MERGING THE REAL AND THE VIRTUAL: AN EXPLORATION OF INTERACTION METHODS TO BLEND REALITIES

by

JEREMY HARTMANN

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirements for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2022

EXAMINING COMMITTEE MEMBERSHIP

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Derek Reilly Associate Professor, Faculty of Computer Science, Dalhousie University
Supervisor	Daniel Vogel Associate Professor, School of Computer Science, University of Waterloo
Internal Member	Edward Lank Professor, School of Computer Science, University of Waterloo
Internal Member	Craig Kaplan Associate Professor, School of Computer Science, University of Waterloo
Internal-external Member	Colin Ellard Professor, Department of Psychology, University of Waterloo

AUTHOR'S DECLARATION

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

STATEMENT OF CONTRIBUTIONS

Sections of this document have appeared in publications or are forthcoming (at the time of writing). In all cases, permission has been granted by the publisher for these works to appear here. Below, the publisher's copyright notice and/or disclaimer is given, with thesis chapter(s) and corresponding publication(s) noted.

Association for Computing Machinery

portions of chapter 4

Jeremy Hartmann, Aakar Gupta, and Daniel Vogel. 2020. Extend, Push, Pull: Smartphone Mediated Interaction in Spatial Augmented Reality via Intuitive Mode Switching. *In Symposium on Spatial User Interaction* (SUI '20). Association for Computing Machinery, New York, NY, USA, Article 2, 1–10. https://doi.org/10.1145/3385959.3418456

portions of chapter 5

Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D. Wilson. 2019. RealityCheck: Blending Virtual Environments with Situated Physical Reality. *In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 347, 1–12. https://doi.org/10.1145/3290605.3300577

portions of chapter 6

Jeremy Hartmann, Yen-Ting Yeh, and Daniel Vogel. 2020. 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). Association for Computing Machinery, New York, NY, USA, 445–458. http://dx.doi.org/10.1145/3379337.3415849

Elsevier & RELX Group plc

Copyright © 2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license https://creativecommons.org/licenses/by-nc-nd/4.0/

portions of chapter 3

Jeremy Hartmann, and Daniel Vogel. 2021. An examination of mobile phone pointing in surface mapped spatial augmented reality. International Journal of Human-Computer Studies, 153, 102662. https://doi.org/10.1016/j.ijhcs.2021.102662

ABSTRACT

We investigate, build, and design interaction methods to merge the real with the virtual. An initial investigation looks at spatial augmented reality (SAR) and its effects on pointing with a real mobile phone. A study reveals a set of trade-offs between the raycast, viewport, and direct pointing techniques. To further investigate the manipulation of virtual content within a SAR environment, we design an interaction technique that utilizes the distance that a user holds mobile phone away from their body. Our technique enables pushing virtual content from a mobile phone to an external SAR environment, interact with that content, rotate-scale-translate it, and pull the content back into the mobile phone. This is all done in a way that ensures seamless transitions between the real environment of the mobile phone and the virtual SAR environment. To investigate the issues that occur when the physical environment is hidden by a fully immersive virtual reality (VR) HMD, we design and investigate a system that merges a realtime 3D reconstruction of the real world with a virtual environment. This allows users to freely move, manipulate, observe, and communicate with people and objects situated in their physical reality without losing their sense of immersion or presence inside a virtual world. A study with VR users demonstrates the affordances provided by the system and how it can be used to enhance current VR experiences. We then move to AR, to investigate the limitations of optical see-through HMDs and the problem of communicating the internal state of the virtual world with unaugmented users. To address these issues and enable new ways to visualize, manipulate, and share virtual content, we propose a system that combines a wearable SAR projector. Demonstrations showcase ways to utilize the projected and head-mounted displays together, such as expanding field of view, distributing content across depth surfaces, and enabling bystander collaboration. We then turn to videogames to investigate how spectatorship of these virtual environments can be enhanced through expanded video rendering techniques. We extract and combine additional data to form a cumulative 3D representation of the live game environment for spectators, which enables each spectator to individually control a personal view into the stream while in VR. A study shows that users prefer spectating in VR when compared with a comparable desktop rendering.

ACKNOWLEDGMENTS

I feel incredibly lucky to have Daniel Vogel as an advisor, whose guidance and encouragement made the completion of this thesis possible. I was also fortunate to have an outstanding and well-rounded committee: Edward Lank, Craig Kaplan, Colin Ellard, and my external examiner, Derek Reilly. There are many students, faculty members, and administrative staff at the University of Waterloo who contributed directly and indirectly. A special thanks to all my coauthors and mentors: Andrew D. Wilson, Stephen DiVerdi, Christian Holz, Cuong Nguyen, Eyal Ofek, Yen-Ting Yeh, Aakar Gupta, Katherine Fennedy, Quentin Roy, Simon T Perrault, Johann Wentzel, Daekun Kim, Hemant Surale, Nikhita Joshi, Antony Albert Raj Irudayaraj, Géry Casiez, and Matthew Lakier; who helped guide and support me. Of course, I owe a debt of gratitude to Lynn, family, and friends who encouraged and supported me throughout.

TABLE OF CONTENTS

Examining Committee Membership ii
Author's Declaration iii
Statement of Contributions iv
Abstract vi
Acknowledgments vii
List of Figures x
List of Tables xii
1 INTRODUCTION 1
1.1 Research Objectives and Overview 3
1.2 Contributions 6
1.3 Dissertation Outline 8
2 BACKGROUND LITERATURE 9
2.1 Definitions of Reality 11
2.2 Systems and Techniques to Blend Reality 14
2.3 Enhancing the Spectatorship of Virtual Environments 17
2.4 Pointing in Immersive Environments 19
2.5 Spatial Interaction On and Around the Body 20
2.6 Hybrid Displays and Collaboration 21
2.7 Summary 23
3 POINTING IN SPATIAL AUGMENTED REALITY 24
3.1 Related Work 25
3.2 Mobile Phone Pointing in SAR 27
3.3 Experiment 1: Ad-hoc SAR Setting 29
3.4 Experiment 2: Simulated SAR Pointing Task 36
3.5 Discussion 41
3.6 Summary 42
4 SMARTPHONE MEDIATED INTERACTION IN SPATIAL AUGMENTED REALITY 43
4.1 Related Work 44
4.2 Design Overview 45
4.3 Implementation and Applications 49
4.4 Study 1: Extended vs Retracted Classification 51
4.5 Study 2: PILOT Usability Evaluation 57
4.6 Discussion and Future Work 58
4.7 Summary 60
5 BLENDING VIRTUAL ENVIRONMENTS WITH SITUATED PHYSICAL REALITY 61
5.1 Related Work 62
5.2 Blending virtual reality with reality 64
5.3 Compositing Real and Virtual Worlds 66
5.4 Compositing Techniques 70
5.5 User Evaluation 75

5.6 Limitations 79
5.7 Discussion and Future Work 80
5.8 Summary 81
6 AUGMENTING A WEARABLE AUGMENTED REALITY DISPLAY WITH AN ACTUATED HEAD-MOUNTED PROJECTOR 82
6.1 Related Work 83
6.2 Augmented Augmented Reality 86
6.3 Usage Scenarios and Applications 89
6.4 Actuated Projector for an AR HMD 93
6.5 AAR Software Toolkit 98
6.6 Developer Toolkit Study 100
6.7 Discussion 102
6.8 Summary 104
7 SPECTATORSHIP OF VIDEOGAME LIVE STREAMS IN VIRTUAL REALITY 105
7.1 Related Work 106
7.2 Enhanced Videogame Spectatorship 107
7.3 System Architecture 110
7.4 Spectator Viewer 115
7.5 User Study 117
7.6 Discussion 122
7.7 Summary 125
8 CONCLUSION 126

BIBLIOGRAPHY 131

LIST OF FIGURES

Figure 1.1 The Virtuality Continuum 2
Figure 1.2 Research Path 4
Figure 2.1 The Sword of Democles. 10
Figure 2.2 Illustration of the VirtualSpace system. 15
Figure 2.3 Illustration and depiction of the RoomAlive system. 16
Figure 2.4 Examples of videogame spectatorship in VR. 18
Figure 2.5 Pointing in table-top spatial augmented reality. 19
Figure 2.6 Demonstration of AR eye-box expansion techniques. 21
Figure 2.7 Examples from the FoveAR system. 22
Figure 3.1 Mobile phone pointing in surface mapped SAR. 25
Figure 3.2 Illustrations of mobile phone pointing techniques. 27
Figure 3.3 Illustration of the SAR experiment environment. 31
Figure 3.4 Error rate of pointing techniques. 33
Figure 3.5 Movement time of pointing techniques. 34
Figure 3.6 Movement time by occlusion of pointing techniques. 35
Figure 3.7 Illustration of experimental setup. 37
Figure 3.8 Movement time by between technique and target view. 40
Figure 4.1 Illustration of push, pull, and extend. 44
Figure 4.2 Interaction state diagram 46
Figure 4.3 Illustration of RST manipulation. 48
Figure 4.4 An example of spatially distributed content in SAR. 50
Figure 4.5 Example scenarios for RST manipulation. 52
Figure 4.6 Illustration of target placements. 53
Figure 4.7 Demonstration of a trial task. 54
Figure 4.8 Depiction of 3D pointing volumes. 55
Figure 4.9 Qualitative study results. 57
Figure 5.1 Visualization of current VR chaperone systems. 62
Figure 5.2 Demonstrative examples of blending techniques. 63
Figure 5.3 Overview of the 3D compositor system. 66
Figure 5.4 Illustration of the VR rendering pipeline. 68
Figure 5.5 Illustration of a depth buffer capture technique. 68
Figure 5.6 An augmented HMD with its composited view. 69
Figure 5.7 Visualization of rendering and compositing techniques. 70
Figure 5.8 Examples of using real-life props as guides. 71
Figure 5.9 A demonstration of colour transfer techniques in VR. 72
Figure 5.10 Geometry interaction between real and virtual items. 73
Figure 5.11 Depicts a virtual space visualized with SAR. 74
Figure 5.12 Room layout used in experimental study. 76
Figure 5.13 Preference ratings across participants. 78
Figure 6.1 System overview and use cases. 83
Figure 6.2 AR persistence example application. 84
Figure 6.3 Illustration of the AAR design space. 87

Figure 6.4 Example demonstrations of AAR scenarios. 89
Figure 6.5 Sharing AR content example. 91
Figure 6.6 Ambient display examples for an actuated HMD. 92
Figure 6.7 Actuated projector HMD. 94
Figure 6.8 View-pair correlation matrix and view frustums. 95
Figure 6.9 Illustration of projector mound geometry. 97
Figure 6.10 Illustration of actuated projector movement. 98
Figure 6.11 Blending virtual objects between a SAR and AR. 99
Figure 6.12 Field of view of a HMD and projector. 100
Figure 7.1 Enhanced videogame livestreaming examples 106
Figure 7.2 Spectator experience levels. 108
Figure 7.3 System diagram. 111
Figure 7.4 Visual representations of different rendering approaches. 116
Figure 7.5 Example interactions with streaming geometry. 118
Figure 7.6 Depictions of the VR spectating task. 119
Figure 7.7 Overall preference ratings. 120

LIST OF TABLES

Table 4.1 Random forest classifier results. 56

1 INTRODUCTION

Virtual, augmented, and mixed reality hold the promise to fundamentally change how we work, live, and play. They can overlay digital information on top of our physical environment, providing us with an immersive interactive experience, or they can replace our environment entirely, transporting us to new and exciting digital worlds. Yet these experiences continue to remain isolated from other people around us. This creates an imbalance between the ones who have access to these technologies and those that are left without such virtual enhancements. An artificial dichotomy of our own making, and one that limits the power and usefulness virtual experiences could bring.

One of the first known uses of the term *virtual reality* is attributed to the French poet and playwright Antonin Artaud, who in 1938 described his avant-garde work in theatre as "la réalité virtuelle" [3]. He saw the characters, images, and objects as constituting a type of virtual reality that a spectator takes part in through the act of *looking* [222]. Though this application of virtual reality could be considered peripheral to how we use the term today, it remains important as a reference point alluding to how the virtual is intrinsically part of our physical reality while also simultaneously removed from it. However, it was not until 1968 that this conceptual idea of the virtual would become actualized by Ivan Sutherland's seminal work on AR and VR displays [246].

Research into virtual reality continued to push the state-of-the-art in areas such as computer graphics, optics, and human-computer interaction. Our conceptual understanding of virtual reality became solidified with the publication of Milgram and Kishino's work on the taxonomy of mixed reality displays, which is used as reference point for all current discourse today [168, 169] (Figure 1.1). With the growing number of high-fidelity consumer head-mounted displays (HMD), research into aspects of the continuum have seen a resurgence, where there is renewed interest investigating aspects like haptics [4, 42], environment blending [161], and co-location resolution strategies [156].

While a significant amount of work has investigated the techniques and technology to enable AR and VR [14, 109], there has been less focus on the methods needed to navigate the space between each type of reality from both an interaction and phenomenological perspective. We use the term and concept of "phenomenological experience" to denote this since we are specifically interested in the user's relationship with the system as well as their subjective point of view when navigating between differing states of reality [233]. Phenomenology has historically been used as a tool to investigate the qualitative aspects of a user's personal experience [97], with focus on the appearance, meaning, and experience the user has while "being in the world" [92]. Specifically, we focus on the individual's conscience experience as they transition from one form of reality to another. In particular, we are

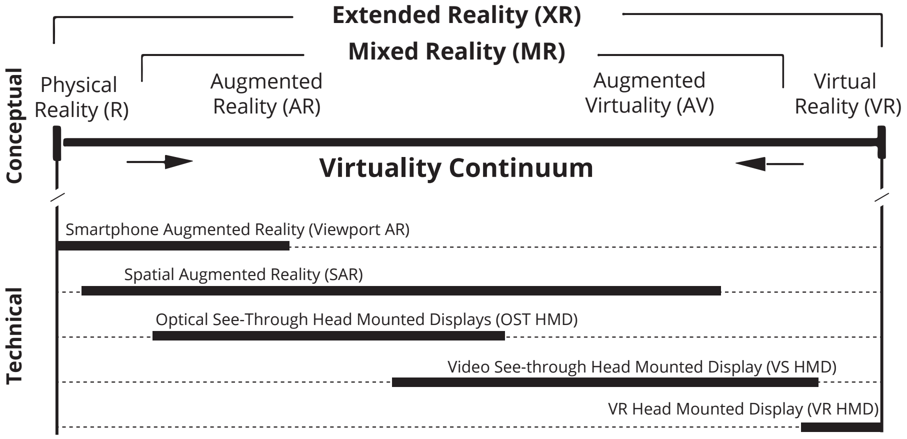

Extended Reality (XR)
Mixed Reality (MR)
Physical Reality (R)
Augmented Reality (AR)
Augmented Virtuality (AV)
Virtual Reality (VR)
Conceptual
Virtuality Continuum
Technical
Smartphone Augmented Reality (Viewport AR)
Spatial Augmented Reality (SAR)
Optical See-Through Head Mounted Displays (OST HMD)
Video See-through Head Mounted Display (VS HMD)
VR Head Mounted Display (VR HMD)

Figure 1.1: The Virtuality Continuum (modified from Milgram et al. [169]). The figure depicts both the conceptual continuum of the real and virtual, and how specific technical implementations map onto them.

interested in how an individual can direct their focus, interactions, and objects across different states of reality.

We position this as distinct from other forms of investigations that look at ways a system behaves or functions to achieve some goal, such as shuffling multiple users around a room while in VR [156]. Another example can be seen in the work by McGill et al. who investigated the effect of different levels of reality on user task performance when wearing a VR HMD [161]. We push these ideas and concepts further by focusing on the techniques, visualizations, and interactions required to enhance a user's transition between different forms and implementations of extended reality with a focus on their subjective points of view from the context of user centred studies.

