel like they were there. This was again possible due to the different visual representations and interactions.

While some limitations exist, this work opened up many interesting questions and opportunities for technological improvement.

Chapter 6

TutoriVR

In the current chapter, as well as subsequent chapters 7 and 8, we will focus on multi-user interaction issues that occur in the context of tasks in VR environments. As we saw in Chapter 3, prior works have studied collaboration in VR systems for accomplishing different tasks. However, these research prototypes and systems have two key limitations - First, they assume access to source code. As we saw in Section 2.2, this is hardly the case with today's commercial softwares. A second limitation of prior systems is that many require collaborating users to use Mixed Reality devices. However, not all users have access to them or want to use them. So in this dissertation, we design interactions for the virtual world that work across applications and do not need access to source code, i.e., application-agnostic; wherever possible, work across interfaces with different visual modalities.

6.1 Preamble

In this chapter, we will look at using Interactive Cross-Dimensional Media to improve a user's understanding of 2D video captures of Virtual spaces. In order to develop concrete insights, we will use a prevalent task carried out in these spaces - the creation of 3D designs, precisely Virtual Reality (VR) painting. VR painting is a form of 3D painting done in a VR space. Being a relatively new kind of art form, there is a growing interest within the creative practices community to learn it. Currently, most users learn using community posted 2D-videos on the internet, a screencast video recording of the painting process by an instructor. This is an example of an asynchronous interaction between an instructor and a learner. While such a video-based approach may suffice for teaching 2D-software tools, these videos by themselves fail in delivering crucial details that a user requires to understand actions in a VR space. This is because the *xspace* of the asynchronous interaction, i.e., the virtual space of the instructor, is perceived asymmetrically by the instructor and the learner. While the instructor operates within it in an embodied manner and has complete control over it, the learner sees a 2D video recording of it.

With the help of a formative study, we identify the challenges this asymmetry poses

Figure 6.1: An overview of our VR-embedded tutorial system. Our video widgets (on the left and below the user) can be rendered directly inside the user's active VR painting application (TiltBrush in this example). The user can access useful tutorial content while painting and can interact with the video widgets to explore and learn at their own pace.

to users using video-based tutorials to learn VR activities. Informed by the results of this study, we develop a VR-embedded tutorial system that supplements video tutorials with appropriate Interactive Cross-Dimensional Media. These media provide 3D and contextual aids directly in the user's VR environment. It allows users to carry out perspective, temporal and dimensional-detail exploration of the virtual tasks and actions being performed in the video. An exploratory evaluation showed that these Interactive Cross-Dimensional Media helped users better understand the VR scene and the instructor's actions recorded in the video.

6.2 Introduction

Painting, sketching, and sculpting in Virtual Reality (VR), henceforth referred to as "VR design", are emerging forms of artistic expression. The key characteristic of VR design is the ability to move a hand in mid-air to directly manipulate and create 3D shapes. This WYSIWYG paradigm was foreseen by early research [38] and yet has only become widely available recently thanks to advancements in VR optics and spatial tracking technologies. The freedom of expression in VR is perhaps one of the main reasons VR design has received significant interest from creative communities [35].

VR design is a creative process and requires mastery of skills. Through initial conversations with two professional VR artists and observations of online user groups, we found two apparent requirements for designing in VR: VR UI skills (e.g., how to use interface elements such as 3D menus/tools) and VR artistic skills (e.g., what are the creative techniques available). Currently, learning VR design happens mainly through trial and error. Support for VR design tutorials has been limited. Most VR applications only provide onboarding experiences that focus on basic functions. More importantly, users, while in VR, do not have a convenient way to browse and consume tutorials that other users created. This is in stark contrast to the ongoing focus of the HCI community on supporting tutorials for desktop-based design software [50].

The lack of learning aids for VR design motivates our main research question: "How do we design a tutorial system for design tasks in VR?". In this chapter, we focus our investigation on the feasibility of using user-generated screencast videos of a VR workflow as tutorials in VR. Video is prevalent in online VR user communities and on VR applications' learning webpages [149]. Its apparent benefit is the low overhead cost of authoring: a user only needs to hit the "record" button once to capture. Other users can then view this capture and learn about the author's design process in VR. This type of learning is self-paced, holistic, and may encourage meta-cognitive processes [146]. Additionally, screencast video is also application-agnostic—the user does not need to rely on the VR application to provide dedicated APIs for recording. This means that any user can create and share tutorials for any VR applications, techniques, and features.

However, there are inherent limitations to using 2D videos as tutorials in VR. First, traditional video players are not well-integrated into VR design applications. To view the video in VR, a user needs to pause the current painting process to switch to another application. Or worse, they may have to take off the Head-Mounted Device (HMD) to view the video on the desktop computer. Second, it is not clear to what extent users can interpret the spatial nature of 3D VR tasks from a 2D video. Compared to standard desktop software, carrying out tasks in VR spaces involves interaction in a much richer 3D input space and with dramatically novel interface concepts and interaction designs (e.g., teleportation, 3D color picking, and so on[1]).

To address these challenges, we develop *VR-embedded widgets* that leverage Cross Dimensional Media. These widgets can show video content and interfaces directly in the user's VR painting environment so that a user can simultaneously access the tutorial and the painting. To do this, our widgets inject custom graphics and interfaces into the VR application's rendering and event system so that they can be rendered on top of the application in VR and respond to a user's input in VR. Using these widgets as building blocks, we first developed a basic timeline-based video player and examined whether users can learn from a screencast tutorial video while painting in VR. We find that while the presence of a readily accessible video player within a VR design application is valuable, it is not sufficient. Various issues exist in understanding and learning VR tasks from 2D videos. We distill these issues and

[1] http://www.xr.design/

synthesize a set of design implications focusing on 1) UI accessibility 2) Spatio-temporal exploration 3) Depth perception and 4) Instruction awareness.

Subsequently, based on these findings, we introduce a novel tutorial system for VR design called TutoriVR. Our system takes as input a screencast video captured from both eyes of the instructor and a data log consisting of activity traces of the instructor's VR device, such as controller poses, HMD poses, and button presses. Both of these are trivially accessible in today's VR systems. Using these materials, TutoriVR delivers an enhanced learning experience through leveraging different Cross-Dimensional Media and interactions. First, we enhance the tutorial user's depth perception by giving them a choice to view a stereoscopic video representation of the tutorial. We supplement it with a 3D rendering of controllers and active painting strokes. This allows for increased perspective exploration by enabling the users to view the stroke creation process from different perspectives. In this, we enable the users to better understand controller interactions by highlighting them for an extended duration allowing for enough time for users to understand any quick actions. Finally, we allow for an increased instructional awareness of users through a widget that follows the user's field of view. We conducted a preliminary evaluation to assess TutoriVR, and we found that users found the Cross-Dimensional Media and interactions of TutoriVR useful to complete more critical steps in a tutorial than when using the baseline video player.

In summary, the main contributions of this chapter are:

- Exploring challenges associated with learning VR design through 2D video-based tutorials
- A VR-embedded widget system that supplements and enhances the understanding of video-based tutorials using Interactive Cross-Dimensional Media

6.3 Formative study

We carried out a formative study to observe and identify challenges faced by users who learn VR painting using video-based tutorials. A fair amount of prior work [30, 60, 64, 147] deals with studying how users follow video tutorials to learn to use design software in conventional desktop computing. However, none of these works deal with using video tutorials to learn the operation of design tools specific to VR. More broadly, none of them deal with using a VR environment embedded video player to learn to do actions within a VR space. Carrying out tasks in VR spaces involves interaction in a richer 3D input space as opposed to the 2D input space of conventional desktops. Hence our motivation for the formative study was that learning to paint in VR using 2D video-based tutorials may pose additional challenges compared to learning conventional desktop software.

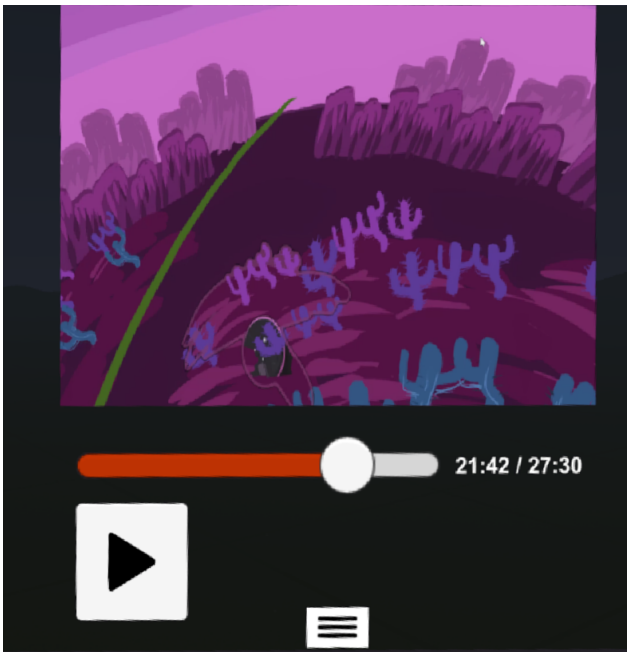


Figure 6.2: Basic video player used in formative study

Development of the Baseline tutorial video player

Currently, only limited solutions exist for users to access a video tutorial directly in VR. The most common way is to switch between VR and watching the video on a desktop computer. A user can also open a VR application in VR, but that would require pausing the VR painting application to carry out any interactions with the video player. Both these approaches disrupt the user's flow of learning and painting.

Since we were already aware of this issue, we didn't want it to influence the findings of the formative study. The formative study aimed to find issues in understanding the 2D videos themselves and not the issue of being able to play and access these videos. So, to facilitate an integrated experience of watching and interacting with videos within the same VR painting environment, we built a custom video player that can work in an embedded manner within the environment of any VR application.

This video player was built using the Unity engine and SteamVR library, which supports its deployment in various standard VR platforms. Our system was developed for the Oculus Rift device. A preview of this embedded video player can be seen in Figure. 6.2. Users can interact with our system using a standard VR laser pointing technique. We included basic video features such as timeline seeking and play/pause. Additionally, to avoid blocking the user's painting in VR, we enabled users to change the UI position. In pilot testing, we found that users frequently needed to check the tutorial by triggering the play/pause button on the UI. This may lead to muscle fatigue [24]. Thus, we also added a play/pause shortcut

button on the user's controller.

Participants

We recruited six participants (5 males and 1 female) through online forums and mailing lists having a community of users with a prior experience in VR painting. Each study lasted for a maximum of 90 minutes, and the users were compensated with a $50 gift card.

Tasks

The study had three tasks focused on different aspects of learning VR painting. The first task aimed at understanding tool usage in VR. The second task involved choosing different tools and creating an object from scratch. The third task required users to follow intricate hand movements and techniques to create a bird-shaped knot for jewelry art. A short video (2-3 minutes) was used for each task. The videos were screencast recordings of an expert VR artist's workflow with a voice-over. The participants were asked to follow the instructions in the video and create their own version of the painting while following a think-aloud protocol. Each task had a maximum time limit of 15 minutes.

Measurement

After each task, the users had to fill out a questionnaire, which asked them to rate their ability to understand the instructions in the video. They were also asked to rate how easy it was for them to navigate and follow these instructions using the video player. Finally, they were asked to rate whether the video player eased the learning process compared to how they usually learn VR painting. Then there was a 5-10 minute semi-structured interview about the user's experience doing the task. Participants' feedback was collected after the study.

Findings of formative study

Embedded baseline video player

Our embedded baseline video player received primarily positive responses in the formative study. Five of the six users rated that our embedded video player eased the learning process compared to how they usually learn to VR paint. The users felt that the embedded player offered an integrated experience to learn VR painting using video tutorials. At the same time, they continued to remain in the creative workflow of their painting process. They appreciated that the video player allowed them to make instant visual comparisons of their work with the tutorial and make quick modifications. In addition, users liked the ability to position the video player at their convenience and that the video player is rendered on top of anything else in the VR environment. Although, in theory, rendering the player on top of VR graphics may introduce discomforts due to depth conflict [132], none of our users

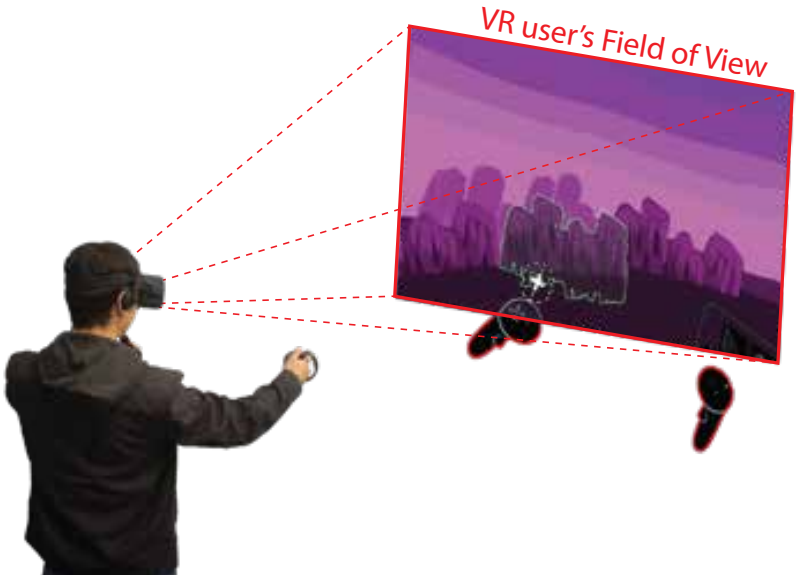


Figure 6.3: VR user may perform actions with the controller, when it may not be in the field of view of the instructor. Such controller actions are absent in the video.

reported or showed signs of it. A possible explanation is that the mobility of the video player allowed users to move it around to avoid blocking their content, thereby preventing depth conflicts. However, depth conflict issues could occur in larger and more cluttered VR spaces, in which case the adjustment solutions proposed by Nguyen et. al [132] could be applied. Alternatively, instead of overlaying directly over the VR content, we could also composite these graphics in accordance with the depth-buffer of the VR scene. This would mean that our custom interfaces get truly integrated with the VR scene. This is more technically sophisticated and has been explored in greater detail in Chapter 8.

We also noted that some improvements were needed to the player - though users could move the video player in z-direction to change its size, they preferred to have a more explicit control. Another major request from users was to have a clearer step structure on the timeline to help with navigation; this is a well-known issue from prior research in video tutorials [147].

Controller interactions

One of the recurring problems that the users faced was understanding the interactions of the VR controllers in the video. Similar to the mouse cursor, the VR controller is the primary input device in most VR applications. Whenever a user gets stuck trying to perform the low-level actions, it was observed that the user actively looks for the VR controller in the video to figure out what action was done with it. This approach, however, is not always feasible due to various reasons. First, controllers may not be in the field of view of the

instructor, in which case the controllers are absent in the video (Figure 6.3). Button presses on the controllers can be too quick to notice, even when presented in the video. This would require users to precisely navigate to the frame where the button was pressed to identify it. Users also found it challenging to follow actions that required a coordinated combination of consecutive button presses and hand gestures. For instance, in the first task, one needs to move a selected object by touching it, hold down a button on the controller, and then move their hands to move the selected object. Many users did not understand this multi-step action through watching videos because it was easy to miss the timing of at least one of those actions.

In summary, we identify three primary sources that lead to the occurrence of these interaction problems while following video tutorials to learn VR design tools:

- **Unfamiliar button mapping:** Learning to perform actions in VR environment requires users to learn an unfamiliar button-action mapping. Since interactions in VR are not standardized, there is no preconceived and established notion of what all buttons mean, how they work and what purpose they serve. This is in contrast to controllers like mice where clicking/dragging type interactions are standardized.

- **Visiblity problem:** There is a lack of ability to visually see the controller interactions followed by an instructor to achieve a particular output.

- **Richer input space:** We have two 6-DoF tracked controllers for interacting with a 3D space. This opens up a richer input space with multiple interactions that need to occur in tandem with button presses, which may not be conveyed effectively in the video depending on the instructor's viewpoint.

Understanding 3D information

Throughout the tasks, users could not figure out the relative depth between the controllers and the different elements of the VR painting environment. This prevents users from being able to reason and answer questions like "How far away from the object should I draw something?", "Is the controller touching an artifact or not?", "Is the drawn stroke an ellipse or a circle?". In a particular selection task, where the instructor selects an artifact of the painting by touching the artifact and pressing a button on the controller, none of the users realized from the video that the controller needed to be in contact with the artifact. All users reported that as the main hurdle in understanding and completing that task.

In addition, all users faced problems understanding the 3D motion of the instructors' hand gestures. When presented with an action involving the creation of a complex stroke, users felt that they had no way of replicating the stroke by seeing the video alone and hence chose to create their version that looked visually similar to that seen from the viewpoint of the instructor. However, there was no easy way to compare their stroke with that drawn in the video and analyze it from different perspectives. In addition to the problem of viewpoint, the creation of such strokes can be very fast as well as be occluded by itself or other elements

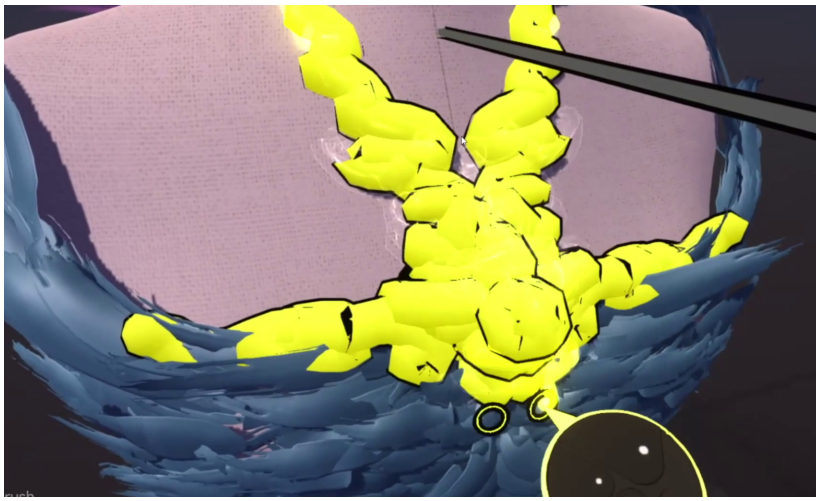


Figure 6.4: A formative study task involved replicating a bird-shaped knot for a jewelry art

in the VR painting environment. This prevents awareness of the stroke's complexity, leading users to misunderstand the stroke structure and draw them incorrectly. At least one of these problems was observed in all the users when they attempted our third task, which involved tying a bird-shaped knot for a jewelry art (Figure 6.4).

The above problems can be traced back to the fact that the videos are inherently 2D, where information relating to the depth and perspective may not be apparent without explicit instructor elicitation or cues like lighting, shadows, and occlusion [152]. However, in many paintings and painting applications, some of these cues could be entirely missing, and explicit elicitation by instructors is generally absent due to the activity-focused, screencast nature of these recordings.

Missing out on instructions

Sometimes users miss important changes in the video, such as tool selection or mode changes. From user interviews, it was realized that this happened for at least one of three reasons.

- It was very common for users in the VR environment to look away from the video player while carrying out their actions, thereby missing important instructions if the instructor did not explicitly voice out of any important step.
- Painting being a creative practice, often gets users immersed in their own painting process that they miss hearing an important step while the video is constantly played in the background.
- Due to the lack of a good tutorial navigation mechanism, the user skipped an important step while navigating the video.

6.4 Design Implications

Based on these observations from the formative study, we derive a generic set of design implications for a screencast video-based tutorial system for Virtual Reality painting.

Accessibility of the video player: The tutorial video player should be accessible within the VR painting environment of the user. This would give users a quick visual reference for the tutorial steps, allow for easy verification of their work, and help retain the user's workflow during the painting process. In addition, the player must also address the typical problems faced by any video-based tutorial system, such as video navigation by explicitly marking important events and enabling quick access to those sections.

Spatio-temporal exploration: The system should offer the ability for users to explore VR-specific spatial actions, including controller poses, button interactions, trajectories, and the stroke creation process from different perspectives. The system should then retain this information for an extended period so that users can still observe and interpret it even if actions are ephemeral. This would give users the spatial and temporal freedom to explore, analyze and understand the various complex actions in the VR space.

Relative Depth Perception: User should be aware and instantly be able to perceive the relative depth of an instructor's action. Though the perspective exploration supports understanding the absolute shape and trajectory of actions carried out by the instructor with their controllers, it may not quite convey the awareness as well as information regarding the spatial occurrence of those actions, especially in terms of depth, relative to the other elements present in the VR environment. For instance, questions like "Is the controller touching an object or not?", "Did the controller move in z-direction? If yes, by how much?", it may be hard to answer by just using 2D videos without additional information on relative depth.

User awareness of the tutorial instructions: As with most 3D spatial tasks, user's workspace in VR is much larger than their field of view. This makes it easy for the user to be engaged in a task, looking away from the video player. In this case, users can miss a tutorial's critical steps. Therefore, the tutorial system must provide awareness aids to the user regarding important events and help the user keep track of the tutorial while minimally disrupting the user's activity.

6.5 *xspace* and asymmetry for tutorials of VR tasks

As outlined in Section 4.3, to be able to design collaborative interactions for Mixed Reality environments, we will begin with identifying the *xspace* for the task. Here the task is to understand a tutorial of an activity carried out in a virtual space (i.e., VR painting) and be able to recreate it. Since the users do not interact with one another in real-time, this is a form of Asynchronous collaboration. From the formative study, the *xspace* consists of the virtual scene in the video, spatial actions carried out by the instructor, and the controller actions carried out by them.

The *asymmetry* between the users stems from multiple sources: (1) While the instructor sees the environment and their actions in 3D and can look around, the learner can neither see them in 3D nor have the freedom to look around. (2) Since the instructor holds the controller, their proprioception and haptic feedback allow them to make controller actions even if they do not see it. These actions may involve spatial actions in 3D or even button presses. Therefore, we need to design and develop interfaces that would mitigate these asymmetries between the users.

6.6 Improved video tutorial system

Building on the design implications developed before, TutoriVR system seeks to address the problems uncovered in our formative studies. It achieves this in an application-agnostic and platform-agnostic manner through leveraging data available in all the commonly used VR systems.

To generate the necessary metadata for TutoriVR, we provide minimal interfaces that run in the background and records the instructor's VR painting workflow.

- **Screencast video + VR system traces**. An active VR application usually mirrors what the user sees in a desktop window. We developed an executable script that one can use to record a screencast video together with a time-stamped log of the VR system traces. This log includes the HMD and controller 3D positions, orientations, and the controller button press events. This logging mechanism is supported in the two standard VR development frameworks (e.g., Oculus SDK and OpenVR).

- **Author annotations**. The tutorial author can also optionally provide annotations in an XML file to indicate important events or emphasize certain segments in the recording. The important events are of three types - 1.) Critical events that users should pay attention to 2.) Tool/mode changes 3.) Step transitions.

TutoriVR packages this metadata into an embedded tutorial system in the user's VR application environment (Figure 6.5). It has four main components - An improved video player, a Stereo widget, a Perspective thumbnail widget, and an Awareness widget.

Improved baseline video player

Before we focus on the issues stemming from the videos, let's address the improvements we noted earlier for the baseline video player itself. First, the size of the video player needs to be adjustable. A large UI provides more details but may block the user's FOV in VR. A small UI is less obtrusive but is less visible. We let users resize the video player by adjusting a vertical slider (Figure 6.5 V.2). Second, important events and the tutorial's temporal structure should be more explicit. We explicitly visualize author's annotations on the timeline (Figure 6.5). Detailed explanations of these icons are provided in Table

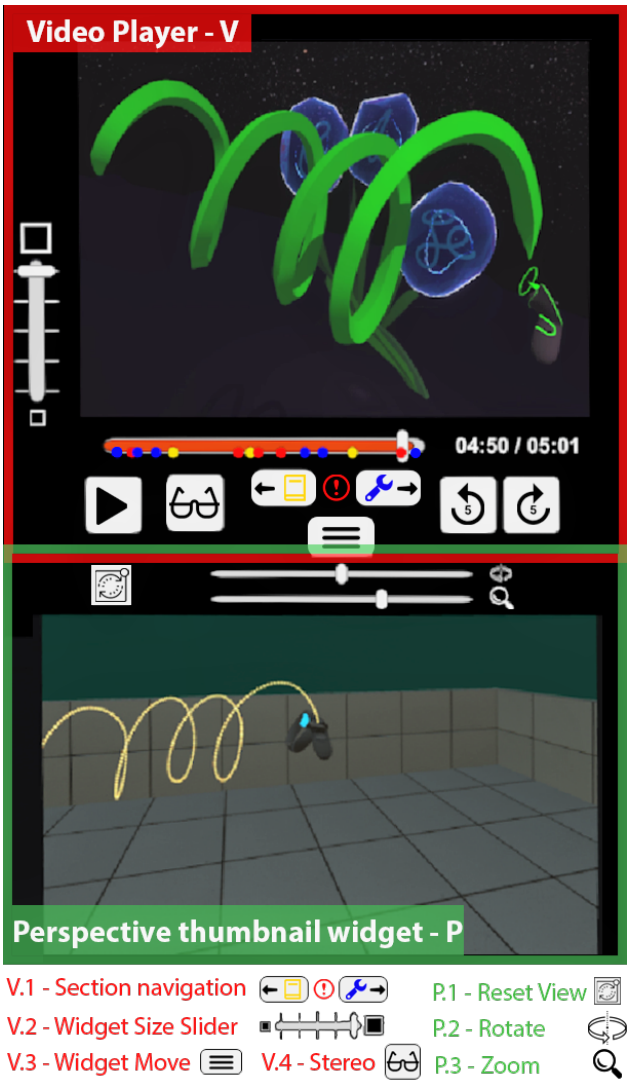


Figure 6.5: The main panel of the TutoriVR system. Up (V): The improved video interface; Down (P): The Perspective thumbnail view, which simulates the actions and interactions of the instructor's controller to help users better understand the video content

6.1. These icons provide navigation cues and allow users to be aware of the important events emphasized by the author [147]. Finally, we added various timeline navigation aids. Seeking a video timeline in VR using the touch controller might be slow and tiring because of hand jitters [125, 24]. We added the -5s/+5s seeking buttons and the section navigation buttons (Figure 6.5 V.1) to aid in video navigation. While these improvements are not novel, they make the interface more usable and would help shed light on problems inherent to understanding the video itself.

Event	Icon	Marker
Tool change		
Critical event		
Step change		

Table 6.1: Tutorial author's annotations, their corresponding icons and timeline markers

Stereo video visualization

As observed in the formative study, a 2D screencast video does not capture the relative depth between the artist's controller and the virtual objects in the scene (Figure 6.6. a). To enhance depth perception/z-dimensional perception of the instructor's *xspace* from the video, TutoriVR allows a learner to view the video in Stereo through a toggle button (Figure 6.5 V.4). Prior works have found stereoscopy to be the most effective depth cue in VR manipulation tasks [86].

To achieve this, TutoriVR's capture application, while recording the instruction, records the feed rendered to each of the eyes of the instructor. Note that this stereo rendering is a strong depth cue for the instructor themselves in their own VR environment. During playback, we use a graphics shader to separate the left and right views from the video image and render them to the viewer's left and right lenses (Figure 6.6. b). The rendering is corrected for the IPD of the learner's HMD and the principle of operation is similar to that of a 3D TV. In this way, TutoriVR implicitly provides access to the depth of the scene without requiring source code access to the original VR application of the instructor.

However, this approach may incur perceptual problems that are well-known for stereoscopy, such as eye straining or window violation [180]. Therefore, we implemented the standard "black-tape" technique to reduce window violation [5]. So, we had this widget inactive by default and expect that the user would only use it when necessary. This stereo-capable video widget is an example of a Cross-Dimensional Medium that allows for dimensional-detail exploration of the instructor's *xspace* and mitigates the asymmetry in depth perception.

Perspective thumbnail widget

Besides depth perception, the second set of issues identified broadly relate to difficulties in understanding the instructor's interactions in the *xspace*. These issues can be further

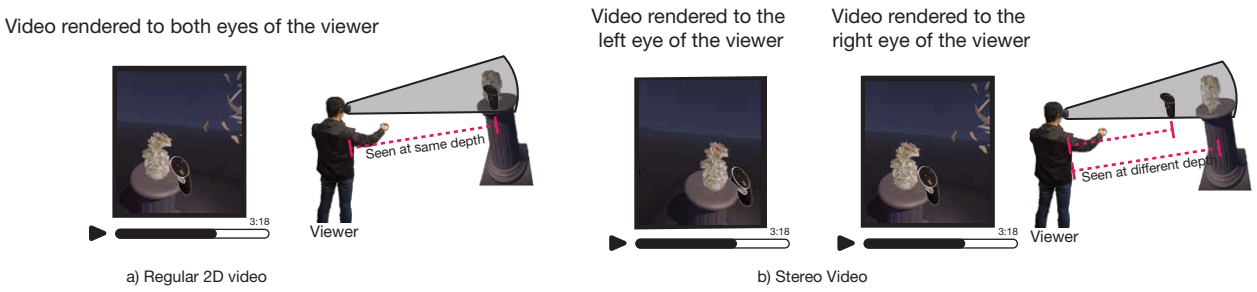


Figure 6.6: a) With a regular video, the same video texture is rendered to both eyes of the viewer b) With a stereo video, different video textures are rendered to each eye. This enhances depth perception in the video and is the principle of operation of TutoriVR's stereo visualization

sub-categorized by the source of difficulties 1) Perspective of the video capture, 2) Complex Hand Motions by the instructor, 3) Rich Input space of VR controllers, and 4) Visibility of Controllers.

To solve these issues, TutoriVR leverages a second Cross-Dimensional Medium delivered through a different embedded widget - the perspective thumbnail widget (Figure 6.5 P). It is a significant part of our system to mitigate the asymmetry between the users in the perception of stroke structure and controller interactions. The widget is placed directly below the video player, providing perspective, temporal, and dimensional-detail exploration of actions carried out by the VR instructor in their *xspace*.

Using the logs generated with our capture-time system, we simulate the movements and interactions of the controller in the video in a separate window below the main video player (Figure 6.5 P). Here, we render motion trails to the virtual controllers whenever a stroke is drawn and highlight button press events in blue (Figure 6.7 right). The thumbnail view mirrors the viewpoint and rotation of the tutorial author in the video. However, the user can manually change rotation and zoom for further inspection (Figure 6.5 P.2 and P.3).

In this way, this widget offers three advantages for better understanding the painting process:

- **Focus-Context exploration of the stroke structure:** The widget allows users to view and explore the stroke creation process from different perspectives, free of occlusion by other elements. The user can switch to different perspectives using the zoom/rotate sliders present in the widget as well as snap back to the instructor's viewpoint. This medium enables seeing an isolated and static visual of an ephemeral, dynamic action. They can then resume playing to see the creation from that chosen perspective. This allows for focus-context exploration of such spatio-temporal actions. Upon pausing the video, the medium provides focused information about the stroke being drawn. It removes other spatial elements in it, as well as draws out the stroke across all time. In contrast, the video player's video provides the context for this stroke.

- **Controller interactions:** The virtual controller mirrors the motions and actions of the instructor's controllers. Whenever the instructor in the tutorial video presses a controller button, the corresponding buttons are highlighted in the virtual controllers. Seeing what is being pressed and the controller's motion in a separate and isolated space can help users better understand the Spatio-temporal relationship of complex controller interactions involving button presses.

- **Stabilized rendering**: The medium also solves the issue of understanding spatial motions from the shaky first-person video feeds. This is because the walls and grid lines rendered in the medium provide important pictorial and perspective depth cues [152]. These cues may not be there in the original VR environment that the VR instructor operates in. The rendering also offers an effectively increased field of view compared to the video. Finally, it stabilizes it by suppressing the noisy and drastic head movements in the first-person video recordings.

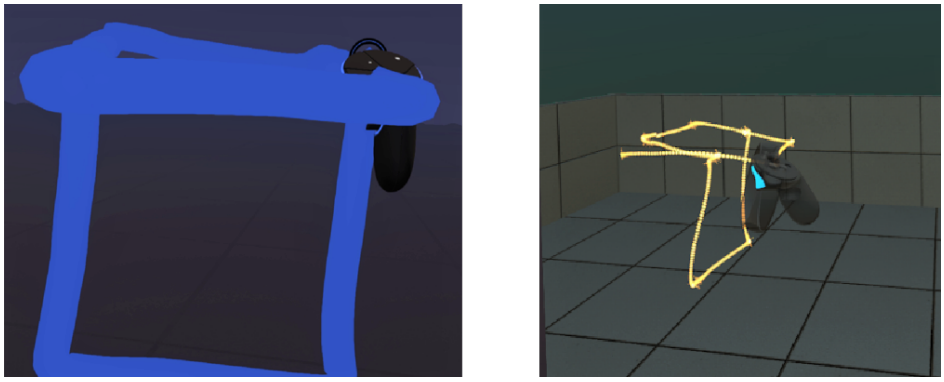


Figure 6.7: Demonstration of the Perspective Thumbnail Widget. Left: Ambiguous cube-like strokes when viewed on a video; Right: The cube strokes become more interpretable when viewed from a different perspective in the Perspective Thumbnail Widget.