In this thesis, we present our observations, studies, and software systems that explore extended realty from an implementation and conceptual standpoint. We first lay down a foundation with an investigation of pointing, one of the most fundamental areas of human-computer interaction studies. Our goal is to understand pointing and manipulation tasks within a projection-based subset of AR called Spatial Augmented Reality (SAR). We specifically look at pointing within the context of a smartphone using three techniques that utilize different aspects of the real and the virtual: raycast, which enables the users to interact with the virtual content at a distance; viewport, which bridges a virtualized reality view on the phone with the augmented environment the user is in; and direct, which provides direct interaction with the virtual content through physical contact with reality. Then, based on observational data, we extend this to explore the techniques needed to bridge interactions between the virtual content on the phone with the content that is in SAR.

Second, we investigate how the user experience can be enhanced through the blending of a virtualized reality [117] with a virtual environment. We explore the effect blending has on specific tasks that span physical and virtual reality, which include manipulation of physical objects, communication between co-located users, and the awareness between virtual and physical environments. We then explore the tools and frameworks needed to enable these kind of interactions.

Third, we explore the user's experience when viewing virtual objects through particular technical implementations of augmented reality, and propose a design space that seeks to enumerate the specific advantages these implementations have on the user experience. We combine an optical see-through based AR display with an inside-out SAR system to create an environment in which the user can manipulate digital data across private and public contexts. We describe an implementation of such a system and explore specific usage scenarios that take advantage of its unique qualities across the two implementations of AR.

Finally, we investigate the performer-spectator relationship between a videogame streamer and the remote user who is watching it in a virtual environment. We examine the effect VR has on the spectator experience when viewing the videogame stream under immersive and non-immersive contexts. From this, we explore the systems and infrastructure needed to make such an experience possible.

1.1 RESEARCH OBJECTIVES AND OVERVIEW

The high-level research objective of thesis can be stated as:

Investigate the systems, techniques, and interactions that guide the user's subjective experiences, awareness, and tasks between different states of reality.

We investigate this as a series of primary research problems that address specific issues relating to our high level research objective. We split the questions between four categories. An overview can be seen in Figure 1.2. The first category of questions addresses pointing, attention, and object transfer between a physical reality and a virtual environment using a mobile phone:

(a) How does spatial augmented reality affect pointing when using a mobile phone?

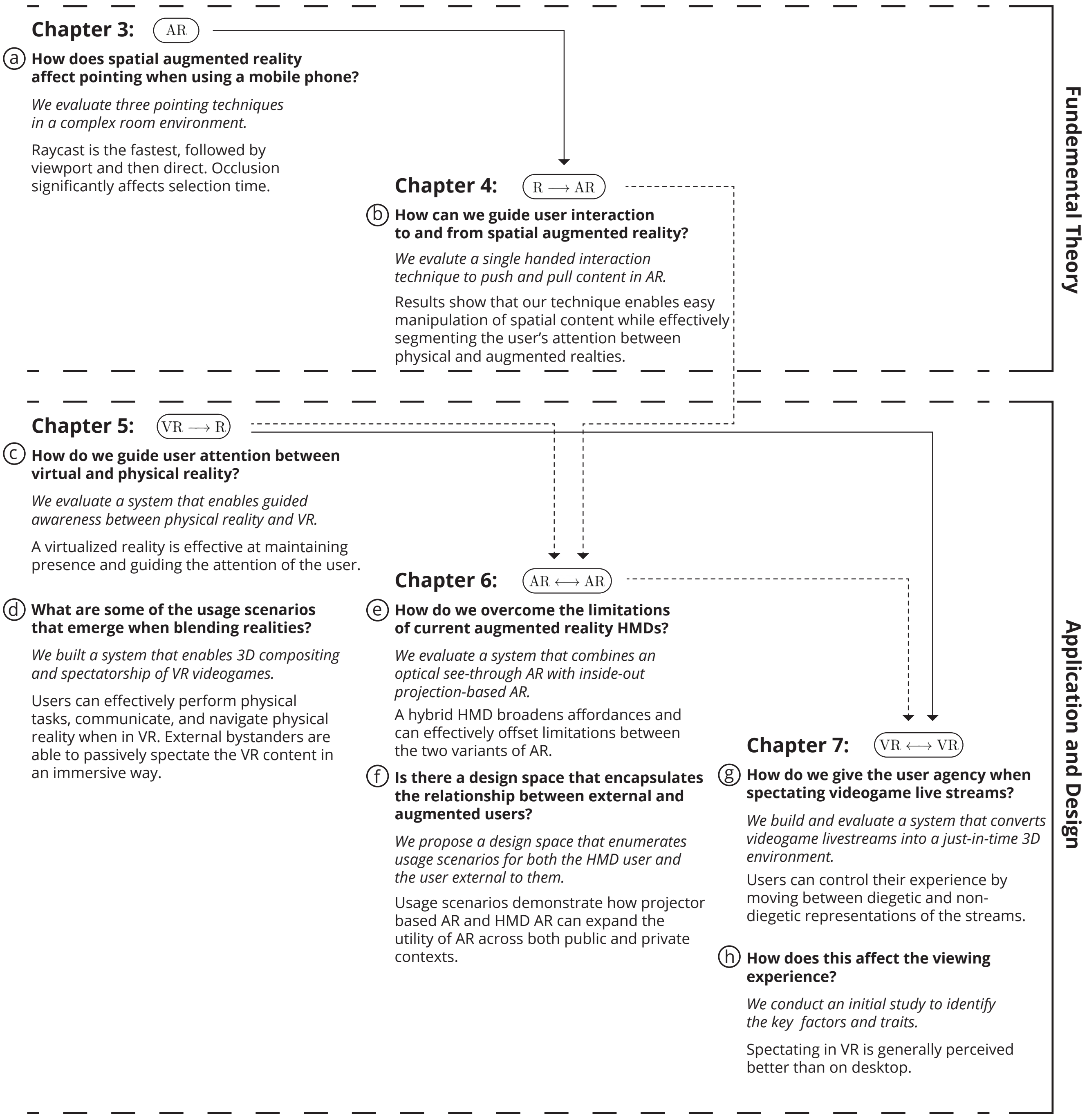
(b) How do we guide user interaction and attention between spatial augmented reality and a mobile phone?

The second category of questions investigates the isolation the user experiences when in a virtual reality environment:

(c) How do we guide user attention between virtual and physical reality?

(d) What are some of the techniques and usage scenarios that emerge when doing so?

The third category of questions explore the design space, usage scenarios, and user experience when transitioning between two variants of augmented reality:



Chapter 3:

(a) **How does spatial augmented reality affect pointing when using a mobile phone?**

We evaluate three pointing techniques in a complex room environment.

Raycast is the fastest, followed by viewport and then direct. Occlusion significantly affects selection time.

Chapter 4:

(b) **How can we guide user interaction to and from spatial augmented reality?**

We evalute a single handed interaction technique to push and pull content in AR.

Results show that our technique enables easy manipulation of spatial content while effectively segmenting the user's attention between physical and augmented realties.

Chapter 5:

(c) **How do we guide user attention between virtual and physical reality?**

We evaluate a system that enables guided awareness between physical reality and VR.

A virtualized reality is effective at maintaining presence and guiding the attention of the user.

(d) **What are some of the usage scenarios that emerge when blending realities?**

We built a system that enables 3D compositing and spectatorship of VR videogames.

Users can effectively perform physical tasks, communicate, and navigate physical reality when in VR. External bystanders are able to passively spectate the VR content in an immersive way.

Chapter 6:

(e) **How do we overcome the limitations of current augmented reality HMDs?**

We evaluate a system that combines an optical see-through AR with inside-out projection-based AR.

A hybrid HMD broadens affordances and can effectively offset limitations between the two variants of AR.

(f) **Is there a design space that encapsulates the relationship between external and augmented users?**

We propose a design space that enumerates usage scenarios for both the HMD user and the user external to them.

Usage scenarios demonstrate how projector based AR and HMD AR can expand the utility of AR across both public and private contexts.

Chapter 7:

(g) **How do we give the user agency when spectating videogame live streams?**

We build and evaluate a system that converts videogame livestreams into a just-in-time 3D environment.

Users can control their experience by moving between diegetic and non-diegetic representations of the streams.

(h) **How does this affect the viewing experience?**

We conduct an initial study to identify the key factors and traits.

Spectating in VR is generally perceived better than on desktop.

Figure 1.2: Research path showing research problems, activities, and main results. Bold text is the research problem statement; italic text is the research activity; and the final block of text is the high-level contribution that influences the next phases of research. A sold line (——) indicates a direct connection, where previous research directly influences the next. A dashed line (- - - -) indicates an indirect connection, where there is high-level influence not tied to specific results. The capsules indicate the direction of attention and interaction from one reality to another.

(e) How can we effectively design a system that blends the user's experience between physical reality, spatial augmented reality, and augmented reality?

(f) Is there a design space that can encapsulate the relationship between external and augmented users of this system?

The final category investigates the user experience when blending between two virtual realities that are composed of diegetic (part of the videogame environment) or non-diegetic (part of the spectators environment) instances of the virtual environment generated when watching live streaming videogames:

(g) How do we design a system to give a user agency when spectating videogame live streams in virtual reality?

(h) What aspects of physical and virtual reality effect the user's viewing experience?

To address these research problems, we took the following steps (also illustrated in Figure 1.2). For each step we investigate points along the Virtuality Continuum (Figure 1.1), where the transitions between points are denoted by directional (→ and ←) or bi-directional (↔) arrows:

1. To answer the first question (a), we conduct a study in a spatial augmented reality environment in order to evaluate three pointing techniques using a mobile phone. The three techniques utilize the properties of the mobile phone to enable raycast, viewport, and direct pointing. $\boxed{AR}$

2. To investigate mobile phone pointing and manipulation in a spatial augmented reality environment (b), we designed an interaction technique that can guide user attention and their digital content from a mobile phone to a spatially augmented version of their environment. $\boxed{R \longleftrightarrow AR}$

3. To answer questions (c) and (d), we built a system that merges the physical reality around the user with a virtual videogame environment. We specifically investigate how the attention of the user can be guided from a VR context to the real world by blending elements of reality into different videogame environments. $\boxed{VR \longrightarrow R}$

4. For questions (e) and (f) we seek to understand the user's experience when they are simultaneously within a spatial augmented reality and a optical see-through (OST) HMD augmented reality experience. To investigate this, we built a hybrid HMD that combines an OST display with a projection-based spatial augmented reality projector. We show how the advantages of one can offset the other through a design space that enumerates specific usage scenarios that take advantage of these specific forms of AR. $\boxed{AR \longleftrightarrow AR}$

5. To answer the last two questions (g) and (h), we design a system that enables spectators of videogame live streams to have agency over their

experience by letting them be "in" the game as it happens. This lets them navigate between different visualizations of the videogame stream, moving from non-diegetic representations to diegetic ones while in both a VR and desktop context. $\boxed{VR \longleftrightarrow VR}$

1.2 CONTRIBUTIONS

We summarize our contributions by project. For each, we outline the methodology used and the key results that form our contributions.

Pointing in Spatial Augmented Reality

In chapter 3, we investigate mobile phone pointing in Spatial Augmented Reality. Three pointing techniques are compared: raycast, viewport, and direct. A first experiment examines these techniques in a realistic five-projector SAR environment with representative targets distributed across different surfaces. Participants were permitted free movement, so variations in target occlusion and incident angle occurred naturally. A second experiment validates and further generalizes findings by strictly controlling target occlusion and orientation in a simulated SAR pointing task using an AR HMD. Overall, results show raycast is fastest for non-occluded targets, direct is most accurate, and fastest for occluded targets in close proximity, and viewport falls in between.

Mediated Interaction in Spatial Augmented Reality

In chapter 4, we investigate how mobile phones can be used to mediate the manipulation of smartphone-based content in SAR. A major challenge is in seamlessly transitioning a phone between its use as a smartphone to its use as a controller for SAR. Most users are familiar with hand extension as a way for using a remote control for SAR. We therefore propose to use hand extension as an intuitive mode switching mechanism for switching back and forth between the mobile interaction mode and the spatial interaction mode. Based on this intuitive mode switch, our technique enables the user to push smartphone content to an external SAR environment, interact with the external content, rotate-scale-translate it, and pull the content back into the smartphone, all the while ensuring no conflict between mobile interaction and spatial interaction. To ensure feasibility of hand extension as mode switch, we evaluate the classification of extended and retracted states of the smartphone based on the phone's relative 3D position with respect to the user's head while varying user postures, surface distances, and target locations. Our results show that a random forest classifier can classify the extended and retracted states with a 96% accuracy on average.

Blending Virtual Environments with Situated Physical Reality

In chapter 5, we build and evaluate a system that guides a user's awareness between their virtual reality environment and the physical world. Current VR systems offer "chaperone" rendering techniques that prevent the user from colliding with physical objects. However, without a detailed geometric model of the physical world, these techniques offer limited possibility for more advanced compositing between the real world and the virtual. We explore this using a realtime 3D reconstruction of the real world that can be combined with a virtual environment. Our system allows users to freely move, manipulate, observe, and communicate with people and objects situated in their physical space without losing the sense of immersion or presence inside their virtual environment. We demonstrate our approach with seven existing VR titles, and describe compositing approaches that address the potential conflicts when rendering the real world and a virtual environment together. A study with frequent VR users demonstrated the affordances provided by our system and how it can be used to enhance current VR experiences.

Augmenting a Wearable Augmented Reality Display with an Actuated Head-Mounted Projector

In chapter 6, we investigate the limitations of current optical see-through (OST) HMDs and present solutions that can overcome them. These current devices create an isolated experience for the user, as the virtual environment they are observing is hidden from all external bystanders. Further, current AR displays have a limited field of view and can suffer from vergence-accommodation conflicts between the focal plane of the virtual object and its distance from the user. To address these issues and enable new ways to visualize, manipulate, and share virtual content, we introduce a system that combines a wearable AR display with a wearable spatial augmented reality projector. To explore this idea, our system combines a head-mounted actuated pico projector with a Hololens AR headset. Projector calibration uses a modified structure from motion (SfM) pipeline to reconstruct the geometric structure of the pan-tilt actuator axes and offsets. A toolkit encapsulates a set of high-level functionality to manage content placement relative to each augmented display and the physical environment. Demonstrations showcase ways to utilize the projected and head-mounted displays together, such as expanding field of view, distributing content across depth surfaces, and enabling bystander collaboration.

Spectatorship of Videogame Live Streams in Virtual Reality

In chapter 7, we build and evaluate a system that takes a videogame livestream and converts it into a VR environment. Current approaches capture the player's rendered RGB view of the game, which is then encoded and streamed as a 2D live video feed to a remote audience. We take this 2D view as a starting point and extend this to also capture the depth buffer, camera pose, and projection matrix from the rendering pipeline of the videogame. We then package all this within an MPEG-4 media container for distribution. Combining these additional data streams with the RGB view of the videogame, our system is able to build a real-time, cumulative 3D representation of the live game environment for spectators. This enables each spectator to individually control a personal game view in 3D. We explore the impact of this enhanced spectatorship across a desktop display and a VR HMD to explore what advantages and trade-offs each afford the spectators watching.

1.3 DISSERTATION OUTLINE

The remainder of this document is organized as follows (see also Figure 1.2):

In chapter 2, we summarize relevant background literature pertaining to pointing in large-display and extended reality environments, around body interaction, head-mounted projective displays, hybrid AR display systems, blending different types of realities, and room-scale immersive experiences.

In chapter 3, we describe the methodology, results, and model for pointing in spatial augmented reality with a mobile phone device. We then discuss subjective preference data and then give design recommendations for pointing in SAR.

In chapter 4, we describe an interaction technique that utilizes the advantages of a smartphone display with a spatial augmented reality environment. Two controlled experiments are presented, one where we evaluate our extend gesture and another to collect preference data from users using the technique.

In chapter 5, we present a system that is able to composite a 3D reconstruction of a physical environment within the context of existing VR videogames. We demonstrate how this is used to guide a user's attention between their VR environment the physical world, which enables them to move and manipulate physical objects. We then describe an experiment which investigates the effects of blending between VR and physical reality

In chapter 6, we present a hybrid AR HMD that combines an optical see-through display with an actuated head-mounted projector. We then describe a series of usage scenarios that cover a concise two dimensional design space which is followed by a discussion on the design considerations, limitations, and future extensions.

In chapter 7, we present a system that enables enhanced 3D videogame livestreams for desktop and VR. We then describe the methodology, results, and system to evaluate the spectator's experience when viewing these enhanced videogame streams.

In chapter 8, we discuss the broader context of our findings, draw conclusions, summarize limitations, and suggest possible avenues for future work.

2 BACKGROUND LITERATURE

> The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.
>
> — Ivan Sutherland, 1965 [245]

What started out as an idea from the French poet Antonin Artaud in 1938 [3], would become actualized by Ivan Sutherland in 1968 [246], and would finally become conceptually grounded by Milgram and Kishino in 1994 [168]. From its birth to its present, virtual reality (VR) has become ingrained in the culture, technology, and research of our time. In this thesis, we see virtual reality as a space that the user must navigate within and across from their physical reality. In particular, we are interested in how a computing system can shape the the user's subjective experience when in between two kinds of realities. In this chapter, we will go over relevant works in the areas that cover systems to blend realities, pointing in immersive environments, spatial interaction around the body, and collaboration. But first, we will position our work in the context of virtual reality's origin and disambiguate the terminology we will be using throughout this thesis.

In 1968, Ivan Sutherland created the very first instance of a virtual reality (VR) head mounted display (HMD) in his lab located at the University of Utah. He would later call this the *Sword of Damocles* [246] (Figure 2.1), though it is unclear whether this was intended to be an allusion to the literal hanging contraption or to some unknown future [185]. He used the device to create the first instance of a virtual 3D computer generated environment that was able to simulate a user's dynamic perspective in real-time.

Research into virtual reality continued to push the state-of-the-art in areas such as computer graphics, optics, and human-computer interaction. However, it was not until the late 1980's with the release of the VPL EyePhone and DataGlove that VR truly started to become rooted in the cultural and capital zeitgeist of the time [50]. And, as the industry matured, consumer devices started to become more readily available and VR concepts began to invade popular entertainment media. One salient example of this came in the form of the science-fiction book *Snow Crash* (1992) by Neal Stephenson. *Snow Crash* explored many ideas that are now becoming realized, including social spaces, shared presence, and the themes encapsulated by the "Metaverse." A term that is now seeing a resurgence as it is becoming associated with the growing popularity of the blockchain, WebGL, and the Web 3.0 [163].

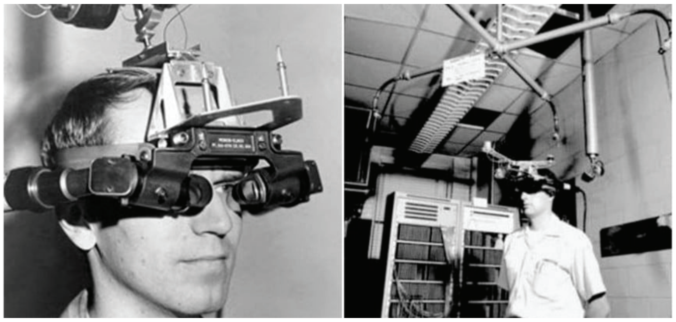


Figure 2.1: Ivan Sutherland's "Sword of Democles". Modified from [202, 246].

Our conceptual understanding of VR, and subsequently augmented reality (AR), mixed reality (MR), and to a larger extent, the superset of extended realities (XR), became solidified with the 1994 publication of Milgram and Kishino's seminal work on the taxonomy of mixed reality displays [168, 169]. Their proposed Reality-Virtuality (RV) Continuum would help frame our discourse around the perceptual aspects of head-mounted displays and to the kinds of technology needed that would allow us to interact with these virtual environments. Many extensions to the RV continuum have been proposed, typically by adding axes orthogonal to the continuum that provide new insights into aspects of human-computer interaction and the design of virtual spaces [85, 112, 153, 167]. Of particular interest is the conceptual framework proposed by Kanade et al. called *Virtualized Reality* [117]. This places the user into a digitally reconstructed version of the physical world that simulates the geometry and texture of the environment they are in. Such a reality relates back to Baudrillard's work in *Simulacra and Simulation* (1981) and his allegory of Borges' fable, where

> ...the cartographers of the Empire draw up a map so detailed that it ends up covering the territory exactly....[which is] the most beautiful allegory of simulation. [7]

But, when that simulation precedes reality, it becomes no longer that of reality but of simulation, a *hyperreality*. These ideas and frameworks would become increasingly more important over the decades as the research community and society explored the virtual from both cultural and technological perspectives.

It was during the time of the 1990's, when a wave of human-computer interaction (HCI) research began to make significant effort evaluating spatial interaction techniques for VR and AR environments [81, 203]. The practice of pointing at objects is intrinsic to how we communicate with other people around us, and it is no surprise that pointing techniques, such as raycasting, were among the first to be explored for selection and manipulation of 3D objects [19, 108]. Variations on the raycasting technique began to be explored, where researchers started to look at the unique ways humans perceive space and objects form a singular point of view [64, 200]. For example, a flashlight metaphor has been used to perform object selection when objects intersect an infinite cone instead of a ray [141], or by adjusting our sense of embodiment, an approximation to direct manipulation can be achieved on virtual objects [25, 204]. Augmented reality began to become more actively researched for specific usage scenarios during this time as well, including collaborative working environments [16, 213, 247] and medical applications [65, 183].

Up to this point, augmented and virtual reality interaction research has mostly used HMDs to provide the user with an immersive environment, however using room-scale projection based methods is also a feasible approach to achieve an immersive virtual environment. In 1993, the Cave Automatic Virtual Environment (CAVE) system was proposed as a way to create immersive virtual environments by projecting images onto a series of perpendicular screens arranged to form a cube [48]. Inspired by this work, Raskar et al. introduced the concept of a *spatially immersive display* that uses projection-based augmentation along with computer vision techniques to understand where each pixel is located and what is being augmented [207]. The combination of computer vision and projector-based augmentation allowed them to correct pixel distortion around non-planar geometry and simulate 3D objects from specific points of view through a process called projection mapping. This projection-based approach to AR was later reconceptualized as Spatial Augmented Reality (SAR) since it compliments other technical implementations of AR that use an optical see-through display. In both cases, a virtual and real environment are blended together using computer vision techniques for the registration of digital objects and light [18].

As research in VR and AR continued, public interest in them began to wane, leading to a period in which very little commercial innovation happened. During this period research shifted away from a specific implementation of VR or AR and started to explore spatial and 3D interaction techniques more generally. This included HMDs [98], projectors [264], large displays [261], and table-top environments [109]. It was not until the onset of mobile AR in the 2010s [210] and the release of the Oculus Rift DK1 in 2013 [187] when growing consumer and research interest in VR started to garner renewed interest. This was shortly followed by head-mounted AR with the release of the original Hololens [223]. With the growing number of high-fidelity consumer HMDs, research into areas relating and complementary to the aspects of the virtual began to increase. This included areas such as haptics [4, 42], blending [161], and user co-location [156]. SAR also has seen renewed interest within research and academic circles, partly due to research coming out of Microsoft and their RoomAlive Toolkit [113, 114].

2.1 DEFINITIONS OF REALITY

One point of confusion around VR, AR, and the larger superset of XR, is in the terminology and definition of what they represent. It is too often that the technical implementation to achieve a type of reality is conflated with the phenomenological experience of the user in that environment. Some of this confusion may stem from the original work by Milgram and Kishino [168, 169], who describe a taxonomy for the technological implementations around mixed reality visual display technology and which continue to be continually misrepresented [236]. We will attempt to disambiguate these difference and come to a common understanding that we will use for the rest of this dissertation. See Figure 1.1 for an overview.

First, we need to disentangle the conceptual idea of extended reality from the specific technological implementation to achieve it. To emphasize this point, consider this thought experiment:

> You are in a windowless room, a table is in the centre and on top or it is a pair of glasses. You walk over, pick up the glasses and put them on. Instantly, in front of you appears a simple rendering of a low-poly cel-shaded planet. As you move closer, you begin to make out the flora that is spread out across its surfaces where the fauna frolick about. And, when you move around it, your perspective of the planet changes, behaving as if it was floating in front of you. It is reacting to the physical environment around you, with proper occlusion, parallax and perspective, and behaving as if it was actually physically there.

This is the quintessential augmented reality experience viewed through a set of *AR glasses*, such as the Hololens [223]. But, what if someone suddenly turned off the lights? The real world would vanish before your eyes and all that would be left is the virtual, a tiny planet floating in the darkness. The questions is then, "Would this still be AR?" We posit that it is not AR but instead a virtual reality (VR). From *your* perspective, you are now floating in a void, stripped from physical reality where the virtual is the only thing you can perceive. This example demonstrates how "AR glasses" can invoke a VR experience and where the problems inherently lie. Common understanding of AR and VR tend to conflate the conceptual with a specific technical implementation, hampering understanding of the space. In the following, we now will carefully define the conceptual meanings across specific types of reality and enumerate on the specific technical implementations that can enable them.

We start with two fundamental and irreducible definitions from which all other types of reality emerge, namely that of virtual reality and physical reality.

PHYSICAL (BASE) REALITY (R) Our *physical reality* is the one from which all other realities are simulated and intrinsically tied to. An observer would consider this to be their "real" reality. Such a reality follows a set of immutable physical laws that can be observed but not changed.

VIRTUAL REALITY (VR) A *virtual reality* is a reality that replicates our phenomenological experience of the physical world inside of a virtual environment. This includes simulating how objects behave when we observe them, including visual and perspective changes, motion parallax, and occlusion, providing the user with a sense of "being in the [virtual] world" [92] around them. A virtual reality is intrinsically computational, where the state of objects that occupy that reality are composed of discrete pieces of information that *can* be known, even if they may not be known by the observer. The term computational is not accidental, as this does not necessarily exclude our physical reality if we were to consider it as being fundamentally "computational," which appears to be more likely than one might like to believe [24].

Our definition takes inspiration from David Chalmers' inquiry into *The Virtual and the Real* [37]. We purposefully leave out "interaction" from our definition as the virtual objects do not necessarily need to react or behave as we might expect as long as the observation of them satisfies our requirements. Based on this, a canonical virtual reality would be one in which an observer is unable to distinguish their virtual reality form their physical reality, as all sensory input is perfectly simulated. Clearly, a canonical experience is not possible, and current VR devices have strong implicit bias towards our visual experience; but this is the bar from which most technical implementations of VR strive to achieve. There are two broad approaches to achieve this; ones using a CAVE-like environment with projectors [48] and ones that use a head-mounted display [188].

AUGMENTED REALITY (AR) An *augmented reality* is a blend between our physical reality and a virtual reality, where the physical reality has precedence over the virtual. The user would consider themselves in the physical world with virtual objects composited within it. Technical implementations typically come in three categories: (1) mobile phone AR, which could be considered a window into the virtual, (2) optical see-through (OST) head-mounted displays (HMD), which accounts for most of the commercial AR headsets, and (3) spatial augmented reality, which uses physical light-emitting projectors.

SPATIAL AUGMENTED REALITY (SAR) *Spatial augmented reality* is a specific technical implementation of AR that combines a visible light-emitting projector with a geometric understanding of the physical world to augment surfaces with digital information. It could be thought of as being a 2D projection of a virtual 3D world onto the surface geometry of our physical reality. The term itself is a bit of a misnomer as there is nothing in SAR that is more "spatial" then other forms of AR, like ones using an optical see-through HMD. And, unlike the others listed, SAR is not a conceptual framework but is used to describe a very particular technical approach to achieve AR.

AUGMENTED VIRTUALITY (AV) An *augmented virtuality* is a blend between our physical reality and that of a virtual reality. In such a context, a virtual reality has precedence over base reality, where the virtual reality is the one the user would perceive to be inside. An example of this would be to composite a physical computer keyboard into a virtual environment as a means to interface with objects in the observer's base reality [161].

MIXED REALITY (MR) *Mixed reality* is a subset of realities that span both AR and AV, but exclude VR and base reality.

EXTENDED REALITY (XR) *Extended reality* is a superset of all possible realities, include AR, AV, VR, and those that occur within our physical world.

VIRTUALIZED REALITY A *virtualized reality* [117] is a copy of our physical reality that retains its geometric structure, colour, and texture. This is distinct from a photo or video in that it is not a 2D projection of a physical space but a reconstruction of it. If we replace our physical reality with a virtualized reality in the Virtuality Continuum, we can achieve an inversion of the space that mixes a copy of reality with a virtual reality.

It is clear from the definitions that certain subset realities, AR, SAR, AV, and MR, are emergent from the combination of VR and R. This is further reinforced by looking at the Virtuality Continuum (Figure 1.1). Interesting scenarios arise if we consider replacing physical reality on the continuum with another virtual reality or even a virtualized reality. In such cases, we get a reflection of the continuum that blends these realities together. An example of this would be blending a virtualized reality, which is a copy of the physical world, with a virtual reality, or blending two virtual realities together. This brings us to our last definition, cross-reality interaction.

CROSS-REALITY INTERACTION A *cross-reality interaction* is an action the user takes to achieve some goal where the action requires the user to process information from two discrete realities as defined by the virtuality continuum and its extensions.

2.2 SYSTEMS AND TECHNIQUES TO BLEND REALITY

A number of works explore methods to combine the real with the virtual, where it is often advantageous to manipulate either the virtual world or the real world representation of it. Reality Skins [229] utilizes the real world as a blueprint to build up a virtual scene by matching the real world against predefined virtual objects contained within a dataset. The construction of the 3D space is posed as a constraint satisfaction problem, where Monte Carlo optimization is used to create a best fit between the virtual objects and the real world. Similarly, Sra et al. proposed a method for building a procedurally generated virtual environment based on a low resolution scan of the real world [239]. This approach is more simplistic than in Reality Skins as the set of constraints imposed on the construction is smaller, giving rise to more variation but at lower fidelity.

Investigating the interaction space between the real world and the virtual, ShareVR [76] explores the communication and interactions that can occur between a non-HMD wearing user and users who are in a virtual environment. The system uses external projectors with a VR HMD to create a space where both the VR user and the non-VR users can share the same virtual experience. Though the system described offers some affordances related to communication and interaction, it only implements pre-defined experiences and cannot integrate with arbitrary virtual worlds. JackIn Space [127] investi-

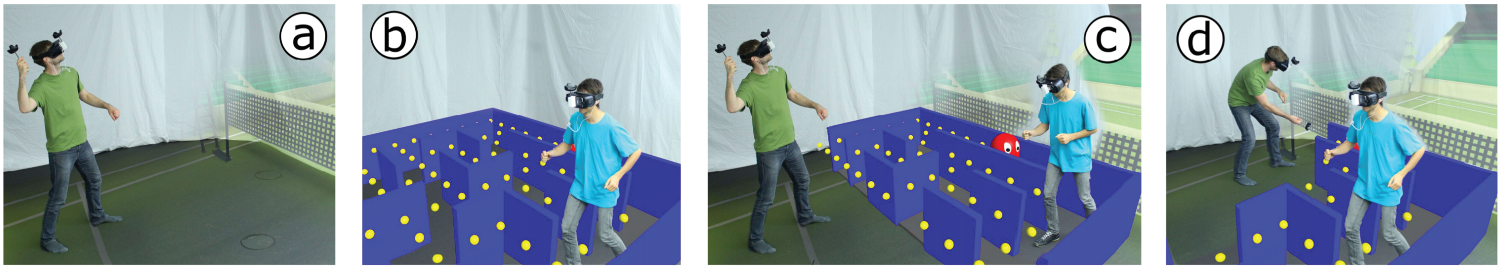


Figure 2.2: VirtualSpace demonstrates how multiple users can share the same physical space while remaining in VR. The figure depicts user (a) playing badminton game while user (b) is playing Pac-Man (from Marweki et al. [156]).

gates the transition between third- and first-person views in the context of telepresence. The system uses Kinect V2 sensors and head mounted cameras, allowing the user to "see" through the eyes of another co-located user or take a third-person vantage point around them. OneReality [218] describes a holistic design space that blends the user's presence across many levels of virtuality. The system uses depth cameras, projectors, and a VR HMD to allow a user to experience a virtual environment from different visual scales (i.e. viewing a city scape diorama from a table-top display or viewing a city street as a pedestrian in VR).