It is important to note that these are rendered in stereo 3D to convey a stronger depth perception. This separate set of VR controllers mimics all the spatial and button actions carried out by the VR instructor in the video capture. It is like a fish tank, with walls and 3D strokes being performed as they appear in the video and buttons getting highlighted as they are pressed in the video tutorial. In addition, it offers alternate visual representations that provide additional dimensional details of these stroke motions. These details are absent in its video counterpart.

The medium further increases the effectiveness in delivering these visualizations by gradually fading the highlights of button presses and the stroke trail. This gives enough time for users to be aware of and understand ephemeral controller actions and prevents the need to

perform a frame-by-frame search of the video to figure out crucial information required for a task.

To achieve these, we used the same technique to achieve stereoscopic rendering. However, to create the stereo textures, we create our own controller simulation scene with walls and grid lines and capture it using a stereo camera rig. The rig outputs the left/right textures used for stereoscopic rendering. It can help users more quickly interpret the motion of the virtual controllers. Additionally, we increase the field-of-view of the stereo camera rig to increase the visibility of the controllers. The final rendered effect is similar to how a learner would observe the controller's actions if they are looking over the shoulder of the instructor.

Awareness widget

The VR environment is much larger than conventional desktops, and it is very easy for users to lose the main video player from their Field of View (FOV) and miss important instructions in a tutorial, especially when they are not explicitly voiced out. For example, when a user is too focused on the painting, they might fail to notice an important action made by the instructor. The user can also choose to place the video player close to the painting, but it may obstruct their workflow. To address this problem, TutoriVR provides a mechanism to non-intrusively provide users with visual information on the progress and key events of the tutorial video (Figure 6.8). Our Awareness widget is an extension of the main video player. Instead of being positioned in the world coordinate, it is rendered view-fixed—attached to the user's view direction, and becomes visible when they are not viewing the main video player. The widget is transparent for most parts, and the user can customize its vertical position so that it does not obstruct their active field of view.

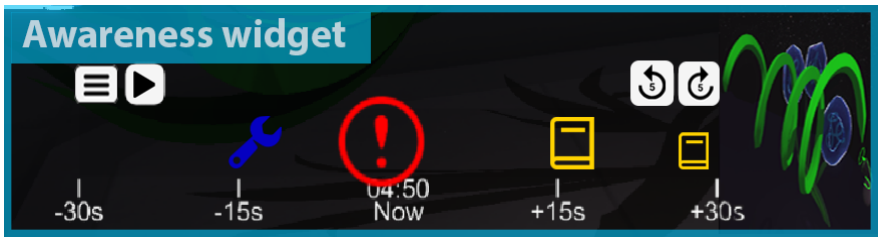


Figure 6.8: Awareness Widget. This widget is view-fixed in VR. It provides a minimal set of important tutorial content to help users be more aware of the tutorial progress.

We render a minimal set of tutorial content on the Awareness widget, including a video thumbnail, current video time, and event icons representing events within a +30s/-30s time window of the video play (Figure 6.8). The icon size increases temporarily as the event approaches the current video time to aid in attracting user attention. Users can also quickly navigate the event by clicking on these icons in this widget. This is similar to how Hologlyph was scaled and used in Loki when the Instructor and Learner work alongside. However, the

Factor	Level	Relevant Physical Space N/A	Relevant Virtual Space Instructor's VR space that includes all components of the VR design
Visual Representation	Same	N/A	
	Limited Representation - Same Output Dimensionality		With TutoriVR - Stereo Video, 3D interactions
	Limited Representation - Reduced Output Dimensionality		
	Video Only		Without TutoriVR (Recorded VR mirror)
	Not Visible		
Interaction Affordance	Same	N/A	
	Limited Representation - Same Input Dimensionality		With TutoriVR - Perspective exploration
	Limited Representation - Reduced Input Dimensionality		
	No Input		Without TutoriVR (Recorded VR mirror)

Figure 6.9: Role of Interactive Cross-Dimensional Media in enhancing asynchronous instruction of VR tasks from 2D videos.

system automatically does it where the Awareness widget is analogous to a miniature version of the Hologlyph.

In summary, the Cross-Dimensional Medium and interactions here highlight the relevant virtual parts of the instructor's *xspace* that are important for the instruction task and provide interactions to facilitate its spatial, temporal, and dimensional-detail exploration. These are summarized in Figure 6.9.

6.7 Exploratory User Study

We evaluated the Cross-Dimensional Media interactions in TutoriVR using an exploratory user study. We wanted to gain initial feedback from users about the improvements we added after the formative study. We also wanted to observe whether users could use these novel interactions to complete VR painting tasks from video tutorials.

We compared TutoriVR with the formative study's improved baseline video player system. Our baseline and TutoriVR conditions included these improvements (i.e., features V.1, V.2, and V.3 in Figure 6.5). We chose a within-subjects study design. We designed two tutorial tasks in VR. These tasks are inspired by the *intense task* condition in a previous tutorial systems research [118]. Specifically, participants were asked to watch a tutorial video in VR and were required to replicate the final results in the video as quickly and as accurately as possible.

Task 1 (2.5 minutes video) introduces the basic features of a VR painting application.

Task 2 (4.25 minutes video) demonstrates a painting workflow, which involves executing a set of complex strokes and embellishments to create an animated painting. To reduce the learning effect, we generated two videos with different concepts but had similar difficulty and length for each task.

Procedure

We recruited 10 participants (2 male, 8 female, age range 19-25) using university and company mailing lists. Participants were compensated with a $50 gift card. 4 participants had regular experience with VR, while all had basic/occasional familiarity. On a scale of 1-5, all but two participants self-rated their art experience as 3 or above and only 2 participants rated their 3D design software experience at 3 or above. The study's total duration was 90 minutes and required participants to use the TiltBrush application in the Oculus Rift VR system. We selected participants with zero or minimal experience with Tilt Brush. We also selected participants that are right-handed and can perceive stereo vision.

Before the study, participants received a 20-minute training on using the VR system and an introduction to the TiltBrush application. The participants then performed Task 1 and Task 2 in order. The order of the systems assigned to a participant was counter-balanced.

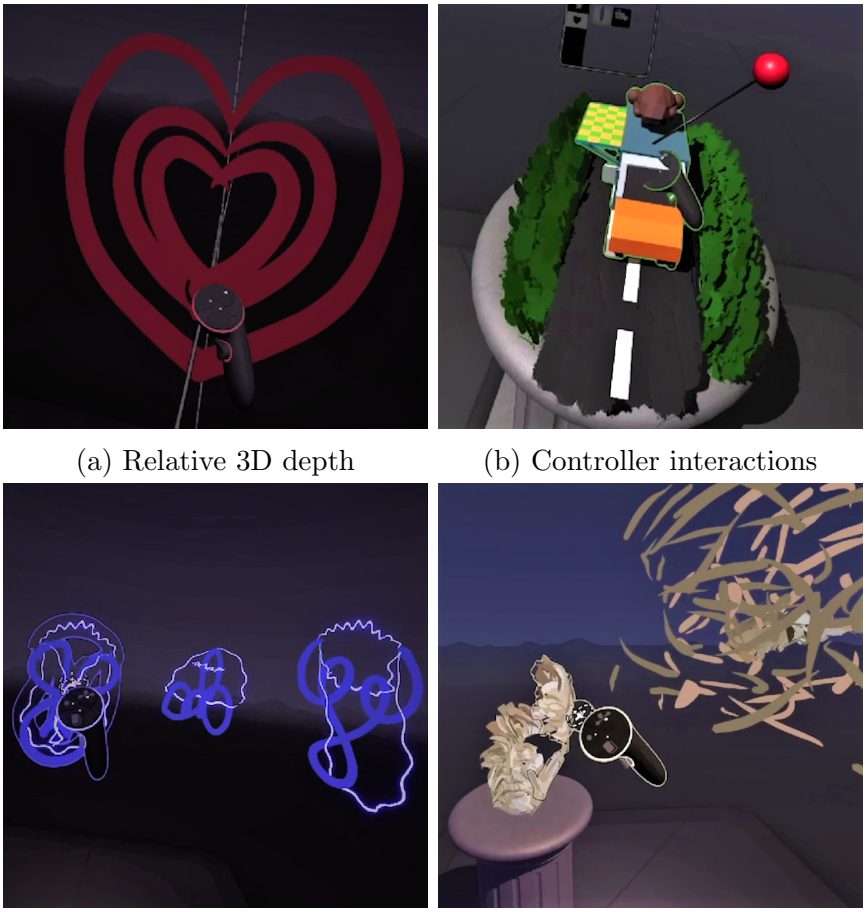
Measurement

At the end of each task, the participants completed a custom questionnaire, and a NASA-TLX questionnaire [76]. The custom questionnaire asked the participants to rate their perception of the tutorial system and task performance. The ratings were based on a 7-point Likert scale, with a higher number indicating easier or more helpful. Additionally, they completed an open-ended feedback questionnaire after each task and the entire study.

In addition to these ratings, we also independently measured the completion rate of each participant in each task. We created a list of critical steps a participant must perform for each task. The critical steps in the tasks were selected ahead of time. They had at least one of the three attributes: the presence of intricate 3D strokes, the component of relative 3D depth, and complex 6-DOF controller interactions. Figure 6.10 shows some examples of critical steps in different tasks. The completion rate is computed by dividing the number of critical steps completed (counted through observation of the participant's performance) by the total number of critical steps.

6.8 Results and Discussion

Figure 6.11 and 6.12 give an overview of the ratings collected after the study. Overall, participants achieved more critical steps ($M = 7.78$, $SD = 3.16$) when using our system compared to the baseline system ($M = 4.88$, $SD = 3.31$). The difference was statistically significant using a paired-samples t-test ($p < 0.02$, $F = 1.1023$, $d = 0.9$). Two participants



(a) Relative 3D depth (b) Controller interactions

(c) Intricate 3D strokes (d) Relative 3D depth

Figure 6.10: Examples of critical steps in the study tasks: (a) Drawing hearts that are on different planes; (b) An action performed with the obstructed controller; (c) Intricate strokes involving 3D loops and coils; (d) assembling a 3D face portrait

failed the tasks (i.e., did not complete any critical steps) using the baseline system. With TutoriVR, on average, across the tasks, users were able to complete 49.2% of steps involving intricate 3D strokes, 55.4% involving relative 3D depth, and 63.8% of 6-DOF controller interactions, in comparison to the baseline where the corresponding numbers were 28.3%, 31.7%, and 41.3% respectively.

In terms of mental workload, there was a small difference between the mean rating of the NASA-TLX. The rating for our system (M = 52.37, SD = 14.30) was marginally higher than the baseline (M = 51.37 , SD = 18.24). The difference was not statistically significant (p = 0.82 , F = 1.62 , d = 0.06). We observed that two opposing effects at work might lead to similar loads. On the one hand, the baseline may have reduced cognitive load due to a lack of awareness of the 3D intricacies involved in the VR painting, causing users to create simpler but incorrect paintings. This is in line with the reported study outcome: users achieved more critical steps in the TutoriVR condition relative to baseline. The TutoriVR system helps users be aware of fine details in the painting. This awareness may demand users to put in extra effort during the tasks.

On the other hand, cognitive load in the TutoriVR conditions may be lower during controller interaction events. Although users in both conditions can notice the presence of controller events on the video, it is much more difficult to follow them in the baseline. Figuring this out is achieved easily in TutoriVR by using the Perspective Thumbnail widget.

In the questionnaire ratings (Figure 6.12), participants were positive about both our system and the baseline system. The differences in ratings, however, were not statistically significant. Qualitative evidence for the ability of the system and its features to help users better understand the tutorial was elicited in the open-ended feedback collected from the participants. Out of 10 participants, 8 users felt the Stereo Widget and the Perspective Thumbnail widgets helped them in the tasks. The awareness widget was designed to be used in free-form exploration tasks with longer task duration, requiring a different kind of study for evaluation. Stereo Widget helped users get better task awareness and assess relative 3D poses of the painting structure. Perspective Thumbnail helped users understand intricate 3D shapes and/or controller interactions. This was also reflected in the written comments by some users:

P2: *"[Stereo widget] was very helpful for understanding the shape and depth of different elements in the video"*

P8: *"[Perspective thumbnail] would have been helpful in this task. Hard to make out the required shape of the loops without the ability to rotate video to see from different perspectives"*

In summary, our study confirms the value of having an embedded tutorial system that leverages the Cross-Dimensional Medium and interactions. Furthermore, the findings show that our improvements were perceived positively by participants. Most importantly, the improvements helped participants replicate more critical steps in the VR tutorial tasks. Since these tasks were designed to be non-trivial and our participants mainly consisted of inexperienced VR painters, these findings provide promising evidence that TutoriVR can aid users in learning to VR paint while being in the VR environment of the painting application.

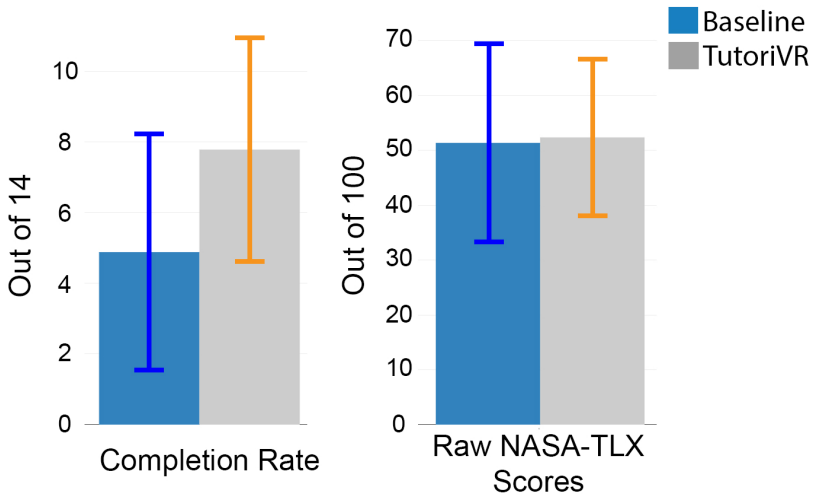


Figure 6.11: Quantitative Scores and NASA-TLX scores

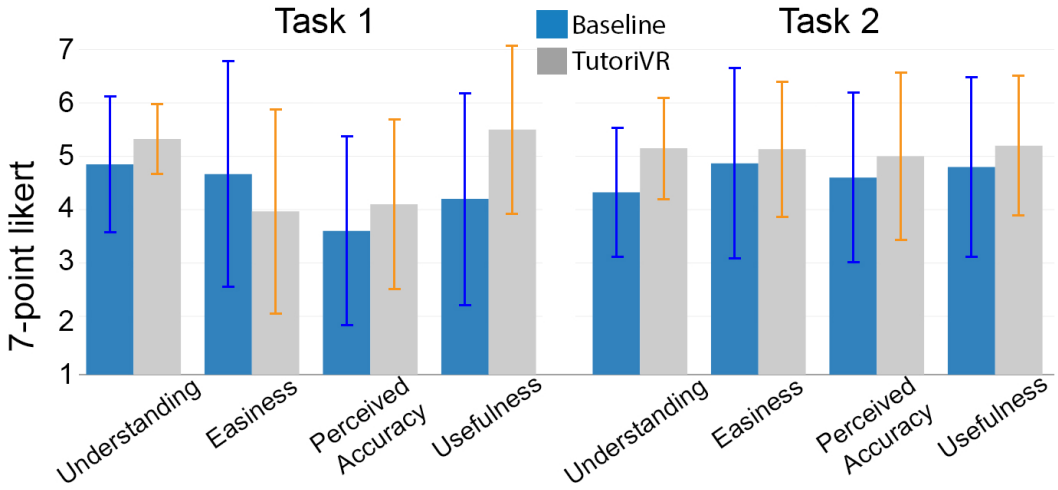


Figure 6.12: Responses to questionnaire questions: *1) Understanding*: "How easy it to understand the tutorial instructions?", *2) Easiness*: "How easy it was to perform the assigned task?", *3) Perceived Accuracy*: "How accurate do you perceive your result compared to the video?", *4) Usefulness*: "How useful was the assigned tutorial system?"

6.9 Limitations and future work

Need for formal Ablation studies

Although we found some promising results in our exploratory study, there are several aspects of this new type of tutorial system that we currently cannot answer. First, we do not know

individual features' specific utility and usability. Therefore, more formal comparisons with dedicated tasks are needed. For example, for the Stereo widget, depth judgment tasks [171] are better at gauging its effectiveness in depth perception. For Awareness Widget, open-ended creative tasks where users tend to focus more on the painting than the tutorial might be more suitable. Second, our study was conducted on a small sample size with a specific VR painting application. Therefore, the findings should be taken with a grain of salt. However, as the first step in evaluating this new type of tutorial, the results are encouraging. We hope to motivate more evaluations with more participants and with other applications in VR (e.g., sculpting, animation, storytelling).

Reactive tutorial systems

We also identify several exciting areas for future work. First, our current system is designed for self-paced learning, where the user decides when and where to access the tutorial. More compelling reactive and intelligent tutorials [50] that provide guidance and feedback can be explored.

Extension to synchronous user interaction

Secondly, the VR-embedded widgets can have applications beyond an asynchronous tutorial interaction between a learner and an instructor. Being embedded into a VR application opens up new possibilities for collaboration and multi-tasking interactions in VR. One example is allowing live interaction by a non-VR user to communicate better and guide a VR user. In this dissertation, we explore this aspect further in the subsequent Chapters 7 and 8.

6.10 Summary

In this chapter, we have taken the first step at exploring the challenges in understanding tasks and actions carried out in a virtual space. Our system was designed based on VR-embedded widgets, which are application interfaces that are observable and interact-able directly inside the learner's own VR application. It allows a learner to learn to VR paint while watching tutorial videos without ever leaving the application. These widgets leveraged Interactive Cross-Dimensional Media to mitigate the *xspace* asymmetry between the learner and the instructor that is typically present in pre-recorded video-based tutorials.

Chapter 7

TransceiVR

In the previous chapter, we saw the challenges of using video-based tutorials for asynchronous interaction between a learner and an instructor. We address them with VR-embedded widgets that leverage Cross-Dimensional Media interactions and properties. Using that, TutoriVR enhanced a user's understanding of videos of VR scenes that an Instructor pre-recorded. But what if one would want to carry out such interactions live? Is it easily possible? Such interactions are messy and hard today. When we played around with TutoriVR, we realized that the VR-embedded widgets could have applications beyond asynchronous video-based interactions. Its technology stack can be adapted for facilitating and enhancing live interaction, communication, and collaboration between a non-VR user and a VR user. This will be the focus of the current chapter.

7.1 Preamble

Virtual Reality (VR) users often need to work with other users who observe them outside of VR with the help of an external display. Communication between them is difficult; the VR user cannot see the external user's gestures. Likewise, the external user cannot see VR scene elements outside the VR user's view. We conducted formative interviews with experts to understand these live asymmetrical interactions and identify their goals and challenges. We identify high-level system design goals to facilitate live asymmetrical interactions. In this chapter, we present TransceiVR, which uses a novel Cross-Dimensional Medium for enabling live asymmetric communication between a VR user and an external user. Through the medium, TransceiVR allows external users to explore the VR scene spatially or temporally, annotate elements in the VR scene at correct depths, and discuss via a shared static virtual display. Similar to TutoriVR, it utilizes VR platform APIs to enable these asymmetric communication interfaces for third-party applications without requiring source code access. An initial co-located user evaluation with 10 pairs shows that TransceiVR makes asymmetric collaborations in VR more effective and successful in terms of task time, error rate, and perceived workload. An informal evaluation with a remote expert gives additional insight

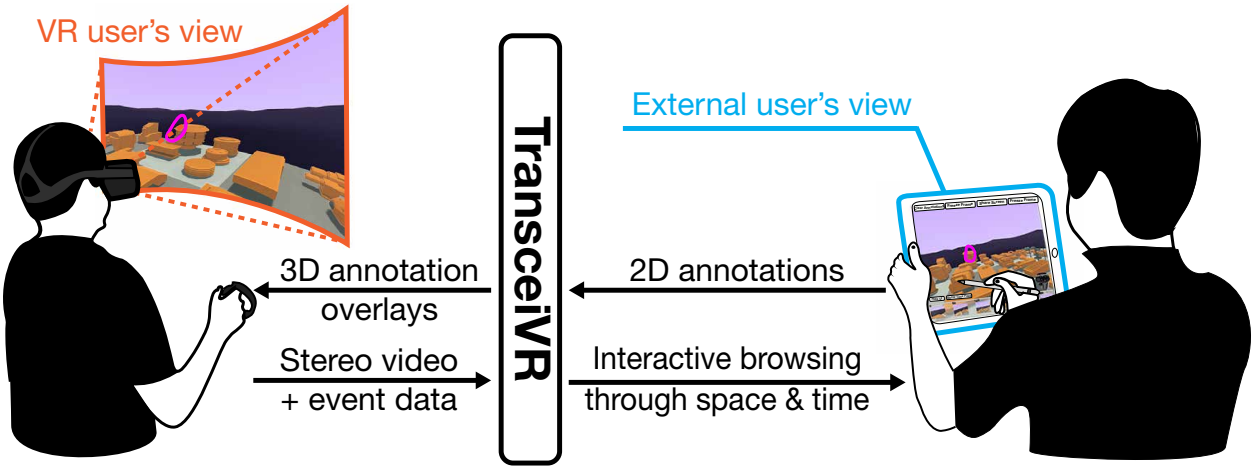


Figure 7.1: Asymmetric VR interaction using TransceiVR: An external user with a tablet can annotate images from the VR user's application; annotations are projected back into VR at the appropriate depth.

into its utility.

7.2 Introduction

Virtual Reality (VR) has gone from research technology to a popular user platform with a growing ecosystem of applications, games, and media. However, the isolation of VR users from their peers is a major problem hindering its adoption today in many domains [67, 69]. While some VR applications are designed to be collaborative, most are conceived as single-user experiences. In line with the focus of this entire dissertation, this chapter focuses on the social interactions that emerge from single-user VR applications. Collaborative use of single-user desktop applications has been well studied, e.g. in "social learning" [124], "occasional meetings" [93], pair programming [37], and game streaming [74, 110, 145, 163]. In many of these, the *xspace* consists of the application state and the participants' speech and body language. These are mutually visible and accessible to everyone, or *symmetric*. For VR applications, similar social interactions can emerge between a person wearing a VR headset and other people being *external* (i.e., not in VR), e.g., a VR artist may want to help teach a new VR user how to paint and sculpt in VR. This communication is *asymmetric* because the VR and external users do not share the same display and input capabilities with respect to the *xspace*. The headset blocks the VR user's view, so they cannot see the external user, and the external user does not have full visibility of the VR scene. This asymmetry creates a communication barrier between them. **In this chapter, unless explicitly specified, any references to asymmetry will refer to this specific asymmetry that arises due to users using different interfaces.**

Some VR systems can mirror the display output to a screen, but such a *VR mirror* only provides a partial and unstable view of the VR user's actions. It is still difficult for the external user to interact with the VR user who is in a 3D virtual space, solely using the *VR mirror*. Another solution is to design VR applications specifically for collaboration, either entirely in VR [53, 117, 131] or through specifically providing interfaces for the external user [69, 68]. However, as mentioned before, single-user applications are unlikely to be rebuilt for multi-user purposes unless there are strong business needs [28].

We conduct formative interviews with experts who regularly participate in such asymmetric interactions with VR users. We distill our findings into design goals that inform the development of software to facilitate live communication between VR users and external users. These high-level design goals focus on enriching the way information regarding the VR environment is delivered to the external user and allowing them to relay information to the VR user.

Developing these design goals into features can require varying levels of access to the underlying VR software and hardware stacks (Figure 2.2). Our formative interviews reveal that asymmetric communication can often occur in closed-source VR applications (e.g., during onboarding or testing a new application). Thus, similar to TutoriVR, we focus our effort on information available at the VR platform level without requiring source code access. These data include the VR stereo camera feed and the VR hardware readings. To render content into the headset, we leverage the ability to inject graphic overlays into VR.

To facilitate asymmetric communication, we propose TransceiVR, a novel Cross Dimensional Medium that aims to bridge this asymmetry of *xspaces* in live communication between a VR user and an external user who may or may not be physically co-located. It mirrors the VR user's feed onto a touchscreen tablet so that the external user can perform interactions directly over the mirrored video feed. A key interaction that the external user can do is annotate the video feed. The medium can then project and place these annotations at the right place in the 3D virtual scene of the VR user. To correctly position the annotations of the external user in the VR scene, we apply an optical flow-based computer vision technique on the mirrored video to estimate a depth map of the VR scene. Alternatively, the medium allows the external user to draw annotations over video frames and share them with the VR user. This can be used like a shared sketch board. Besides interacting with the live video frames that the VR user sees, TransceiVR also supports annotating video frames in the recent temporal and spatial history. This allows the external user to discuss scene elements outside the VR user's current view. TransceiVR also provides an interface for external users to highlight specific buttons on the VR controller. This feature enables fine-grained discussions around the controller, which are common in onboarding activities. Finally, the external user can also share their live web camera feed with the VR user.

We conducted a preliminary evaluation of TransceiVR's Cross-Dimensional Media interactions with 10 co-located user pairs. We found that with these interactions, users completed a VR assembly task faster, with fewer errors, lower perceived workload, and felt it was easier to understand and communicate with their partner. TransceiVR also supports remote operation through any screen sharing and control software. In addition, we carried out a

remote expert evaluation, in which an expert used TransceiVR to teach VR painting to a first-time VR user. From this, we gained additional feedback on the different components of the Cross-Dimensional Medium.

There are two main contributions of this work. First, we report our findings of the problems in live asymmetrical VR interaction through formative interviews and design goals. Second, we present the TransceiVR system that uses Interactive Cross-Dimensional Media to realize these design goals for existing VR applications. A user evaluation of the system shows an improved efficiency in communication in such asymmetric interaction scenarios.

7.3 Formative Interviews

To understand the goals and challenges of asymmetrical VR interactions, we interviewed five expert users: a VR user experience (UX) designer, a VR UX researcher, a graduate student VR researcher, a VR engineer, and a 360° filmmaker. Each had encountered various asymmetrical scenarios in their work, including conducting VR user studies, collecting feedback on new VR user interfaces, testing new VR prototypes, and reviewing 360° films. Besides these primary activities, they have also engaged in other asymmetrical VR interactions during non-work times, such as demoing VR to friends and playing and viewing VR games. The interview started with the experts overviewing their experience with VR. Then they discussed their primary asymmetrical VR activity, using a verbal step-wise walk-through of it. Some used prior video captures or enacted it when appropriate to clarify details. Our questions focused on the goals of the interaction, how it was carried out including its limitations and challenges, the roles of the two users, and their social dynamics. Interviews concluded with open-ended remarks from the experts.

All participants reported using a *VR mirror* as the only way to share display for grounding discussion. *VR mirror* is a standard feature supported in most commercial platforms. It displays what the VR user sees on an external display such as a monitor or a TV screen. However, the interviews identified several problems with this technique supporting asymmetrical VR communication.

A constantly moving first person view

The video feed in *VR mirror* is very jittery, and all our users indicated that following a VR activity this way can be uncomfortable. This is because VR experiences are immersive and often induce frequent, large head motion, leading to unstable video feeds. In VR, this is specifically aggravated due to its reduced field of view, which increases the head movement and the time the VR users take to explore the scene [119].

More issues arise when an external user needs to refer to an object in the VR scene. Since the *VR mirror* shows the VR user's live view, the desired object is visible only when the VR user views it. Otherwise, it will not be visible to the external user. Worse, if the

external users are nearby when discussing a scene element, the VR user may turn towards them, changing views more frequently.

The difficulties of talking about VR scene elements

A typical VR scene contains dynamic events and objects, many of which are alien concepts in the real world (e.g., an interface floating in mid-air). Thus, it is often difficult for the two users to talk about or refer to elements in the VR scene. We identified four unique problems:

1. Transient elements: Many objects in VR may be seen for only a short period. Examples include user interface menus and game objects. The transient nature of these VR elements makes it difficult for the external user to refer to them. For example, an external user may want to instruct a novice VR user on how to choose a new brush style in Tilt Brush [58]. The VR user may make mistakes like pointing at the wrong menus or selecting the wrong brush buttons. Remedying such errors is usually slow: either the VR user has to repeat the actions, or the external user has to describe those actions for the VR user to redo verbally.

2. VR Controller elements: Hand controllers are the primary interaction mechanism in VR. As noted in the previous chapter, naming and design conventions for VR inputs are not yet standardized. For instance, *grip*, *squeeze*, *secondary trigger* and *fist* were four different terms used by the experts in our formative study to refer to the same button in two different VR platforms (Oculus vs. Vive). One may verbally describe the button by its shape and position. But this often fails in VR applications that do not render a virtual depiction of the hand controllers [150]. In such cases, the external user may resort to instructing the VR user on specific finger(s) to hold down.

3. Gestural elements: A common hassle mentioned by experts is when the external user has to instruct gestural movements to the VR user. These are essential in many VR applications [165] and may be used for common spatial actions like painting, aiming, or teleportation. For these, our experts reported using verbal description of something the VR user might have prior knowledge about (e.g., "*move your hand like a sine curve*" or "*picture yourself as Ironman having boosters in your hands and try to push to fly*"). Alternatively, they may hold VR user's hand to physically guide movement. But they remarked that this is sometimes infeasible, since it may break social boundaries and make either of them uncomfortable.

4. Directional & attentional elements: It is common for the external user to direct the VR user's attention toward a particular scene area. Some examples include "*move this box to this position*" or "*pay attention to that region.*" But these directions are not effective because the VR user does not see the external user's pointing gestures. Understanding a peer's activity is a key requirement for effective shared conversation [164]. Our experts reported workarounds based on verbal egocentric references such as "*Turn to your 3'o clock*" or "*look to your left*". But they found it to be cognitively demanding because they have to constantly take the changing perspective of a VR user. They can also give inaccurate, guessed directions to objects that are not immediately visible to them. Remedying these

errors leads to lengthy back and forth conversations between the two users. Elements with distinct shapes, colors, and sizes, can be used as landmarks for guidance but may fail for VR scenes that are too cluttered or bland.

High-level design goals

We distill feedback from the formative interviews into a set of high-level design goals:

DG1 - Static and stable VR view: Enable external users to quickly access, on demand, a static and a stabilized visual of the VR environment. This will allow the external user to have a more detailed and unhurried look at the feed rather than a constantly moving view that is hard to keep track of.

DG2 - Support independent exploration: Allow external users to independently explore the different views of the scene despite where and what the VR user is currently looking at. Additionally, allow external users to independently access the recent past without relying on the VR user to repeat actions.

DG3 - Augment conversation with spatial referencing: Enable external users to quickly and directly [89, 160] refer to scene elements in the VR view. The external user should be able to talk about 3D objects, interfaces, interaction mechanics, and the spatiality of the scene as if they are directly engaging with the VR experience.

DG4 - Support multi-modal context sharing: Provide communication beyond words, such as body language, actions, sketches, and scene contexts that are being observed by either the external or the VR user. These elements would provide a shared context and conversational grounding for both parties. This could also help in elevating the feeling of co-presence amongst the users.

xspace for live interaction between a VR user and a external user

We will begin with identifying the *xspace* for the task. Here users need to interact, discuss and collaborate to complete a VR task. This is a form of synchronous collaboration. From the formative study, we see that *xspace* consists of the virtual scene of the VR application, actions, and gestures of the VR user as well as that of the external user in the context of the virtual space of the VR user.

The asymmetry between the users stems from two sources: (1) While the VR user sees the virtual environment and their actions in 3D and can look around, the external user can not see them in 3D and have no control over the viewport of the video feed. (2) Since the VR user wears a headset, they cannot see anything that the external user does. Whereas it is easy for the VR user to point to and show components of the virtual space to the external user. We need to design and build interactions that can help mitigate these two asymmetries.

7.4 TransceiVR System

Using the design goals (DG) as guidelines, we developed the TransceiVR system that uses novel Cross-Dimensional Media interfaces to mitigate the asymmetry and facilitate efficient communication between a VR user and an external user. An external user can view the scene of the VR user on a tablet device and detach from the *VR mirror* paradigm by freezing any view (DG 1 - stable VR view), browsing previous views, either in space (*Angle frames*) or time (*Time frames*) (DG2 - independent exploration).

Perhaps the most important Cross-Dimensional Media interaction of TransceiVR is the annotations. External users can annotate directly over the video feed using the table interface. The medium here allows users to specify the *output dimensionality* (Table 4.1) of the annotations:

- It can be projected directly and rendered at the correct depth into the 3D VR space (Figure 7.3 L, R), or
- The annotations can be drawn as 2D sketches over the video frame, which can then be shared with the VR user as an annotated image (Figures 7.8 L, R))

Each of these choices has its advantages. The direct 3D projection can be useful for quick and easy referencing of objects in the VR scene (DG 3 - spatial referencing). When the VR user is not looking at the right part of the scene, they are guided to it through 3D arrows and spatial audio cues. On the other hand, the annotated image can be used for more detailed discussion, especially about virtual objects that may be non-existent or dynamic in the VR scene (DG 4 - multi-modal context sharing).

Besides this, external users can also indicate controller button actions that a VR user should take in their interface using a 2D UI that TransceiVR renders on the table. Corresponding buttons are highlighted over the 3D virtual controller model that the VR user holds (DG 3). Finally, the VR user can also choose to see a webcam video feed of the external user (DG 4).

To implement these functions, TransceiVR operates similar to TutoriVR, which runs as an application overlay on the SteamVR platform. Several VR systems such as Oculus, Vive, and Windows Mixed Reality either support SteamVR or offer similar APIs. The platform API includes access to the stereo-view feed of the VR user and the ability to inject 2D overlays and interfaces into the VR scene. The left and right eye feed and head tracking are used to compute a depth map of the scene and corresponding 3D coordinates of the different scene elements (in the frame of reference of the VR user's room). In addition, these are used to compute the position of new 3D overlays, which are rendered using sophisticated VR overlay injection techniques. Figure 7.2 gives an overview of these operations in the TransceiVR system.

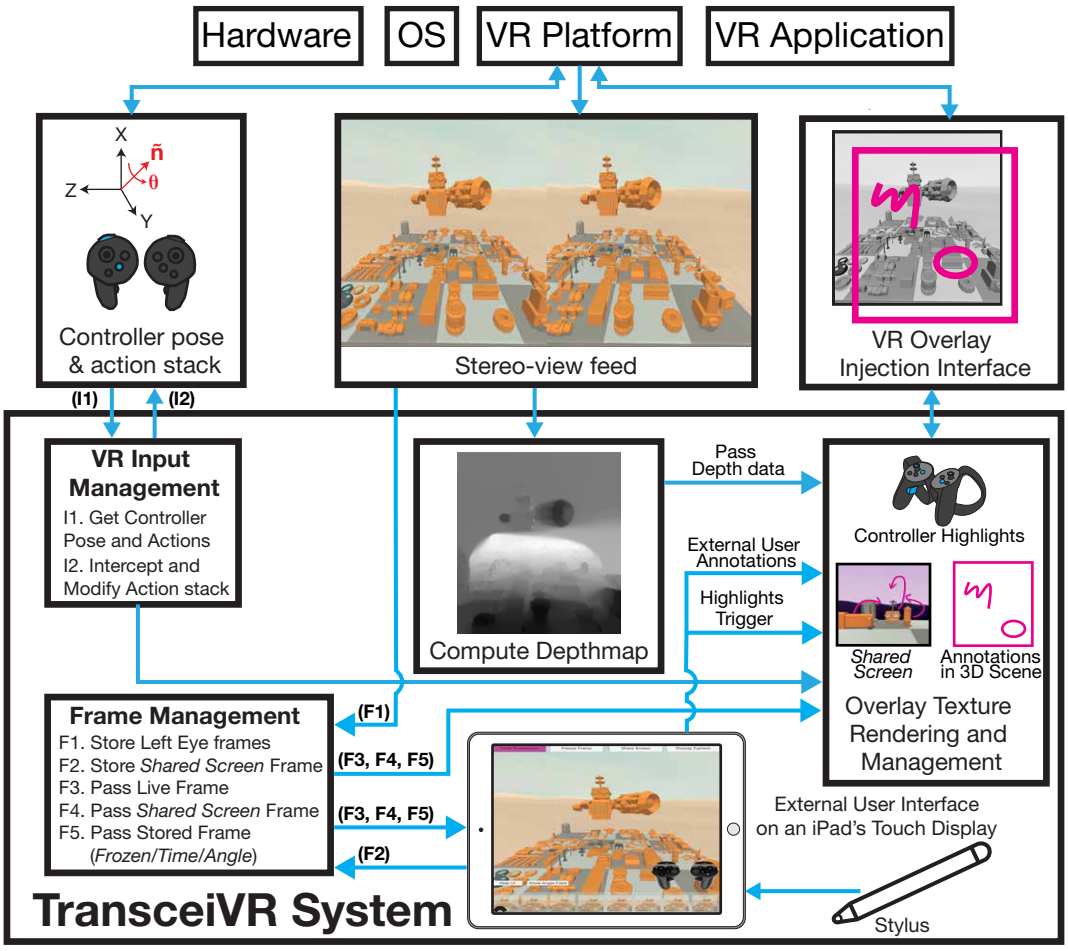


Figure 7.2: TransceiVR intervenes at the VR platform level and assumes access to controller pose, actions, and a stereo view feed. It uses the stereo view to compute a depth map of the scene. VR controller pose and actions are intercepted to enable the VR user's interactions with TransceiVR. The external user uses the tablet interface to annotate, share screens, access frames, and trigger highlights. All these data are used to render corresponding information as overlays in the VR scene through the VR platform's overlay injection pipeline.

External tablet interface

External users browse the VR scene and provide input through a multi-touch tablet interface. The tablet is connected as a multi-touch external display to the PC on which the VR application is being run. Thus, our current implementation is incompatible with standalone VR headsets. The tablet is the primary interface for the external user and accepts input using either a mouse or a stylus. We use an Apple Pen and iPad in our implementation. While we discuss TransceiVR in the context of collocated interactions, the system is equally functional with a remote external user who can access the interface via remote desktop soft-

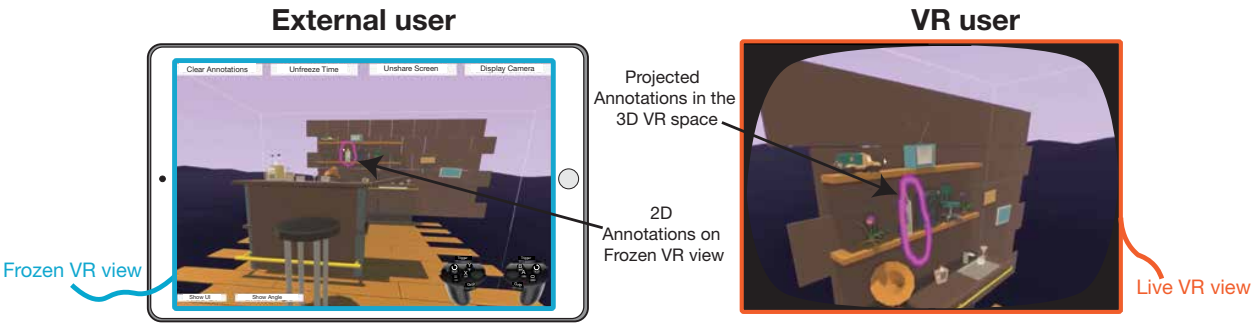


Figure 7.3: Left: Annotations made over a 2D video for referring to an object in the VR space; Right Corresponding annotations projected to the right 3D position in the VR space

ware and a voice channel. In our user evaluation, we use TransceiVR locally, whereas, for the expert evaluation, we used the remote configuration. By default, the tablet interface mirrors the first-person view of the VR users.

Annotations

Our formative interviews indicate that referring to VR scene elements is difficult and is a top problem in such interactions. Often, it requires the external user to use detailed verbal instructions to direct the gaze and attention of the VR user. TransceiVR overcomes this hurdle by allowing the external user to draw annotations over the video feed directly and have TransceiVR project and place them meaningfully in the VR scene (Figure 7.3). The tablet interface detects sketches and renders them in the VR scene at the appropriate depth and orientation, making it look like they are a part of the scene. This leverages findings from prior work [7, 6, 172] that suggest that flat tablet surfaces provide an efficient interface to sketch elements into a VR scene. However, there are issues with this approach: 1) Continuous motion of the VR user's view makes it difficult to sketch over, and 2) The sketches are done on a 2D surface but need to be rendered in the VR scene at the correct 3D position.

TransceiVR solves the first issue by temporarily freezing the mirrored view if the external user begins sketching on the tablet. This allows them to view and annotate a static, *frozen frame*. The second issue is less trivial to solve. In contrast to prior work where such interfaces were part of custom-built applications, TransceiVR is designed to work in an application-agnostic manner with existing VR applications. These applications do not provide access to the depth of elements in the scene. Hence TransceiVR lacks direct access to it. In the previous chapter, with TutoriVR, we simply rendered the stereo-feed as a *3D TV display* in VR scene without computing numerical depth values of points in the scene. In TransceiVR, we explicitly estimate scene depth using binocular disparity and create a real-time depth map of the VR scene. We then use it to render annotations at the correct depth. This is a crucial function of the Cross-Dimensional Medium, as it allows for the property of meaningful

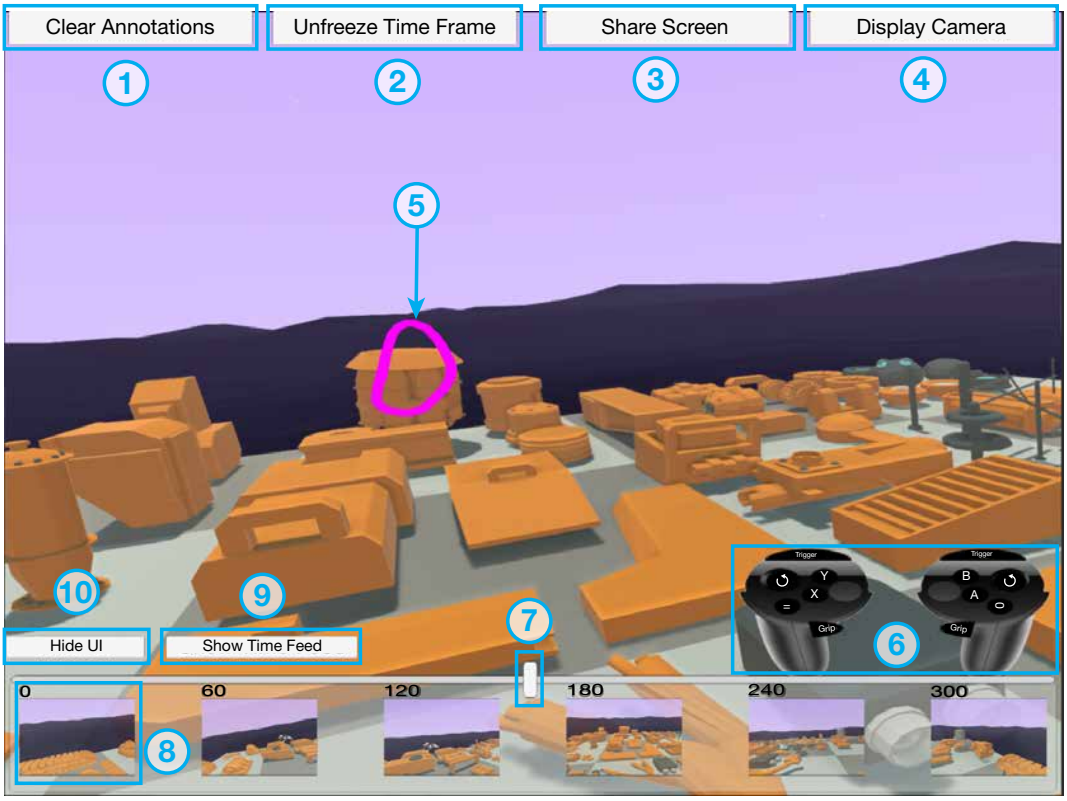


Figure 7.4: The iPad interface's elements; (1)-(4), (9), (10) - UI Buttons; (5) - A user drawn annotation; (6) - *Controller Panel*; (7) - VR user yaw direction indicator; (8) - *Angle frame* at 0° yaw. Note: UI Text emphasized for clarity.

translation across visual representations. It translates the annotation information from a 2D video representation seen by the external user to the 3D virtual world seen by the VR user, thereby mitigating the communication barrier due to the asymmetry in the way the users perceive and interact with the *xspace*.

Depth map generation

From the computer vision literature for stereo images [77], we know that for a stereo camera setup containing two identical cameras that are separated only along the x-axis of the image plane, there exists a relationship between disparity d of two pixels p_1 and p_2 in the two images corresponding to the same world point P, the distance between the two cameras C, the focal length f, and the real-world depth Z of the point P through the following equation.

$$d = \frac{Cf}{Z} = Ap_x + B = \frac{K}{Z} \tag{7.1}$$

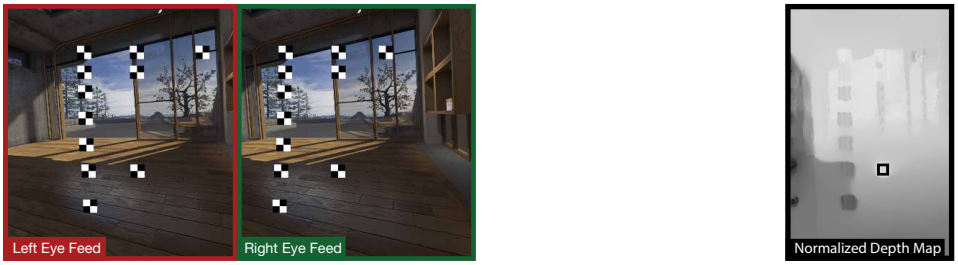


Figure 7.5: (L) The stereo feed of the calibration process being performed in SteamVR Home. (R) The normalized map of the flow vector x-components. The black box shows the region of a single calibration square across which the flow vector values are averaged.

In our prototype system, the Oculus Rift renders with an asymmetric FOV [136], and in such a case, disparity d is a linear function of x-component of optical flow vectors $\mathbf{p}$ computed between left and right eye images (Figure 7.5(R)). We use OpenCV's implementation [139] of the Brox optical flow algorithm [19] that can be executed on a GPU in real-time. This function varies depending on the specific projection matrix for rendering the VR scene on a specific headset. Where $K = Cf$, and A and B are constants that vary depending on the specific VR hardware being used and the resolution of the VR feed being used for computing the optical flow. By applying appropriate transforms Eq. 7.1 can be rewritten with different constants as, Eq. 7.2. Where $p_{x\infty}$ is the x-component of the flow vector for a pixel at distance $Z \to \infty$.

$$Z = 10^{C_1} 10^{-M_1 \log(p_x - p_{x\infty})} \tag{7.2}$$

For TransceiVR to automatically compute the constants, $p_{x\infty}$ and M_1 and C_1 for different hardware, we created a calibration program that overlays a 2×2 grid of black and white squares at various fixed distances and positions in the world as shown in Figure 7.5 (L) and use it to fit Eq. 7.2. We estimate $p_{x\infty}$ as the flow vector's x-component of an overlay placed at 100m which is treated as an infinite distance. Ideally, we should expect $M_1 = -1$ and $C_1 = 1$, but in practice deviations might occur due to the discretization of pixels and other factors. For our setup, parameter values were $M_1 = -0.9968046$, $C_1 = 1.119834$. Note that the optical flow vector values near the left edge of the image are not accurate—these are regions of the left eye's feed that are not seen by the right eye and are extrapolated by the Brox algorithm to obtain the flow vectors. Rendering annotations at correct depth instantly conveys the location of the desired point to the VR user and ensures that the VR user sees the correct location even when they move around and change perspective. If an annotation is made outside the current FOV of the VR user, a 3D direction arrow appears in the scene and guides the VR user's gaze towards the direction of annotation until it appears in their FOV (Figure 7.6(R)). When the annotations are no longer relevant, they can be cleared using a button in the tablet interface (Figure 7.4 (1)).

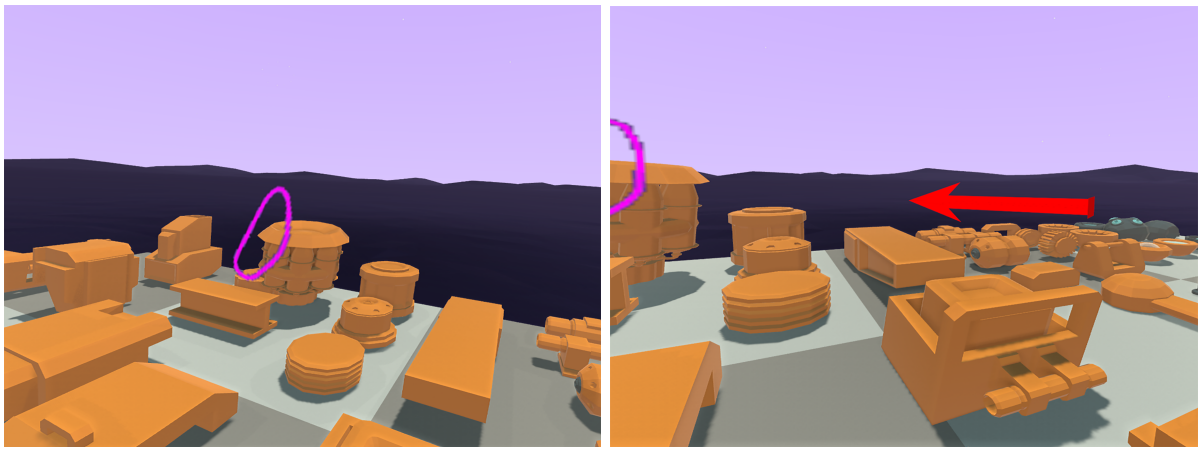


Figure 7.6: (L) A rendered annotation in the VR environment. (R) A 3D directional arrow indicator (red) appears when a newly created annotation is outside the field of view.

Actions with Controller

Sometimes the external user has to instruct the VR user about specific buttons and associated actions on the VR controller. This is crucial and common, especially when an expert external user trains a novice VR user to use an application. This is common in public demos, testing of early-stage prototypes, workplaces, and home environments, where a frequent VR user gets an interested person to try it. Pointing to buttons of VR controllers is a special case of pointing to virtual objects in the VR scene. Still it has some unique characteristics that require additional consideration: controllers are a primary interaction mechanism in VR applications and are almost always in motion. As identified in the previous chapter, they may not be present in the VR user's video feed either because they are outside the field of view or because the application does not render a virtual model of the controllers. In TransceiVR, a *Controller panel* (Figure 7.4 (6)) is used for referring to controller buttons. The external user taps the button(s) of interest in the panel, and colored blinking highlights appear over the location of the controller buttons in the VR scene that visualize these buttons to the VR user. In applications that do not render any virtual controllers, TransceiVR renders a proxy model instead of it. This simple, meaningful translation of interaction from a tap on 2D UI representation of the controllers to 3D highlights on the actual controller is a key property of the Cross-Dimensional Medium here.

Spatial and Temporal Exploration

External users cannot view and refer to virtual objects outside the VR user's current field of view. Additionally, referring to dynamic elements is tough due to their reduced temporal persistence in the VR feed. To aid with these, TransceiVR captures and stores the frames of the VR feed from the recent past (2 mins, one frame per second). External users can then visit

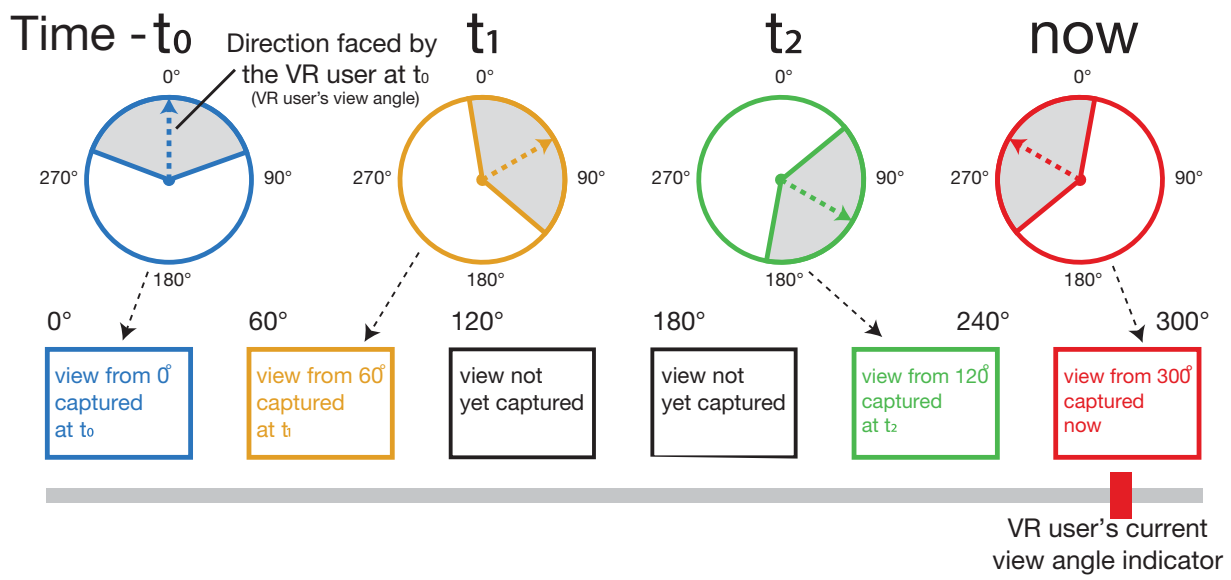


Figure 7.7: *Angle frames* (or view) are captured and updated when the VR user turns their head around and cross certain fixed angles

a prior frame to examine it or to discuss a virtual element that has vanished. We term these frames as *Time frames*. TransceiVR also allows accessing frames based on VR headset's orientation angle (yaw angle), referred to as *Angle frames* (see Figure 7.7). This allows external users to view and interact with last seen frames at specific view angles and allows external users to get a glimpse of the VR environment around the VR user, annotate and refer to objects outside the VR user's FOV. *Angle view* also supports egocentric directional guidance by providing the external user with a direction indicator(see Figure 7.4 (7)). The indicator marks the VR user's current view angle on an angle scale placed above the *Angle frames*. It provides a direct mapping for the external user, who can use this to guide the VR user's relative movements, e.g. "*Turn left/right*", or more precisely as, "*Turn about 30° to your right.*" Users can toggle between the two types of frames (Figure 7.4 (9)). As with any Cross-Dimensional Medium, interactions that allow for spatial and temporal explorations of the *xspace* by the external user are important to mitigate the asymmetry between them and the VR user.

Share screen

The communication between the two users may involve more than directional references and point to objects. Users need to discuss, plan and execute actions in the VR scene. TransceiVR supports this requirement by allowing users to share a static frame of the VR user's feed. Both users can share a copy of a static frame of the VR scene with the other user. The VR user invokes it using the hand controllers (Joystick button), while the external

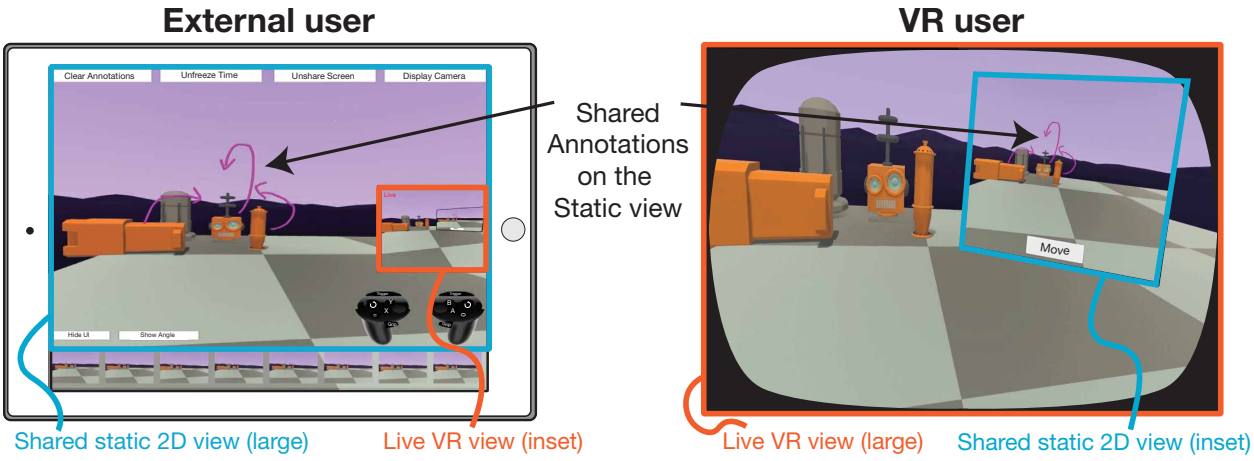


Figure 7.8: Example use of *share screen* to discuss and plan tasks - External user interface having a shared static view with annotations, having an inset of the live VR view (L) and VR user's feed with the movable display of the shared static view with annotations (R). Note: UI Text emphasized for clarity

user can press a button on the tablet interface (Figure 7.4 (3)). When shared, the frame appears on the tablet interface for the external user and as an overlay display in the VR user's scene. The VR user can place it at a convenient location in the scene. The *shared screen* contains the annotations made by the external user and a laser cursor for the VR user. Both users can see these. While a frame is being shared, the live feed of the VR user is also shown as an inset in the tablet interface. The external user can choose to share a static *Frozen frame* from live feed, an *Angle frame* or a *Time frame*. Applications of the *share screen* include discussing actions, complex strokes, or gestures. An example of using this to discuss tasks is in Figure 7.8. As mentioned earlier, the external user's ability to control the dimensional-detail of their annotation interaction is a key property of the Cross-Dimensional Medium.

Viewing the External Environment

Experts have reported that the VR user can lose awareness of the external user and the environment in longer interactions. Prior work such as RealityCheck [79] focuses on this issue from the viewpoint of safety and the ability to interact with the physical environment. We are more concerned with increasing the interaction efficiency and the ability to communicate complex information such as gestures and body language, as well as the need to increase social engagement among the interacting users. TransceiVR allows the VR user to look at the external user through a live camera feed rendered on an overlay display. This is accessed and viewed in a similar manner to the *share screen* feature (see Figure 7.4 (3)). While other prior solutions like overlaying the 3D physical environment in the VR scene can be employed,

these techniques require access to the VR application and/or require sophisticated hardware setups that make them impractical for many users. *Share screen* and camera feed allow for higher fidelity information transfer and can enhance the social engagement between users.

7.5 User Study

We conducted a co-located user study to investigate if the cross-dimensional interactions in TransceiVR can effectively aid users in a scenario where an external user is guiding a VR user with an application designed for single-user usage. We compared the task completion time, error rate, and other subjective metrics in a within-subjects design, comparing TransceiVR to the baseline condition—standard *VR mirror*.

Procedure

We recruited 20 users (8 female, 12 male, age range 18-57 years) in 10 pairs (VR user and external user), using university mailing lists. Three external users had no prior experience with VR, while the rest had used it a few times. In the case of VR users, four had used VR occasionally (a couple of times per month), while the rest used VR regularly (multiple times a week). We used a within-subjects study design to evaluate and counter-balanced the order of conditions and tasks. The study duration was 60 minutes. Each user was compensated with a $25 Amazon gift card. Participants used the *Blocks* [59] 3D-modeling application to perform a robot assembly task, using a pre-existing set of building blocks (see Figure 7.9a). They were asked to select and manipulate them to assemble a given robot design (see Figure 7.9b,c). We sourced over 100 parts for the task from Jarlan Perez's collection for *Bots with Blocks Challenge* [17] to match an existing task using Google Blocks closely.

Participants were given initial 5-minute training on the usage of VR and safety instructions on sharing the physical space, and then a 5-8 minute training on the usage of the *Blocks* application and the TransceiVR interface. Pairs then completed the two robot design tasks, with each task in a different condition. The time limit per task was 12 minutes. The external user guided the VR user to assemble the robot. We provided the user with an assembly manual to position the external user as an expert. Thus, the two users had to communicate actively to complete the task. We added a more challenging sub-task towards the end of the assembly, in which the VR user had to draw a specific foreign character (Tamil language) on the robot. None of our users knew the language, so the character seemed arbitrary to them. Since we knew the language, it was easy to evaluate the drawing qualitatively. This sub-task simulates a scenario where the two users have to work together on something both have no prior expertise on.

Measures

We measured task success, completion time, and error rate (number of incorrect elements in the finished robot). In addition, after each task, participants rated their subjective expe-

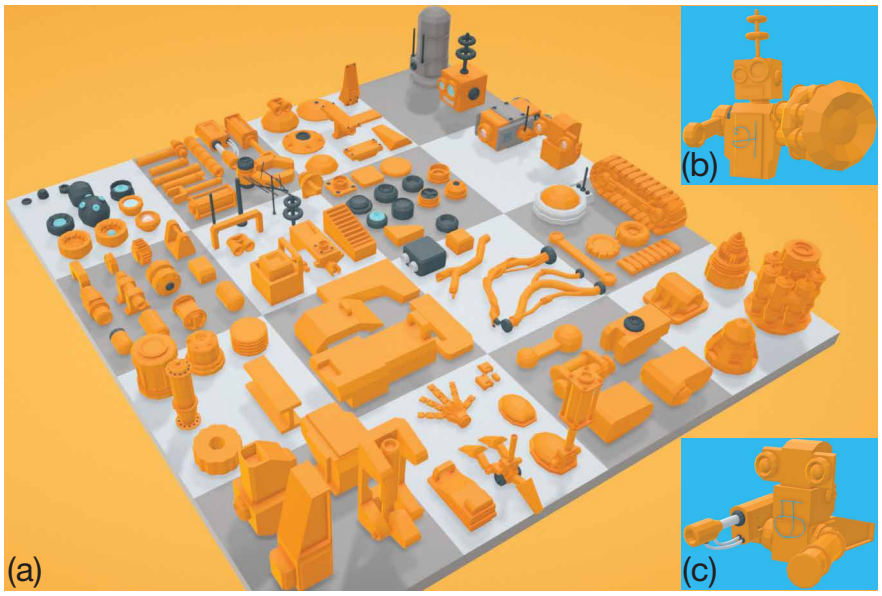


Figure 7.9: a) The set of building blocks provided to users for the tasks; b) and c) Final output of tasks

rience on 5-point Likert scales. The first set of questions asked the user about the ease of understanding (for VR users) or conveying (for external users), the different aspects of communication (referred objects, where to look, conversational grounding, indicated controller buttons, complex strokes) (C1-C5 - see Figure 7.10, top). Then they rated their level of agreement with a set of summary statements. (S1-S5 - see Figure 7.10, bottom).

Participants also responded to the NASA-TLX instrument [76] after each condition that measured their perceived workload and the SUS [18] questionnaire that measured the interface usability. After the entire study, participants were asked to rate the usefulness of the different features of the system in carrying out an efficient conversation with their partners. In the end, they provided open-ended feedback on the experience.

7.6 Results

With TransceiVR pairs outperformed the baseline condition with respect to task success, task completion time, and error rate.

Task Success: When a pair completes all steps of the task correctly within the given time, the task performance is marked as a success. There was a significant difference in task success between TransceiVR (10 of 10 pairs succeeded) and the baseline (only 4 of 10 pairs) ($p<.05$, Fisher's exact test).

Task Completion Time: All pairs of participants took less time to complete tasks with TransceiVR compared to the baseline (incomplete tasks were stopped after the task

	Likert Scale Statements for VR User: It was easy to understand each of the following....
	For External User: It was easy to convey each of the following to the VR user....
C1	Pointing out or referring to objects
C2	Directing where to look or turn
C3	Conversational grounding for actions and instructions
C4	Referring to controller buttons
C5	Communicate complex strokes and symbols
	Common Likert Scale Statements
S1	I was able to carry out efficient communication with my partner
S2	I found it easy to understand my partner's actions/words/instructions
S3	My partner found it easy to understand my actions/words/instructions
S4	The experience was engaging
S5	The experience was fun

Figure 7.10: The set of Likert Scale statements used. Questions S2 and S3 were borrowed from Harms et al.'s questionnaire[75] that measures perceived message understanding in communication.

time limit of 12 mins or 720s and counted as such). There was a significant difference in task completion time between TransceiVR (M=516.8s, SD=108.6061) and the baseline condition(M=678.1s, SD = 71.56) (paired sample t-test t(9)=4.38, p<0.01,d=1.38) - see Figure 7.12.a.

Error Rate: Each task's robot assembly had 7 major parts; we counted the number of incorrect parts in the final output. There were significantly fewer errors in the TransceiVR condition (M=0.5, SD = 0.71) than in the baseline condition (M=1.8,SD=0.79) (paired sample t-test t(9)=4.99, p<0.01,d=1.58) - see Figure 7.12.b. These errors were mostly due to placing parts in incorrect orientations, picking up wrong parts, and incorrect design strokes.

Subjective Ratings: Besides the differences in quantitative data, there were also statistically significant differences in qualitative data. Participant's ratings of perceived workload as measured by NASA-TLX was lower for TransceiVR (external user - M=39, SD = 13.97; VR user- M=25, SD = 12.57) than for the baseline condition for both users (external user - M=52.49, SD = 15.55; VR user- M=37.5, SD = 13.21) (paired sample t-test for external user - t(9) = 3.71, p<0.01, d=1.17; VR user - t(9) = 3.45, p<0.01, d=1.09) see Figure 7.12.c.

The median pooled Likert-scale ratings for questions C1-C5 about the ease of understanding by the VR user of different elements involved in communication were the same in the system condition (Median = 4) compared to the baseline condition (Median = 4). However, a Wilcoxon signed-rank test still shows a significant difference (W = 428, Z = 3.06, p<0.01, r = 0.43) because the distribution skews more positive for TransceiVR. A similar

analysis for the ease of conveying this information for external users has a higher median for our system condition (Median = 4) compared to the baseline condition (Median = 3). The Wilcoxon signed-rank test shows that there is a significant difference (W = 613.5, Z = 4.56, p<0.01, r = 0.64). These findings are further supported by participant's Likert scale responses to their agreement to statements S1-S5 mentioned earlier (see Figure 7.11.b,c).