The blending of haptics with virtual environments can enhance interaction with a virtual space, making it feel more "real". For example, there has been research into the use of real world proxies as substitutes for virtual objects [94, 231]. Here the real object is mapped to a virtual counterpart using an optimization algorithm to obtain high correspondence between the two. This allows the user to interact with the virtual object through the haptic feedback of the real one. Redirection between a virtual and real world correspondence is explored in work by Cheng et al. [42], where their system helps guide the user's hand to a physical plane that approximately aligns with a virtual object. They demonstrate techniques in several mock virtual environments, including an escape-the-room style game and a virtual spaceship. Extending this, Zhoa and Follmer demonstrate continuous hand retargeting between virtual and real objects [281]. An algorithm is proposed that provides continuous haptic retargeting between a real world object and a virtual approximation of it by finding a spatial mapping that maximizes smoothness and minimizes mismatch between them.

Object Avoidance

Within a virtual world, there will always be some conflict that exists with the real. Current "chaperone systems" found in modern day VR HMDs attempt to avoid this by compositing a grid on top of the virtual scene with the assumption that the user's immediate space is free of physical obstacles. Several works look at ways to diminish the spatial conflict between virtual and

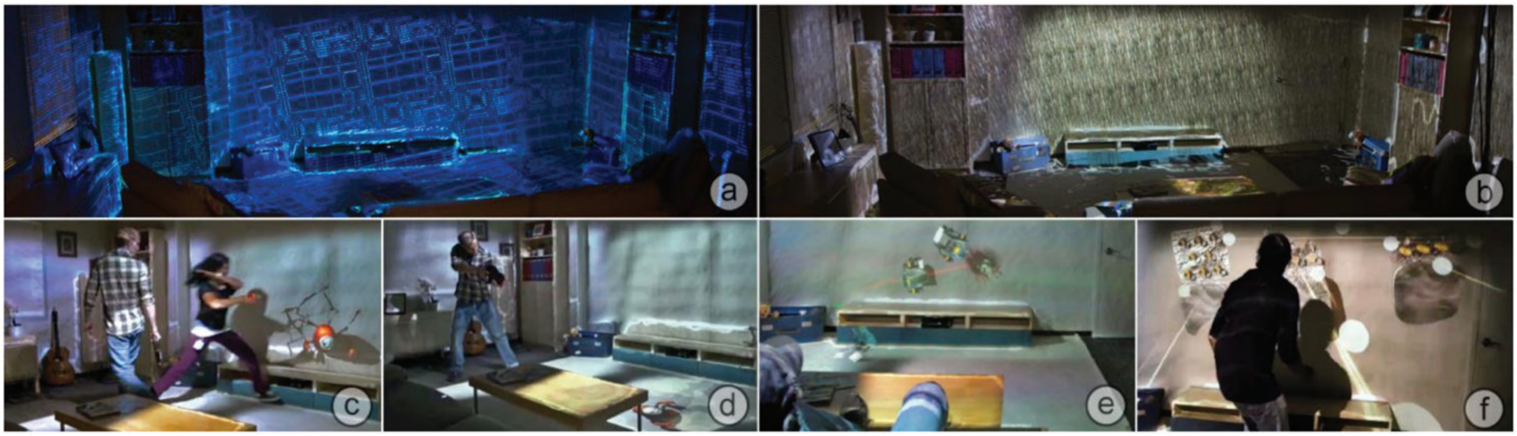


Figure 2.3: RoomAlive uses projection-mapping to create an immersive spatial augmented reality environment. Images (a) to (f) showcase how the same room can be augmented using different virtual scenes (from Jones et al. [114]).

real spaces. For example, VirtualSpace [156] creates a set of design guidelines for using a single physical space with multiple virtual reality experiences (Figure 2.2). The system works by cutting the physical space into tiles, where there is a one to one correspondence between the tiles and each of the VR users. Through a provided API, the tiles can be shuffled around the physical space and *maneuvers* are implemented so that the VR users avoid colliding with one another.

Redirected walking is another way to resolve conflict with the physical world [193, 243]. The goal of these systems is to create virtual spaces that feel larger then the physical space the user is currently occupying. This is accomplished through the subtle manipulation of the virtual scene, by either rotating or translating it slightly. Recent work uses eye saccades to make the user oblivious to these changes by forcing the eye to rapidly change fixation points by introducing tiny specks of light within their field of view. During the rapid change in fixation, the virtual world is instantly rotated, forcing the user to change walking direction [242]. Avoidance of other co-located VR users is explored by Lacoche et al. [134], where different in-game representations of the other players, like a cylindrical grid, a ghost-like avatar, or a highlighted location on the floor, are used for purpose of safety and awareness.

Room-scale experiences

Room-scale experiences have been explored in the context of spatial augmented reality (SAR) [114] and VR [43, 143]. These systems provide the user with custom interactive experiences tailored to a large interactive space. The RoomAlive system [113, 114] explored room scale projection mapping for entertainment. A multi-projector setup with depth cameras maps the environment so digital content can be projected all around the user, creating a visually immersive experience. Interaction was facilitated through the use of direct contact with the physical surfaces in the space (Figure 2.3). Petford et al. [197] created a system that utilizes a half-dome mirror to provide room scale projection while reducing the number of projectors needed to one. However, the fidelity of this system is low due to how light spreads across the room, making the effective resolution per wall a fraction of the projector's native resolution.

In the area of VR, RemixedReality [143] took a reconstruction of a physical space and explored design and user experience implications through the manipulation of the underlying polygonal mesh. The system uses a voxel grid to record state changes for groups of adjacent vertices in order to facilitate interactions. For example, a user could manipulate their spatial perspective to be more cylindrical, erase parts of the reconstructed world, or record segments of reconstructed reality for playback. In order to investigate the use of haptics in a virtual environment, Cheng et al. [43] used human workers to dynamically build props around the user that match the geometry in-game. The system used real actors to manipulate office divider like objects allowing them to recreate the scene in VR. They demonstrated the system on predefined VR experiences in a large space where multiple human actors where used to create the game scene in real-time around the user.

2.3 ENHANCING THE SPECTATORSHIP OF VIRTUAL ENVIRONMENTS

The role of the spectator is asymmetrical to that of the performer, where the primary means of participation is accomplished through the simple act of *looking* [222]. There are intrinsic asymmetric qualities to the roles the streamer and their spectators have within the medium of videogame live-streaming. This relationship between the spectator and the content was first formally explored in the field of film theory, where semiotics was used to break down and decode the hidden meanings produced on screen [89]. Though these conceptual frameworks are only tangentially applicable to Human-Computer Interaction (HCI), this conceptual model of the spectator is important in how we frame the *user* within a computing system, as there is an inherent asymmetric dichotomy that exists between the creator streaming the content (the streamer) with the spectator observing it.

Spectatorship of videogames could arguably be attributed to the rise of arcades, where a group of people would gather around and watch others play [184]. This type of behaviour would later shift to ad hoc meetups known as LAN events [111]. In contrast, spectating videogames remotely is a relatively new phenomenon that has risen in popularity with streaming platforms such as Twitch [107] and YouTube [105]. The onset of *Let's Play* videos have further popularized the medium [73], where the total market capitalization of the videogame industry now exceeds that of both movies and sports combined [270].

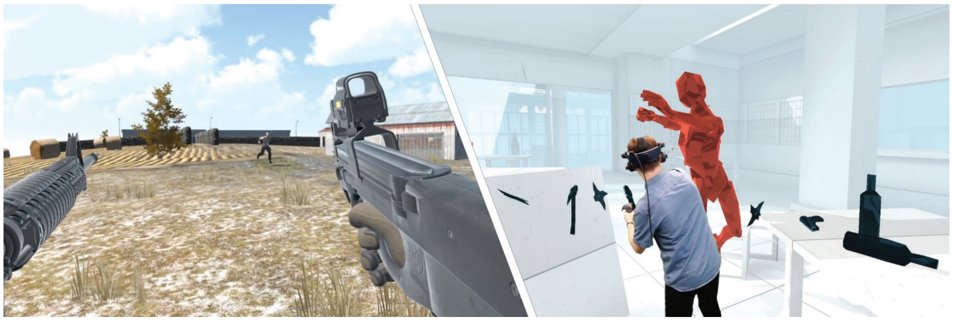


Figure 2.4: Examples of videogame spectatorship in VR from both a first- (left) and third- (right) person perspective (modified from Emmerich et al. [60]).

Videogame Spectatorship

There is a significant amount of research around the motivations, preferences, and reasons why people watch others play videogames [60, 80, 232]. For the most part, these investigations fall under two contexts: when the spectator is collocated with the player and when they are remote.

Collocated gaming and spectatorship has been studied in the context of audiences [119] to smaller intimate at-home play with only a few people [250]. To describe the relationships between the spectators and players, Downs et al. [55] proposed that the spectator can take on the role of a bystander, audience member, or player where participation can range from passively watching to active engagement [158]. Recently, it is becoming more common for games to blur what role a spectator can have within the context of a game, giving them direct control over minor aspects or even making them a critical part of the game's design [69, 255].

In contrast to collocated spectatorship, watching others play games remotely is becoming an increasingly popular passtime, one that is comparable to traditional sports [44, 80, 122, 206]. To better understand the motivations behind why people engage in spectating activity, Sjöblom and Hamari looked at intrinsic and extrinsic factors that motivate users to watch others play videogames online [232]. They found that the total number of hours watched is positively associated with information seeking, tension release, and affective motivations. Expanding this to VR, Emmerich et al. investigated the live-streaming of VR games and found first- or third-person perspectives of the VR streamer can affect the spectator's overall experience [60] (Figure 2.4). Their findings suggest that a third-person perspective of the VR player is not as effective as the view taken directly from the HMD, and can sometimes be detrimental to the viewing experience. However, this was limited to a fixed perspective with no spectator agency over the view. In Chapter 7, we build on these insights to explore the inverse problem, VR users spectating non-VR videogame livestreams.

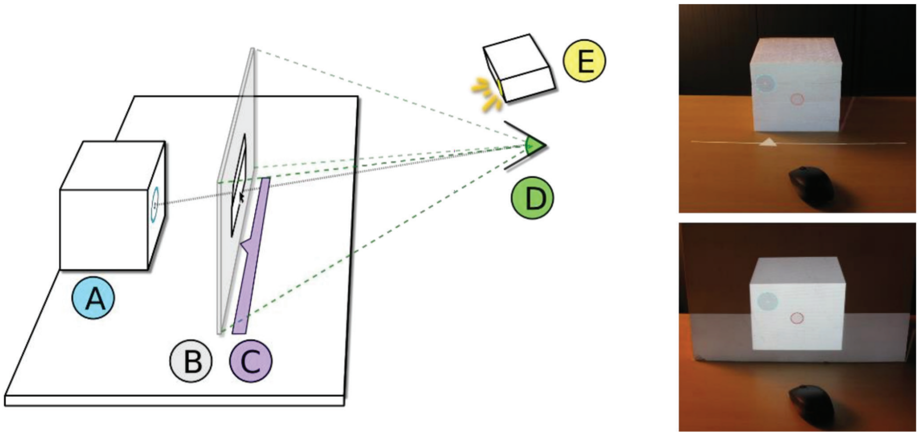


Figure 2.5: Experimental setup of a tabletop pointing experiment using spatial augmented reality (from Gervais et al. [71]).

2.4 POINTING IN IMMERSIVE ENVIRONMENTS

Pointing is a fundamental human trait used to communicate intent with other collocated people. There is a significant body of work that evaluates pointing under different technological modalities. For the most part, these have been conducted in environments other than SAR: large displays, multi-display environments, tabletops, and viewport AR. Although some VR and 3D user interface pointing studies have investigated handheld device pointing, none have investigated it with SAR. Within the larger context of this thesis, pointing can be used to allow the user to communicate their intent in the transition between a virtual and physical environment or used to partition pieces of a reality inside a virtual space.

VR and 3D User Interface Pointing

There is extensive research on the ways to target and point at objects in VR [23, 49]. Early work by Bowman et al. investigated 3D techniques like raycasting and non-linear arm-extension techniques (Go-Go [25]), as well as others like world-in-miniature visualizations for direct manipulation of objects [26]. Cashion et al. evaluated selection techniques in five game-like virtual environments with varying degrees of object density and dynamics [34]. They compared four 3D object selection techniques: Racyast, SQUAD, Zoom, and Expand. In later work they evaluated an auto-selection technique against Expand and Bendcast [35]. All these techniques are variations on raycasting within a virtual environment. Teather and Stuerzlinger studied effects of stereo displays and passive haptics on target selection. They found that performance degrades when targets are placed stereoscopically in front of the screen [249], but passive haptics significantly improves pointing throughput in target acquisition tasks [248].

Pointing in Immersive Environments

Existing research into SAR environments includes RoomAlive [114] and IllumiRoom [113], where the intent is to immerse the user in artificial environments through spatially mapped graphics and view-dependent illusions. Here the user is completely surrounded with dynamic content and every surface is utilized to maintain the immersive experience.

Exploring tabletop SAR environments, Spindler et al. created augmented environments in which users can peek into 3D worlds through hand-held lenses [237, 238]. Here the user can explore an augmented table-top that renders a 3D scene. Using a paper lens, the user is able to peer inside this scene from their particular perspective or interact with the content on the table itself using a handheld lens.

Research into ubiquitous intelligent light sources has also been explored as a medium to transmit data and information within an environment [256]. LuminAR explores this concept with an actuated desktop lamp that can interact and display content within an office setting by moving around the scene and projecting onto different surfaces and objects [142].

Gervais et al. [71] evaluated pointing in a small desktop SAR environment while seated and stationary. A mouse and tablet were used, and the environment included targets on different faces of objects. They report Fitts' law holds when selecting targets on abnormal geometry. However, the fixed user position, small environment, and input modes presented a very simple SAR context (Figure 2.5). *Mano-a-Mano* [12] examined selection in a large room-sized SAR environment, however, these were preformed in mid-air with a wand. The task is focused on perception rather than performance: one participant uses a small pole to point at one of several rendered floating spheres, and an adjacent participant reports which sphere they are selecting. The other participant then verbally dictates what sphere they were pointing at. A formal study showed that users are able to identify targets within a radius of 12cm.

2.5 SPATIAL INTERACTION ON AND AROUND THE BODY

It has been proposed that areas around the body can be broken up into three distinct areas: pericutaneous, peripersonal, and extrapersonal layers [58, 96]. Each layer describes how we view ourselves in relation to the objects situated around us, and prior work has investigated how these layers can expand the set of affordances offered to the user. For example, the peripersonal space that surrounds the body is easily reachable by the hand [259]. This space had been imagined as containing hidden digital information that can be viewed through a mobile phone's screen [280], as a means to explore multi-layered panorama images [251], or as a way to track a phone within a body-centric coordinate system [124].