In the post-study questionnaire, participants rated the usefulness of individual components of TransceiVR. The highest-scoring component were the annotation system (Median=5) followed by the *Share Scree*n (Median=4) and *Controller Panel* (Median=3). Verbal communication that complemented TransceiVR was also rated high (Median=5). Besides being useful, scores of the SUS questionnaire indicated that our system was usable by the VR user (M=75.5, SD=16.7) and the external user (M=69, SD=16.63). Corresponding SUS scores were less for the baseline for both the VR user (M=67, SD=20.13) and the external user (M=54.75, SD=14.6). A significant difference was found only for the external user (paired sample t-test - t(9) = 2.3056, p<0.05, d = 0.73).

7.7 Informal Expert Feedback

To gain additional qualitative insight into how TransceiVR might be used in a more realistic context, we tested TransceiVR with a VR expert in a remote setting and collected informal feedback. The expert is a well-known VR artist whose work is featured in the Tiltbrush application, and leads a prominent VR art group. They taught Tiltbrush to a first-time user of Virtual Reality. The pair were asked to use TransceiVR over video conferencing (*Zoom*), with the expert operating our tablet interface, which has screen sharing, input sharing, and audio. After receiving an initial orientation with TransceiVR, the expert spent 60 minutes teaching the novice both the operation of VR controllers and the basic commands, tools, and elementary 3D painting techniques in Tiltbrush.

Overall, the expert found that TransceiVR helped decrease their dependence on potentially ambiguous and problematic verbal descriptions compared to the status quo *VR mirror*: "*The fact that I had a real technical aid, along with it, I felt less beholden to my words because words can be really problematic, especially when trying to communicate within VR and in terms of instruction.*"

The expert primarily relied on two features: the *controller panel*, and *share screen*. With the *controller panel*, the expert indicated to the novice, which buttons to press on each controller to invoke desired functions. This was essential as the novice had no prior experience with VR controllers. The expert used annotations on *shared screen* as the primary method of instruction. While annotating the VR scene directly seemed exciting to the expert, Tiltbrush starts with a blank canvas and thus has almost no objects in the environment to anchor annotations to. In addition, the expert preferred the instructions to appear alongside the novice in VR rather than directly in the workspace, which might hinder novice's actions.

Finally, the expert also used *Time frames* and static *Frozen Frames*. Many operations in Tiltbrush require users to choose options in a menu. To help the novice with this task, the

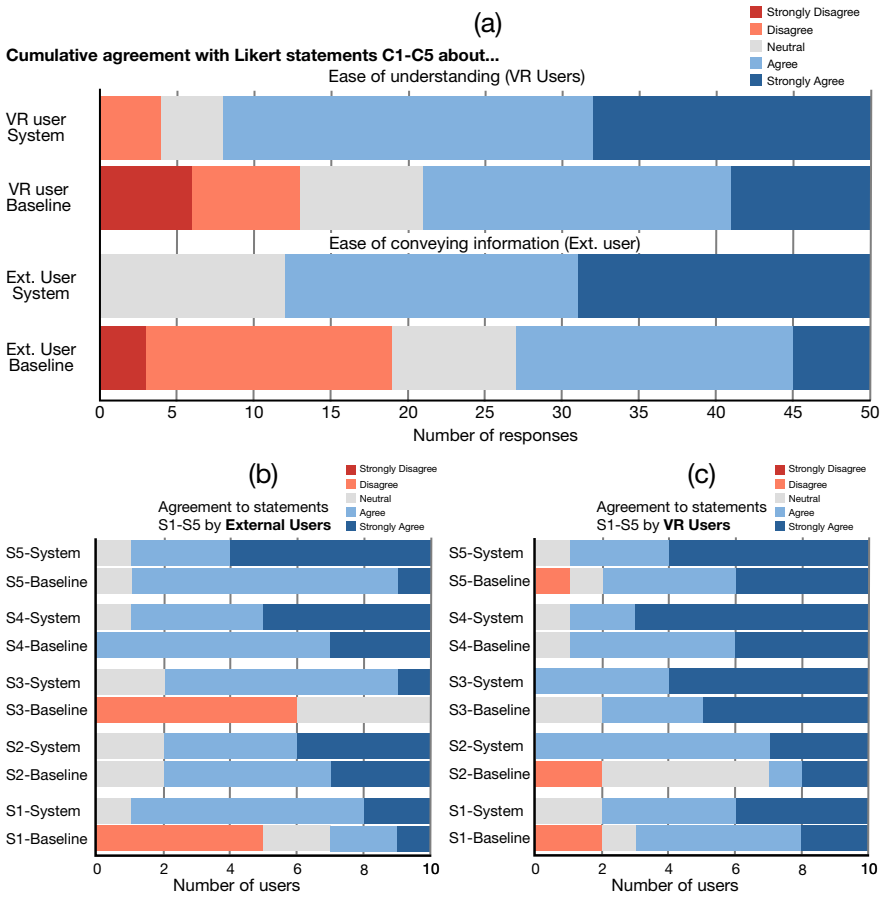


Figure 7.11: Qualitative measures. a) VR user ratings for ease of understanding and external user's rating for ease of conveying the five different elements involved in communication (higher is better) ; C1-5 (Ref Figure 7.10). b) Likert Scale agreement to S1-5 (Ref Figure 7.10) by the external user and c) by VR user (higher is better).

expert frequently selected specific *Time frames* where the menu options were present and screen-shared these frames with the VR user. While *Angle frames* was not used in this task, the expert commented that they could be useful in tasks that require guiding users towards specific spatial locations in a well-structured and populated environment.

Besides the largely positive experience with the system, the expert suggested improvements to TransceiVR, mostly focused on improving the UI that could enhance the user experience - such as better feedback for transition across modes and visual distinction between different feature elements, being able to control timing of controller highlights to show the difference between hold, taps and double taps, and also providing haptic feedback on controllers when a button is referenced on it. However, in summary, they found the offered interactions "crucial". They mentioned that a tool like TransceiVR could open up new avenues of remote tutoring, guidance, and "co-creation" in VR applications.

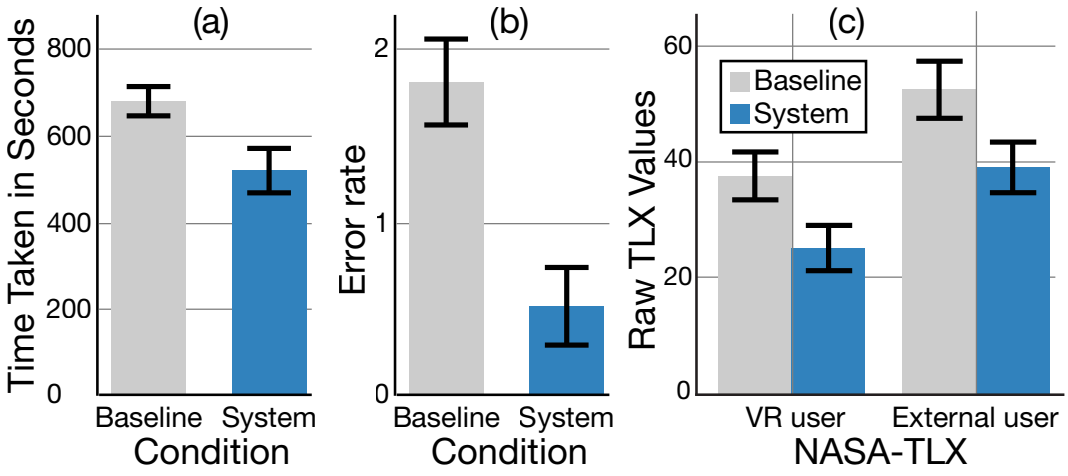


Figure 7.12: Quantitative measures (a,b): a) Time taken to finish, and b) no. of errors in the final output (lower is better); Qualitative measure - c) Raw NASA-TLX for external and VR users (lower is better). Error bars show standard error, $SE = \sigma/\sqrt{n}$.

7.8 Discussion

Overall, the quantitative results from our co-located user study showed that the cross-dimensional interactions of TransceiVR can effectively aid users in an asymmetrical VR-external collaboration scenario. Pairs that used TransceiVR were able to complete tasks faster and with better performance. When asked to rate the overall experience, eighteen out of twenty participants agreed that TransceiVR made communication easy and efficient (Figure 7.11.b,c, S1 and S2). It shows that supporting effective communication was a key factor that led to TransceiVR's success. This finding is important because TransceiVR ran on top of an existing VR application that was designed only for single-user usage. Yet, our participants could easily communicate and collaborate over it. Participants found the experience to be fun and engaging (Figure 7.11.b,c S4 and S5) with lower mental workload (Figure 7.12.c). These findings confirmed the feasibility of our approach and contributed a new use case (e.g., asymmetrical communication) to the growing body of work on retrofitting closed-source VR applications [79, 186, 201].

Looking at individual features of TransceiVR and how they were used in the study, TransceiVR's annotation system received the highest rating from participants. The annotation system as a whole consists of annotating over *Time frames*, *Angle frames* and the *Frozen frame* of a live feed. These annotations made over a 2D video frame are then meaningfully translated and rendered in the 3D VR environment. In the assembly task of the user study, the external user, being the only one aware of how the desired part looks, needed first to locate it in the VR scene. To do that, they used the *Angle Frames* feature and the live video feed to explore the scene on their own. Once they located the desired part, they annotated directly over it. When the annotation was not in the VR user's view, TransceiVR guided

the VR user towards it using the 3D directional arrow (Figure 7.6). Thus, the combination of these tools augmented the external user's ability to understand the VR scene and to convey gestural and directional instructions. These interactions added benefits and facilitated more effective communication compared to the baseline system. Without TransceiVR, users reported it was challenging to observe the shaky VR feed and resorted to verbal communication, which often created confusion between them. We did not see any effect of the external user's local audio on user study task performance. This might be due to the goal-oriented and time-constrained nature of the assembly task. Also, the user study and the expert evaluation task did not entail using an external user's camera feed.

The next highly rated feature of the user study is the *Share Screen*. We designed this feature to complement the annotation system in situations where placing annotations directly on the environment of the VR user might be too distracting or create unnecessary clutters. With *Share Screen*, the external user can send a static image of the VR scene with annotations on that image. The static form of this feature allowed the VR user to place it at a convenient spot for their own reference as they continue with their own work. We observed heavy usage of *Shared Screen* during the foreign language character drawing step of the co-located user study since directly annotating in the environment could get in the way of the VR user's drawing as well as its different parts could get projected at different places depending upon depth.

Another interesting use of TransceiVR was to combine *Shared Screen* and *Time Frames*. We observed this user pattern in the remote expert evaluation. In contrast to Blocks, the Tiltbrush environment of the expert evaluation was rather bland and did not have many scene landmarks for the in-environment annotations. Moreover, the VR painting lesson required the novice user to interact with highly dynamic and transient interface elements like menus, buttons, and strokes. In such scenarios, *Time Frames* complemented the use of *Shared Screen* because it can "freeze" these aforementioned dynamic elements. And then, the expert user can use *Shared Screen* to communicate instructions to the novice user. As a result, the expert frequently used these two features to mark menu options and on 3D strokes made by the novice user. The former helped the novice trigger the right tool. The latter helped improve the novice's stroke quality and techniques. Users learned on their own this unique combination of leveraging temporal exploration and dimensional-detail specification of annotations.

The user study's least used and rated feature was the *Controller panel*. This is because, during the user study's onboarding stage, the VR users received a controller orientation to use the *Blocks* application. However, in the expert evaluation, we observed heavy usage of this feature by the remote expert because the painting lesson required the novice user to learn a lot of different Tiltbrush menus, buttons, and advanced shortcuts throughout the lesson.

We see that the different cross-dimensional interactions of TransceiVR have varying trade-offs. In our experience performing these evaluations and using TransceiVR ourselves, we realized that no feature by itself is sufficient. Instead, a combination of features complementing each other in function is required. This combination varies based on the characteristics of

Factor	Level	Relevant Physical Space N/A	Relevant Virtual Space VR user's space that includes all virtual elements in it
Visual Representation	Same	N/A	
	Limited Representation - Same Output Dimensionality		
	Limited Representation - Reduced Output Dimensionality		With TransceiVR - Spatial exploration through choice of Viewports
	Video Only		Without TransceiVR (VR Mirror)
	Not Visible		
Interaction Affordance	Same	N/A	
	Limited Representation - Same Input Dimensionality		With TransceiVR - Annotate and project to VR space, share 2D annotations
	Limited Representation - Reduced Input Dimensionality		
	No Input		Without TransceiVR

Figure 7.13: Role of Interactive Cross-Dimensional Media in facilitating synchronous collaboration between a non-VR user operating a 2D UI and a VR user

the VR scene and the type of information that needs to be communicated in it.

In summary, during a live interaction, the Cross-Dimensional Medium of TransceiVR allows an external user to better interact with the VR user regarding the elements of the virtual environment. It does this by meaningfully translating the external user's interactions across dimensions, from a 2D tablet screen space to the 3D virtual space of the VR user. Users also used different combinations of interactions such as dimensional-detail specification, perspective exploration (using *Angle frames*) and temporal exploration (using *Time frames*) to their advantage. These benefits are summarized in Figure 7.13.

7.9 Limitations and Future Work

Overlays were planar

A key interaction in TransceiVR is the annotation system. Due to the current SteamVR injection pipeline limitations, only 2D overlays (placed at the correct 3D pose) are supported. This prevented real-time rendering of annotations with 3D shapes into the VR scene. Moreover, TransceiVR does not track when users in the VR scene teleport or change its scaling since these are application-level changes that are not passed on to the VR platform. TransceiVR's annotations are rendered according to the system-level world space coordinates, and these go out of place when such interactions occur. To better handle such situations and increase the richness of annotations, we recommend that VR platform APIs allow for the rendering of dynamic 3D content, and provide access to information regarding

the scale and pose of the main camera rendering the VR scene. In the subsequent Chapter 8, we overcame this problem and discussed how we could achieve the rendering of dynamic 3D content without access to the VR application's source code.

Technological challenges for networking and achieving scalable interactions

The experience of using TransceiVR remotely also raises several questions on how systems should tailor functionality to support remote asymmetric interaction specifically. We uncovered several avenues for future work by conducting several additional remote sessions in which an author assumed the role of an expert in VR, communicating with remote novices outside of VR. Firstly, tools will have to intelligently make tradeoffs between latency, image resolution, and framerate to work in realistic settings. We focus on this networking aspect and study this in the subsequent chapter 8. Besides this, we have not yet investigated the role of other modalities such as spatial audio. For example, audio localization could help external users make statements such as "*Turn towards me*". Finally, we have thus far only focused on dyadic interactions, but remote access enables larger groups of participants. This introduces new dynamics and challenges of control sharing across the different users.

7.10 Summary

In this paper, we identified challenges and goals of asymmetric interactions between VR users and external collaborators. We formulated design goals for systems focused on improving such interactions and presented a design space of implementation approaches. We chose a common scenario in the space and built TransceiVR, a system that leverages Interactive Cross-Dimensional Media to mitigate communication barriers between VR and external users. We believe that important asymmetric VR tasks such as on-boarding, instruction, guidance, and co-creation in VR can all be effectively supported by TransceiVR. While our study focused on the co-located user, we also tested these interactions remotely. Our initial experiences with remote asymmetric VR guidance highlight important areas for future work. We hope our findings can inspire future extensions to multi-user settings and richer multi-modal interactions.

Chapter 8

DreamStream

In Chapter 6, we saw how a Cross-Dimensional Medium consisting of stereo renderings and 3D stroke reconstructions enhanced the 3D understanding of a virtual environment from a recorded video by another user. Then in Chapter 7, with TransceiVR, we saw how another Cross-Dimensional Medium could allow for live interaction between an external user and a VR user regarding elements of a VR scene. It does this through a meaningful translation of interactions across 2D and 3D visual representations together with the dimensional-detail specification, spatial and temporal exploration. However, neither of these systems allows another user to be "embodied" in the virtual scene of the VR user and feel like they were there in the scene with them. While these systems help accomplish certain learning, guidance, and collaboration tasks, they do not necessarily provide a richer spectating experience. Loki system discussed in Chapter 5, achieves these for physical environments. However, what if we want interactions and user experience similar to Loki but for virtual environments? We already saw numerous limitations and challenges in these previous chapters. First, it is hard to obtain a true 3D depth of a VR scene that can allow reconstruction. Second, it's hard to achieve streaming of depth and image data over the internet, allowing for low-latency interactions required for synchronous multi-user interactions. Third, it's hard to render 3D objects in VR scenes that respect the depth buffer of the scene rather than just overlay them. In this chapter, we focus on solving these issues that will allow us to have a richer and more interactive spectating for virtual environments, similar to that offered by the Cross-Dimensional Medium in Loki. However, since we are trying to achieve this for virtual environments, we will motivate the design of interactions from the challenges we uncovered in the formative studies in Chapter 6 and 7.

8.1 Preamble

Today spectating and streaming virtual reality (VR) activities typically involve spectators viewing a 2D stream of the VR user's view. Streaming 2D videos of the gameplay are popular and well-supported by platforms such as Twitch. However, the generic streaming of

full 3D visual representations is less explored. Thus, while the VR player's experience may be fully immersive, today's spectators are limited to 2D videos. This asymmetry lessens the overall experience for spectators, who may be eager to spectate in VR. DreamStream uses a Cross-Dimensional Medium that puts viewers in a 3D reconstruction of the virtual environment of the VR application. This allows them to be embodied in the experience and look "over the shoulder" of the VR player. Spectators can view streamed VR content immersively in 3D using a VR headset. They can independently explore the VR scene beyond what the VR player sees and ultimately cohabit the virtual environment alongside the VR player. Spectators can also choose to use a 2D screen-based desktop interface, allowing all the interactions possible through the VR headset. The spectator interactions using these interfaces are meaningfully translated to the scene of the VR player. In addition, DreamStream provides a spatial awareness of all its spectators for the VR player. Like other systems in this dissertation, DreamStream retrofits and works with existing VR applications. We discuss the design and implementation of DreamStream and conduct three qualitative informal evaluations. These evaluations shed light on the strengths and weaknesses of the Cross-Dimensional Media interactions in DreamStream for interactive spectating. Our participants found that DreamStream's VR viewer interface offered increased immersion and made communicating and interacting with the VR player easier.

8.2 Introduction

Virtual Reality (VR) is increasingly popular, with gaming, collaboration, design, and media consumption applications. As VR use increases, a growing population of VR users shares their VR activity with multiple spectators through live media streams. This is popularly done for entertainment, instruction, and guidance. Such spectating practices can influence the overall VR user experience, as they have done for gaming before VR [163, 106]. New social dynamics emerge from such spectating practices, at different levels, including one-one, small groups, and crowds at scale. Prior work [106, 178, 167, 29] has studied the varying motives of the spectators, as well as the roles they take in these different settings. The role of a spectator can range from someone who observes the activity to someone who actively engages with other spectators and the VR player itself.

Present-day systems that enable live streaming of VR activity stream live 2D video feeds of the VR player's first-person view. This has the advantage of codecs, platforms, and user bases firmly established in today's live streaming practices. However, as we saw in the previous chapters, streaming 2D video of VR content introduces an asymmetry between the VR player and the spectators. The VR player can access an immersive and stereo view while the spectators view a 2D video on a conventional display. In the previous chapters, we discussed issues using 2D videos to spectate and interact with a VR player. Spectating VR content via a 2D display is perhaps expected, given that few people today own sophisticated VR hardware. However, as VR becomes more accessible and more mainstream, we expect there will be more demand to spectate VR in VR. For someone with a VR headset, a VR

Figure 8.1: DreamStream allows spectators to view live 3D reconstructions of a VR game either through VR or a 2D Desktop interface. A VR spectator (represented by a green avatar) watches the live stream of a VR player (represented by a red avatar) who is playing Skyrim. Besides seeing what the VR player sees, the spectator can also see the ambient parts of the scene from a perspective of their choice. DreamStream also composites a live 3D reconstruction of the player.

activity's vanilla 2D spectating experience may be disappointing and even puzzling.

Some VR applications include support for multiple users by adopting a *multiplayer approach* that was discussed in section 2.3. In these, the spectators would be located within the virtual environment of the VR activity and would have the ability to look around them freely. They may even be able to move independently in the virtual space of the VR player. These experiences are usually multiplayer VR games and live VR social spaces. In these, the spectator must download an entire application onto their system. This contains all the 3D assets of the virtual environment and spectator interactions. Thus, during run-time, spectators and the VR player cohabit in a shared virtual space, and such a system facilitates a range of interactions between them. This is similar to how multiplayer games work today. As we discussed earlier in section 2.3, the drawback of this *multiplayer approach* is that it requires every spectator to own a full copy of the player's VR application. This can be problematic, as the game be large in size, require installation time, and may pose

specific hardware requirements. "Drop-in" access typical of live streaming platforms is difficult or impossible. Furthermore, the VR application must be developed to support such multi-spectator interactions.

This chapter discusses design considerations when developing systems for spectating a VR activity. The proposed DreamStream system leverages a novel Cross-Dimensional Medium that allows sharing of VR activity in existing commercial VR applications. It retrofits these applications using a novel approach discussed in Section 8.3. This approach partly mimics the live *streaming approach*, in which the virtual assets are present only at the VR player's computer. DreamStream streams custom video textures (*2.5D frames*) that comprise color and depth components in real-time. However, DreamStream uses these custom textures at the spectator's end to create and render a live 3D reconstruction of the virtual scene. Spectators can now view and interact with the streamed content immersively using a VR HMD or an interactive desktop-based 2D UI. In addition to spectating only what the VR player sees, a spectator can also explore the scene independently and view regions outside the VR player's field of view. DreamStream's approach decouples the spectators' view from that of the VR player and thus enables user interactions that are typical of the *multiplayer approach*. For instance, to visualize the presence of one another in the space, spectators, as well as the VR player, are rendered as virtual avatars in the scene. Optionally, DreamStream can render a live 3D capture of the VR player's body in addition to their abstract avatar. Such a digital representation of users allows spectators and the VR player to cohabit and feel co-present in the 3D virtual environment of the application. Thus, in summary, the *DreamStream approach* facilitates a user experience and interface style similar to ones found in a *multiplayer approach* but uses the scalable technological pipeline of the *streaming approach*, allowing for "drop-in" access. With this combination approach, the Cross-Dimensional Medium of DreamStream allows for increased perspective and dimensional-detail exploration. Also, it satisfies Design goals DG1 and DG2 that we synthesized in section 7.3.

Like other systems in this dissertation, a key feature of DreamStream is that it uses interventions made at the *VR platform level*, enabling it to work with existing commercial VR applications without source code modification. Advantages of this is discussed in detail in Section 2.2. We tested DreamStream with seven popular VR applications - Skyrim, TiltBrush, Blocks, TrainerVR, BeatSaber, Waltz of the Wizard, and Fallout 4 (Figure 8.2).

We conducted three informal qualitative evaluations: (1) A qualitative evaluation of the Cross-Dimensional interactions of DreamStream with four professional VR streamers (experts), (2) An initial user evaluation with eight participants playing the role of spectator, comparing DreamStream with a standard VR mirror and (3) A final user evaluation with twelve participants in which DreamStream was compared with the TransceiVR system that was discussed in the previous chapter. Study results highlight how these interfaces impact a spectator's ability to experience, understand, interact with and enjoy a VR stream. Together, they show that DreamStream's Cross-Dimensional Media interactions show promise in enabling better VR spectating experiences.

Our main contribution is the DreamStream's novel streaming approach and the DreamStream's Cross-Dimensional Media that carries out interventions at the *VR platform level*,

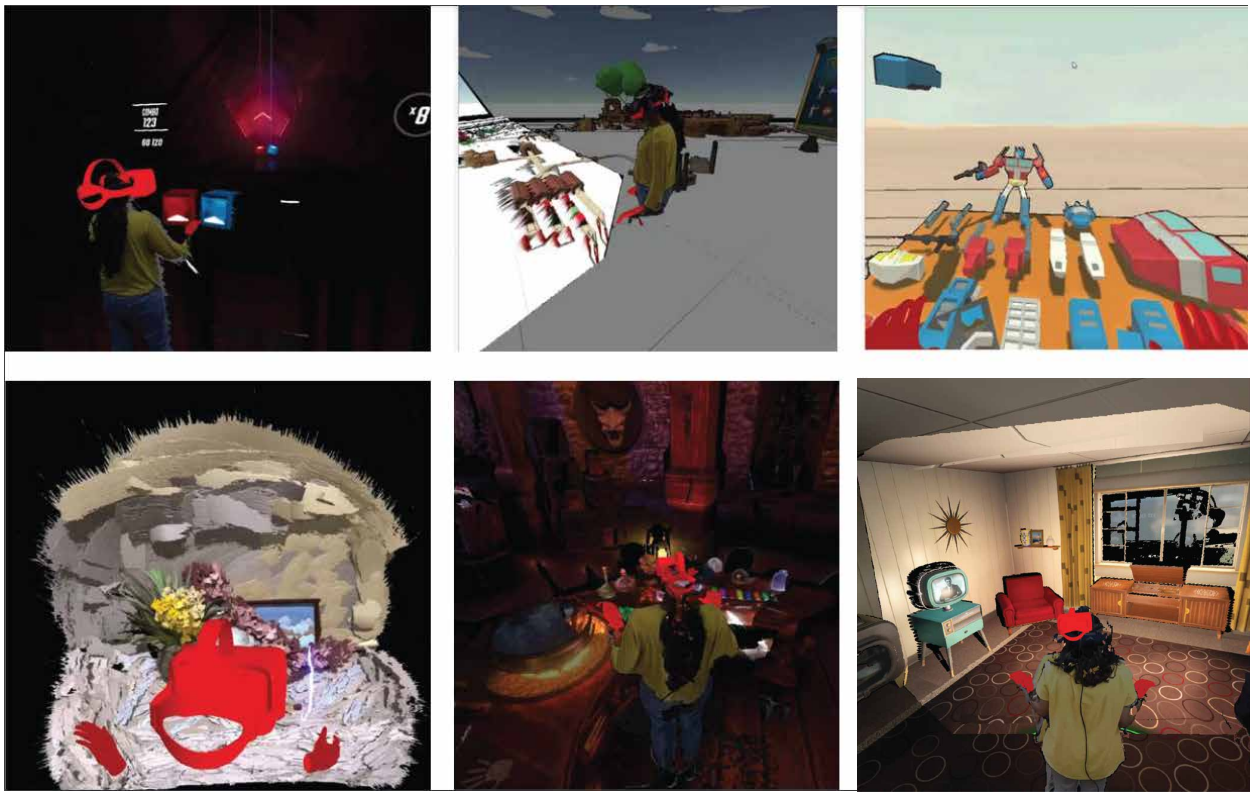


Figure 8.2: A glimpse of other applications that have been tested with DreamStream - (Top) - Beatsaber, TrainerVR, Blocks; (Bottom) - Tiltbrush, Waltz of the Wizard, Fallout 4

uses 3D capture systems and leverages depth textures to stream 3D views at scale via H.264. In addition, DreamStream implements immersive and 2D interfaces, allowing spectators of a VR activity to cohabit the space of the VR player, offering interactions similar to a multi-user system.

xspace for live interaction between a VR user and a spectating user

As before, we will begin identifying the *xspace* for the task. Here the goal is to enrich the spectating experience for a spectator of a VR activity and possibly allow them to interact with the VR player. The *xspace* for this is the same as that for TransceiVR, which is the virtual scene of the VR application, actions, and gestures of the VR user as well as that of the external user in the context of the virtual space of the VR user. However, with DreamStream, we will use the insights from the formative and user study of TransceiVR to mitigate further the asymmetry between the users, especially the asymmetries that result from the lack of the 3D perception, 3D interactions, and embodiment of the spectator. While

in TransceiVR, we designed interactions that allow for better collaboration. Here our goal is to enhance the spectating experience of a VR activity.

8.3 DreamStream's novel streaming approach - 2.5D video streaming and 3D reconstruction

In section 2.3, we saw three popular approaches that the systems of today use for allowing users to spectate VR activities - *Multiplayer approach*, *Streaming approach* and the *Cloud approach*. We saw where and how the spectator view was rendered for each of these approaches. We also saw what interactions and interfaces each approach provides to the spectators.

In this chapter, we introduce a novel approach that adopts many features and the technology stack of *streaming approach* but also allows the spectators and the player to have experiences and interactions typical of the *multiplayer approach*. In Section 2.2 we have discussed different implementation approaches for developing collaborative software for VR. As with other systems in this dissertation, we focus on developing interventions at the VR platform level so that they can work with existing VR applications without source code/API access. The ability to retrofit existing applications is an important consideration for our system design. To achieve this, we introduce software interventions that modify the OpenVR library allowing DreamStream to change camera positions, capture depth information and render custom 3D graphics into a VR scene. We use this approach to enable an enhanced, interactive spectating experience for existing VR applications.

Where and how is the spectator view rendered?

A key idea of the *DreamStream approach* is to embed the actual depth buffer information that was used to render the video frames of the VR activity (Figure 8.4-d,i). The depth buffer or "z-buffer" is commonly generated in today's 3D graphics rendering pipelines. It captures much of the 3D information of the scene from the player's viewpoint. For each pixel in an image, the corresponding pixel in a z-buffer contains information of the "depth into the scene". However, because it does not give depth to surfaces behind visible objects, it is not a complete representation of the 3D scene. It also does not contain information on any screen-space effects that may be rendered in the scene. Therefore, we apply appropriate transformations to these depth buffers to reduce noise during the encoding and decoding process and embed them alongside color (RGB) video frames. We call these depth-embedded video frames as *2.5D video frames*. Examples of these depth buffers and 2.5D video frames are shown in Figure 8.4.

To transmit this 2.5D representation, we leverage standard H.264 encoding, streaming, and decoding pipelines. This allows a partial 3D reconstruction of the VR scene that a player sees. Therefore, unlike interactions typical of the *streaming approach*, every spectator can now take different and independent perspectives from which they view this 3D reconstruction. For the Cross-Dimensional Medium of DreamStream, we also propose using a secondary 2.5D

stream that fills the spectators' view with 3D information regarding the surrounding ambient environment of the VR scene. The VR user may not view that. This is integrated with the 3D reconstruction of what the VR user sees and provides added context. These ensure that the external user has a stable view and is capable of independent spatial exploration of the VR scene. These are essential interactions for a Cross-Dimensional Media and satisfies high-level design goals: "*DG1 - Static and stable VR view*" and "*DG2 - Support independent exploration*" that we synthesized in Section 7.3 using the formative studies conducted for TransceiVR. Furthermore, spectators can view and interact with these using a VR headset or a 2D desktop user interface. This is a form of dimensional-detail exploration, also a key interaction for a Cross-Dimensional Medium.

In Section 2.3, we saw different visual representations in which the spectator's view can be rendered - 2D videos, stereo video, 360 videos, 360 stereo videos, and complete 3D reconstructions. Depending on the hardware available to the spectator, a user may choose to view using either of these visual representations. It is important to note that there may be multiple spectators who have different hardware and may prefer to use one visual representation over the other. Since all these spectators can spectate simultaneously, it becomes important for the system to gracefully degrade the experience tailored to the capabilities of the user's hardware while maximizing the information perceivable by them. To this end, DreamStream, as a Cross-Dimensional Medium, allows spectating using both 2D desktop UI as well as through a VR headset by providing interfaces to view and explore parts of the complete 3D environment of the VR player through providing 3D mesh reconstructions of them.

The frame rate at which the 3D reconstruction is updated matches that achieved by existing *streaming approaches*. However, in our approach, the user does not suffer from motion sickness issues because the final rendering now occurs locally at frame rates sufficient for VR (e.g., 90Hz) and because now the spectator controls their view. Moreover, we demonstrate that existing VR applications can be retrofit to use this technique. A drawback of this approach is that, depending upon the quality of encoding and decoding, the 3D reconstruction can exhibit certain artifacts, particularly around discontinuities in depth. In section 8.4, we discuss our approaches to mitigate these through appropriately handling the depth buffer.