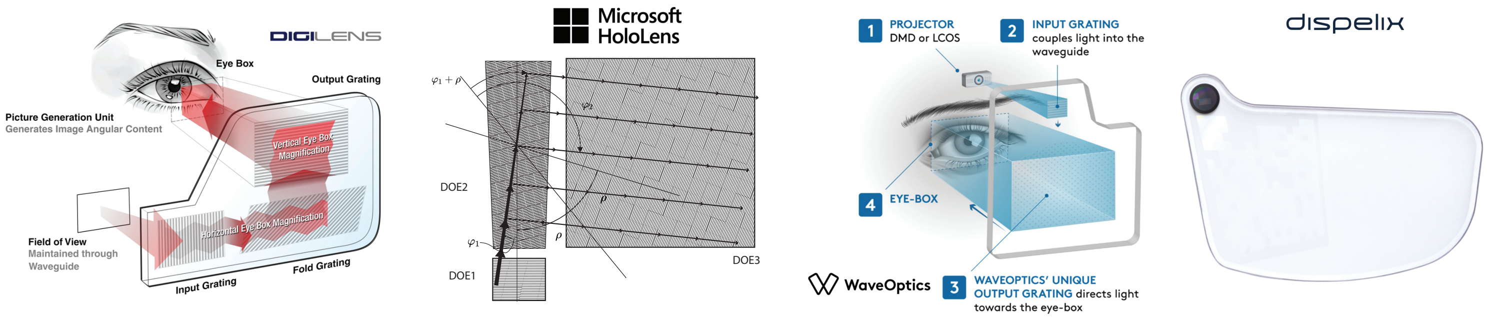

DIGILENS
Eye Box
Output Grating
Picture Generation Unit
Generates Image Angular Content
Vertical Eye Box Magnification
Field of View
Maintained through Waveguide
Horizontal Eye Box Magnification
Input Grating
Fold Grating
Microsoft HoloLens
$\varphi_1 + \rho$
φ_2
DOE2
ρ
φ_1
ρ
DOE1
DOE3
1 PROJECTOR DMD or LCOS
2 INPUT GRATING couples light into the waveguide
4 EYE-BOX
WaveOptics
3 WAVEOPTICS' UNIQUE OUTPUT GRATING directs light towards the eye-box
dispelix

Figure 2.6: Demonstration of 2D pupil replication architectures that use eyebox expansion techniques (modified from Kress [131]).

Augmented Reality Wearable Projector Systems

Of all places to mount a projector, the human body appears at first glance to be an odd choice. However, attaching projectors so close to the body can provide interesting new opportunities for interaction and visualization [171]. For example, if a projector is positioned on the shoulder, the situational awareness of an environment can be maintained while under high-stress military activity [160]. Another example is OmniTouch [83], a shoulder-worn depth-sensing and projection system that can transform everyday surfaces into interactive screens. This allows the user to use the space around their body as an ad-hoc display, where they can interact with digital information through touch. The Ambient Mobile Pervasive Display (AMPD) explored these concepts further by investigating how ad hoc displays can aid specific tasks for both indoor and outdoor environments [269].

Of the many possible on-body mounting locations, on or near the head has been of particular interest. A fixed, front facing head-mounted projector can be used to directly augment the physical environment to reproduce the effect of wearing an optical see-through AR HMD [99, 123]. Scape [98] showed how a "head-mounted projective displays" (HMPD) can enable multi-user collaboration in AR. Krum et al. found this allowed for more natural depth cues [132], while Kade et al. demonstrated it uses in entertainment contexts like shooting games [116]. Genç et al. showed how a head-mounted projected image of static and dynamic content can be effective when the user is in motion [70].

2.6 HYBRID DISPLAYS AND COLLABORATION

Current optical see-through AR HMDs typically use a diffractive grating waveguide combiner with two-dimensional eyebox expansions [131]. This is the technology being used in popular consumer headsets like the Hololens [165] and MagicLeap One [148]. An alternative to this is to use a low powered head-mounted projector that utilizes a small laser pico projector with an optical beam splitter. When the light reflects off of surfaces using a retro-

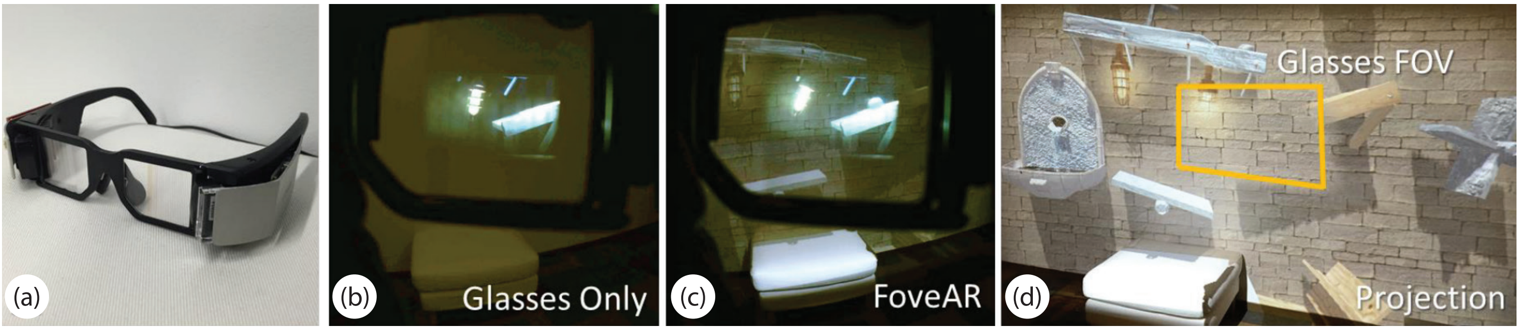


Figure 2.7: Examples from the FoveAR system, showing the (a) AR glasses, (b) the virtual content as seen through the AR glasses, (c) the combination of SAR and AR, and (d) the SAR environment (modified from Benko et al. [11]).

reflective coating, the resulting light is combined to produce the illusion of a spatial object [99]. A variation of this removes the optical beam splitter to use the projector directly, augmenting the environment through direct illumination [116]. All variants of AR offer the user unique affordances that can be thought of as complementary. Figure 2.6 gives an overview of optical see-through (OST) AR display variations.

Asymmetric HMDs

Researchers have investigated a variety of ways to overcome the limitations of current generation AR HMDs [148, 165]. One of the known limitations of these devices is in their field of view (FoV). Though manufacturers are not entirely transparent on the exact specifications, it has been reported that the Hololens 2 FoV is at maximum only 29° vertical [91]. One way researchers have addressed this is by introducing a sparse peripheral display that is arranged around the display to give the user in AR a low-fidelity sense of what is happening outside their current view [75, 276].

Combining different technical implementations of AR together is an interesting direction of work that has seen recent exploration (see Figure 1.1). For example, Zhou et al. used both SAR projection and AR HMDs to overlay just-in-time information on-top of industrial equipment for their workers [282]. Using a depth reconstruction of an environment has also shown to be useful. Maimone et al. used a SAR reconstruction, depth data, and AR glasses to enhance communication and telepresence for remote communication scenarios [152]. Enhancing the look of a real object with a combination of SAR and AR augmentation has shown to be effective. HySAR [79] used a the AR HMD to render view-dependent specular reflections and SAR to render view-independent diffuse reflections on real objects to enhance overall realism.

Of particular interest is the work that has used SAR environments with OST-HMDs for collaboration. UbiBeam++ [125] used a table-top SAR environment with multi-user AR headsets for collaboration and gaming. They described a toolkit to enable these kinds of interactions and demonstrated how it can be used for table-top gaming. FoveAR [11] investigated a room-scale SAR environment with an OST-HMD for purposes of gaming and to increase the immersion of the AR user inside the virtual environment. They combined the view-independent rendering of a virtual scene with view-dependent graphics to overcome the FoV limitations of current headsets (Figure 2.7).

Other types of asymmetric systems have been used to explore collaboration between the users wearing an AR or VR HMD to those users that are external to them. This includes Tabletop+HMDs [103, 149, 240], Room-scale SAR+HMDs [76], portable displays+HMDs [277], and self-contained displays attached to HMDs [38, 77, 151]. Attaching a projector directly to an HMD can provide usage scenarios not possible through other approaches. ShARe [110] combined a mid-sized projector with Hololens HMD to explore the asymmetries between the HMD user and an external onlooker. HMD Light [262] explored the use of a projector attached to a roaming VR HMD. They explored the interactive capabilities between the VR user and the collocated users around them.

2.7 SUMMARY

This chapter covered a range of systems, pointing methods, interaction techniques, and systems that utilize AR, SAR, VR, or a combination to create novel contributions to the research community. Our work builds on these existing explorations to create systems and techniques that explore the user's experience while within and between realities as outlined in Figure 1.1. The following chapters will detail specific contexts, scenarios, and implementation details that we use to achieve novel interaction across the Virtuality Continuum.

3 POINTING IN SPATIAL AUGMENTED REALITY

Spatial Augmented Reality (SAR) [207, 208] places digital content directly into a real physical environment. One application of SAR is to create immersive environments that differ significantly from physical reality [113], often for gaming or virtual teleportation [194]. This typically involves covering and hiding large portions of real surfaces and objects with textures, often creating illusions of virtual 3D objects [53, 84, 114, 190]. In contrast, SAR can be applied in a more integrated and subtle way, where real surfaces and objects are selectively augmented with 2D digital information. We refer to this as "surface mapped" SAR, since it relates to SAR surface shading [209]. Essentially, every surface becomes a potential display without the limitations that AR glasses impose, like a limited field of view.

Such an environment could be used to facilitate cross-device interaction, for example content from a mobile phone can be spread into underutilized spaces for the purpose of awareness (like weather conditions), notifications (like upcoming meetings), visualizations (like maps), or sharing content (like photos). Techniques already exist to track the 6-DOF position of a phone [180, 189, 262] and to detect when it touches a surface [82, 145, 224]. Enabled by this, the phone could be a ubiquitous input device for surface mapped SAR.

Mobile phone pointing has been explored with large displays [225], multi-display environments [28], hand-held projectors [175], and "viewport AR" in relatively planar scenes from a fixed perspective [22, 216, 217]. In general, mid-air device pointing in AR and VR has assumed immersive or floating 3D targets [12, 248], while work examining surface mapped SAR has kept the user at a fixed location in a small desktop setting [71], or in an essentially empty room [195].

In this chapter, we compare three popular mobile phone pointing techniques in surface mapped SAR. The techniques are adapted from other contexts: *raycasting* from large displays, *viewport* selection from mobile AR, and *direct* contact of the phone from tabletops. In our first experiment, we evaluate mobile phone pointing in a realistic projection-based SAR environment (Figure 3.1). The results identify key characteristics that influence pointing performance: the degree of target occlusion due to environment geometry, the target view angle relative to the user, and the amount of user movement required. Our second experiment tests these key factors in a highly controlled simulation of SAR pointing tasks using a stereo AR head-mounted display.

In summary, we contribute empirical evidence for the relative performance of mobile phone pointing in SAR, showing *raycast* is fastest for non-occluded targets, *direct* is most accurate, and fastest for occluded targets in close proximity, and *viewport* falls in between.

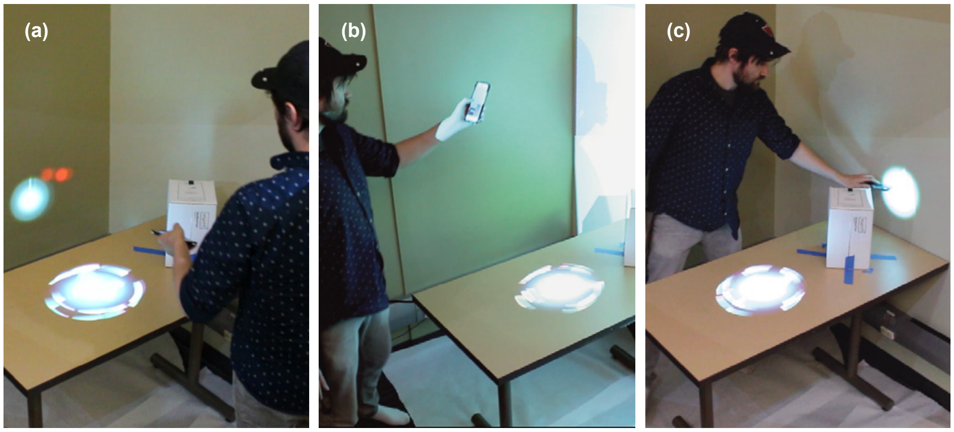


Figure 3.1: Mobile phone pointing in the surface mapped SAR environment used in Experiment 1: (a) the raycast technique uses an invisible ray emanating from the phone to point at the desired target, with a tap on the phone screen to select it; (b) the viewport technique views targets through a simulated rear camera and selection is by tapping on the target on the touch screen; and (c) the direct pointing technique uses the phone itself to directly touch a target.

3.1 RELATED WORK

In the previous chapter, we explored a large body of work that highlights both broad and detailed investigations into pointing across a range of environments. Here, we will provide an overview of these investigations and give broader context where necessary. Broadly speaking, this chapter relates to previous evaluations of mobile phone pointing, including evaluations using similar mid-air hand-held devices like laser pointers. For the most part, these have been conducted in environments other than SAR, specifically large displays, multi-display environments, tabletops, and viewport AR. Although some VR and 3D user interface pointing studies have investigated hand-held device pointing, they focus on 3D virtual targets, not 2D targets fixed to planes of differing orientations and positions like our surface mapped SAR environment.

VR and 3D User Interface Pointing

Target selection in VR has been extensively studied in the past [23, 49]. Examples of this include the raycasting [25], Go-Go methods [204], and world-in-miniature [26] which can be thought of as a form of direct pointing. Boritz and Booth used a 6-DOF device within a virtual environment to evaluate how monoscopic and stereoscopic displays affect the selection time of 3D objects during a pointing task [23]. They found that stereoscopic displays performed better overall. Cashion et al [34, 35] evaluated selection techniques in five game-like virtual environments with varying degrees of object density and dynamics. Teather and Stuerzlinger studied effects of stereo displays and passive haptics on target selection, and found objects viewed in front of the display degrade performance, but passive haptics improves throughput [248, 249].

Pointing in a surface mapped SAR is different than 3D immersive pointing. In this type of SAR, there is no illusion of 3D, all targets are placed directly on top of the geometry contained in the physical environment. As such, they will conform to the mostly planar surface the targets are projected on. This means Fitts' models designed for 3D, like the trivariate model or Murata and Iwase's model [181], are incompatible since surface mapped 2D targets do not have depth or ordered direction vectors. Our evaluation only considers targets that are strictly conforming to geometric surfaces.

SAR and AR Pointing

Rohs and Oulasvirta [216, 217] evaluated "magic lens mobile phone pointing" at near-planar scenes, like distant buildings. Pointing is done through the camera-view of the phone, where the phone is first positioned in physical space near the target, then fine-tuned in virtual space in the phone display. We refer to this general type of interaction as viewport pointing. They proposed and tested a two-part model based on Fitts' law, that splits physical and virtual pointing phases into two terms.

Gervais et al. [71] evaluated pointing in a small desktop SAR environment while seated and stationary. A mouse and tablet were used, and the environment included targets on different faces of objects. They report standard Fitts' law holds when selecting targets on abnormal geometry. MeetAlive [63] used mouse pointing in a SAR environment to facilitate meeting productivity, which was largely contained to four flat walls and a large boardroom table. Similarly, Petford et al. [195] compared mouse and raycast pointing in a similar SAR environment that was constrained to four flat walls and a ceiling. They found the mouse to be fastest for targets in front of the user and racyast for targets behind. *Mano-a-Mano* [12] examined selection in a large room-sized SAR environment using a wand for mid-air selection. In contrast, Molyneaux et al. [175] demonstrate direct touch and indirect shadow-based interaction techniques in a projector-based SAR system.

Pointing studies in AR and VR have focused almost exclusively on immersive 3D object pointing [12, 95, 249] or AR pointing at near-planar scenes [22, 216, 217]. Raycast, viewport, and direct mobile phone pointing techniques have been used with large displays, MDEs, and AR, but to our knowledge, never compared directly in SAR. In fact, few pointing studies have been conducted in SAR at all. Gervais, Frey, and Hachet [71] used a small desktop SAR environment, limited target variations, and unique interaction techniques controlled by a conventional mouse or tablet. Benko, Wilson, and Zannier [12] conducted hand-pointing tasks within SAR, but this is in context of a view dependent rendering of 3-D objects. In contrast, we investigate popular mobile phone pointing techniques in a larger and more complex SAR environment.

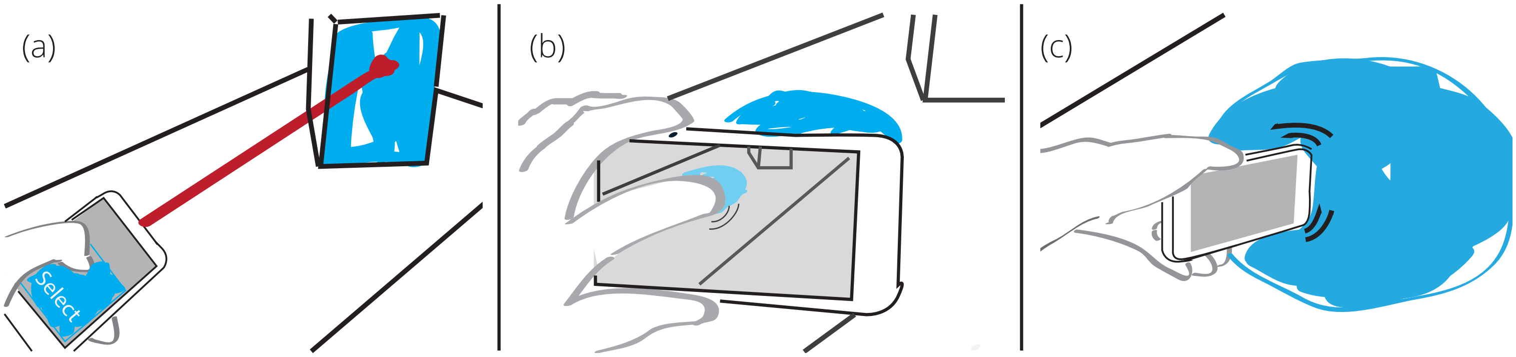


Figure 3.2: Mobile phone pointing techniques: (a) raycasting; (b) viewport; (c) direct.

3.2 MOBILE PHONE POINTING IN SAR

We briefly describe our surface mapped SAR technical infrastructure, then provide details for the three mobile phone pointing techniques to be compared.

SAR System and Environment

The setup occupies a corner of a room, occupying approximately 4 × 4 meters of floor space (Figure 3.1 and 3.3). Mounted in the ceiling are 5 digital projectors, 6 Microsoft Kinect cameras (each connected to an IntelNUC Core i7-7567U), and a 10-camera Vicon (Vera/Bonita) tracking system. Tracker 3.5.0 software on a dedicated server tracks the 6DOF position of a mobile phone and a person's head. The phone tracking object is a custom-printed phone case with seven 6.4mm spherical markers, and the head tracking object is a baseball cap with five markers attached to the brim.

The main server (Windows 10, Core i7-6850K) is connected to the Vicon server and IntelNUCs using a 10Gb intranet. Projectors and Kinects are calibrated using the RoomAlive toolkit [114], with the 3D room reconstruction imported into Unity3D. Manual adjustments to geometry position, and design tricks like texture blending and transparency, compensate for limited precision of projector alignment and room reconstruction. A Unity3D 5.6 application processes tracked objects, enables two-way mobile phone communication, and renders projection-mapped content with all projectors at 60 FPS using twin GTX 1080 GPUs.

The mobile phone is a Google Pixel 2 running Android Nougat 7.1 (5.0" display, 149 × 74 × 11 mm including case). A custom app enables the server to render a simple interface for experiment control and communicates status such as current motion tracking confidence.

Mobile Phone Pointing Techniques

Using the system above, we created three mobile phone pointing techniques suitable for SAR.

RAYCAST POINTING Previous mobile phone raycasting techniques used a laser [182, 226], or a geometric ray based on 3D tracked position [115] as we do. To use the technique, the user holds the mobile phone with either their right or left hand, points the front end at a target, and taps the screen with their thumb to select (Figure 3.2a). Since we accurately track the mobile phone's 3D position, and we have a 3D scan of the environment geometry aligned with the real world, we use a virtual ray to test intersections with virtual surfaces and objects. At the point of intersection, a red cursor is displayed on the surface. This allows the user to remain fully attentive on the virtual content that is in the SAR environment.

VIEWPORT POINTING Using the phone's camera like a viewport to select content is a common approach for virtual content selection. Implementations, like in Rohs and Oulasvirta's ([216, 217]) work, use a single fixed crosshair at the centre of the screen for selection, but we chose a more versatile method where targets are selected anywhere in the viewport [22]. To use the technique, the user roughly frames the desired target using the phone like a camera, then taps the desired target in the display (Figure 3.2b).

Typically, a live camera feed with computer vision tracking is used for viewport techniques. However, mapping a touch to a precise physical world location using a live camera view is challenging, and can be unstable and hard to accurately control using current AR methods. To avoid these potential confounds, our system uses a 3D rendering of the camera view synchronized with the SAR server, providing a view into a virtualized reality [117] of the environment. By configuring the virtual camera to use the same 60° field-of-view as the real phone camera, and the 3D scanned and calibrated room geometry creating a one-to-one mapping between physical and virtual worlds, an accurate rendered camera view can be produced.

Boring et al. [21] enhanced a standard viewport technique with several variations of zoom control (combined with selective frame freezing), tuned for selecting targets on distant displays with direct touch from a finger. Using direct touch on the display has the advantage of not requiring the user to precisely aim the phone, making the action more direct than the precise phone aiming required with raycasting. Tapping on small targets in the phone display does introduce a fat finger problem [230], but the user is free to move closer to the target to increase its overall size in the display. This worked well for our studies, but if the target is very small or the user is unable to move closer, then enhanced viewport techniques like Boring et al. propose can be used. One reason we do not use a zoom-in technique in our study is to keep the viewport technique under examination elemental, robust, and simple to use. Notably, most current viewport-style AR mobile phone applications using Apple's ARKit or Android ARCore also do not use zoom, so our viewport implementation is ecologically valid.

DIRECT POINTING Direct mobile phone input has primarily been used in the context of tabletops [224] and large displays [82], where the mobile phone acts as an extension of a person's hand. To use the technique, the user holds the mobile phone with either their left or right hand and physically taps the currently active target with a corner, side, or face (Figure 3.2c). Contact of the mobile phone to surfaces and objects is triggered when a bounding box constructed around the phone intersects with the scanned 3D geometry of the environment. This required the user to directly interface with the physical reality around them.

3.3 EXPERIMENT 1: AD-HOC SAR SETTING

This within-subjects experiment compares the three mobile phone pointing techniques in a realistic ad-hoc surface mapped SAR environment. For each technique, the participant is free to move around the space during the selection task of the 2D targets. Instead of strictly controlling target size and position in the traditional sense, we create a constrained but representative environment geometry and select target sizes and positions to represent content that might exist in a future where SAR is ubiquitous. A start target controls the user's initial position, but they can move freely after to select the required target. With the same task conditions across techniques, we examine how different segmentations of targets, such as by size and position, by initial target occlusion, or by target view angle, affect key task performance metrics like time, error, and user movement. Results lead to the identification of key characteristics, including target occlusion and target view angle, that are investigated in a controlled setting in Experiment 2. We also use the data from this experiment to develop and test a predictive model of SAR pointing based on these same key characteristics, explained later in this paper.

Participants

We recruited 18 participants, ages 21 to 50, 13 male, 5 female, 1 left-handed. All used a mobile phone every day. Remuneration was $10.

Apparatus

The SAR system described above is used in an environment containing a table with a small box on top (Figure 3.3). The 61 × 59 × 122 cm table is positioned orthogonal to one wall and sits approximately 56cm from a parallel wall. On top of the table sits a 17 × 27 × 20 cm cardboard box rotated approximately 30 degrees and sits 20 cm from the wall. A large portion of the floor, the two corner walls, the table, and box were all covered by the light produced by the projectors, which were orientated to minimize shadows and maximize coverage over all surfaces in the environment. The system was calibrated within a 1 cm tolerance and all input tracking and target hit detection for measured trials used the Vicon, which is accurate within 1 mm. If tracking was lost during a trial, the phone notified the participant by vibrating and turning the screen red. However, during the experiment, tracking was rarely lost.

Our surface mapped SAR setup can be thought of as a Multi-Display Environment (MDE) with approximately 11 "displays", which are different surfaces in the room with very different sizes, orientations, and positions (with some hidden). Figure 3.3 illustrates these surfaces. There are 2 large walls (large grey areas), 2 surfaces along different baseboard mouldings (rectangular grey area below wall surfaces), 1 floor surface, 1 table top surface, 2 table edge surfaces (thin green rectangles), and 3 surfaces forming the two sides and top of the box (shown in orange). In the experiment, we use most surfaces as one type of target (with the exception of floor and table top) and for large surfaces, we also display smaller circular targets mapped into a surface. These are explained below. Similar to the work by Molyneaux et al. [175], we restricted our experiment to a room-sized environment as this more closely replicates a scenario where SAR would be used.

LOGGING AND METRICS During trials, the system logged when each target was selected and whether selection errors occured to calculate the primary dependent variables of *Movement Time* and *Error Rate*. In addition, other data was logged: the position and orientation of the participant's head, the phone, and each target; all touchscreen input; and technique events and states (such as the 3D raycast cursor position). These are used to calculate a dependent variable for *Head Movement*, and a metric called the *visibility ratio* that determines how much of the target is occluded from the participant's perspective.

Two variations of *visibility ratio* are determined as follows. The system uses Unity to render 224 × 224px views of the full 3D scene from two virtual cameras (created in the same Unity scene) attached to the participant's head, one matching head orientation as a proxy for gaze, and the other oriented to the next target to be selected. For each camera, there are two rendering passes: one only containing the target, and the other containing the target with all scene objects that may occlude it. The proportion of the target in the second render relative to the first is the *visibility ratio* for each virtual camera.

Task

We imagine an environment where users interact with content on any surface. Consider a SAR office: pointing at a wall could place a large visualization like a map, pointing at the edge of a desk could silence a notification displayed there, pointing at the floor could open an application for viewing photos, and so on.

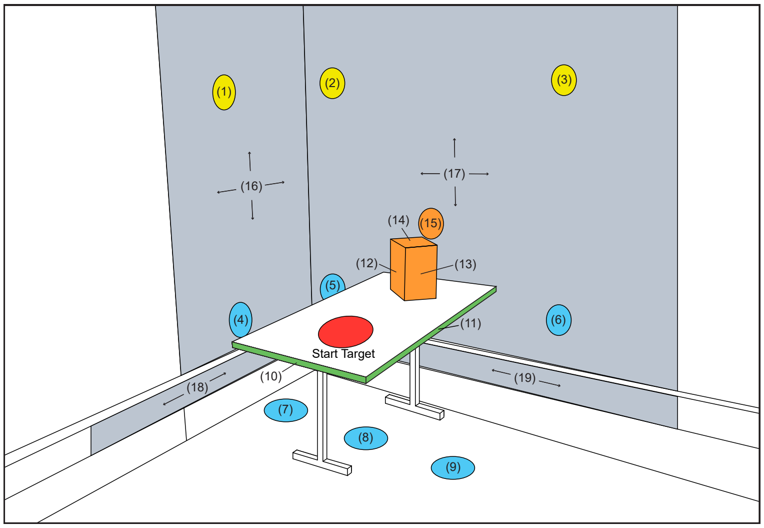


Figure 3.3: Illustration of the SAR experiment environment showing target sizes and locations. For this figure, targets coloured yellow are in the HIGH group, gray are in the LARGE group, orange are MID group, green are TABLE group, and light blue are in the LOW group. The start target is the red circle on the table, and the user's start position is located 3 meters away from each corner wall, placing them directly opposite of where the walls intersect.

The experiment task was to select two targets in sequence as quickly and accurately as possible. Targets were rendered on real surfaces using our SAR system. The targets were bright and easy to locate. Auditory feedback was given for successful and unsuccessful target selections. The participant had to successfully select the target to complete the trial, but all trials with one or more errors are noted in the log. The first target was a circular *start target* ($r = 18$cm). The centre of the target was placed at a 30cm offset from the edge of the table. The second measurement target could be either a *circle* ($r = 13$cm) or a *rectangle* of varying dimensions. There were 19 targets grouped into five types: HIGH, MID, LOW, TABLE, and LARGE. Target positions, shapes, and sizes are illustrated in Figure 3.3 and explained below:

HIGH — Composed of three circular targets positioned slightly above an average person's gaze ($\sim$176 cm) See Figure 3.3, yellow targets 1, 2, and 3.

MID — Composed of a circular target placed on the wall behind the box, and three rectangular targets mapped to the three sides of the $17 \times 27 \times 20$ cm box. See Figure 3.3, orange targets 12, 13, 14, and 15.

LOW — Composed of six circular targets placed on the floor or on the wall 20 cm below the table height. See Figure 3.3, light-blue targets 4, 5, 6, 7, 8, and 9.

TABLE — Composed of two long thin rectangular targets placed along the front and side edges of the table each approximately 3 cm high and between 50 and 100 cm wide). See Figure 3.3, green targets 10 and 11.

LARGE — Composed of four rectangular targets: two large rectangles covering the entire wall each approximately 300 cm by 300 cm and two rectangles conforming to the shape of two baseboards each approximately 25 cm by 100 cm, the bottom 30 cm above the floor. See Figure 3.3, grey walls and baseboards 16, 17, 18, and 19.

Each target was chosen to replicate realistic scenarios that may be encountered in future SAR environments. The motivation for rectangular targets was to analyze pointing on full faces of geometry (like walls and edges). We give example applications above. The circular targets represent specific content locations. In SAR, the dimensions of targets are complicated by the user position and other geometry, but the model we develop later accounts for actual target size as it appears in the environment by considering view angle and occlusion.

Unlike classic Fitts' studies [71, 249], we do not use a variation of the *Ergonomics of human-system interaction – Part 411: Evaluation methods for the design of physical input devices* [61] task. The ISO standard uses a radial set of circles around a centre point, but given the amount of geometric variation within SAR, any attempt to enforce a controlled circular pattern mapped onto the environment would render the control of distance and size nearly impossible. Considering that SAR is conforming to the physical environment, we designed this initial study to investigate pointing at targets representing possible real-world content placement. In a second study that follows, we use AR to simulate key SAR pointing task configurations with strict controls on target width, location, and size.

Design and Protocol

The design is fully within-subjects. The primary independent variables are TECHNIQUE (3 levels: VIEWPORT, RAYCAST, and DIRECT) and TARGET (19 different targets spanning five categories: HIGH, MID, LOW, TABLE, and LARGE). The ordering of TECHNIQUE for each participant was counter-balanced using a Latin square. For each TECHNIQUE, the participant completed 5 BLOCKS of 19 TARGET selection tasks presented in random order. Recall that each target selection begins with a fixed start target, so each task sequence from start target to measurement target is a measurement trial.

Before the start of the experiment, each participant was given brief instructions on how to use each of the techniques, and told to be as fast and as accurate as possible. Participants were free to move around the space, but were required to return to the starting position at the beginning of each block. No other instructions were given. For each technique, a short practice session preceded the five blocks of measured trials. Each participant completed a short post-experiment questionnaire rating each technique on four subjective measures using a 1 to 10 scale: ease-of-learning, comfort, ease-of-use, and overall performance. The entire session lasted approximately 30 minutes.

In summary: 3 TECHNIQUES × 5 BLOCKS × 19 TARGETS, resulting in 285 data points per participant.