In summary, depth buffers have several advantages as a representation of scene geometry:

1. They are ubiquitous in current graphics pipelines
2. They capture much of the 3D information of a scene from a given perspective
3. Unlike vertex buffers (meshes), they are more suited to conventional image compression and streaming approaches

Our current prototype gains access to the depth buffer by intercepting calls made at the VR platform level (i.e., OpenVR). We note that VR platforms could enable a wide variety of novel compositing techniques and Mixed Reality scenarios [79] if the depth buffer and related information were exposed more directly. Furthermore, because the depth buffer is so

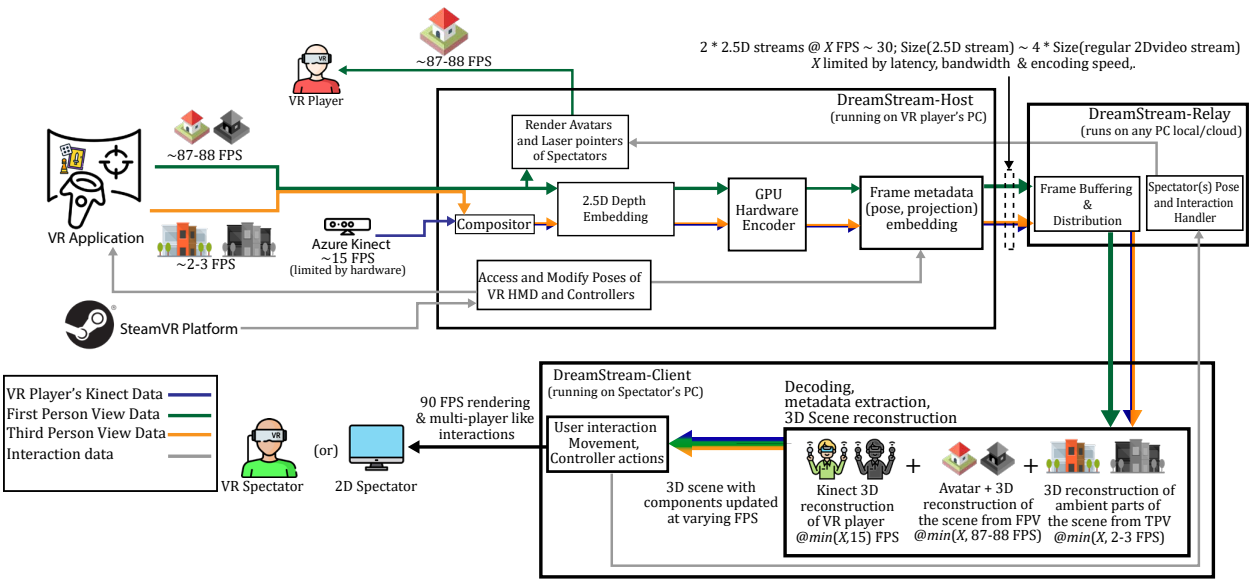


Figure 8.3: In DreamStream, the 3D reconstruction of the VR scene seen by the spectators has three main graphical components, each of which may operate at different frame rates: (1) FPV data of the VR player, (2) TPV data of ambient parts of the VR scene. TPV is controllable by the spectator, and (3) Live Kinect data of the physical body of the VR player. These reconstructions corresponding to these data are rendered spatially coherently, with their different parts updated at different rates. The spectator, however, has a full 90 FPS interaction over viewing and navigating these parts.

commonly used, we propose that it may be as generically useful for streaming, just as 32-bit color video is today.

In section 4.3, we mentioned that works in this dissertation would assume some depth data. In TutoriVR, we implicitly leveraged depth from Stereo video feeds. In TransceiVR, we computed an explicit depth map of the VR scene using computer vision techniques that leveraged optical flow in stereo feeds (Section 7.4). Those depth maps were sufficient to compute scene depth for the correct placement of annotation overlays. However, they tend to be very noisy and cannot be used to generate an accurate 3D reconstruction of the scene. Hence to generate the 3D reconstruction used for rendering the spectator's view, with DreamStream, we directly utilize the ground truth depth buffer that was part of the VR application's graphics rendering pipeline.

What interactivity is offered to the spectator?

In DreamStream, we have a Cross-Dimensional Medium that uses the visual representation of a 3D reconstruction of the virtual environment. Therefore, similar to the complete 3D environments present in *multiplayer approaches*, with DreamStream, spectators can adopt

different perspectives in the scene and perceive different parts of the action. To facilitate co-presence in such scenarios, these systems typically use an avatar-based representation of each user in the space. These avatars can be abstract or real-life replicas of the users and can vary in degrees of freedom depending on the fidelity of the pose that is tracked for each user. Accordingly, in DreamStream, the spatial presence of a spectator is communicated to the VR player and other spectators through low-fidelity abstract avatars. However, the VR player's spatial presence and actions are communicated by compositing a live 3D colored mesh reconstruction of them into the VR scene. This data is acquired from Kinect cameras in the VR player's space. This can be contrasted with approaches that use a conventional 2D camera and a fixed overall viewpoint, such as Liv. The depth images received from depth cameras such as the Kinect are naturally suited to be combined with the VR scene's depth buffer. They use very similar rendering techniques.

8.4 DreamStream System

The DreamStream system has three components that operate coherently at different locations: DreamStream-Host, DreamStream-Client and DreamStream-Relay. The DreamStream-Host program runs on the VR player's computer. The DreamStream-Client runs on a spectator's computer, handles the rendering for the spectators and provides the interface for them. Spectators can use it either through a VR headset or a regular 2D UI. Finally, a DreamStream-Relay program hosted in the cloud acts as a relay between the DreamStream-Host and the multiple DreamStream-Client applications. The DreamStream-Host streams the player's VR environment data to the multiple client applications and in turn, receives from them the pose and interaction information of the different spectators.

DreamStream-Host: VR player side of DreamStream

The VR player side of DreamStream has three main functions: (1) intercept the video and depth textures from the VR application, (2) transform and stream them as 2.5D frames to the DreamStream-Relay, and (3) handle the rendering of spectator-related information (such as their avatars and interactions) into the player's VR headset.

DreamStream targets applications built to work with SteamVR. These applications use the OpenVR library to interface with VR hardware. The library is implemented by the openvr_api.dll file. We use C++ vTable injection techniques like the ones proposed in RealityCheck [79], to build a custom DLL file that intercepts and runs the DreamStream-Host program within the functions calls made by the VR application.

Intercepting RGB and depth information

Similar to RealityCheck, calls to the functions `IVRCompositor::Submit`, `ID3D11DeviceContext::OMSetRenderTargets` and `ID3D11DeviceContext::ClearDepthStencil` are intercepted to access the rendered video frames and depth buffers sent by the VR application

to the VR player's headset. However, at the time of writing this, multiple VR applications appear to use additional rendering pipelines that do not follow either of the three rendering pipelines outlined by RealityCheck; e.g., the depth buffer for both eyes may be populated and processed only after the call to `Submit` is called for the right eye. In such cases, we use additional per-application heuristics to obtain the textures.

Primary 2.5D video frames

Primary 2.5D video frames are used to reconstruct and render a high frame rate 3D mesh feed of the VR user's First Person View (FPV) in the DreamStream's Cross-Dimensional Medium. To do this, the depth buffer and the video frame for both eyes must be prepared for streaming. Today, H.264 is the most common video encoding scheme. While this can be used to encode the color video frames, it generates noisy artifacts when directly applied to the depth buffer. This is due to two underlying factors. Firstly, H.264 algorithms are designed to preserve video's visual appearance and introduce compression artifacts when applied to depth buffers. To address this, Sonoda and Grunnet-Jepsen proposed a technique based on an inverse-colorization technique that maps depth values to the hue space [181]. We apply this operation to the depth buffers using custom compute shaders before embedding them alongside the video stream.

A second issue is that the depth buffer VR applications typically have 10-bit channel values, whereas H.264 commonly works with RGB textures with 8 bits per channel. Discretizing 10-bit to 8-bit values leads to visible discontinuities during 3D reconstruction. Therefore, we split the depth buffer values into three regimes: Background, Far-Foreground, and Near-Foreground, using predefined distance-based thresholds tuned for specific interest ranges. We arrived at these threshold values through empirical testing on different VR applications. These can be changed at run-time if required. We drop all depth buffer information closer to the threshold for Near-Foreground and further than the threshold for Background. Typically these influence the least significant bits (LSBs) and most significant bits (MSBs) of the depth buffer value. The range of depth buffer values is thus effectively reduced so that it may be encoded. The Far-Foreground and Near-Foreground ranges then focus on different exclusive (non-overlapping) 8-bit ranges of the depth buffer. We then apply the inverse-colorization technique to these and stitch them together with the RGB texture of the video feed to obtain a 2.5D video frame. The stitched figure is shown in Figure 8.4. d.

These operations are applied to VR users' FPV video feed to generate the Primary 2.5D video frames. The rate at which these frames are encoded and streamed is maximized through programming patterns that leverage parallel computing. The rate is, however, limited by other factors such as latency, bandwidth, and the GPU's encoding speed. In our setup, we achieve about 30 FPS. This is on par with the widely used video streams today.

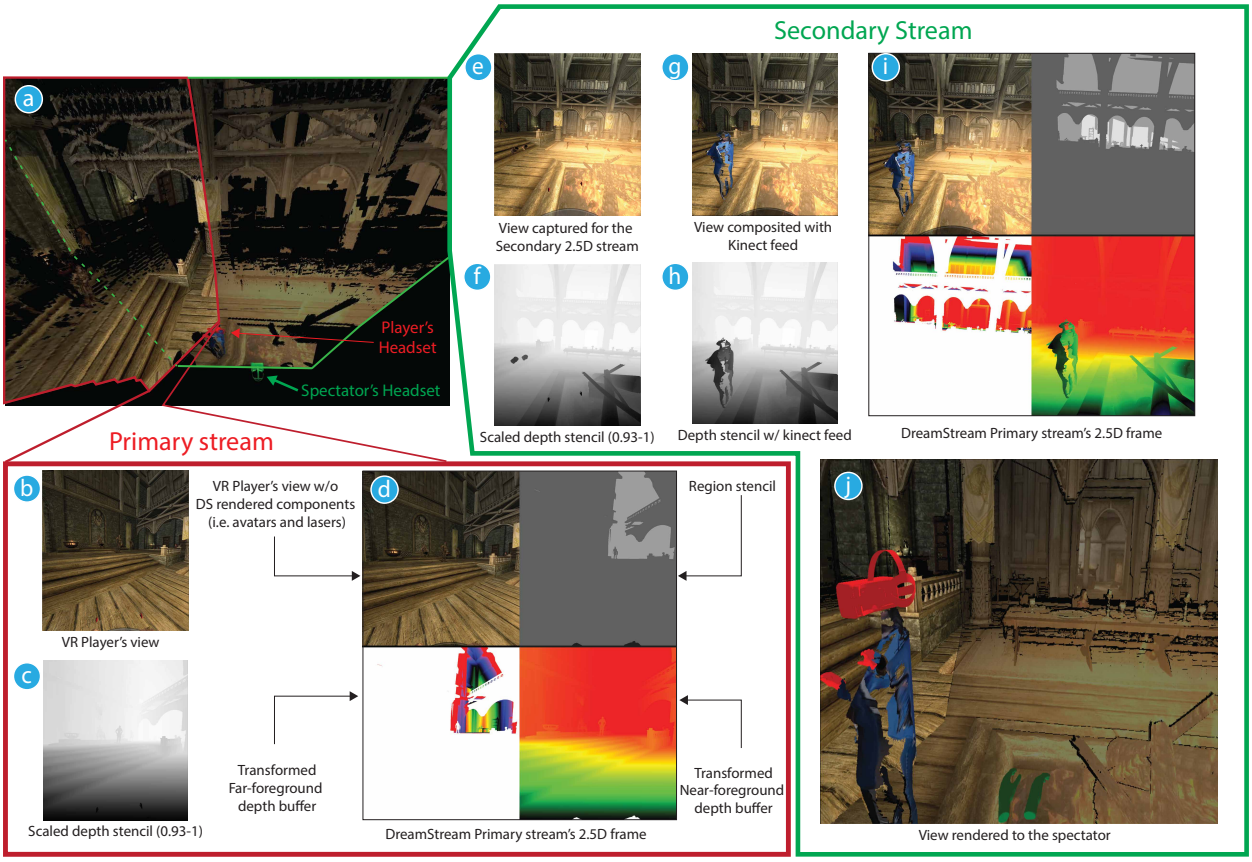


Figure 8.4: (a) 3D reconstruction of Primary and Secondary 2.5D frames. The head poses of the VR player and the spectator are represented as red and green avatars. (b) First-person video feed that VR player sees. (c) Depth stencil that has been re-scaled to the 0.93-1 range. This is where most scene data exists in the current scene. (d) A frame of DreamStream's primary 2.5D stream. Top-Left has the RGB feed from the user; Top-Right has the Stencil indicating the region the pixel belongs to (i.e. near/far-foreground, too close, too far/Background); Bottom-Left contains the Depth information of the far-foreground range; Bottom-Right contains the Depth information of the near fore-ground range. (e,f) RGB feed and depth stencil of the view seen from the overridden position corresponding to the spectator's view. (g,h) Kinect feed is being composited into the RGB and depth textures. (i) A frame of DreamStream's secondary 2.5D stream. This is similar to the primary stream. (j) View seen by the VR spectator based after reconstruction at DreamStream-Client

Secondary 2.5D video frames

Besides reconstructing the VR scene from the FPV of the VR player, DreamStream reconstructs parts of the VR scene and the VR player's physical body from a Third-Person View (TPV). It does this by creating Secondary 2.5D frames that composite the *ambient* parts of the VR scene at a low frame rate and a 3D Kinect feed of the VR player at a higher frame rate.

DreamStream captures the *ambient* parts of the VR scene surrounding the VR user from a Third Person View (TPV). Spectators can control the TPV for which they need the *ambient* parts of the scene to be rendered. These *ambient* parts are rendered for a single frame by momentarily overriding the VR player's HMD pose reported by the VR platform to the VR application. We only alter its pose to capture from the chosen TPV but otherwise do not modify the rendering process. This TPV frame is hidden from the VR user by momentarily presenting the previous frame that was rendered using their actual HMD position in the VR world. Since the TPV frames are captured at the cost of FPV frame rate of the VR player, the *ambient* capture rate is limited to 2-3 FPS. Due to this, the effective frame rate of the VR user and the primary stream's frame rate drops slightly (up to 87-88 FPS). From our experience using the system, this drop in frame rate is often unnoticed by the VR player and spectators.

In addition to capturing ambient parts of the VR scene, we also capture live 3D Kinect-based depth capture of the VR player as per their corresponding location in the VR space (see Figure 8.6(i)). We calibrate and acquire the Kinect depth capture through a modified version of the RoomAlive Toolkit [96]. The Kinect feed is also captured from the same TPV chosen by the spectator.

The captured ambient parts of the VR scene from the chosen TPV and the 3D Kinect feed from the same TPV are then composited to obtain a single RGB frame and depth buffer. This composited capture is then used to generate the secondary 2.5D frame using the same procedures for generating the Primary 2.5D frames. This can be seen in Figure 8.1.

Taken together, the primary and secondary 2.5D streams contain high frame rate 3D data of VR players' main focus of interaction in the physical and virtual environments. The low frame rate 3D data of VR users' ambient environment provides added context. Qualitatively, the idea is that spectators have an "over the shoulder" view of the VR player from within the scene of the VR application. The Field of View (FoV) of the secondary 2.5D frame is the same as that of the primary 2.5D frame, which is determined by the VR headset and the application. However, taken together, the primary and secondary 2.5D frames increase the effective FoV perceived by the spectators compared to using only the primary stream.

Encoding textures

Once the primary and the secondary 2.5D video frames are generated, the next task is to encode them. DreamStream encodes these streams in parallel threads and leverages separate hardware video encoders available on a GPU. For our setup, we use the NVIDIA GTX 1080

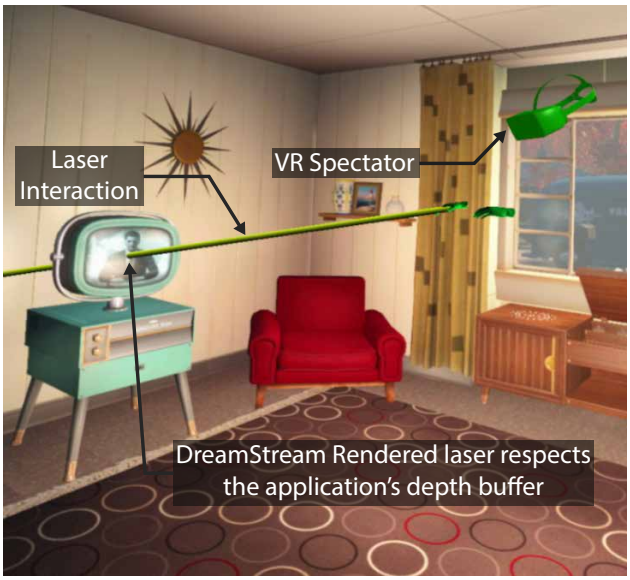


Figure 8.5: The VR player's view represents a VR spectator with a green avatar. The spectator is using a DreamStream's laser pointer to point to a TV present in a room in Fallout 4 VR. Note that the laser respects the depth buffer of the scene and can actually pass through objects in it as if it was rendered by the game itself.

and its NVENC API that supports concurrent hardware encoding of up to two video textures with a maximum pixel resolution of 4096×4096. Once the video frames are encoded, we append a binary payload containing the view and projection matrices associated with that frame. These are required for successfully reconstructing the frame as a 3D mesh at the correct pose.

Rendering into the VR scene

Besides transmitting the streams, DreamStream also renders spectators' avatars and interactions into the VR scene. To do this, we leverage the color texture and the depth buffer. These avatars denote the spatial location of each spectator. The VR spectators' avatar contains their head and hand poses depicted by headset and hand models. These poses are obtained using the spectators' VR HMD, and hand controller pose. For spectators using the 2D UI, their avatar is represented using only a headset model that denotes the pose from which they view the VR scene. All spectators also have access to a laser pointer to point to different parts of the VR scene. This can be seen in Figure 8.5. The laser pointer can be triggered either by pressing a button on the VR controller (for VR spectator), using right click on the mouse, or pressing a specific key on the keyboard (for 2D UI spectator).

DreamStream-Relay

The relay is an intermediary between the DreamStream-Host and all the DreamStream-Client programs. It can achieve NAT traversal as well as efficient distribution to multiple spectators. Upon connecting to the relay server, a DreamStream-Client is assigned a unique client ID and is initialized with the stream's information required for 3D reconstruction, such as texture size and the VR camera's calibration parameters. A buffer for storing and distributing encoded frames is allocated for each spectator client. This buffering mechanism ensures that spectators are not affected by intermittent network issues that any arbitrary spectator may face. It also keeps track of the slowest client and dynamically changes the encoding rate at run-time.

Besides distributing frames, the relay server exchanges and synchronizes information about the spectator's viewport poses. Hand controllers poses, and corresponding actions to the VR player and other spectators. This is sent through a communication channel separate from the ones used for the primary and secondary frames.

DreamStream-Client

Every spectator runs the DreamStream-Client program, which is initially unaware and agnostic of the inner details and specifics of the VR application streaming the data. When connected, it automatically obtains the stream's metadata from the DreamStream-Host through the DreamStream-Relay and uses it to initialize itself.

Once the stream is initialized, for each frame, it separates the encoded frame data and metadata associated with that frame. The per-frame metadata usually consists of the camera parameters and the position from which the frame was rendered. It then uses a GPU-based decoder to decode the encoded primary and the secondary streams to get the corresponding 2.5D frames and the metadata associated with each frame.

The metadata and 2.5D frames are used to reconstruct the three parts of the scene in a spatially coherent manner - The parts in the first-person view of the player, ambient parts of the VR scene as seen from the third-person view of the spectator, and the live Kinect feed of the VR user. The reconstruction is done through shaders and takes the form of a colored 3D mesh. The spectators can change the viewport used for generating the ambient parts of the VR scene or disable it entirely.

A spectator can view the reconstructed 3D scene through a VR headset or an interactive 2D UI (Figure 8.6). In the 2D UI, DreamStream offers four viewing modes: (1) Free Camera View allows for free and unconstrained movement of a spectating camera in the 3D space; (2) Orbit View also allows for free movement but always orients towards the VR user. The controls are mapped in such a manner as to orbit the user. (3) Over the shoulder (Follow) view positions the spectator camera behind the VR user. The horizontal distance, height, and pitch of the camera can vary in this mode; finally the (4) First-person view replicates what the VR user sees. In addition to regular camera controls, the 2D UI offers an option for the view to be stabilized when in First Person View or Over the Shoulder View (Figure

Figure 8.6: DreamStream's interactive 2D viewer; (a)-(d) allows for switching to different viewing modes. (e) toggles the ambient reconstructions, and (f) toggles the stabilization in follow and first-person view. Region (g) is the reconstruction from the primary 2.5D video frame. These are parts that the player sees. Region (h) is the reconstruction from the secondary 2.5D frame. These are rendered dimmer in color and are parts not seen by the player. This region is also updated at a low frame rate. Region (i) is a live 3D reconstruction of the player using the secondary 2.5D frames but is updated at a high frame rate. (j) shows the headset's red avatars and the player's hands.

8.6.f). This is performed by smoothing camera motion with an empirically tuned exponential filter. In all viewing modes, the ambient parts of the scene are captured from the point of view of the 2D viewer.

For viewing with the VR headset, the user can use their hand controllers to navigate the scene. By default, the view for rendering ambient parts of the VR scene is set such that the entire body of the VR player is seen. However, viewers can manually override this by holding a specific button on the controller and using it to point at parts of the scene they want to see, rendering the ambient parts in that region. The interaction is analogous to how one would use a flashlight to illuminate unseen dark regions.

As in the player's view, a spectator can see avatars of other spectators in the scene. When viewing through a 2D UI, the viewport of the 2D viewer is used as a proxy for the spectator's spatial location in the scene. When viewing through a VR viewer, the HMD and hand controller poses are used to render the avatar. As mentioned earlier, all spectators have access to a laser pointer that can be toggled on and off by pressing a key on the VR controller, keyboard, or mouse.

The abstract avatars, Kinect reconstructions, and 3D reconstructions through a VR viewer and the 2D viewer are the different visual representations used by the Cross-Dimensional Medium in DreamStream. The different viewing modes in the 2D viewer and the "*flashlight*" interaction in the VR interface are the Cross Dimensional Media interactions that allow for independent spatial exploration by the spectator. Additionally, the ability to choose between using a 2D interface or a VR viewer while being able to perform the same interactions with both is a key aspect of the dimensional-detail interaction. It also highlights the Cross-Dimensional Medium's property of meaningful translation of interactions across interfaces.

8.5 Evaluation

To better understand the opportunities and shortcomings of the Cross-Dimensional Media of DreamStream, we conducted three sets of informal evaluations: an expert evaluation with four experts, an initial evaluation with eight users, and a revised evaluation with twelve users. We first describe the methodology and measurements for these. We found common themes in our expert study and in both user studies. We discuss these together in a common subsection.

Expert Evaluation

We recruited four professional VR steamers as experts (E1-E4) in our study; at the time of the study, they had 12K, 120K, 380K, and 3.3K subscribers on their streaming channel, and each has been streaming VR content for the last 6-7 years. We omit other information about our experts to ensure their anonymity. We interacted remotely with each streamer for about 70 minutes, and as a token of appreciation, we gave them a gift certificate of $60. At the start of the study, we conducted a semi-structured interview in which we elicited details

on the state of the art of VR streaming, its goals, challenges, and the workarounds that they employ to solve those. This served two purposes: (1) it helped us better understand these, and (2) it helped the streamers collect their thoughts on these so that they could evaluate our tool. After demoing DreamStream through Zoom, we asked them if it might solve any of the issues they had mentioned earlier and whether they could anticipate any issues using the system.

Initial User Evaluation

We conducted an initial evaluation with eight participants (I1-I8) (5 Male, 3 Female, Age range 20-28). They all had prior experience with VR and games, used VR, and played games at least a twice every month. In this study, we compared DreamStream with the Standard VR mirror. Each study lasted 90 minutes, and the participants were compensated with a $40 gift card for their time. Per our campus regulations, to minimize Covid-related risks, an author of this chapter took the role of a VR player and played Skyrim in VR. A participant spectated and interacted with them. Participants were told that their goal was to have fun, occasionally interact with the player and aid them in making key decisions in the game. Beyond spectating, spectator interactions included identifying objects, navigating terrains, and helping choose other game player actions. Participants used DreamStream-Client through a laptop placed 10ft away from the player. Though DreamStream can work over the internet, for the study, the DreamStream-Host and the DreamStream-Client interfaced through the campus network. The author's audio was heard directly by the participants, whereas the game audio was channeled in stereo through Zoom. They spectated through three different interfaces: (1) Standard video mirror, (2) DreamStream's Interactive Desktop-based 2D interface (DS-I2D), and (3) DreamStream's VR interface (DS-VR). Each participant spent roughly 12-15 minutes with each of the interfaces (within-subjects), and the order of the interfaces was counterbalanced.

Final User Evaluation

Our Final user evaluation consisted of 12 participants (P1-P12) (7 Male, 5 Female, Age range 23-31). All of them had used VR at least a couple of times in the last 6 months. In addition, they play games at least a couple of times every month. In this evaluation, we compared DreamStream with TransceiVR [185], which shares similar functionality for its overall goal of enabling asymmetric collaboration in VR. In line with the original implementation, TransceiVR was deployed and presented through an iPad-based touchscreen interface with a stylus. The evaluation was conducted similarly as our initial evaluation.

Measures

For our final evaluation, our questionnaire consisted of 5-point Likert scale questions rating the following elements: (1) Easy to communicate to the VR player, (2) Easy to point/refer to objects to the VR player, (3) Easy to direct the VR player, (4) Easy to understand

the VR user and their actions, (5) Easy to understand the VR scene, (6) Easy to explore independently, (7) Spectating was enjoyable, (8) They felt immersed in the game and (9) They felt alongside the VR player. A Likert Rating of 1 corresponds to 'strongly disagree' and 5 to 'strongly agree'. Intermediate values were labeled accordingly. In our initial evaluation, we used similar but slightly different Likert scale questions, in which we asked participants to compare pairs of interfaces directly. However, in our final evaluation, we asked participants to rate one interface at a time since this is more amenable to standard analysis techniques.

They also completed the NASA-TLX instrument that measured the user's perceived workload. Using 5-point Likert scale questionnaires, we asked the participants about the role of stabilization, the utility of ambient reconstructions, and the player's 3D body reconstruction on their spectating experience. In the end, we conducted a semi-structured interview with participants to understand how the interfaces compared. We also welcomed any open-ended feedback on the interfaces and their study experience. In this chapter, we only report the Likert scale responses and the NASA-TLX responses of the final user evaluation.

Final User Evaluation Results

This subsection reports the statistically significant results from our final user evaluation.

Analysis Methodology: Since Likert-scale ratings are ordinal values, we first performed a Friedman test to determine if the interface condition had an overall effect on the measured rating. If an overall effect was found, we performed a post-hoc pairwise exact Wilcoxon-Pratt signed-rank test between conditions to see if there was a significant difference. For the NASA-TLX scores, we first carried out a Repeated measures ANOVA to check for the overall effect of the interface. We then performed post-hoc pairwise t-tests to check for individual differences. All p-values used for testing were adjusted with Bonferroni correction.

Communication: Friedman test revealed a significant effect of interface condition on the participant's ease of communication to the VR user ($\chi^2(2) = 19.5$, $p<0.05$). Post-hoc pairwise Wilcoxon signed-rank tests showed statistically significant differences between DS-I2D and TransceiVR ($p<0.05$, with a large effect size - $r = 0.61$) as well as between DS-VR and TransceiVR ($p<0.05$, with a large effect size - $r = 0.62$). Aggregate statistics are : $\mu_{TransceiVR} = 2.92$, $\sigma_{TransceiVR} = 1.16$, $\mu_{DS-I2D} = 4.33$, $\sigma_{DS-I2D} = 0.78$, $\mu_{DS-VR} = 4.67$, $\sigma_{DS-VR} = 0.49$.

Friedman test also revealed a significant effect of interface on participant's ease of referring to objects in the VR scene ($\chi^2(2) = 18.57$, $p<0.05$) as well as ease of directing the VR player($\chi^2(2) = 14.6$, $p<0.05$). Furthermore, the pairwise Wilcoxon tests for the ease of referring to objects in the VR scene found a statistically significant difference between DS-I2D and TransceiVR ($p<0.05$, $r = 0.52$) as well as between DS-VR and TransceiVR ($p<0.05$, $r = 0.58$). Similarly, Wilcoxon tests for ease of directing the VR player also showed significant differences between DS-I2D and TransceiVR ($p<0.05$, $r = 0.52$) as well as DS-VR and TransceiVR ($p<0.05$, $r = 0.58$).

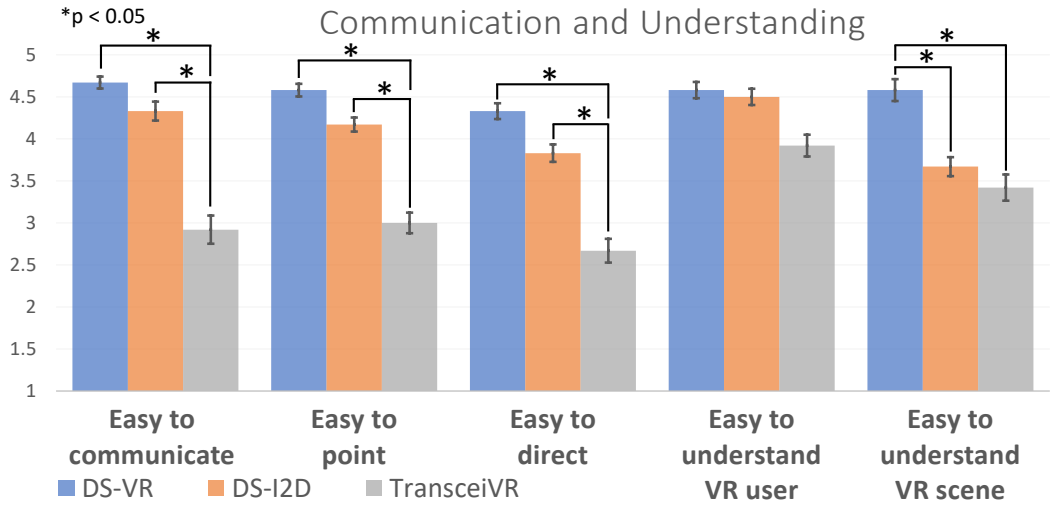


Figure 8.7: 5-point Likert scale responses of participant self-rated ease of (1) Communication with VR player, (2) Pointing to objects in the scene, (3) Directing the VR player, (4) Understanding VR player's actions, and (5) Understanding the VR scene. The error bars denote Standard Error ($\sigma/\sqrt{n}$). A Likert Rating of 1 corresponds to 'strongly disagree' and 5 to 'strongly agree'. All intermediate ratings were labeled accordingly. All significance values were calculated at $p < 0.05$ after Bonferroni correction.

These results indicate that our participants found it easier to communicate, point and direct with both DS-2D and DS-VR when compared to TransceiVR.

Spectator's understanding of VR scene and player: Friedman test revealed a significant effect of interface condition on the spectator's ease of understanding the VR user's actions ($\chi^2(2) = 6.28$, $p<0.05$) as well as their ease of understanding the VR scene. Posthoc Wilcoxon tests did not yield any statistically significant differences in the ease of understanding the VR user's actions.

On the other hand, Wilcoxon tests did yield a statistically significant difference for ease of understanding the VR scene between DS-VR and DS-I2D ($p<0.05$, $r = 0.63$) as well as between DS-VR and TransceiVR ($p<0.05$, $r = 0.56$). Aggregate statistics are: $\mu_{TransceiVR} = 3.42$, $\sigma_{TransceiVR} = 1.08$, $\mu_{DS-I2D} = 3.67$, $\sigma_{DS-I2D} = 0.78$, $\mu_{DS-VR} = 4.58$, $\sigma_{DS-VR} = 0.9$.

This implies that the choice of interface did not make a difference in easing our participants' understanding of the VR user's actions. However, they found it easier to understand the VR scene with DS-VR when compared to both DS-I2D and TransceiVR.

Scene exploration: Friedman test revealed a significant effect of interface condition on the spectator's ease of exploring the VR scene ($\chi^2(2) = 10.55$, $p<0.05$). The pairwise Wilcoxon tests showed a statistically significant difference between DS-VR and TransceiVR ($p<0.05$, $r = 0.54$). Aggregate statistics are: $\mu_{TransceiVR} = 2.25$, $\sigma_{TransceiVR} = 1.21$, $\mu_{DS-VR} = 3.83$, $\sigma_{DS-VR} = 1.11$, $\mu_{DS-I2D} = 3.25$, $\sigma_{DSI2D} = 1.29$.

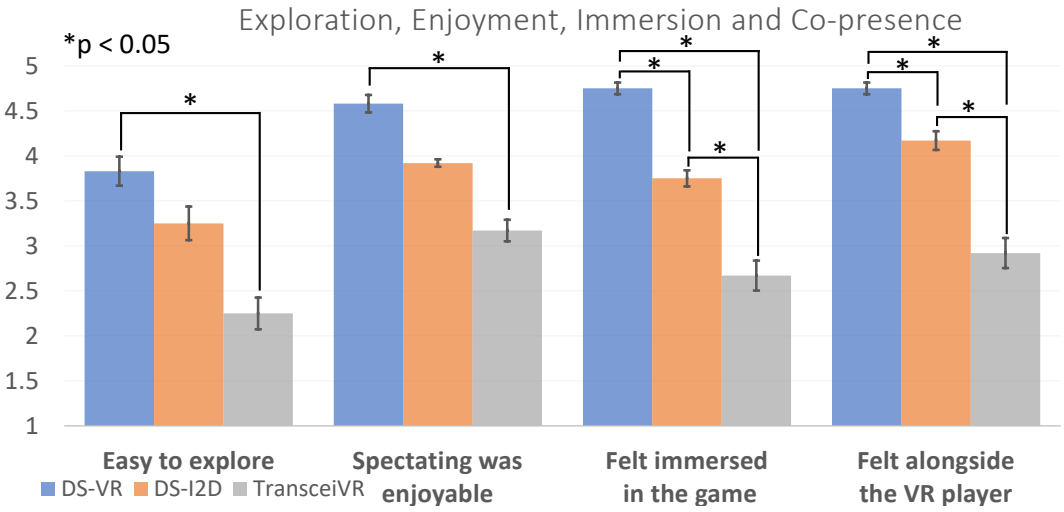


Figure 8.8: 5-point Likert scale responses of participant rating of (1) Ease of independent exploration of the scene, (2) Enjoyable spectating experience, (3) Feeling immersed in the game with VR player, and (4) Feeling present alongside the VR player in the game. The error bars denote Standard Error ($\sigma/\sqrt{n}$). A Likert Rating of 1 corresponds to 'strongly disagree' and 5 to 'strongly agree'. All intermediate ratings were labeled accordingly. All the significance values were calculated at $p < 0.05$ after Bonferroni correction.

This implies that our participants found it much easier to independently explore the VR scene using DS-VR when compared to TransceiVR.

Enjoyment: Friedman test revealed a significant effect of interface condition on the spectator's agreement that the spectating experience was enjoyable ($\chi^2(2) = 15.6$, $p<0.05$). The pairwise Wilcoxon tests showed a statistically significant difference between DS-VR and TransceiVR ($p<0.05$, $r = 0.61$). Aggregate statistics are: $\mu_{TransceiVR} = 3.17$, $\sigma_{TransceiVR} = 0.84$, $\mu_{DS-VR} = 4.58$, $\sigma_{DS-VR} = 0.67$.

This implies that our participants found the spectating experience more enjoyable using DS-VR compared to TransceiVR.

Immersion and Co-presence: Friedman test revealed a significant effect of interface condition for spectator's agreement that they felt immersed inside of the game during the spectating experience ($\chi^2(2) = 21.54$, $p<0.05$) as well as for their agreement that they felt alongside the VR player ($\chi^2(2) = 20.15$, $p<0.05$). For both ratings, pairwise Wilcoxon tests found statistically significant differences between all pairs.

For Immersion, the difference between DS-I2D and DS-VR was significant at $p<0.05$, $r = 0.62$; between DS-I2D and TransceiVR at $p<0.05$, $r = 0.59$; and between DS-VR and TransceiVR at $p<0.05$, $r = 0.63$. Aggregate statistics are: $\mu_{TransceiVR} = 2.67$, $\sigma_{TransceiVR} = 1.15$, $\mu_{DS-I2D} = 3.75$, $\sigma_{DS-I2D} = 0.62$, $\mu_{DS-VR} = 4.75$, $\sigma_{DS-VR} = 0.45$.

For the agreement to feeling alongside the VR player, difference between DS-I2D and DS-VR was significant at $p<0.05$, $r = 0.54$; between DS-I2D and TransceiVR at $p<0.05$,

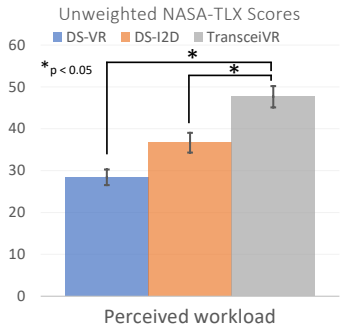


Figure 8.9: NASA-TLX scores that measured participants' perceived workload while using each of the interfaces. The error bars denote Standard Error ($\sigma/\sqrt{n}$). Note: A Higher perceived workload for an interface means that the participants found it more difficult to use. The maximum score possible is 100.

$r = 0.59$; and between DS-VR and TransceiVR at p<0.05, $r = 0.64$. Aggregate statistics are: $\mu_{TransceiVR} = 2.92$, $\sigma_{TransceiVR} = 1.16$, $\mu_{DS-I2D} = 4.17$, $\sigma_{DS-I2D} = 0.72$, $\mu_{DS-VR} = 4.75$, $\sigma_{DS-VR} = 0.45$.

These results imply that our participants felt that they were most immersed in the game and alongside the player when they used DS-VR, followed by DS-I2D and then TransceiVR.

NASA-TLX: We used the unweighted version of the NASA-TLX scores. A Higher perceived workload score for an interface means that the participants found it more difficult to use. Mauchly's test did not show a violation of sphericity against the interface conditions ($W(2) = 0.65$, $p = 0.12$). With one-way repeated-measure ANOVA, we found a significant effect of interface on the NASA-TLX scores ($F(2,22)=11.51$, $p<0.05$, $\eta^2_{partial} = 0.51$). Pairwise t-test results found a significant difference between DS-I2D and TransceiVR ($t(11) = 4.19$, $p<0.05$, Cohen's $d = 1.21$) as well as between DS-VR and TransceiVR ($t(11) = 4.056$, $p<0.05$, Cohen's $d = 1.17$). Aggregate statistics are: $\mu_{TransceiVR} = 47.64$, $\sigma_{TransceiVR} = 17.68$, $\mu_{DS-I2D} = 36.67$, $\sigma_{DS-I2D} = 16.36$, $\mu_{DS-VR} = 28.4$, $\sigma_{DS-VR} = 12.97$.

This indicates that our participants perceived a higher workload to interact when they used TransceiVR compared to both DS-VR and DS-I2D.

Discussion of User and Expert Evaluation

In this subsection, we qualitatively discuss and compare the different interfaces.

Overall interface preference

In the initial user study, five of eight participants rated the VR interface of DreamStream (DS-VR) as their most preferred viewing mode. In addition, two preferred Follow view of the Interactive 2D viewer (DS-I2D), and one preferred the standard VR mirror. In our final user

evaluation, nine of twelve participants rated the VR interface of DreamStream (DS-VR) as their first preference, one each preferred first-person view and follow view of the interactive 2D viewer (DS-I2D), and one preferred TransceiVR. In further subsections, we break down the factors that provide insights into this preference.

Understanding the VR scene and actions of the VR player

In our final study, we found no statistically significant difference among the three interfaces in the spectator's ease of understanding the player's actions. However, we did find statistically significant differences in the spectator's ease of understanding the VR scene when using DS-VR compared to when using DS-I2D and TransceiVR. We observed no significant difference between DS-I2D and TransceiVR.

For the spatial exploration interactions of the Cross-Dimensional Medium, TransceiVR uses *angle Frames* to allow a spectator to view parts of the VR scene other than that of the VR user's current view. These are static frames captured when the VR player last looked along certain directions. While these can be useful for collaboration in relatively static scenes, they are less so in the more dynamic environments of games: terrain and other elements frequently change, and players often teleport and change orientation through in-game controls. Both these interactions tended to break the *angle frames* interaction in TransceiVR. For example, the VR player can be looking straight and exiting a cave into an open terrain. However, since the VR player had not turned around in this process, *angle frames* would still display the outdated frames of the cave. Another issue is that the TransceiVR system relies on changes in the pose of the headset in order to update the *angle frames*. Therefore, any change in orientation or position caused by using the controller (such as teleportation) will not be tracked by TransceiVR, and a previously captured frame's view direction may now be unrelated to the current view. These interactions are common in VR games and interfere with the spectator's understanding of the scene.

P3: "*I would have loved the TransceiVR interface if the images below [Angle frames] had updated in real-time. But they were delayed. So I could not use them appropriately and ultimately did not use it.*"

P11: "*With TransceiVR, the latency of interactions was too high with moving objects. When searching for chickens, for example, the [angle frames] on the bottom were not updating fast enough.*"

P6: "*With TransceiVR, there's a lot of frustration when the frames [angle frames] are not updating. It happens when the player has not looked in a direction for a while. It is a few more layers away from the player.*"

While with DS-I2D, ambient reconstructions of the Cross-Dimensional Medium gave spectators a comparatively more frequently updated view of the player's surroundings. However, there was still no significant difference in scene understanding between TransceiVR and DS-I2D. Both DS-VR and DS-I2D rendered the same information. However, participants felt using the medium through VR (i.e., DS-VR) to be easier than using it through the desktop interface (i.e., DS-I2D) for understanding the VR scene. They attributed this to an increased

feeling of immersion and embodiment. This allowed them to better understand and perceive the depth and scale of the scene as a whole, as well as the ease of interaction and independent exploration of the Cross-Dimensional Medium.

P4: "*In [DS-VR], I felt that I was much more on the same plane as the [VR Player]. I had the same ability as the [VR Player], so I could easily relate to the way [VR Player] was looking around in the world. So instead of having this omniscient view [DS-I2D], I felt more like another person in the scene. I was able to perceive the same things in the same way as the [VR Player] did. I could get a more sense of depth, and I was able to better perceive my relative distance to the [VR Player].*"

This was also noted by one of our experts,

E3: "*I would say that [DS-VR] fixes the fact that normally viewers can't see the 3D effects in the depth. Like, which is the big thing. It never comes across properly on a 2d screen. [DS-VR] means they're in VR with the [VR Player]. They're going to get the 3d depth of that, and completely understand the world they are in. It also means that, they can help the [VR Player]. They can guide them. So it actually adds more interaction than a standard stream, which are just text chats.*"

Communicating with the VR player

In our final study, participants found communicating with the VR player easier using either interface of the DreamStream's Cross-Dimensional Medium, i.e., DS-I2D and the DS-VR, than TransceiVR. Spectators similarly found it easier to point to scene elements and direct the VR player's attention. There were no statistically significant differences between DS-I2D and DS-VR interfaces. We observed similar responses in our initial study, in which participants found it easier to communicate, point and direct using DS-VR and DS-I2D compared to VR mirror. While the initial study's results were not surprising, we found the results of the final study to be surprising. The Cross-Dimensional Medium of TransceiVR is primarily designed to facilitate communication with the VR user, so we expected that annotation and shared screen features would enable levels of communication as good as both DreamStream media interfaces.

From the study, we found that the lack of effectiveness of TransceiVR is due to two key factors. The first is due to the difficulty in understanding the VR scene, as described earlier in this section. The lack of regularly updated views in the more dynamic scenes and interactions of games makes it more difficult for spectators to quickly and easily parse the VR scene. This impacts their ability to communicate with the VR player regarding the VR scene easily.

Second, with TransceiVR, spectators can communicate with the VR player by creating annotations on the live feed. However, these annotations do not track the changes in player orientation. As a result, annotations quickly move out of position as the player teleports, translates, or rotates in the game. Furthermore, there is no mechanism to correct the positions of these annotations after placement, so they must be cleared and re-drawn if they persist over time. Furthermore, the live freed is frozen when the annotation is begun and

only resumes after a 500ms time out. Alternatively, spectators can share annotated frames as separate screenshots to be placed in the scene, but this is more involved and cumbersome than a WYSIWYG annotation interaction.

As with TransceiVR, DreamStream has access to only the system-level VR poses of the player's HMD and their controllers. With the Cross-Dimensional Medium of DreamStream, both VR and I2D interfaces offer dynamic laser pointers to the spectator. This interaction aims to achieve a similar purpose as TransceiVR's static annotations but can be used for more dynamic interactions. Laser pointers are rendered into the live feed of the scene, and spectators can quickly adapt their pointing in response to changes in the scene. Furthermore, the laser pointer interaction is simpler, easier to trigger (right-click on DS-I2D, and trigger on DS-VR), less time-consuming. Therefore, it is more suited to fast-paced VR experiences with dynamic elements and interactions. Experts, as well as users from both our studies, noted this difference.

I5: "*In VR, the laser pointers are so nice because it is easier and faster for me to use 360 degrees and then just tell the [VR Player] to look for my laser pointers.*"

E2: "*The laser pointers in [DreamStream] could really help with new kinds of engagement. I mean, this is so much more engaging than just the chat.*"

P11:"*In the VR experience, I think a laser pointer was more accurate in terms of having feedback and it felt more like I could directly communicate with the [VR Player] about the scale of their surroundings.*"

P5: "*The pointing in TransceiVR freezes the view and I do not see the live scene anymore. If the player then moves, the annotation does not follow them nicely. Whereas in the DreamStream there is a real-time pointer which is [more apt and helpful].*"

In summary, while TransceiVR allows for conveying more complex information, it requires more effort and time and fails for dynamic scenes. However, DreamStream's laser pointer allows for conveying simple information with less effort and works for dynamic scenes. DreamStream's laser pointer interaction in the DS-I2D and DS-VR interfaces is yet another instance of the property of the Cross-Dimensional Medium's meaningful translation of interactions across visual representations.

Immersion, Enjoyment, Co-presence, and Independent exploration

In our final study, we see statistically significant differences across all the interfaces with respect to the spectator ratings of immersion in the game and their rating of feeling alongside the VR player. They felt most immersed and felt most alongside the VR player when using the medium in DreamStream compared to the one in TransceiVR. Even within the DreamStream medium, the DS-VR interface was rated higher than the DS-I2D for both aspects. We observed similar results with our initial study participants. We believe this is because the DS-VR interface allowed participants to have a more symmetrical spectating experience of the *xspace* compared to the other interfaces. In addition, DS-I2D interface allowed for more control over their viewport compared to TransceiVR. Immersion is also influenced by the participant's ability to perceive and understand the VR scene and communicate with the

VR player, as discussed above. DS-I2D and DS-VR interfaces are different visual representations of the same medium. So while they offered similar information, participants noted two key differences between the interfaces that made the DS-VR more immersive.

Firstly, with DS-VR, participants could better understand the depth and scale of things in the environment and the ambient surroundings in which the VR player operated. P4's comment (in the subsection that discussed understanding) also reflects this. I1 also says:

I1: "*[DS-VR] helped me feel more immersed, I think because I felt like I was fully in the environment. I didn't feel like just an external spectator.*".

A second key difference was that participants felt DS-VR controls were easier, more natural, and intuitive.

P3: "*[DS-VR] was much easier to use, and I could feel I was with the [VR Player] more than the iPad [TransceiVR] or Desktop [DS-I2D] interfaces. When I want to move or rotate my view quickly with iPad or Desktop, it's pretty hard. When controlling using the mouse or keyboard, its like different from the real world. In real world, when I look around, I just turn like this. I was able to do that in VR. But with the Desktop and the iPad interface, I had to click or do some action.*"

This was also seen with ease of spatial exploration of the Cross-Dimensional Medium. We found a statistically significant difference only between DS-VR and TransceiVR. Within the DreamStream medium, both interfaces, DS-VR and DS-I2D allow for free spatial exploration of the VR scene, and hence we did not expect a difference between them. However, for the same reasons mentioned for differences in understanding, we expected both the interfaces to DS-VR and DS-I2D to outperform TransceiVR. However, this was not the case. From participant comments, we understand that while the DS-I2D interface allowed for free exploration of the VR scene, there was a cost involved when they had to switch back and forth between what the player was seeing and their exploration. They noted that this interaction cost in DS-I2D is much higher than DS-VR's. In DS-I2D this meant changing viewing modes or using mouse/keyboard to manually change the viewports every time.

P8: "*In VR [DS-VR] I only had to look away to detach from the [VR Player's] view. When I wanted to see the [VR Player's] view, I could easily turn back. It was really easy for me to switch back and forth between independently exploring and looking at what the [VR Player] did.*"

We believe that the better perception, ease of control and an overall more symmetrical experience offered by the DS-VR interface likely lead to it being rated more immersive. Due to this, participants also felt more like they were with the player, alongside them.

I7:"*VR appealed to me the most, because it felt like we [VR Player and I7] were together doing something. We were experiencing the same thing*".

E3 summarized this as: "*The idea that [in DS-VR], they can see what I'm seeing and they can move independently around the scene, maybe play the game with me, can actually see things I missed out, and point to it. These are just groundbreaking.*"

In terms of enjoyment, the only significant difference was between the DS-VR interface and TransceiVR, where participants found DS-VR more enjoyable. Participants rated the DS-I2D interface somewhere between these two interfaces, but the pairwise comparison with

either insignificant. Through our interviews, we attribute this to the fact that the VR interface of the medium offered increased immersion as well as more intuitive usage.

Perceived Workload

As we had noted briefly in previous chapters, all experts we interviewed for this work also mentioned the issue of shaky video feeds, and lower field of view (FOV) of the VR scene as the two biggest issues that spectators face with VR video streams. Experts noted that they currently resort to withholding information from the spectators for the same reason.

E1: "*One of the unwritten rules of streaming - Don't talk about something [spectators] can't or didn't see. They are already shackled by the fact that they're looking through a flat display with a reduced FOV*".

We had similar issues to address during the development of Cross-Dimensional Media in TutoriVR and TransceiVR. In TutoriVR, this was addressed partially by the stabilized 3D stroke rendering in the Perspective Thumbnail widget, whereas in TransceiVR we did this through appropriate freeze frames. The Cross-Dimensional Medium of DreamStream addresses this issue more natively by providing interfaces to the spectator that allows them to explore the 3D reconstructions of the virtual space independently. This decouples the VR player's head motion from the spectator's and makes the spectator feeds easier to watch.

Accordingly, we observed a significant difference between the perceived workload measured by NASA-TLX between TransceiVR and the DreamStream's interface. We also saw a significant difference between the perceived workload of DS-VR and DS-I2D interfaces. We believe this is due to the relatively larger number of keyboard controls and viewing modes in the latter. A few participants recommended that a joystick controller be used as an alternate input. DS-I2D interface instead of the conventional keyboard and mouse.

In summary, DS-VR was better for viewing and interaction. DS-I2D was also good for viewing but made interaction more difficult. TransceiVR was difficult to view and also made interaction more difficult.

Reconstructions of Ambient parts of the scene

All experts agreed that reconstructing the ambient parts of the scene from a third-person view and the spectator's ability to control the view adds value to the spectating experience. E1: "*I think the value of this [Ambient reconstructions] for something like Half Life Alyx or Skyrim really stands out because it gives the spectator the peripheral fill-in information that, even if less accurate, even if less frequently refreshed, is very valuable. For example, they can see if there's a Wolf running up behind me, even if it was like low frame rate.*"

User study participants had somewhat more varied opinions on its value. In the 5-point Likert-Scale questionnaires, Eight of twelve participants agreed or strongly agreed that ambient information made it easier to understand the scene. Three were neutral, and one disagreed. Participants who agreed felt that ambient information helped because they did not have to rely on the player's view.

I3: "*[Ambient reconstructions] made it more comfortable to see the full scene without, a black void. So it was more enjoyable that way as well. It also gave me a little more awareness of where I was and where I was facing*".

However, they also noted challenges: First was its lower frame rate and this resulted in issues some times. P11 noted that they turned off ambient temporarily when the scene had too many fast moving objects.

P11: "*I enjoyed it more with the ambient reconstructions. I tried to keep it on most of the time, mainly because not only was it enjoyable, but it helped orient me a little bit more. In action scenes I had to turn it off so that I could focus on that person that the [VR Player] was trying to hit, but then I'd turn it back on if I wanted to just check up the surroundings and navigate through to the top of that mountain. Ambient also helped me forget about how enclosed the field of view was. So it was definitely helpful for just making the field view less limiting for my experience.*"

The second issue we observed was that Skyrim's rendering occasionally would omit objects that are not in the player's field of view. For example, this will cause far-away terrain to pop out of view as the player moves towards it. In our user evaluations, some participants noticed this effect. Aside from these issues, P1 noted that they would not use ambient reconstruction because they felt it gave them more information than they preferred.

P1: "*The fun part about spectating games is the shared experience. So I don't find a lot of value in looking at other parts of the scene, I guess. So just like being with the [VR Player] and seeing what they see, I think is the most fun part of the shared gaming experience.*"

Player live 3D reconstruction

Participants' opinions on the value of the VR player's live 3D reconstruction were varied. In the 5-point Likert-Scale questionnaires, seven out of twelve participants agreed or strongly agreed that the VR player's 3D reconstruction made the experience enjoyable, while three were neutral and one disagreed. Eight felt that it made it easy to understand the VR user's actions, while three disagreed (one remained neutral). Some participants felt that the rendering of the player added social value to the experience but not much value in understanding their actions. On the other hand, they felt that it sometimes gave an idea of what the VR player can physically do, such as whether an object was within reach.

P6: "*It definitely helps to see see how a friend [VR Player] is behaving in real life when they are playing the game. With productivity of interactions, it only marginally helps. I definitely have a better picture of how and what the player is doing. Whether he is capable of reaching certain points. I couldn't figure out the emotions because their facial expressions were covered. It helps understand the body gestures, but still I am missing the face.*"

While some participants, such as P4 felt it added to the co-presence of the experience, they also noted that the player reconstruction sometimes occludes their view.

P4: "*It helped that I could feel that I was with the [VR Player], the feeling that we [VR Player and P4] are together was really enhanced with that. But whenever the [VR Player] is occluding some of my views, it was not nice.*"

They also noted that this could potentially break the immersion of the game.

P1: "*Having the [VR Player] sitting there kind of took me out of the world of the game. Like, I know, the [VR Player] is running around murdering chickens and stuff. But then the [VR Player] is just like sitting in their chair that doesn't really connect.*"

Preference within spectating modes in 2D displays

Within the different modes of spectating using the DS-I2D interface, participants had widely varying preferences and described the trade-offs between them. Therefore, we counted the number of users who ranked each viewing mode as their first or second preference interface. the highest was Follow view with six users, followed by free view with three users. Follow view was popular because it allowed participants to sit back and passively observe the action.

Follow view also provided a greater field of view, with minimal controls of height, distance, pitch, and yaw of the camera. Often these were set once, switching to other modes if they wished to explore the scene.

I4: "*I stuck to the follow and free view the most. In that, I can not only see the [VR Player], but I also see the things around them that they aren't seeing. That gives me a much better perspective and allows me to direct them better...I would switch to the free view was primarily to explore and interact*"

P6: "*The Follow view synchronizes my view with the players view. That's very nice, and eases burden from using arrow keys. Follow view also gives more information than first person view. First-person view always restricts me with the players view*"

The camera stabilization deployed in DS-I2D interface received mixed feedback. While some users liked that it smoothed any jarring motion, they also complained that it often broke the authenticity of a fast-paced action by the VR player.

Value of platform level support

All experts expressed strong positive opinions about DreamStream's ability to these Cross-Dimensional interactions while still operating at a platform level. They highlighted that it is hard to gain any traction for such a spectating experience without this. For example, three of the experts cited *vReal*, a system providing some functionality similar to DreamStream but requiring developer support in each game.

E1: "*Even if one provided them[Developers] the perfect tool set and said, just write one line of code, one wouldn't even get to 80%, coverage of all the apps.*"

This was also noted by participants who had played or watched similar games before.

P6: "*Having spectators view it in VR gives an entirely different experience. Some games are bound to be played as a single person. It's quite hard to allocate tasks [to spectators]. This interface [DS-VR] helps with that. It allows playing a single player game together [VR player and spectators]. It allows experiencing what a player experiences.*"

Impact of noise and defects in scene reconstruction

During the interviews, we asked all participants how much the noise and defects in scene reconstruction affected their experience of the medium. All except one participant felt that though noise and defects affected their experience, the increased immersion and interactivity improved the experience. Some users noted that artifacts were not uniform across distances and perspectives. For example, P3 noted that, in DS-VR, where they do not have an option to be in the first person, the noise seems more pronounced when they took perspectives far away from that of the player. However, they also noted that the added immersion and interactivity are worth it. Other participants, as well as experts we interviewed, had similar opinions. Other participants as well as experts we interviewed had similar opinions. E4: "*I do not think that the artifacts would detract a crazy amount from the viewer experience. The immersion that viewers gain from this would help so much more*".

8.6 Limitations

Artifacts in 3D rendering

In section 8.3, we listed the advantages of using depth buffers for enabling the three-dimensional reconstruction and visualization for DreamStream's Cross-Dimensional Medium. The key advantage is that it leads to a generic approach for streaming 3D data that leverages existing video codecs for compression and works across many, if not most, VR applications. Furthermore, transmitting the depth buffer has the added advantage that it may be used to perform further rendering or compositing at either the player or spectator sites, such as rendering avatars.

The main limitation of this approach is that the depth buffer is only a 3D representation of the scene from the VR player's point of view. As a result, a spectator standing on the side of the VR player may notice gaps in the geometry (see Figure 8.4a). These gaps will appear larger as the spectator moves further away from the VR player, as noted by our users. Theoretically, the secondary ambient reconstruction resulting from the third-person view (controlled by the spectators) can effectively fill these gaps. However, presently DreamStream is limited in how often this alternate view is rendered and updated. We note that it may be possible to cache texture and geometry information to fill these gaps using computer vision and video analysis techniques.

The second related problem is that when rendering 3D objects from a depth buffer, care must be taken around discontinuity in depth at the edges of objects. Spurious geometry that straddles the edge of an object and the background scene can lead to highly visible artifacts. This can be reduced by omitting triangles that exhibit an unreasonably large change in depth, but eliminating all such artifacts without introducing new ones is difficult. Our shaders used a simple thresholding technique that eliminated all triangles with a large height above a certain fixed value. However, this is not sufficient, and again, this might be addressed by employing computer vision techniques for object segmentation.

Both of these problems might be addressed by analyzing scene geometry as it is drawn. However, this requires knowledge of how geometry is laid out in memory, and how the shaders of these VR applications work. This will be difficult to do in general.

A final issue is that during the rendering of the reconstructions captured using the third-person view, the game may skip the rendering of some objects. Such optimizations are common in 3D rendering, and DreamStream had no control over it.

Scaling to multiple users

While DreamStream's Cross-Dimensional Medium supports multiple users, our user study focused only on one-one interactions between a VR player and a spectator. This is a limitation of our user study. In practice, multiple spectators may use DreamStream to spectate the VR player, and DreamStream's tech stack supports it. Each of them can either use a VR headset or a desktop interface and would be represented in the scene through avatars. A further study is required to shed light on how DreamStream performs in these multi-user scenarios. During the feedback with experts, a challenge they noted is that, unlike text chat, having hundreds of user avatars in a virtual space might not be feasible. It then becomes important for the player to either manually moderate or for the system to automatically determine which spectators the player can see and have beside them. It is also unclear how the interactions among spectators might play out in such a scenario. Nevertheless, interesting access control and social dynamics emerge, requiring careful design choices. For example, a few "super-spectators" may be admitted, whereas the rest can be part of the general audience and view it as they would in a 3D theatre.

8.7 Conclusion

In this chapter, we identified key considerations for designing VR spectator systems and used the insight we gained from prior chapters. We presented DreamStream, which operates at a platform level and use a novel Cross-Dimensional Medium to achieve immersive and interactive spectating for VR experiences. The chapter also proposed a novel approach for sharing and distribution of 3D data that leveraged depth buffers. The system used depth buffers to capture, transmit and reconstruct the geometry of the 3D scene. Using this approach, DreamStream allows viewers to spectate using both 2D desktop-based UI as well as, immersively using a VR-based viewer, allowing for dimensional-detail exploration. The medium allowed users of either interface to interact with the VR player. In addition, the medium meaningfully translated them to make them appear as if the spectators and their interactions were an integral part of the VR scene. Through our user and expert studies, we found that the VR viewer was the preferred interface for the users and offered multiple advantages over others. We hope these findings and discussions inspire future progress in developing VR spectating systems that offer increased immersion.

Chapter 9

Conclusions

9.1 Restatement of contributions

This dissertation explored guidance and collaborative interactions in Mixed Reality environments. It made the following conceptual contributions:

- Extended Spaces (*xspaces*)
  - The concept of Extended Spaces (*xspaces*) and an adapted CSCW matrix that introduces the notion of users being in symmetric or asymmetric *xspaces*.
  - The matrix allows for better classification and understanding of collaborative work using Mixed Reality interfaces.
- Interactive Cross-Dimensional Media (XD media)
  - Novel media and associated interaction techniques that allow users to carry out spatial, temporal, and dimensional-detail explorations of information of interest.
  - A design methodology that leverages the notion of *xspaces* and *XD media* to design systems that enhance user communication and collaboration by mitigating their asymmetry.

The dissertation then showed how these concepts could aid collaboration and guidance in Mixed Reality through four interactive systems and their evaluations:

- A system that facilitates remote instruction of physical tasks using Mixed Reality (Loki).
  - A design space of bi-directional Mixed Reality based remote instruction of physical tasks.
  - Interfaces and interactions that allow users to switch to different configurations of the design space.

  - An Interactive Cross-Dimensional Medium that leverages spatial data, video, annotations, and playback to help connect users as they teach and learn physical tasks.
  - A qualitative evaluation which showed that participants successfully used the system in nuanced ways that exercised the different modes and interactions of the medium to communicate with an instructor within a single learning session.

- A system that facilitates asynchronous video-based instruction of design tasks in VR (TutoriVR).
  - Formative studies that explored the challenges of using video-based tutorials for accomplishing VR design tasks.
  - VR-embedded widgets, which are application interfaces that are observable and interactable directly inside existing commercial VR applications.
  - An Interactive Cross-Dimensional Medium that mitigates the *xspace* asymmetry between the learner and the instructor for video-based instruction of Virtual tasks.
  - An exploratory user evaluation (n=10) showed that the system helped users better understand the VR scene and the instructor's actions recorded in the video tutorials.

- A system that facilitates synchronous interaction between a non-VR external user and a VR user for collaborative tasks in VR (TransceiVR).
  - Formative interviews with experts that explored the goals and challenges of asymmetrical VR interactions.
  - A touch-tablet interface with an Interactive Cross-Dimensional Medium allows an external user to perform different interactions - Perspective and temporal exploration, Annotate the VR scene, Share annotated virtual displays, and Trigger highlights on controllers.
  - A user evaluation of the system (n = 10 pairs) which showed an improved efficiency in communication between the external user and VR user in such interaction scenarios.

- A system that facilitates immersive and interactive spectating of activities carried out in VR (Dreamstream).
  - A novel streaming approach which carries out interventions at the VR platform level, uses 3D capture systems, leverages depth buffers to stream 3D views at scale via conventional encoders such as H.264.
  - An Interactive Cross-Dimensional Medium with immersive and 2D interfaces, which allows the spectators of a VR activity to cohabit and be embodied in the

virtual scene of the VR player as avatars, take independent viewports to explore the scene, and refer to scene elements using laser pointers.

– A user evaluation (n=12) showed that DreamStream's VR viewer interface offered increased immersion and made it easier to communicate and synchronously interact with, and feel co-present with the VR player.

9.2 Strengths of Cross-Dimensional Media

From our experience working on these interactive systems, we now reflect on the key strengths and challenges of Cross-Dimensional Media. We note three key strengths specific to Cross-Dimensional Media interactions that are valuable for guidance and collaboration in Mixed Reality environments: (1) Dimensional Exploration, (2) Inter-dimensional translation, and (3) *xspace* Mutual awareness.

Dimensional Exploration

Cross-Dimensional Media interactions allow users to vary and explore the different dimensions through the different visual representations of the underlying data. Each representation has its merits. Today, 2D videos are popularly understood and a well-supported visual representation to scalably store, transmit and exchange information. Almost any media device today supports the recording and playback of video-based information. Hence, it is inevitable to avoid their usage. However, in the context of conveying information about a Mixed Reality space, they suffer from numerous issues such as shaky first-person view feeds, information occlusion, lack of depth information, lack of ambient spatial awareness and the inability to transition to alternate perspectives. In our interactive systems, we have explored other visual representations beyond 2D video that solve some of these issues: alternate scene viewports, stereo 3D videos, reconstructed 3D interactions of a user, and 3D point cloud reconstructions of a scene. These representations allow for the communication of information that may be ambiguous in a 2D video.

Loki allows a user to view multiple 2D video feeds and a 3D point cloud reconstruction of another user's space. Users can freely explore the 3D point cloud at different scales. They can miniaturize it and get a bird's eye perspective. In addition, they can "jump" into it, and experience it on a full first-person scale (1:1).

In TutoriVR, we have a recorded video of an instructor performing activities in VR. The video contains 3D interactions that the instructor performs in the VR space, as well as 3D information of the scene. Some of this information is lost when converted to a 2D video. For a learner in VR, TutoriVR uses Stereo 3D to enhance the depth of information in regular 2D video captures. In addition, it reconstructs in 3D, the key user gestures and their interactions with their VR controller. During the reconstruction, it stabilizes the VR user's movements, leaving out any other occluding and unnecessary objects that may be present in the original video.

In TransceiVR, we have one user in VR, who is capable of 3D perception and interaction. The other user is external to VR, operating a 2D UI, and is not easily capable of 3D perception or interaction. During a live interaction, TransceiVR allows an external user to independently take alternate viewports into the scene and annotate them. External users can choose to perform annotations in a 2D format where they draw over video frames and share them with the VR user. They can also do it in a 3D format, where their annotations are projected into the VR scene.

Finally, with DreamStream, we have one user who is spectating a VR activity and multiple spectators who can spectate it either immersively using a VR headset or, using a 2D UI with the help of conventional desktop interfaces. Both interfaces perform a 3D reconstruction of the virtual scene along with the 3D live capture of the VR player's body. They allow the spectators to take independent perspectives of the scene. The VR viewer allows the spectators to view the scene in an embodied manner and move around it as if they were there. On the other hand, the 2D UI - based desktop viewer provides a variety of viewing modes that the spectators can choose between depending on their viewing goals. Besides seeing what the VR player sees, the spectators can also explore the parts of the scene that are outside the VR player's view, which are updated live. In contrast, with TransceiVR, the alternate viewports are not live and are updated only when the VR player turns along a specific direction.

Together with dimensional exploration, spatial and temporal exploration are valuable in all these scenarios. However, the latter two have been well studied in prior works and are not specific to Mixed Reality environments. Hence, we skip their discussion here.