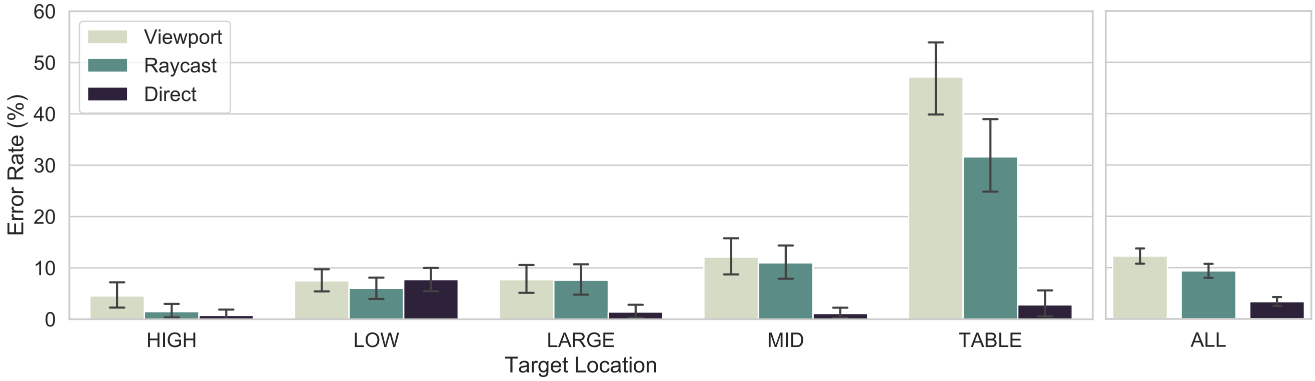


Figure 3.4: *Error Rate* for each TECHNIQUE by: TARGET type (left); all target types combined (right). Error bars in all figures are 95% CI.

Results

Repeated measures ANOVA and posthoc t-tests with Holm correction were used for all measures. When sphericity is violated, degrees of freedom are corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$). Time data was aggregated using the median to account for a skewed distribution, and a BoxCox transformation [27] corrected non-normal time data when necessary. 78 outliers more than 3 standard deviations from the mean target time were removed (1.5%).

LEARNING EFFECT We are interested in practised performance, so we verified there were no large differences in task times across subsequent blocks. There was no effect of BLOCK on *Movement Time* for RAYCAST ($F_{4,68} = 1.96$, $p < .10$) or DIRECT ($F_{4,68} = 0.32$, $p < .85$). However, there was a small effect on BLOCK for VIEWPORT ($F_{4,68} = 2.67$, $p < .03$), but corrected post hoc tests did not detect a significant result (all $p \geq .44$). There was no significant effect found in error rate across all BLOCKS. With no strong learning effects present, all blocks were retained in the analysis below.

ERROR RATE The *Error Rate* is the proportion of trials in which one or more errors occurred. Overall, direct input is least error prone and using a viewport is most error prone (Figure 3.4-right). There is a significant main effect for TECHNIQUE ($F_{2,34} = 20.32$, $p < .001$) with post hoc tests finding DIRECT has fewest errors (3.3%), followed by RAYCAST (9.3%), then VIEWPORT (12.2%) (all $p < .002$).

Direct input had as few, or fewer, errors than raycasting, while viewport typically had as many, or more, errors than raycasting (Figure 3.4-left). A significant interaction between TECHNIQUE and TARGET ($F_{3.08,52.46} = 21.37$, $p < .0001$) with post hoc tests showing that for HIGH target types, VIEWPORT (4.5%) has more errors than both RAYCAST (1.4%) and DIRECT (0.7%) (all $p < .035$). For all other target types, DIRECT is significantly less error prone ($p < .01$) with

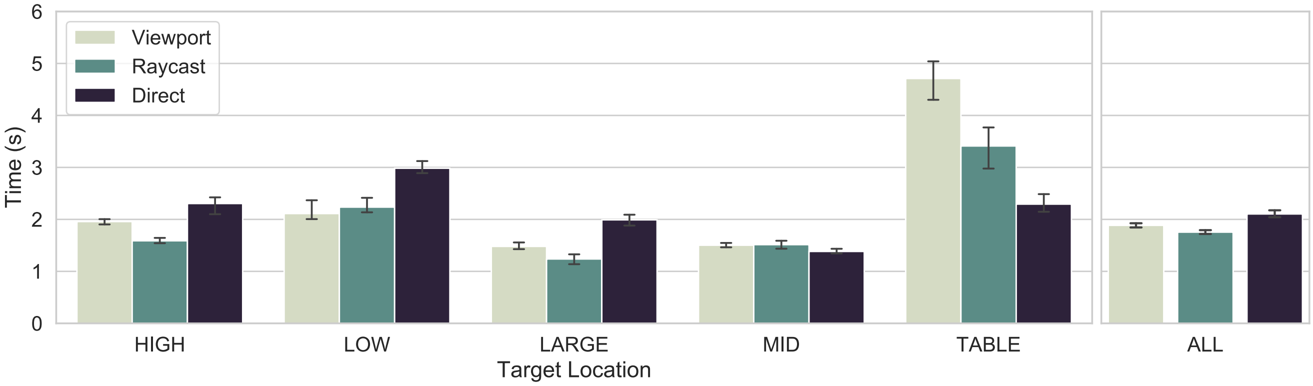


Figure 3.5: *Movement Time* for TECHNIQUE and TARGET types (left). MT for TECHNIQUE on all combined types (right).

the exception of LOW, likely due to the difficulty of reaching to tap on floor targets. A pronounced difference is for TABLE targets, where DIRECT (2.8%) has an order of magnitude fewer errors than RAYCAST (31.6%) and VIEWPORT (47.2%) (both $p < .001$).

MOVEMENT TIME The *Movement Time* is the duration from moment the start target is selected until the moment the measurement target is selected. Overall, raycasting is fastest and direct input slowest (Figure 3.5-right). There is a significant main effect for TECHNIQUE ($F_{2,34} = 7.39$, $p < .002$), with post hoc tests finding the difference between each technique significant ($p < .001$): RAYCAST (1.75s) is slightly faster than VIEWPORT (1.89s) and DIRECT (2.10s).

When considering target types, raycasting is fastest for large and high targets, direct input is fastest for targets on the table, while viewport is comparable, or slightly slower, than the fastest technique for all target types, except when targets are on the edge of the table (Figure 3.5 left). A significant interaction between TECHNIQUE and TARGET ($F_{8,136} = 69.59$, $p < .0001$) with post hoc tests finding differences between all techniques and target types (all $p < .03$), except LOW, which had no difference between VIEWPORT and RAYCAST ($p = .41$). Highlighting salient results: RAYCAST was fastest for HIGH (1.59s) and LARGE (1.24s) targets, but no significant effect was found between RAYCAST (2.24s) and VIEWPORT (2.11s) for LOW; DIRECT is fastest for both MID (1.38s) and TABLE (2.29s). For targets on the table edge, VIEWPORT is slower than the other techniques with 4.7s on average.

OCCLUDED TARGETS Our experiment protocol does not strictly control for occluded targets, but the diverse target types we test within a reasonably complex geometric setting of objects and surfaces naturally leads to trials in which there is some visual occlusion of the measurement target. To examine the effect of naturally occurring target occlusion, we create a new independent variable. Whole or partially occluded measurement targets are identified at the moment the start target is selected using the *visibility ratio* metric, calculated

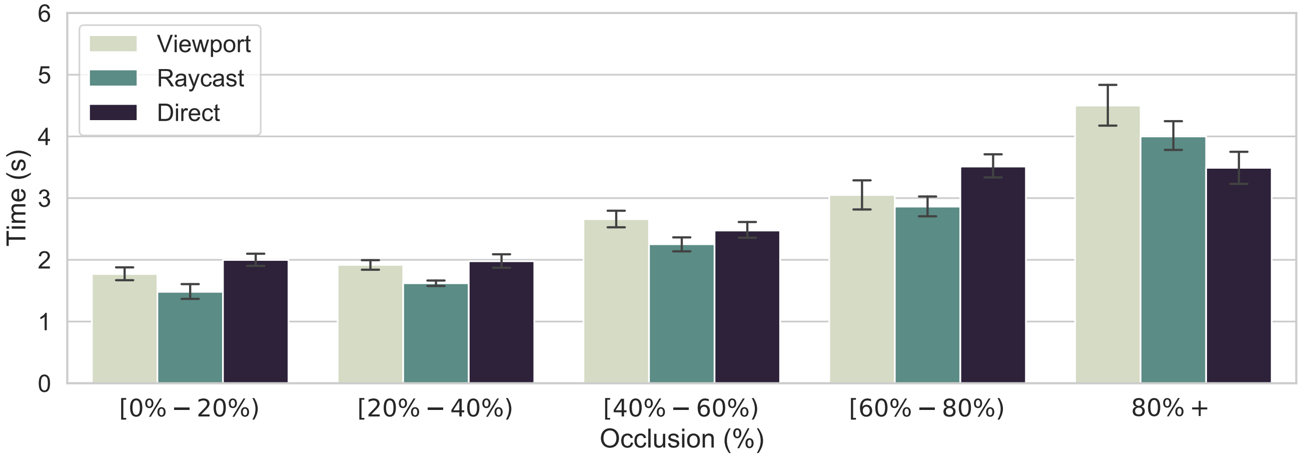


Figure 3.6: *Movement Time* by OCCLUSION by TECHNIQUE.

from the user's head (see Apparatus section). We use this to create a five-level OCCLUSION factor, with each level representing a 20% bin (see x-axis of Fig. 3.6).

There is a significant interaction between TECHNIQUE and OCCLUSION ($F_{2,5029} = 31.78$, $p < .0001$) on *Movement Time* (Figure 3.6). Post hoc tests show that target occlusion has no effect on DIRECT input for movement times for the $[0\% - 40\%)$ and $[60\% - 100\%]$ bins ($p \geq 0.32$). In contrast, there is an effect for VIEWPORT and RAYCAST, for which movement time steadily increases over each bin by an average amount of 0.68s and 0.6s respectively ($p < 0.006$).

SUBJECTIVE RATINGS After the main experiment was completed, the participant rated each technique from 1 (worst) to 10 (best) for four subjective measures. Data for each was transformed using Aligned Rank Transform [271] to correct non-normality, but no main effect for TECHNIQUE was found for any subjective measure. Combined average scores across techniques are 9.1 for *ease-of-learning*, 8.0 for *comfort*, 8.0 for *ease-of-use*, and 7.6 for *overall performance*. We expected direct input to be rated lower due to higher physical effort, but our data does not support this.

Discussion

We found important differences among the three techniques. Direct input may be slower overall for tested conditions, but it also had the lowest error rate, except for targets near the floor. In some cases, like the targets on the table edge, on the box, or behind the box, direct input was faster and had an order of magnitude lower error rate compared to the other techniques. Perhaps because that particular group of targets where narrower then the others, physical interaction made it easier to control. On the other hand, raycast was fastest overall, and as fast or faster than the other techniques for all target types except in the table group. For the most part, viewport had comparable, or only slightly worse time and error compared to raycast. Notably, viewport was as fast as raycast for targets on or near the floor, possibly due to how the mobile phone's camera naturally points down when holding it.

Overall, our results suggest raycast or viewport are good overall pointing methods in SAR, but direct input should still be considered for small targets that are within arms reach or less. Further, a hybrid technique may also be possible. Analogous to Parker, Mandryk, and Inkpen's TractorBeam [191], a method that transitions between raycast and direct pen input on a tabletop, a hybrid technique could be designed for mobile phone pointing in SAR using the context of the space and proximity of the user to surfaces. For example, if the phone contacts a surface or object, then a direct input selection is made. Otherwise, raycast or viewport pointing could be used depending on the particular use case of the task. In particular, viewport does not suffer from self-occlusion, so could be used when targets are hidden by the user's shadow and blocking a projected image from being seen [86].

The effects of target occlusion on movement time, and differences in head movement distance, especially to compensate for occlusion, suggests these are important factors affecting pointing time in SAR. In the next experiment, we strictly control these factors to better understand their effect.

3.4 EXPERIMENT 2: SIMULATED SAR POINTING TASK

The goal of this second experiment is to validate results of Experiment 1 in a more controlled SAR pointing task. To achieve high control over target placement, occlusion, and view angle, we simulate specific conditions of a SAR pointing task by rendering targets and occluding geometry in an AR HMD. The pointing context under investigation still remains surface mapped SAR since the targets are 2D, just as they would be if mapped onto real 3D surfaces. We test a reduced range of target distances compared to Experiment 1 which resulted in a decreased number of factors. This made the study practical to run within a limited time period, however it does mean our results are more representative of a best case task in terms of arm reach.

Using an AR HMD is much more practical and flexible than actuating the physical environment itself [41], or creating a physical layout of real objects and targets with constraints for the participant's initial position. Simulating SAR in AR enables target consistency across a diverse set of participants: we can place targets and objects around the user so that the distances, height, occlusion, and size are exactly the same for each participant regardless of their height or where they stand. There are limitations to this approach. The field-of-view of the HMD is smaller then the human eye, wearing an HMD can be uncomfortable and requires a tether attached to a computer, and there is no natural tactile feedback in the direct condition. However, we took steps to minimize these aspects by using a wide 90° field-of-view AR HMD, the HMD is light since it is tethered eliminating the need for heavy batteries, we

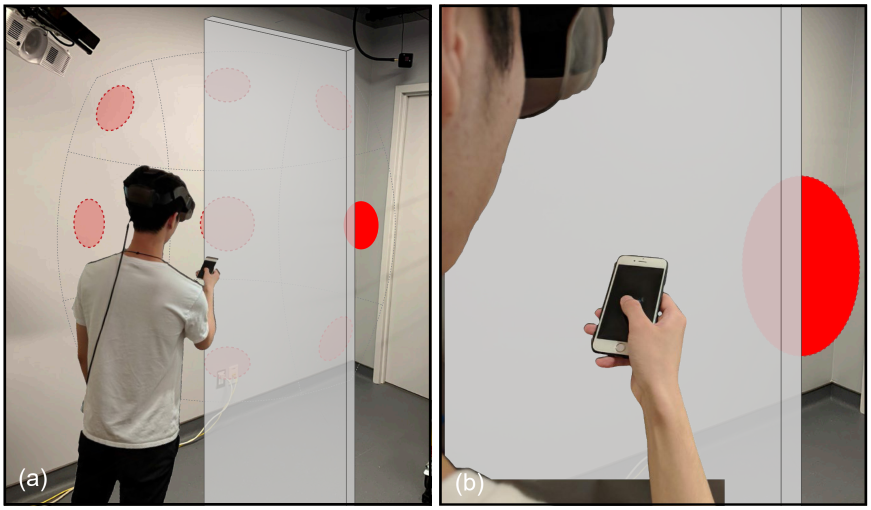


Figure 3.7: Illustration of the Experiment 2 simulated SAR pointing task using an AR HMD and real phone: (a) near target positions; (b) participant point of view showing partially occluded target. Note the real phone screen was used for input and output (there is no virtual overlay).

were careful routing the HMD tether to avoid obstruction, and we used phone vibration to simulate physical surface contact with the direct technique.

Participants

We recruited 12 participants, ages 19 to 28, 10 male, 2 female, 10 were right-handed. Overall, they reported using a mobile device an average of 3.5 hours a day. Participants received $15 for their time. This experiment was conducted 2 months after Experiment 1, and no participants participated in both experiments.

Apparatus

The Unity3D software running on the server was modified to render targets and geometry to a Meta2 AR HMD (2550 × 1440 px, 90° FOV), which is tracked with the Vicon which ensures that targets and visuals are precisely placed and remain stable relative to HMD movement. Meta2 depth compression (a known issue with the headset at that time) was corrected to simulate a real world view by applying a logarithmic function to the target and occluding geometry positions. The room was empty, neutral, and clear of unnecessary clutter. All SAR environment surfaces and targets are rendered in the HMD and illuminated to ensure easy identification in the environment. Using rendered virtual targets means there is no physical feedback in the direct technique. We vibrate the phone when it contacts a virtual surface to compensate. These considerations combine to make perception of the task in AR reasonably similar to SAR.

The same Pixel phone was used, and in all conditions, the real phone screen was used for input and output (there is no virtual overlay). For example, in the viewport technique, the actual phone screen renders a view of the same controlled 3D geometry (obstructions and targets) used to render the AR HMD. The rendering simulates what would be seen from the phone's real back camera.