Translation of interactions across representations

We have different visual representations of an *xspace*, and a user may be operating on any of the representations. It then becomes important to ensure that any interaction carried out over a specific visual representation is translated in a meaningful manner to other visual representations, especially to those that have a different *output dimensionality*.

In TutoriVR, key information relevant to the VR activity is recorded through stereo captures and logging of the VR HMD and controller poses. These are used to render meaningful and valuable visual representations for the learner - stereo 3D, reconstructed 3D interactions, and 2D video. Since this is asynchronous and uni-directional communication, translation is required only from the VR instructor to the VR learner and not the other way.

In the case of Loki, TransceiVR, and DreamStream, this is not the case. They involve synchronous communication between the two users. 2D annotations over 3D geometry made in TransceiVR must be projected into the correct 3D pose in the VR scene. Alternatively, 2D annotations made over specific frames must be rendered as a shared screen in the VR scene.

Similarly, in Loki, annotations made over a virtual 3D point cloud representation of a space are translated and rendered at the right position in the real-world physical space, when viewed through an Augmented Reality Headset. This is also done for user presence in

a space. If a user is viewing the point cloud of a space from a specific viewpoint, then they are rendered at the corresponding position in the real world as a virtual avatar. This mapping is also done in the reverse direction, from the real world to the 3D point cloud reconstructions. In the case of DreamStream, we have similar interactions where user representations and interactions - Kinect depth captures, avatars, and laser pointer interactions are translated meaningfully between the actual virtual space and its 3D reconstruction.

Mutual Awareness

Cross-Dimensional Media is valuable in facilitating communication and guidance between users operating across asymmetric *xspaces*. In all our works, users operated in asymmetric *xspaces*, and we found that it is important always to provide an ambient awareness of activities carried out by another user. In TutoriVR, this is provided by an awareness widget that anchors the instructor's tutorial video to the FOV of a user as they wander in their own *xspace*. In Loki, miniaturized Hologlyphs provided this, which always showed the live feed of the other user's *xspace*. In TransceiVR, when the external user is not primarily viewing the live feed of the VR user, such as when they share and annotate screens, then live view of the VR user is shown through an inset overlaid over the shared screen. Finally, in DreamStream, users always saw a live 3D reconstruction of the VR scene. To distinguish between the different components, the region that was viewed by the player was always rendered brighter. While not core to Cross-Dimensional Media interactions, these mutual awareness mechanisms are, however, essential when users operate across asymmetric *xspaces*.

9.3 Challenges to Cross-Dimensional Media

Access to calibrated 3D data and multiple camera viewports

A significant challenge for facilitating Cross-Dimensional Media interactions is that it requires access to various 3D information in a virtual and/or a physical environment. Physical environments must be instrumented with depth cameras such as Kinect or iPhone depth cameras. To acquire multiple viewports into the scene, one may need to install more than one camera, which need to be calibrated to get spatially coherent 3D meshes. In virtual environments, there are currently no straightforward and universally accepted mechanisms to get depth information from a VR scene or to spawn spectator cameras at arbitrary locations in the scene without application source access.

Besides the access to the depth data, another concern is obtaining a noise-free 3D reconstruction. Today, most depth-based reconstructions suffer from the issue of depth shadows and noise. Hence, there is a need for high-precision depth data and algorithmic techniques that can improve the quality of reconstructions.

Sensing human activity

When users operate across asymmetric *xspaces*, we see that it is important to keep track of their actions so that they can be communicated to other users. In our works, for activities carried out in a virtual space, we leveraged the HMD and hand pose tracking available through today's VR systems. However, this misses the nuances of facial expressions and body poses. For activities carried out in a physical space, we use the Kinect's 3D point cloud as a proxy. However, this suffers from noise, depth shadows, and occlusions. Solutions such as full-body tracking suits can go a long way in capturing data on human activity, but these are also inconvenient and expensive. Recent works exploring tracking and reconstruction of human activities through RGB cameras [25, 114] could prove valuable to the future of Cross-Dimensional Media interfaces.

Information density versus Cognitive load

When interacting with Cross-Dimensional Media, a user has access to much more information and UI when compared to just a video, such as different dimensional representations and associated UI to browse through and interact with them. When not designed carefully, the UI for these could bloat quickly and hamper the user experience. Therefore, depending upon the nature of the collaboration task, one must carefully balance the trade-off between the density of information presented to the users and the cognitive load required to perceive and interact with this information. This balancing is usually done by identifying the minimum information required for accomplishing a collaboration task's goals, then designing the media interactions around that.

Will there be asymmetric *xspaces* in the *Metaverse* future?

Cross-Dimensional Media interactions help bridge the gap in guidance and collaboration across asymmetric *xspaces*. With the emergence of online virtual spaces such as *Horizon Workrooms* and *Mozilla Hubs*, one may speculate a Metaverse future, where there is no asymmetry; in which everyone uses Mixed Reality displays, operates in virtual worlds, performs virtual tasks, and views all information in 3D. We believe that such a future is not possible. There will always be tasks carried out in the physical world, where one may need to collaborate with users over the internet. Similarly, 2D media and displays will continue to play an active role in many tasks. As Buxton [23] states: "*Everything is best for something and worst for something else*". The new challenge is to figure out how these 3D virtual worlds will interface with our 2D digital interfaces and 3D physical spaces. We hypothesize that Interactive Cross-Dimensional Media will pave the way for this multiplicity.

9.4 Extensions to the dissertation work

This section discusses ways in which the concepts, interfaces, and technologies presented in this dissertation can be extended and improved.

Multi-user and Cross-device interactions

Two key properties of a Cross-Dimensional Medium are that - (1) it has different visual representations of the same data and hence can be viewed using different interfaces, and (2) it can meaningfully translate user interactions across these representations. The most interesting of these interactions are those that go across dimensions, say from VR/AR interfaces to tablets/PCs and vice-versa. These inter-dimensional translations lend themselves naturally to studying how Interactive Cross-Dimensional Media can play a role in facilitating collaborative tasks that require the use of different devices. Such cross-device interactions may be performed by multiple users or a single user who requires the data to be viewed and interacted with, using different devices. A rich history of ongoing research has explored *cross-device interactions* mostly in the context of 2D displays and touch surfaces. Some such work include AirConstellations [116], SurfaceFleet [21], SymbiosisSketch [7], OmniDesk [111], Micro-mobility and F-formations [115], Tablet-grasp + Micro-mobility [199]. Concepts from these works need to be re-visited and explored in the context of Mixed Reality devices and interactions. Some questions to ask are: *Where and how could Mixed Reality devices aid and enhance productive workflows today? How do we design facilitate seamless transition of interactions from Mixed Reality interfaces to conventional interfaces? What types of information needs to be shown using which interfaces? How do we design interfaces that do not add to the cognitive load of the users?*

Non-visual modalities of information presentation

As a natural extension to multi-user and cross-device interactions, it could be important to think about *What types of data need to be shared?* and *How will these data be shared across devices?*. Works in this dissertation focused on challenges that arise from collaboration where the data is dominantly visual. For Cross-Dimensional Media, we relied on four streams of data - RGB video frames, depth data corresponding to the frames, viewport in the physical or virtual space from which the frame was rendered or seen, and finally, gestural or controller actions made by a user.

Visual data is a major component of Mixed Reality systems, but it is just one of the many types of data that users exchange. There are other modalities and types of information that we have not explored, such as spatial audio, text, haptic, smell, etc. Across the dissertation, we saw and addressed challenges in the visual modality. Similarly, these other types of data and modalities will come with their own challenges that need to be considered and addressed. Since each device may have varying computing and interaction capabilities, it is

vital to consider the technological challenges surrounding capturing relevant data, sharing it across devices, and establishing standards for it.

Scaling to large groups of users

This dissertation focuses on collaborative interactions that apply to small groups of users as defined by Grudin [66]. The canonical CSCW matrix and the Extended Space-Time matrix proposed by this work all assume this. These concepts, as well as the different Cross-Dimensional Media interactions used in the systems of these users, do not scale to a large number of users. Lee et al.'s Model of Coordinated Action (MoCA) [109] aims to address larger communities of users. It might be a useful future work to extend MoCA for Mixed Reality interactions. A first step might be to redefine "Dimension 2: Physical Distribution" of MoCA with the notion of *Extended Space*.

With conventional desktop interactions, interaction techniques and methods exist to carry out interactions at scale, such as chat polls and live sentiment analysis of text chats [102]. However, spatial interactions that can scale to many users are less studied. Interactions used in this work, such as annotations, avatar embodiment, and laser pointers, would not work when there are 100s of users. For example, in DreamStream, though the *streaming* technology is scalable, having 100s of avatars in the scene would render the VR application unusable. So it is crucial to ask the question: *How do we consolidate and represent spatial interactions of a large group of users who operate in Mixed Reality?* Techniques from prior work such as Spatial Dynamic Voting [57], Density Maps [144, 31], and Interactive spatial visualizations [39] could be extended to aid with such scalable interactions with users of Mixed Reality environments.

Intelligent and Adaptive User Interfaces

The Interactive Cross-Dimensional Media implemented by systems in this dissertation do not implement any smart and live adaptions to a user's actions. Through our experience working with these systems, we note a couple of avenues where this could be useful.

Context-aware annotations

Annotation interactions were an important part of the Cross-Dimensional Media implemented in Loki (Chapter 5) and TransceiVR (Chapter 7). In these systems, annotations made over a specific *visual representation* were meaningfully translated to other representations. However, these translations were made purely based on geometric relationships and transforms. In TransceiVR, this was done by translating the 2D coordinates of the annotations from the tablet screen space to the 3D coordinates of the VR scene. There are different ways to map gestures on a 2D screen to a 3D space [134], and in TransceiVR, we used the *median depth plane* approach. In this, the annotations were placed on a plane parallel to the plane of the 2D viewport and placed at the median depth of the region enclosed by the

annotation. In Loki, annotations made over a point cloud representation were mapped to the physical space through appropriate Translation, Rotation, and Scale matrices.

These systems did not know the intent of these annotations and relied only on geometry. This can fail in different scenarios. For instance, in TransceiVR, if one were to draw an arrow to refer to an object, instead of anchoring the annotation to the object pointed to by the arrow, the annotation would be naively projected into the scene. Similarly, if one were to annotate to underline an object, the naive projection could lead to an incorrect result. Moreover, these annotations are static, which means that if the scene changes, or the object on which the annotation is made moves, then these annotations do not move along with those objects.

In the future, it would be valuable to have systems that could interpret the intent of a user's annotations and embed them into the different visual representations in a context-aware manner. This requires answering questions such as: *What is the relationship between the shape of an annotation and its intent? How would we need to adapt the annotations based on changes in the physical or virtual? Can the users interact with the annotation itself?* Such questions have been explored in the context of 2D interfaces by systems such as ChalkTalk [135] and RealitySketch [175]. It would be valuable to extend these to suit the 3D nature of Mixed Reality environments.

Reactive tutorials and active guidance

Both Loki and TutoriVR focus on delivering instructions for tasks. Loki allows for a learning experience where the learner may be actively coached by the instructor because of their ability to switch to different styles of instruction.

However, TutoriVR can only deliver passive instructions wherein the learners must browse through the material themselves and learn it without additional support from the software. This means there is no scaffolding, evaluation of progress, and active guidance. It may be valuable to have systems that can actively guide and tutor the learner, using the data available in the Cross-Dimensional Medium.

Both TutoriVR and Loki have access to all the actions of the instructor. Now consider a hypothetical interface that allows a learner to choose a segment of the instructor's recorded instruction for learning which has a particular stroke of interest. A reactive tutorial system [100] can extract the stroke data of the instructor from the medium and use it to actively guide the learner to learn that stroke.

Interactive Cross-Dimensional Media has access to 3D data of the scene and the users' controller pose and interaction data. In addition, it allows for the presentation of these data in various visual representations. Thus, the media can naturally support the development of systems that can provide learners with reactive tutorials and active guidance.

Platform-level support

This dissertation contributes three systems - TutoriVR (Chapter 6), TransceiVR (Chapter 7) and DreamStream (Chapter 8) all of which implemented Interactive Cross-Dimensional Media at a platform level. These systems needed to access 3D data related to the VR scene and user interaction data. Additionally, these systems had to overlay interfaces and information into a VR scene.

TutoriVR implicitly captured and presented the 3D scene data by presenting Stereo Feeds like that of a 3D TV. In TransceiVR, we explicitly compute an approximate depth map of the scene by applying computer vision techniques on the stereo feeds. Finally, in DreamStream, we use sophisticated DLL injection techniques to obtain the exact depth buffer of the scene.

We leveraged platform-level APIs to access the VR user's controller and HMD data, such as its pose and button actions. However, a major caveat of these data is that they do not track when users in the VR scene perform actions such as teleportation or change their scaling. This is because these are application-level changes that are not passed on to the VR platform.

For the rendering of content, TutoriVR and TransceiVR used platform-level graphic overlay methods. However, this renders only planar overlays at a specified 3D pose. Moreover, these overlays do not respect the depth buffer of the scene and thus render over every other element irrespective of the depth. In contrast, DreamStream can render 3D objects into the scene that respect the scene depth. However, as before, this requires sophisticated DLL injection techniques.

Our systems show the value of accessing the 3D data of the VR scene, information regarding the scale and pose of the main camera rendering the VR scene, and the ability to render dynamic 3D content. We hope that the software in the VR platforms of the future will expose these data and interactions more readily for use by researchers and developers.

Opportunities with 3D data

The systems across this dissertation relied heavily on 3D data of VR scenes for providing different kinds of *visual representations*.

Representation

For representing the 3D data, Loki, TransceiVR, and DreamStream used depth maps. Section 8.3 notes its advantages. However, a consequence of using these depth maps is that it leads to depth shadows. This is because a depth buffer is only a 3D representation of the scene from the VR player's point of view. As it is viewed from a perspective away from the point of view, gaps and crevices begin to appear. In recent years, there are various 3D inpainting techniques [159]. Such works could be repurposed to solve the depth shadow issues when using depth buffers.

To mitigate these issues, it would be valuable to explore other ways in which 3D data can be stored, e.g., layered mesh representations [20].

Transmission

Besides storage, it is also essential to ensure that the data can be compressed for reliable transmission. This was a second source for the artifacts in the 3D reconstructions in our systems. Today's encoding-decoding techniques are not suited for depth data. Loki uses lossless compression to achieve this. However the compression achieved is not much and is feasible only for local networks today. In DreamStream, we use H.264 compression. However, H.264 compression is optimized for preserving the visual look of images and not for the structure of depth data. A key limitation of H.264 was that it uses only 8 bits per channel. However, for efficient 3D reconstruction, one could require anywhere from 12-32 bits of precision. We could overcome this by using other encoders that allow for higher bits per channel, such as the HEVC. But this still optimizes for preserving information critical for visual perception. An alternate approach is to leverage neural networks. In recent years, neural networks have improved in accuracy and computational efficiency. Models such as auto-encoders for encoding textures could be better than conventional encoders. An example of this approach is the work by Habibian et al. [73]. Future work needs to explore how these autoencoders can be tweaked to suit the needs of depth buffers as opposed to images.

9.5 Summary

This dissertation explored the challenges users of Mixed Reality interfaces face when they want to spectate, guide, and collaborate with other users. We show that the current techniques to analyze computer-supported cooperative work falls short for Mixed Reality interactions because they have rigid definitions of a workspace that is informed by the characteristics of a *Physical Space*. To address this, we propose the concept of an *Extended Space (xspace)*, which redefines the notion of a workspace by taking into consideration the task-relevant physical as well as the virtual elements of the spaces that users operate in. Using this, we define an Extended Space-Time matrix that introduces the concept of users being in symmetric or asymmetric *xspaces*. When users operate in asymmetric *xspaces*, there exist barriers to effective communication and collaboration. To mitigate these barriers, we propose Interactive Cross-Dimensional Media (XD Media), which are novel media and associated interaction techniques that allow users to carry out spatial, temporal, and dimensional-detail explorations of information of interest. Using these concepts, we analyze challenges in four scenarios where users guide and collaborate using Mixed Reality interfaces - (1) Remote instruction of physical tasks, (2) Asynchronous video-based instruction for VR design tasks, (3) Synchronous video-based collaboration for VR tasks and (4) Synchronous and interactive spectating for VR tasks. For each scenario, we build systems that leverage Interactive Cross-Dimensional Media to address challenges and present their evaluations.

Bibliography

[1] Matt Adcock and Chris Gunn. "Annotating with 'Sticky' Light for Remote Guidance". In: *ACM SIGGRAPH ASIA 2010 Posters*. SA '10. Seoul, Republic of Korea: ACM, 2010, 62:1–62:1. ISBN: 978-1-4503-0524-2. DOI: 10.1145/1900354.1900423. URL: http://doi.acm.org/10.1145/1900354.1900423.

[2] Leila Alem and Jane Li. "A Study of Gestures in a Video-mediated Collaborative Assembly Task". In: *Adv. in Hum.-Comp. Int.* 2011 (Jan. 2011), 1:1–1:7. ISSN: 1687-5893. DOI: 10.1155/2011/987830. URL: http://dx.doi.org/10.1155/2011/987830.

[3] Fraser Anderson et al. "YouMove: Enhancing Movement Training with an Augmented Reality Mirror". In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 2013, pp. 311–320. ISBN: 9781450322683. DOI: 10.1145/2501988.2502045. URL: https://doi-org.libproxy.berkeley.edu/10.1145/2501988.2502045.

[4] JH Erik Andriessen. *Working with groupware: understanding and evaluating collaboration technology*. Springer Science & Business Media, 2012.

[5] Jerome Ardouin et al. "Design and evaluation of methods to prevent frame cancellation in real-time stereoscopic rendering". In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, Mar. 2011, pp. 95–98. ISBN: 978-1-4577-0063-7. DOI: 10.1109/3DUI.2011.5759223. URL: http://ieeexplore.ieee.org/document/5759223/.

[6] Rahul Arora et al. "Experimental Evaluation of Sketching on Surfaces in VR". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 5643–5654. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025474. URL: http://doi.acm.org/10.1145/3025453.3025474.

[7] Rahul Arora et al. "SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 185:1–185:15. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173759. URL: http://doi.acm.org/10.1145/3173574.3173759.

[8] Nikola Banovic et al. "Waken: Reverse Engineering Usage Information and Interface Structure from Software Videos". In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST '12. Cambridge, Massachusetts, USA: ACM, 2012, pp. 83–92. ISBN: 978-1-4503-1580-7. DOI: 10.1145/2380116.2380129. URL: http://doi.acm.org/10.1145/2380116.2380129.

[9] Mayra D Barrera Machuca et al. "Multiplanes: Assisted freehand VR drawing". In: *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17*. New York, New York, USA: ACM Press, 2017, pp. 1–3. ISBN: 9781450354196. DOI: 10.1145/3131785.3131794. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041377794%7B%5C&%7Ddoi=10.1145%7B%5C%%7D2F3131785.3131794%7B%5C&%7DpartnerID=40%7B%5C&%7Dmd5=a1405e893abaa7d58a85c81f9d4914c1%20http://dl.acm.org/citation.cfm?doid=3131785.3131794.

[10] Patrick Baudisch, Nathaniel Good, and Paul Stewart. "Focus plus Context Screens: Combining Display Technology with Visualization Techniques". In: *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. UIST '01. Orlando, Florida: Association for Computing Machinery, 2001, pp. 31–40. ISBN: 158113438X. DOI: 10.1145/502348.502354. URL: https://doi.org/10.1145/502348.502354.

[11] Steve Benford et al. "Embodiments, avatars, clones and agents for multi-user, multi-sensory virtual worlds". In: *Multimedia Systems* 5.2 (1997), pp. 93–104.

[12] Steve Benford et al. "Supporting cooperative work in virtual environments". In: *The Computer Journal* 37.8 (1994), pp. 653–668.

[13] Steve Benford et al. "Three Dimensional Visualization of the World Wide Web". In: *ACM Comput. Surv.* 31.4es (Dec. 1999), 25–es. ISSN: 0360-0300. DOI: 10.1145/345966.346021. URL: https://doi.org/10.1145/345966.346021.

[14] Oliver Bimber and Ramesh Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. Natick, MA, USA: A. K. Peters, Ltd., 2005. ISBN: 1568812302.

[15] *Black Hat Cooperative Game*. 2016. URL: https://www.teamfuturegames.com/.

[16] Sanne MBI Botden et al. "ProMIS augmented reality training of laparoscopic procedures face validity". In: *Simulation in healthcare* 3.2 (2008), pp. 97–102.

[17] *Bots with Blocks Challenge*. 2019. URL: https://medium.com/@JarlanPerez/bots-with-blocks-challenge-50f3e1dde810.

[18] John Brooke et al. "SUS-A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.

[19] Thomas Brox et al. "High Accuracy Optical Flow Estimation Based on a Theory for Warping". In: *Computer Vision - ECCV 2004*. Ed. by Tomás Pajdla and Jiří Matas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 25–36. ISBN: 978-3-540-24673-2.

[20] Michael Broxton et al. "Immersive Light Field Video with a Layered Mesh Representation". In: *ACM Trans. Graph.* 39.4 (July 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392485. URL: https://doi.org/10.1145/3386569.3392485.

[21] Frederik Brudy et al. "SurfaceFleet: Exploring Distributed Interactions Unbounded from Device, Application, User, and Time". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 7–21. ISBN: 9781450375146. DOI: 10.1145/3379337.3415874. URL: https://doi.org/10.1145/3379337.3415874.

[22] Andrius Budrionis et al. "An evaluation framework for defining the contributions of telestration in surgical telementoring". In: *Interactive journal of medical research* 2.2 (2013), e2611.

[23] Bill Buxton. Jan. 2007. URL: http://www.billbuxton.com/multitouchOverview.html.

[24] Marcio C. Cabral, Carlos H. Morimoto, and Marcelo K. Zuffo. "On the Usability of Gesture Interfaces in Virtual Reality Environments". In: *Proceedings of the 2005 Latin American Conference on Human-computer Interaction*. CLIHC '05. Cuernavaca, Mexico: ACM, 2005, pp. 100–108. ISBN: 1-59593-224-0. DOI: 10.1145/1111360.1111370. URL: http://doi.acm.org/10.1145/1111360.1111370.

[25] Zhe Cao et al. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". In: *IEEE transactions on pattern analysis and machine intelligence* 43.1 (2019), pp. 172–186.

[26] Christer Carlsson and Olof Hagsand. "DIVE A multi-user virtual reality system". In: *Proceedings of IEEE virtual reality annual international symposium*. IEEE. 1993, pp. 394–400.

[27] Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. "Mutual Human Actuation". In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Québec City, QC, Canada: Association for Computing Machinery, 2017, pp. 797–805. ISBN: 9781450349819. DOI: 10.1145/3126594.3126667. URL: https://doi.org/10.1145/3126594.3126667.

[28] Li-Te Cheng et al. "Retrofitting Collaboration into UIs with Aspects". In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. CSCW '04. Chicago, Illinois, USA: Association for Computing Machinery, 2004, pp. 25–28. ISBN: 1581138105. DOI: 10.1145/1031607.1031612. URL: https://doi.org/10.1145/1031607.1031612.

[29] Gifford Cheung and Jeff Huang. "Starcraft from the Stands: Understanding the Game Spectator". In: May 2011, pp. 763–772. DOI: 10.1145/1978942.1979053.

[30] Pei-Yu Chi et al. "MixT: Automatic Generation of Step-by-Step Mixed Media Tutorials". In: *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. New York, New York, USA: ACM Press, 2012, p. 93. ISBN: 9781450315807. DOI: 10.1145/2380116.2380130. URL: http://dl.acm.org/citation.cfm?id=2380130%20http://dl.acm.org/citation.cfm?doid=2380116.2380130.

[31] Luca Chittaro and Marta Serafini. "Visualization of User's Behavior in Indoor Virtual Environments Through Interactive Heatmaps". In: *Human-Computer Interaction – INTERACT 2021*. Ed. by Carmelo Ardito et al. Cham: Springer International Publishing, 2021, pp. 600–609. ISBN: 978-3-030-85613-7.

[32] Elizabeth F Churchill and Dave Snowdon. "Collaborative virtual environments: an introductory review of issues and systems". In: *Virtual Reality* 3.1 (1998), pp. 3–15.

[33] Allan Collins, John Seely Brown, and Ann Holum. "Cognitive apprenticeship: Making thinking visible". In: *American educator* 15.3 (1991), pp. 6–11.

[34] Allan Collins, John Seely Brown, and Susan E Newman. "Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics". In: *Knowing, learning, and instruction*. Routledge, 2018, pp. 453–494.

[35] Josh Constine. *Oculus "Quill" Turns VR Painting Into Performance Art*. 2016. URL: https://techcrunch.com/2016/01/26/oculus-quill/.

[36] M.R. Cutkosky et al. "PACT: an experiment in integrating concurrent engineering systems". In: *Computer* 26.1 (1993), pp. 28–37. DOI: 10.1109/2.179153.

[37] Sarah D'Angelo and Andrew Begel. "Improving Communication Between Pair Programmers Using Shared Gaze Awareness". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 6245–6290. ISBN: 9781450346559. DOI: 10.1145/3025453.3025573. URL: https://doi.org/10.1145/3025453.3025573.

[38] Michael F. Deering. "HoloSketch: a virtual reality sketching/animation tool". In: *ACM Transactions on Computer-Human Interaction* 2.3 (1995), pp. 220–238. ISSN: 10730516. DOI: 10.1145/210079.210087.

[39] Jonathan D. Denning, William B. Kerr, and Fabio Pellacini. "MeshFlow: Interactive Visualization of Mesh Construction Sequences". In: *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11* 30.4 (2011), p. 1. DOI: 10.1145/1964921.1964961. URL: http://portal.acm.org/citation.cfm?doid=1964921.1964961.

[40] P. Dewan and J. Riedl. "Toward computer-supported concurrent software engineering". In: *Computer* 26.1 (1993), pp. 17–27. DOI: 10.1109/2.179149.

[41] Alan Dix et al. "Human-computer interaction". In: *Harlow ua* (2000).

[42] Morgan Dixon and James Fogarty. "Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2010, pp. 1525–1534.

[43] Volodymyr Dziubak et al. "Maestro: Designing a System for Real-Time Orchestration of 3D Modeling Workshops". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. Berlin, Germany: Association for Computing Machinery, 2018, pp. 287–298. ISBN: 9781450359481. DOI: 10.1145/3242587.3242606. URL: https://doi-org.libproxy.berkeley.edu/10.1145/3242587.3242606.

[44] James R. Eagan, Michel Beaudouin-Lafon, and Wendy E. Mackay. "Cracking the Cocoa Nut: User Interface Programming at Runtime". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: Association for Computing Machinery, 2011, pp. 225–234. ISBN: 9781450307161. DOI: 10.1145/2047196.2047226. URL: https://doi.org/10.1145/2047196.2047226.

[45] Katharina Emmerich et al. "Streaming VR Games to the Broad Audience: A Comparison of the First-Person and Third-Person Perspectives". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450380966. URL: https://doi.org/10.1145/3411764.3445515.

[46] Douglas C. Engelbart and William K. English. "A Research Center for Augmenting Human Intellect". In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: Association for Computing Machinery, 1968, pp. 395–410. ISBN: 9781450378994. DOI: 10.1145/1476589.1476645. URL: https://doi.org/10.1145/1476589.1476645.

[47] Barrett Ens et al. "Revisiting collaboration through mixed reality: The evolution of groupware". In: *International Journal of Human-Computer Studies* 131 (2019), pp. 81–98.

[48] *Eye in the Sky - VinLia Games*. 2017. URL: https://www.vinliagames.com/.

[49] Ching-Ling Fan et al. "A Survey on 360° Video Streaming: Acquisition, Transmission, and Display". In: *ACM Comput. Surv.* 52.4 (Aug. 2019). ISSN: 0360-0300. DOI: 10.1145/3329119. URL: https://doi.org/10.1145/3329119.

[50] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. "Sketch-sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: ACM, 2011, pp. 373–382. ISBN: 978-1-4503-0716-1. DOI: 10.1145/2047196.2047245. URL: http://doi.acm.org/10.1145/2047196.2047245.

[51] PM Fitts and MI Posner. "Human performance (Basic concepts in psychology)". In: *Belmont Calif.: Brooks/Cole (in eng)* (1967), p. 162.

[52] *Fologram*. URL: https://fologram.com/.

[53] Mike Fraser et al. "Supporting Awareness and Interaction Through Collaborative Virtual Interfaces". In: *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*. UIST '99. Asheville, North Carolina, USA: ACM, 1999, pp. 27–36. ISBN: 1-58113-075-9. DOI: 10.1145/320719.322580. URL: http://doi.acm.org/10.1145/320719.322580.

[54] Taichi Furukawa et al. "TeleSight: Enabling Asymmetric Collaboration in VR Between HMD User and Non-HMD Users". In: *ACM SIGGRAPH 2019 Emerging Technologies*. SIGGRAPH '19. Los Angeles, California: ACM, 2019, 26:1–26:2. ISBN: 978-1-4503-6308-2. DOI: 10.1145/3305367.3335040. URL: http://doi.acm.org/10.1145/3305367.3335040.

[55] Susan R. Fussell, Leslie D. Setlock, and Robert E. Kraut. "Effects of Head-mounted and Scene-oriented Video Systems on Remote Collaboration on Physical Tasks". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 513–520. ISBN: 1-58113-630-7. DOI: 10.1145/642611.642701. URL: http://doi.acm.org/10.1145/642611.642701.

[56] William W. Gaver. "The Affordances of Media Spaces for Collaboration". In: *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*. CSCW '92. Toronto, Ontario, Canada: Association for Computing Machinery, 1992, pp. 17–24. ISBN: 0897915429. DOI: 10.1145/143457.371596. URL: https://doi.org/10.1145/143457.371596.

[57] K. Goldberg et al. "Collaborative online teleoperation with spatial dynamic voting and a human "Tele-Actor"". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*. Vol. 2. 2002, 1179–1184 vol.2. DOI: 10.1109/ROBOT.2002.1014703.

[58] Google. *TiltBrush*. Visited 19-Sep-2019. 2019. URL: http://www.tiltbrush.com/.

[59] *Google Blocks*. 2022. URL: https://arvr.google.com/blocks/.

[60] Floraine Grabler et al. "Generating photo manipulation tutorials by demonstration". In: *ACM Transactions on Graphics* 28.3 (July 2009), p. 1. ISSN: 07300301. DOI: 10.1145/1531326.1531372. URL: http://dl.acm.org/citation.cfm?id=1531372%20http://portal.acm.org/citation.cfm?doid=1531326.1531372.

[61] T P Grantcharov et al. "Randomized clinical trial of virtual reality simulation for laparoscopic skills training". In: *British Journal of Surgery* 91.2 (Nov. 2003), pp. 146–150. ISSN: 0007-1323. DOI: 10.1002/bjs.4407. eprint: https://academic.oup.com/bjs/article-pdf/91/2/146/36728451/bjs4407.pdf. URL: https://doi.org/10.1002/bjs.4407.

[62] Saul Greenberg and David Marwood. "Real Time Groupware as a Distributed System: Concurrency Control and Its Effect on the Interface". In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. CSCW '94. Chapel Hill, North Carolina, USA: Association for Computing Machinery, 1994, pp. 207–217. ISBN: 0897916891. DOI: 10.1145/192844.193011. URL: https://doi.org/10.1145/192844.193011.

[63] Chris Greenhalgh et al. "Applications of temporal links: Recording and replaying virtual environments". In: *Virtual Reality, 2002. Proceedings. IEEE*. IEEE. 2002, pp. 101–108. URL: https://doi.org/10.1109/VR.2002.996512.

[64] Tovi Grossman and George Fitzmaurice. "ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: ACM, 2010, pp. 1515–1524. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753552. URL: http://doi.acm.org/10.1145/1753326.1753552.

[65] Tovi Grossman, Justin Matejka, and George Fitzmaurice. "Chronicle: capture, exploration, and playback of document workflow histories". In: *SIGCHI conference on Human factors in computing systems*. 2010, pp. 143–152. ISBN: 9781450302715. URL: http://dl.acm.org/citation.cfm?id=1866054.

[66] J. Grudin. "Computer-supported cooperative work: history and focus". In: *Computer* 27.5 (1994), pp. 19–26. DOI: 10.1109/2.291294.

[67] Jan Gugenheimer et al. "Challenges Using Head-Mounted Displays in Shared and Social Spaces". In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI EA '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–8. ISBN: 9781450359719. DOI: 10.1145/3290607.3299028. URL: https://doi.org/10.1145/3290607.3299028.

[68] Jan Gugenheimer et al. "FaceDisplay: Towards Asymmetric Multi-User Interaction for Nomadic Virtual Reality". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 54:1–54:13. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173628. URL: http://doi.acm.org/10.1145/3173574.3173628.

[69] Jan Gugenheimer et al. "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*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 4021–4033. ISBN: 9781450346559. URL: https://doi-org.libproxy.berkeley.edu/10.1145/3025453.3025683.

[70] Pavel Gurevich et al. "TeleAdvisor: A Versatile Augmented Reality Tool for Remote Assistance". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 619–622. ISBN:

978-1-4503-1015-4. DOI: 10.1145/2207676.2207763. URL: http://doi.acm.org/10.1145/2207676.2207763.

[71] Carl Gutwin and Saul Greenberg. "A descriptive framework of workspace awareness for real-time groupware". In: *Computer Supported Cooperative Work (CSCW)* 11.3 (2002), pp. 411–446.

[72] Carl Gutwin et al. "Supporting Informal Collaboration in Shared-Workspace Groupware." In: *J. Univers. Comput. Sci.* 14.9 (2008), pp. 1411–1434.

[73] Amirhossein Habibian et al. "Video compression with rate-distortion autoencoders". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7033–7042.

[74] William A. Hamilton, Oliver Garretson, and Andruid Kerne. "Streaming on Twitch: Fostering Participatory Communities of Play Within Live Mixed Media". In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 1315–1324. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557048. URL: http://doi.acm.org/10.1145/2556288.2557048.

[75] Chad Harms and Frank Biocca. "Internal consistency and reliability of the networked minds measure of social presence". In: (2004).

[76] Sandra G Hart and Lowell E Staveland. "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research". In: *Advances in psychology*. Vol. 52. Elsevier, 1988, pp. 139–183.

[77] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. New York, NY, USA: Cambridge University Press, 2003. ISBN: 0521540518.

[78] Jeremy Hartmann, Yen-Ting Yeh, and Daniel Vogel. "AAR: Augmenting a Wearable Augmented Reality Display with an Actuated Head-Mounted Projector". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 445–458. ISBN: 9781450375146. DOI: 10.1145/3379337.3415849. URL: https://doi.org/10.1145/3379337.3415849.

[79] Jeremy Hartmann et al. "RealityCheck: Blending Virtual Environments with Situated Physical Reality". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: ACM, 2019, 347:1–347:12. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300577. URL: http://doi.acm.org/10.1145/3290605.3300577.

[80] Steven J Henderson and Steven K Feiner. *Augmented reality for maintenance and repair (armar)*. Tech. rep. Columbia Univ New York Dept of Computer Science, 2007.

[81] Steven J. Henderson and Steven K. Feiner. "Augmented reality in the psychomotor phase of a procedural task". In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 2011, pp. 191–200. DOI: 10.1109/ISMAR.2011.6092386.

[82] Hickeys. *Kinect for windows - windows apps*. URL: https://developer.microsoft.com/en-us/windows/kinect.

[83] Jim Hollan and Scott Stornetta. "Beyond Being There". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: Association for Computing Machinery, 1992, pp. 119–125. ISBN: 0897915135. DOI: 10.1145/142750.142769. URL: https://doi.org/10.1145/142750.142769.

[84] Julian Hough et al. "Timing and grounding in motor skill coaching interaction: Consequences for the information state". In: *Proceedings of the 19th SemDial Workshop on the Semantics and Pragmatics of Dialogue (goDIAL)*. 2015.

[85] *How tech companies can help upskill the U.S. workforce*. Sept. 2021. URL: https://hbr.org/2018/02/how-tech-companies-can-help-upskill-the-u-s-workforce.

[86] H.H. Hu et al. "Visual cues for imminent object contact in realistic virtual environments". In: *Proceedings Visualization 2000. VIS 2000*. November 2015. IEEE, 2000, pp. 179–185. ISBN: 0-7803-6478-3. DOI: 10.1109/VISUAL.2000.885692. URL: http://ieeexplore.ieee.org/document/885692/.

[87] Weidong Huang, Leila Alem, and Franco Tecchia. "HandsIn3D: Augmenting the Shared 3D Visual Space with Unmediated Hand Gestures". In: *SIGGRAPH Asia 2013 Emerging Technologies*. SA '13. Hong Kong, Hong Kong: ACM, 2013, 10:1–10:3. ISBN: 978-1-4503-2632-2. DOI: 10.1145/2542284.2542294. URL: http://doi.acm.org/10.1145/2542284.2542294.

[88] Weidong Huang et al. "HandsInTouch: Sharing Gestures in Remote Collaboration". In: *Proceedings of the 30th Australian Conference on Computer-Human Interaction*. OzCHI '18. Melbourne, Australia: ACM, 2018, pp. 396–400. ISBN: 978-1-4503-6188-0. DOI: 10.1145/3292147.3292177. URL: http://doi.acm.org/10.1145/3292147.3292177.

[89] Edwin L Hutchins, James D Hollan, and Donald A Norman. "Direct manipulation interfaces". In: *Human-computer interaction* 1.4 (1985), pp. 311–338.

[90] H. Ishii. "TeamWorkStation: Towards a Seamless Shared Workspace". In: *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work*. CSCW '90. Los Angeles, California, USA: Association for Computing Machinery, 1990, pp. 13–26. ISBN: 0897914023. DOI: 10.1145/99332.99337. URL: https://doi.org/10.1145/99332.99337.

[91] Hiroshi Ishii and Minoru Kobayashi. "ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: Association for Computing Machinery, 1992, pp. 525–532. ISBN: 0897915135. DOI: 10.1145/142750.142977. URL: https://doi.org/10.1145/142750.142977.

[92] Hiroshi Ishii, Minoru Kobayashi, and Kazuho Arita. "Iterative Design of Seamless Collaboration Media". In: *Commun. ACM* 37.8 (Aug. 1994), pp. 83–97. ISSN: 0001-0782. DOI: 10.1145/179606.179687. URL: https://doi.org/10.1145/179606.179687.

[93] Shahram Izadi et al. "Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media". In: *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST '03. Vancouver, Canada: Association for Computing Machinery, 2003, pp. 159–168. ISBN: 1581136366. DOI: 10.1145/964696.964714. URL: https://doi.org/10.1145/964696.964714.

[94] Christopher M Janelle et al. "Maximizing performance feedback effectiveness through videotape replay and a self-controlled learning environment". In: *Research quarterly for exercise and sport* 68.4 (1997), pp. 269–279.

[95] Robert Johansen. *Groupware: Computer support for business teams*. The Free Press, 1988.

[96] Brett Jones et al. "RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units". In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, pp. 637–644. ISBN: 9781450330695. DOI: 10.1145/2642918.2647383. URL: https://doi.org/10.1145/2642918.2647383.

[97] Shunichi Kasahara and Jun Rekimoto. "JackIn: Integrating First-Person View with out-of-Body Vision Generation for Human-Human Augmentation". In: *Proceedings of the 5th Augmented Human International Conference*. AH '14. Kobe, Japan: Association for Computing Machinery, 2014. ISBN: 9781450327619. DOI: 10.1145/2582051.2582097. URL: https://doi.org/10.1145/2582051.2582097.

[98] Rubaiat Habib Kazi et al. "Draco: Bringing Life to Illustrations with Kinetic Textures". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 351–360. ISBN: 9781450324731. DOI: 10.1145/2556288.2556987. URL: https://doi.org/10.1145/2556288.2556987.

[99] Rubaiat Habib Kazi et al. "Kitty: Sketching Dynamic and Interactive Illustrations". In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, pp. 395–405. ISBN: 9781450330695. DOI: 10.1145/2642918.2647375. URL: https://doi.org/10.1145/2642918.2647375.

[100] Caitlin Kelleher and Randy Pausch. "Stencils-Based Tutorials: Design and Evaluation". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. Portland, Oregon, USA: Association for Computing Machinery, 2005, pp. 541–550. ISBN: 1581139985. DOI: 10.1145/1054972.1055047. URL: https://doi.org/10.1145/1054972.1055047.

[101] David Kirk and Danae Stanton Fraser. "Comparing Remote Gesture Technologies for Supporting Collaborative Physical Tasks". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: ACM, 2006, pp. 1191–1200. ISBN: 1-59593-372-7. DOI: 10.1145/1124772.1124951. URL: http://doi.acm.org/10.1145/1124772.1124951.

[102] Konstantin Kobs et al. "Emote-Controlled: Obtaining Implicit Viewer Feedback Through Emote-Based Sentiment Analysis on Comments of Popular Twitch.Tv Channels". In: *Trans. Soc. Comput.* 3.2 (Apr. 2020). ISSN: 2469-7818. DOI: 10.1145/3365523. URL: https://doi-org.libproxy.berkeley.edu/10.1145/3365523.

[103] Iwan de Kok et al. "A multimodal system for real-time action instruction in motor skill learning". In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 2015, pp. 355–362.

[104] David A Kolb. *Experiential learning: Experience as the source of learning and development*. FT press, 2014.

[105] Jan Kolkmeier et al. "With a Little Help from a Holographic Friend: The OpenIMPRESS Mixed Reality Telepresence Toolkit for Remote Collaboration Systems". In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. Tokyo, Japan: Association for Computing Machinery, 2018. ISBN: 9781450360869. DOI: 10.1145/3281505.3281542. URL: https://doi.org/10.1145/3281505.3281542.

[106] Simone Kriglstein et al. "Be Part Of It: Spectator Experience in Gaming and Esports". In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–7. ISBN: 9781450368193. DOI: 10.1145/3334480.3375153. URL: https://doi-org.libproxy.berkeley.edu/10.1145/3334480.3375153.

[107] André Kunert et al. "Photoportals: Shared References in Space and Time". In: *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work amp; Social Computing*. CSCW '14. Baltimore, Maryland, USA: Association for Computing Machinery, 2014, pp. 1388–1399. ISBN: 9781450325400. DOI: 10.1145/2531602.2531727. URL: https://doi.org/10.1145/2531602.2531727.

[108] Hideaki Kuzuoka. "Spatial Workspace Collaboration: A SharedView Video Support System for Remote Collaboration Capability". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: ACM, 1992, pp. 533–540. ISBN: 0-89791-513-5. DOI: 10.1145/142750.142980. URL: http://doi.acm.org/10.1145/142750.142980.

[109] Charlotte P. Lee and Drew Paine. "From The Matrix to a Model of Coordinated Action (MoCA): A Conceptual Framework of and for CSCW". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*. CSCW '15. Vancouver, BC, Canada: Association for Computing Machinery,

2015, pp. 179–194. ISBN: 9781450329224. DOI: 10.1145/2675133.2675161. URL: https://doi.org/10.1145/2675133.2675161.

[110] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. "Expanding Video Game Live-Streams with Enhanced Communication Channels: A Case Study". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 1571–1576. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025708. URL: http://doi.acm.org/10.1145/3025453.3025708.

[111] Zhen Li et al. "SMAC: A Simplified Model of Attention and Capture in Multi-Device Desk-Centric Environments". In: *Proc. ACM Hum.-Comput. Interact.* 3.EICS (June 2019). DOI: 10.1145/3300961. URL: https://doi.org/10.1145/3300961.

[112] David Lindlbauer and Andy D. Wilson. "Remixed Reality: Manipulating Space and Time in Augmented Reality". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173703. URL: https://doi.org/10.1145/3173574.3173703.

[113] Haohua Lyu et al. "WebTransceiVR: Asymmetrical Communication Between Multiple VR and Non-VR Users Online". In: *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 2022, pp. 1–7.

[114] Shugao Ma et al. "Pixel Codec Avatars". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 64–73.

[115] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. "Cross-Device Interaction via Micro-Mobility and f-Formations". In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST '12. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2012, pp. 13–22. ISBN: 9781450315807. DOI: 10.1145/2380116.2380121. URL: https://doi.org/10.1145/2380116.2380121.

[116] Nicolai Marquardt et al. "AirConstellations: In-Air Device Formations for Cross-Device Interaction via Multiple Spatially-Aware Armatures". In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. UIST '21. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 1252–1268. ISBN: 9781450386357. DOI: 10.1145/3472749.3474820. URL: https://doi.org/10.1145/3472749.3474820.

[117] Sebastian Marwecki et al. "VirtualSpace - Overloading Physical Space with Multiple Virtual Reality Users". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 241:1–241:10. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173815. URL: http://doi.acm.org/10.1145/3173574.3173815.

[118] Justin Matejka, Tovi Grossman, and George Fitzmaurice. "Ambient Help". In: *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, 2011, p. 2751. ISBN: 9781450302289. DOI: 10.1145/1978942.1979349. URL: http://dl.acm.org/citation.cfm?id=1979349%20http://dl.acm.org/citation.cfm?doid=1978942.1979349.

[119] Michael Venturino Maxwell J. Wells. "Performance and head movements using a helmet-mounted display with different sized fields-of-view". In: *Optical Engineering* 29.8 (1990), pp. 870 - 877 –8. DOI: 10.1117/12.55672. URL: https://doi.org/10.1117/12.55672.

[120] *Microsoft Cloud*. URL: https://www.microsoft.com/he-il/enterprise/manufacturing.

[121] Helen C Miles et al. "A review of virtual environments for training in ball sports". In: *Computers & Graphics* 36.6 (2012), pp. 714–726.

[122] Paul Milgram et al. "Augmented reality: A class of displays on the reality-virtuality continuum". In: *Telemanipulator and telepresence technologies*. Vol. 2351. International Society for Optics and Photonics. 1995, pp. 282–292.

[123] Peter Mohr et al. "Retargeting Video Tutorials Showing Tools With Surface Contact to Augmented Reality". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. New York, New York, USA: ACM Press, 2017, pp. 6547–6558. ISBN: 9781450346559. DOI: 10.1145/3025453.3025688. URL: http://dl.acm.org/citation.cfm?doid=3025453.3025688.

[124] Emerson Murphy-Hill et al. "How Do Users Discover New Tools in Software Development and Beyond?" In: *Comput. Supported Coop. Work* 24.5 (Oct. 2015), pp. 389–422. ISSN: 0925-9724. DOI: 10.1007/s10606-015-9230-9. URL: https://doi.org/10.1007/s10606-015-9230-9.

[125] Brad A Myers et al. "Interacting at a Distance: Measuring the Performance of Laser Pointers and Other Devices". In: *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*. 4. New York, New York, USA: ACM Press, 2002, p. 33. ISBN: 1581134533. DOI: 10.1145/503376.503383. URL: http://portal.acm.org/citation.cfm?doid=503376.503383.

[126] Alok Mysore and Philip J. Guo. "Torta: Generating Mixed-Media GUI and Command-Line App Tutorials Using Operating-System-Wide Activity Tracing". In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Québec City, QC, Canada: Association for Computing Machinery, 2017, pp. 703–714. ISBN: 9781450349819. DOI: 10.1145/3126594.3126628. URL: https://doi.org/10.1145/3126594.3126628.

[127] Toshio Nakamura and Takeo Igarashi. "An Application-independent System for Visualizing User Operation History". In: *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. UIST '08. Monterey, CA, USA: ACM, 2008, pp. 23–32. ISBN: 978-1-59593-975-3. DOI: 10.1145/1449715.1449721. URL: http://doi.acm.org/10.1145/1449715.1449721.

[128] Jackie Neider, Tom Davis, and Mason Woo. *OpenGL programming guide*. Vol. 478. Addison-Wesley Reading, MA, 1993.

[129] *Nemesis Perspective*. 2017. URL: https://www.evocatgames.com/nemesis-perspective/.

[130] Cuong Nguyen and Feng Liu. "Making Software Tutorial Video Responsive". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. New York, New York, USA: ACM Press, 2015, pp. 1565–1568. ISBN: 9781450331456. DOI: 10.1145/2702123.2702209. URL: http://dl.acm.org/citation.cfm?doid=2702123.2702209.

[131] Cuong Nguyen et al. "CollaVR: Collaborative In-Headset Review for VR Video". In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Québec City, QC, Canada: Association for Computing Machinery, 2017, pp. 267–277. ISBN: 9781450349819. DOI: 10.1145/3126594.3126659. URL: https://doi.org/10.1145/3126594.3126659.

[132] Cuong Nguyen et al. "Depth Conflict Reduction for Stereo VR Video Interfaces". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 64:1–64:9. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173638. URL: http://doi.acm.org/10.1145/3173574.3173638.

[133] Christopher Niederauer et al. "Non-Invasive Interactive Visualization of Dynamic Architectural Environments". In: *Proceedings of the 2003 Symposium on Interactive 3D Graphics*. I3D '03. Monterey, California: Association for Computing Machinery, 2003, pp. 55–58. ISBN: 1581136455. DOI: 10.1145/641480.641493. URL: https://doi-org.libproxy.berkeley.edu/10.1145/641480.641493.

[134] Benjamin Nuernberger et al. "Interpreting 2D gesture annotations in 3D augmented reality". In: *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. 2016, pp. 149–158. DOI: 10.1109/3DUI.2016.7460046.

[135] Gabriel Barbosa Nunes and Ken Perlin. "Atypical: A Type System for Live Performances". In: *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Québec City, QC, Canada: Association for Computing Machinery, 2017, pp. 61–62. ISBN: 9781450354196. DOI: 10.1145/3131785.3131812. URL: https://doi.org/10.1145/3131785.3131812.

[136] *Oculus Asymmetric Field of View*. 2019. URL: https://developer.oculus.com/documentation/quest/latest/concepts/unity-asymmetric-fov-faq/.

[137] *Oculus Medium*. 2018. URL: https://www.oculus.com/medium/learn/.

[138] Ohan Oda et al. "Virtual Replicas for Remote Assistance in Virtual and Augmented Reality". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, pp. 405–415. ISBN: 9781450337793. DOI: 10.1145/2807442.2807497. URL: https://doi.org/10.1145/2807442.2807497.

[139] *OpenCV Brox Optical Flow*. 2019. URL: https://docs.opencv.org/3.4.2/d7/d18/classcv_1_1cuda_1_1BroxOpticalFlow.html.

[140] Sergio Orts-Escolano et al. "Holoportation: Virtual 3D Teleportation in Real-Time". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, pp. 741–754. ISBN: 9781450341899. DOI: 10.1145/2984511.2984517. URL: https://doi.org/10.1145/2984511.2984517.

[141] Doug Palmer et al. "Annotating with Light for Remote Guidance". In: *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*. OZCHI '07. Adelaide, Australia: ACM, 2007, pp. 103–110. ISBN: 978-1-59593-872-5. DOI: 10.1145/1324892.1324911. URL: http://doi.acm.org/10.1145/1324892.1324911.

[142] *Panoptic Asymmetrical VR game*. 2019. URL: http://panopticgame.com/.

[143] Ken Perlin, Zhenyi He, and Karl Rosenberg. "Chalktalk: A Visualization and Communication Language–As a Tool in the Domain of Computer Science Education". In: *arXiv preprint arXiv:1809.07166* (2018).

[144] Thies Pfeiffer. "Measuring and Visualizing Attention in Space with 3D Attention Volumes". In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA '12. Santa Barbara, California: Association for Computing Machinery, 2012, pp. 29–36. ISBN: 9781450312219. DOI: 10.1145/2168556.2168560. URL: https://doi.org/10.1145/2168556.2168560.

[145] Karine Pires and Gwendal Simon. "YouTube Live and Twitch: A Tour of User-generated Live Streaming Systems". In: *Proceedings of the 6th ACM Multimedia Systems Conference*. MMSys '15. Portland, Oregon: ACM, 2015, pp. 225–230. ISBN: 978-1-4503-3351-1. DOI: 10.1145/2713168.2713195. URL: http://doi.acm.org/10.1145/2713168.2713195.

[146] Catherine Plaisant et al. "The design of history mechanisms and their use in collaborative educational simulations". In: *Proceedings of the 1999 conference on Computer support for collaborative learning - CSCL '99*. Vol. 11. 99. Morristown, NJ, USA: Association for Computational Linguistics, 1999, 44–es. ISBN: HCIL Technical Report No. 99-11. DOI: 10.3115/1150240.1150284. URL: http://portal.acm.org/citation.cfm?doid=1150240.1150284.

[147] Suporn Pongnumkul et al. "Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: Association for Computing Machinery, 2011, pp. 135–144. ISBN: 9781450307161. DOI: 10.1145/2047196.2047213. URL: https://doi.org/10.1145/2047196.2047213.

[148] Kevin Ponto, Joe Kohlmann, and Michael Gleicher. "Effective Replays and Summarization of Virtual Experiences". In: *IEEE Transactions on Visualization and Computer Graphics* 18.4 (Apr. 2012), pp. 607–616. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.41. URL: http://dx.doi.org/10.1109/TVCG.2012.41.

[149] *Quill*. 2018. URL: https://quill.fb.com/tutorials/.

[150] *Quill*. 2020. URL: https://quill.fb.com/.

[151] Abhishek Ranjan, Jeremy P. Birnholtz, and Ravin Balakrishnan. "Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 1177–1186. ISBN: 9781595935939. DOI: 10.1145/1240624.1240802. URL: https://doi.org/10.1145/1240624.1240802.

[152] Stephan Reichelt et al. "Depth cues in human visual perception and their realization in 3D displays". In: ed. by Bahram Javidi et al. April. Apr. 2010, 76900B. ISBN: 9780819481542. DOI: 10.1117/12.850094. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.850094.

[153] Richard K Reznick. "Teaching and testing technical skills". In: *The American journal of surgery* 165.3 (1993), pp. 358–361.

[154] T. Richardson et al. "Virtual network computing". In: *IEEE Internet Computing* 2.1 (1998), pp. 33–38. DOI: 10.1109/4236.656066.

[155] Hazim Sadideen and Roger Kneebone. "Practical skills teaching in contemporary surgical education: how can educational theory be applied to promote effective learning?" In: *The American Journal of Surgery* 204.3 (2012), pp. 396–401.

[156] Manolis Savva et al. "ReVision: Automated Classification, Analysis and Redesign of Chart Images". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: ACM, 2011, pp. 393–402. ISBN: 978-1-4503-0716-1. DOI: 10.1145/2047196.2047247. URL: http://doi.acm.org/10.1145/2047196.2047247.

[157] Steven Schkolne, Michael Pruett, and Peter Schröder. "Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools". In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '01*. New York, New York, USA: ACM Press, 2001, pp. 261–268. ISBN: 1581133278. DOI: 10.1145/365024.365114. URL: http://portal.acm.org/citation.cfm?id=365114%20http://portal.acm.org/citation.cfm?doid=365024.365114.

[158] Weiming Shen, Qi Hao, and Weidong Li. "Computer Supported Collaborative Design: Retrospective and Perspective". In: *Comput. Ind.* 59.9 (Dec. 2008), pp. 855–862. ISSN: 0166-3615. DOI: 10.1016/j.compind.2008.07.001. URL: https://doi.org/10.1016/j.compind.2008.07.001.

[159] Meng-Li Shih et al. "3d photography using context-aware layered depth inpainting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8028–8038.

[160] Ben Shneiderman. "Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces". In: *Proceedings of the 2Nd International Conference on Intelligent User Interfaces*. IUI '97. Orlando, Florida, USA: ACM, 1997, pp. 33–39. ISBN: 0-89791-839-8. DOI: 10.1145/238218.238281. URL: http://doi.acm.org/10.1145/238218.238281.

[161] Roland Sigrist. "Visual and auditory augmented concurrent feedback in a complex motor task". In: *Presence* 20.1 (2011), pp. 15–32.

[162] Roland Sigrist et al. "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review". In: *Psychonomic bulletin & review* 20.1 (2013), pp. 21–53.

[163] Thomas Smith, Marianna Obrist, and Peter Wright. "Live-streaming Changes the (Video) Game". In: *Proceedings of the 11th European Conference on Interactive TV and Video*. EuroITV '13. Como, Italy: ACM, 2013, pp. 131–138. ISBN: 978-1-4503-1951-5. DOI: 10.1145/2465958.2465971. URL: http://doi.acm.org/10.1145/2465958.2465971.

[164] David N. Snowdon, Elizabeth F. Churchill, and Alan J. Munro. "Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction". In: *Collaborative Virtual Environments*. Ed. by Elizabeth F. Churchill, David N. Snowdon, and Alan J. Monro. London: Springer, 2001. Chap. 1, pp. 3–17.

[165] Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. "LightGuide: Projected Visualizations for Hand Movement Guidance". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 179–188. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2207702. URL: http://doi.acm.org/10.1145/2207676.2207702.

[166] Maximilian Speicher et al. "360Anywhere: Mobile Ad-hoc Collaboration in Any Environment Using 360 Video and Augmented Reality". In: *Proc. ACM Hum.-Comput. Interact.* 2.EICS (June 2018), 9:1–9:20. ISSN: 2573-0142. DOI: 10.1145/3229091. URL: http://doi.acm.org/10.1145/3229091.

[167] Samantha Stahlke, James Robb, and Pejman Mirza-Babaei. "The Fall of the Fourth Wall: Designing and Evaluating Interactive Spectator Experiences". In: *International Journal of Gaming and Computer-Mediated Simulations* 10 (July 2018), pp. 42–62. DOI: 10.4018/IJGCMS.2018010103.

[168] *Steam :: Steamvr :: Transitioning to openxr*. July 2020. URL: https://steamcommunity.com/games/250820/announcements/detail/2522527900755718764.

[169] Richard Stoakley, Matthew J. Conway, and Randy Pausch. "Virtual Reality on a WIM: Interactive Worlds in Miniature". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 265–272. ISBN: 0201847051. DOI: 10.1145/223904.223938. URL: https://doi.org/10.1145/223904.223938.

[170] Chengzheng Sun and Clarence Ellis. "Operational Transformation in Real-Time Group Editors: Issues, Algorithms, and Achievements". In: *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*. CSCW '98. Seattle, Washington, USA: Association for Computing Machinery, 1998, pp. 59–68. ISBN: 1581130090. DOI: 10.1145/289444.289469. URL: https://doi.org/10.1145/289444.289469.

[171] Geng Sun and Nick Holliman. "Evaluating methods for controlling depth perception in stereoscopic cinematography". In: *Proc. SPIE*. Ed. by Andrew J. Woods, Nicolas S. Holliman, and John O. Merritt. Vol. 7237. Feb. 2009, p. 72370I. DOI: 10.1117/12.807136. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=811252%7B%5C%%7D5Cnhttp://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.807136%20http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.807136.

[172] Hemant Bhaskar Surale et al. "TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: ACM, 2019, 13:1–13:13. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300243. URL: http://doi.acm.org/10.1145/3290605.3300243.

[173] Ivan Sutherland. "The ultimate display". In: (1965).

[174] Ivan E. Sutherland. "A Head-Mounted Three Dimensional Display". In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: Association for Computing Machinery, 1968, pp. 757–764. ISBN: 9781450378994. DOI: 10.1145/1476589.1476686. URL: https://doi.org/10.1145/1476589.1476686.

[175] Ryo Suzuki et al. "RealitySketch: Augmented Reality Sketching for Real-Time Embedded and Responsive Visualizations". In: *SIGGRAPH Asia 2021 Real-Time Live!* SA '21. Tokyo, Japan: Association for Computing Machinery, 2021. ISBN: 9781450390729. DOI: 10.1145/3478511.3491313. URL: https://doi.org/10.1145/3478511.3491313.

[176] John C. Tang and Scott L. Minneman. "Videodraw: A Video Interface for Collaborative Drawing". In: *ACM Trans. Inf. Syst.* 9.2 (Apr. 1991), pp. 170–184. ISSN: 1046-8188. DOI: 10.1145/123078.128729. URL: https://doi.org/10.1145/123078.128729.

[177] Richard Tang et al. "Physio@Home: Exploring Visual Guidance and Feedback Techniques for Physiotherapy Exercises". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: Association for Computing Machinery, 2015, pp. 4123–4132. ISBN: 9781450331456. DOI: 10.1145/2702123.2702401. URL: https://doi.org/10.1145/2702123.2702401.

[178] Burak S. Tekin and Stuart Reeves. "Ways of Spectating: Unravelling Spectator Participation in Kinect Play". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 1558–1570. ISBN: 9781450346559. DOI: 10.1145/3025453.3025813. URL: https://doi-org.libproxy.berkeley.edu/10.1145/3025453.3025813.

[179] Theophilus Teo et al. "Mixed Reality Remote Collaboration Combining 360 Video and 3D Reconstruction". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: ACM, 2019, 201:1–201:14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300431. URL: http://doi.acm.org/10.1145/3290605.3300431.

[180] Kasim Terzić and Miles Hansard. "Methods for reducing visual discomfort in stereoscopic 3D: A review". In: *Signal Processing: Image Communication* 47 (2016), pp. 402–416. ISSN: 09235965. DOI: 10.1016/j.image.2016.08.002.

[181] Sonoda Tetsuri and Anders Grunnet-Jepsen. *Depth image compression by colorization for Intel® RealSense™ Depth Cameras*. URL: https://dev.intelrealsense.com/docs/depth-image-compression-by-colorization-for-intel-realsense-depth-cameras.

[182] Balasaravanan Thoravi Kumaravel and Björn Hartmann. "Interactive Mixed-Dimensional Media for Cross-Dimensional Collaboration in Mixed Reality Environments". In: *Frontiers in Virtual Reality* (2022), p. 9.

[183] Balasaravanan Thoravi Kumaravel and Andrew D Wilson. "DreamStream: Immersive and Interactive Spectating for VR". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517508. URL: https://doi.org/10.1145/3491102.3517508.

[184] Balasaravanan Thoravi Kumaravel et al. "Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence". In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 161–174. ISBN: 9781450368162. DOI: 10.1145/3332165.3347872. URL: https://doi.org/10.1145/3332165.3347872.

[185] Balasaravanan Thoravi Kumaravel et al. "TransceiVR: Bridging Asymmetrical Communication Between VR Users and External Collaborators". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 182–195. ISBN: 9781450375146. DOI: 10.1145/3379337.3415827. URL: https://doi.org/10.1145/3379337.3415827.

[186] Balasaravanan Thoravi Kumaravel et al. "TutoriVR: A Video-Based Tutorial System for Design Applications in Virtual Reality". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300514. URL: https://doi.org/10.1145/3290605.3300514.

[187] Eduardo Velloso, Andreas Bulling, and Hans Gellersen. "MotionMA: Motion Modelling and Analysis by Demonstration". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, pp. 1309–1318. ISBN: 9781450318990. DOI: 10.1145/2470654.2466171. URL: https://doi.org/10.1145/2470654.2466171.

[188] *VR headsets, games, and metaverse life: United States*. URL: https://www.vive.com/us/product/vive-virtual-reality-system/.

[189] Chiu-Hsuan Wang et al. "Slice of Light: Transparent and Integrative Transition Among Realities in a Multi-HMD-User Environment". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 805–817. ISBN: 9781450375146. URL: https://doi.org/10.1145/3379337.3415868.

[190] Jeremy Warner et al. "ElectroTutor: Test-Driven Physical Computing Tutorials". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. Berlin, Germany: Association for Computing Machinery, 2018, pp. 435–446. ISBN: 9781450359481. DOI: 10.1145/3242587.3242591. URL: https://doi.org/10.1145/3242587.3242591.

[191] Gerold Wesche and Hans-Peter Seidel. "FreeDrawer - A free-form sketching system on the responsive workbench". In: *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '01*. New York, New York, USA: ACM Press, 2001, p. 167. ISBN: 1581134274. DOI: 10.1145/505008.505041. URL: http://portal.acm.org/citation.cfm?doid=505008.505041.

[192] *What is augmented reality?* July 2022. URL: https://www.ptc.com/en/technologies/augmented-reality.

[193] Sean White, Levi Lister, and Steven Feiner. "Visual hints for tangible gestures in augmented reality". In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*. Ismar. IEEE, Nov. 2007, pp. 1–4. ISBN: 9781424417506.

DOI: 10.1109/ISMAR.2007.4538824. URL: http://ieeexplore.ieee.org/document/4538824/.

[194] E. Wiese et al. "Investigating the Learnability of Immersive Free-hand Sketching". In: *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. SBIM '10. Annecy, France: Eurographics Association, 2010, pp. 135–142. ISBN: 978-3-905674-25-5. URL: http://dl.acm.org/citation.cfm?id=1923363.1923387.

[195] Julius von Willich et al. "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*. DIS '19. San Diego, CA, USA: ACM, 2019, pp. 487–496. ISBN: 978-1-4503-5850-7. DOI: 10.1145/3322276.3322334. URL: http://doi.acm.org/10.1145/3322276.3322334.

[196] Gabriele Wulf, Charles Shea, and Rebecca Lewthwaite. "Motor skill learning and performance: a review of influential factors". In: *Medical education* 44.1 (2010), pp. 75–84.

[197] Haijun Xia et al. "Spacetime: Enabling Fluid Individual and Collaborative Editing in Virtual Reality". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. Berlin, Germany: Association for Computing Machinery, 2018, pp. 853–866. ISBN: 9781450359481. DOI: 10.1145/3242587.3242597. URL: https://doi.org/10.1145/3242587.3242597.

[198] Ungyeon Yang and Gerard Jounghyun Kim. "Implementation and Evaluation of "Just Follow Me": An Immersive, VR-Based, Motion-Training System". In: *Presence: Teleoperators and Virtual Environments* 11.3 (2002), pp. 304–323. ISSN: 1054-7460. DOI: 10.1162/105474602317473240.

[199] Dongwook Yoon et al. "Sensing Tablet Grasp + Micro-Mobility for Active Reading". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software amp; Technology*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, pp. 477–487. ISBN: 9781450337793. DOI: 10.1145/2807442.2807510. URL: https://doi.org/10.1145/2807442.2807510.

[200] *Zed Mini - mixed-reality camera*. URL: https://www.stereolabs.com/zed-mini.

[201] Yuhang Zhao et al. "SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–14. ISBN: 9781450359702. DOI: 10.1145/3290605.3300341. URL: https://doi.org/10.1145/3290605.3300341.