Task

The task was to select two targets in sequence as quickly and accurately as possible. The first target was a circular *start target* ($r = 18$cm) located at a fixed position directly in front of the user, 150 cm above the floor, oriented towards them. The second target (the *measurement target*) was a red circle ($r = 13$cm). To increase task variability, these targets were placed at different positions relative to the start target. They were distributed on the surface of a hemisphere into 9 radial positions (30° intervals) from a point of origin (the user's head position) at a "near" and "far" distance (67 and 124 cm) relative to the origin like two concentric spheres (Figure 3.7a). The near distance was chosen to be within arms reach and the far distance requires some body movement to reach.

These varied target positions generalize our results when considering the primary factors of occlusion and target view angle. The targets are rendered in midair to simplify the scene and avoid unnecessary rendering, but they are still 2D as though they were mapped into a 3D surface. What is important is their position relative to the participant.

Design and Protocol

The design is fully within-subjects. The primary independent variables are TECHNIQUE (3 levels: VIEWPORT, RAYCAST, DIRECT), target OCCLUSION (2 levels: 100% occluded, 0% occluded), and target view ANGLE (2 levels: 0°, 90°). A target view angle of 0° means the normal of the target points towards the participant and the full target is easily viewed if not occluded. A view angle of 90° means the target normal is orthogonal to the participant's view, where the target appears as a thin sliver until the participant adjusts their head position. To control target occlusion, a large grey wall was rendered between the participant and the target to create the desired occlusion level (Figure 3.7b). Target view angle was controlled by rendering the target normal at the desired angle relative to the participant. The ordering of TECHNIQUE was counter-balanced using a Latin square. For each TECHNIQUE, the participant completed 3 BLOCKS of trials presented in random order.

The instructions, technique practice, and post experiment questionnaire were the same as Experiment 1. Participants were free to move around the space, but were required to return to a starting position at the beginning of each trial. The entire session lasted approximately 60 minutes. In summary: 3

TECHNIQUES × 3 BLOCKS × 2 OCCLUSION levels × 2 ANGLE levels × 17 target positions (8 near and 9 far), resulting in 612 data points per participant.

Results

The same analysis methods from Experiment 1 are used. Similar to the first experiment, 133 (1.8%) outliers were removed.

LEARNING EFFECT There is a significant BLOCK ×TECHNIQUE interaction on *Movement Time* ($F_{1.35,25.71} = 30.91$, $p < .0001$), but not on *Error Rate*. Post hoc tests found block 1 significantly slower than blocks 2 and 3 (both $p < .0001$), suggesting a learning effect in block 1. In all subsequent analysis, we use only blocks 2 and 3 for the best estimation of practised performance.

ERROR RATE There is a significant effect of TECHNIQUE on *Error Rate* ($F_{2,22} = 7.5$, $p < 0.01$). Overall, raycast is least error prone (4%), direct input is the most error prone (11%), and viewport falls in between (9%).

There is a significant effect of TECHNIQUE ×OCCLUSION on *Error Rate* ($F_{2,22} = 10.6$, $p < 0.001$). A post hoc analysis shows that DIRECT is least error prone when targets are non-occluded (1.8%) and most error prone when occluded (20.3%). In contrast, the error rate for both VIEWPORT and RAYCAST remained the same across occlusion levels with no significant effect ($p \geq 0.48$).

There is a significant effect of TECHNIQUE ×ANGLE on *Error Rate* ($F_{2,22} = 5.67$, $p < 0.01$). Post hoc tests show an effect of ANGLE on VIEWPORT ($p < 0.001$) where the error rate is 6% without rotation and 12% when rotated. In contrast, there is no observed effect of ANGLE on RAYCAST or DIRECT.

MOVEMENT TIME Overall, direct input is fastest and raycast is slowest. There is a significant main effect of TECHNIQUE on *Movement Time* ($F_{2,11} = 11.70$, $p < .001$), with post hoc tests finding a significant effect among all techniques ($p < .034$): RAYCAST (2.03s) is slightly slower than VIEWPORT (1.92s) and DIRECT (1.69s).

When considering occlusion and angle factors, viewport is fastest for far targets with the best view angle, direct input is fastest for near targets that have poor viewing angle, while raycast is comparable (or slightly slower) than viewport for far targets with the best view angle (Figure 3.8). A significant interaction between TECHNIQUE, OCCLUSION and ANGLE ($F_{2,22} = 21.61$, $p < .001$) and post hoc tests found varying differences between techniques, occlusion, and angle target. Highlighting the most salient results: DIRECT was fastest for all near and non-rotated targets at 1.14s ($p < .001$), and both RAYCAST (1.57s) and VIEWPORT (1.56s) are essentially tied for far, non-rotated, and non-occluded targets .

SUBJECTIVE RATINGS After the main experiment was completed, the participant rated each technique from 1 (worst) to 10 (best) using four subjective measures: ease-of-learning, comfort, ease-of-use, and overall perfor-

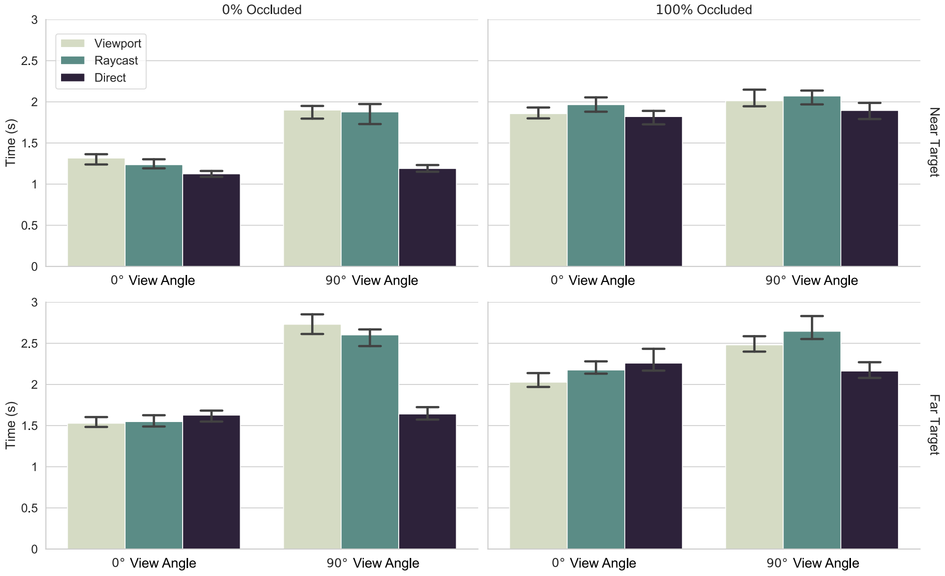


Figure 3.8: *Movement Time* by TECHNIQUE by target view ANGLE for combinations of target OCCLUSION and DISTANCE.

mance. There was a significant main effect of TECHNIQUE on *ease-of-learning* ($F_{2,22} = 9.45$, $p < 0.01$), with post hoc tests finding that direct input was perceived easier (9.0) compared with viewport (6.4) and raycast (7.75). No other subjective measures had significant effects.

Discussion

We found similarities and differences with Experiment 1. Although direct input only had simulated haptic feedback when contacting virtual targets, it still outperformed both raycast and viewport for near and rotated targets. We observed the relative robustness of direct input to rotated targets, with movement time across rotation remaining similar. However, the performance increase for direct could be partly the result of how we structured our experiment. Since our setup creates virtual walls and surfaces, the participant did not have to slow down when hitting the target like they would with a real surface, allowing them to keep their velocity and partially "punch through" the virtual wall to hit targets. People may be unlikely to strike a real surface with a phone using the same speeds and forces. With viewport and raycast, target view angle has a pronounced effect: viewport performed best for far non-rotated targets, while raycast was in-between. This contrasts with Experiment 1, where raycast was fastest with more varied target situations.

Both experiments reveal useful insights into the three pointing techniques under investigation. Experiment 1 provides a more authentic setting, which is complemented by the carefully controlled Experiment 2. Together they provide a more holistic view into how each technique performs under different SAR environment settings.

3.5 DISCUSSION

The results of the two experiments, combined with the model analysis lead to overall findings and design implications.

Direct Input Performs Well When a Target is Nearby

The good performance of the direct technique in several target conditions indicates this type of absolute direct input is well suited to SAR when targets are within close proximity to a user. This can be seen in Experiment 1 for the MID and TABLE target types. For targets in Experiment 2, direct outperforms the other two techniques in most cases, which is different than the pattern in Experiment 1 results. This may be explained by the lack of physical surfaces the user would typically need to navigate, letting them maintain velocity and move through the virtual barriers without the cost of damaging the mobile phone. There are other apparent disadvantages to the direct technique that are not present in the distant pointing methods to consider as well, like how much movement is required when targets are far away. This raises questions regarding the suitability of direct selection in large environments, in which the selection cost increases the farther the target is away from the user's initial position.

Viewport Affordances

In the discussion for Experiment 1, we were cautious to recommend viewport overall. Except for some subjective preference of certain target types, there was no clear reason to choose it over raycast or direct input for a given target context. Though Experiment 2 shows the robustness of viewport for different target types in this more homogeneous target setting. During both experiments, we observed that some participants appeared to be reluctant to adjust their physical proximity when using viewport, and would rather attempt selection even if the target was not optimally viewed by the phone camera. The result was an action of repeated (and rapid) touch selection attempts creating the high error rates for the thin table edge targets (i.e. TABLE target type) in Experiment 1. One unique aspect of viewport is its ability to overlay additional personal or contextual digital information on top of the SAR environment. Though we do not explore this explicitly, it is interesting to note the possible affordances a public and glasses-free SAR environment could have when combined with different *personal viewports*, all occupying the same SAR space. Interesting use cases include multi-person gaming, remote and co-located collaboration, and content sharing. We leave this as another possible direction for future work.

Overall, each technique has advantages and disadvantage when used in a more geometrically complex and large SAR environment. Depending on the context of the task and properties of the target relative to the user, various combinations of raycast, viewport, and direct techniques can be used to accommodate specialized content selection scenarios.

3.6 SUMMARY

In this chapter, we examined fundamental characteristics of device-based interaction in SAR: pointing at surface mapped targets. Our results show how the simplicity and speed of raycasting results in excellent performance for many situations, and how surprisingly versatile a simple method like directly tapping the phone to a target can be in many situations. Our results for our implementation of the viewport pointing method is mixed. In the ad hoc realistic SAR setting of Experiment 1, the viewport could approach raycasting performance, but was never significantly better in the tested tasks. In the controlled and more restricted setting of Experiment 2, the viewport method outperformed raycasting for distant targets that were facing the user. Our conclusion is that each method has beneficial characteristics, and that depending on the expected SAR usage context, a hybrid method or mode-switching technique to switch between methods could be the best solution.

4 SMARTPHONE MEDIATED INTERACTION IN SPATIAL AUGMENTED REALITY

Smartphone-based content and services are now central to many logistical and social aspects of life. However, a small phone screen still constrains how content can be viewed, manipulated, and shared in our immediate physical environment. One solution to constrained screen sizes is the use of external screens in the form of large displays or augmented reality to view and manipulate smartphone content. However, it is not always clear what the relationship is between the phone and content is or how that content is viewed when it is "outside" the phone.

Current phones support television "screencasting" and its subsequent use as a 'remote control'. Researchers have also proposed methods to send phone content to large displays (e.g. [28, 135]). Several other works have proposed using the phone as a pointer for varying forms of external content around a user, including for large displays [182, 225], for head-mounted augmented reality [31, 136], or for projected spatial augmented reality (SAR) [87, 196]. However, these works do not consider the problem of how to seamlessly transition a phone between its use as a smartphone to its use as a remote control for external spatial content. When using regular smartphone operations, such as swipes, taps, or rotations, to push or manipulate external spatial content, some mode switch is needed to avoid conflict between these two use cases. Although this could be accomplished with a dedicated remote control app, this kind of explicit mode switch introduces high friction when switching back and forth between mobile interaction and spatial interaction modes multiple times within a short period.

In this chapter, we propose to use a hand extension as a more implicit mechanism for switching the user's attention back and forth between the smartphone's default usage (interaction in the real world) and its usage as a push and point device for external spatial content (spatial interaction in SAR): when the user extends out their hand, the smartphone switches to spatial interaction mode, and when the user retracts their hand, it switches back to mobile interaction mode. Based on this intuitive mode switch, we describe the design of our interaction technique that enables the user to push smartphone content to an external SAR environment, interact with the spatial content, rotate-scale-translate it, and pull the content back into the smartphone, all the while ensuring no conflict between the mobile interaction mode and spatial interaction use. While similar gestures have been proposed as design techniques [28], there have been limited sensing investigations that demonstrate that such an intuitive mode switch is feasible. We evaluate the classification of extended and retracted states of the smartphone based on the phone's relative 3D position with respect to the user's head while varying user postures, surface distances, and target locations. Our results show that a

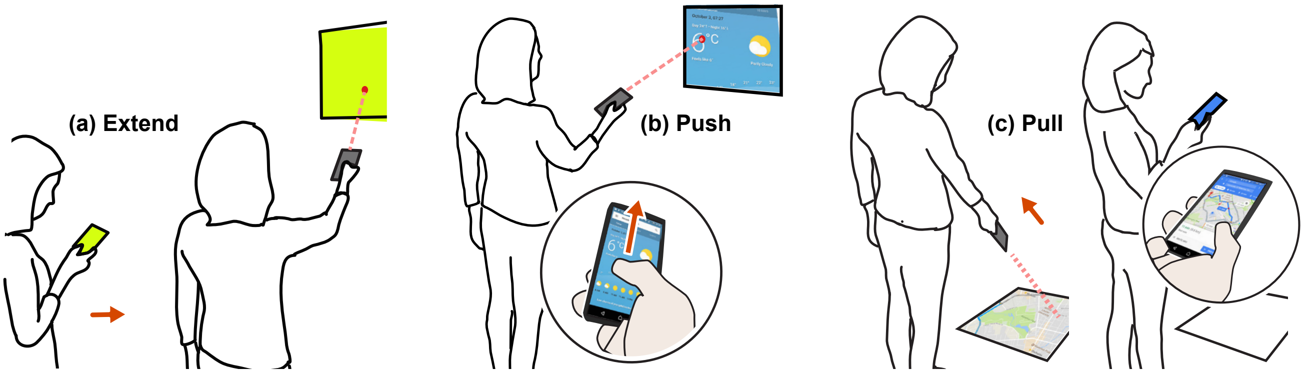


Figure 4.1: Illustration of the basic technique: (a) arm extension switches from mobile interaction mode to spatial interaction mode; (b) extending arm while holding finger on an application screen, then flicking up, pushes content to environment; (c) pointing at content in environment, then flicking down, pulls content to phone application for detailed manipulation.

random forest classifier can classify the extended and retracted states with a 96% accuracy on average.

4.1 RELATED WORK

In Chapter 2, we looked in detail at pointing within immersive environment. Here, we explore pointing for the purpose of mode switching and manipulation of digital content. First, we look at works that use the smartphone as a pointing device for external content manipulation. Second, we investigate around-body interaction that pertains to our usage scenarios.

Smartphone as a Pointing Device

Multiple works have explored the use of smartphones as pointing devices for controlling content on large displays, augmented reality, and spatial augmented reality. Myers et al. [182] investigated large display pointing with a laser equipped Personal Digital Assistant, which has a similar form factor to a mobile phone. Their Semantic Snarfing technique is used for remote laser pointing, and features a method to capture remote content into the phone for detailed manipulation. PointerPhone [225] studied how a laser equipped mobile phone could be used with a large display across six tasks, including similar capture techniques that can transfer external content to the phone's display. Beaudouin-Lafon et al. [9] investigated the use of a mobile phone for interaction in a multi-display environment using unimanual and bimanual gestures. Langner et al. [135] developed a flick-transfer gesture for content sharing to a large display which is combined with a hybrid raycast and orthogonal pointing technique. However, none of these techniques address mode switch as a problem and assume that the user is using an application dedicated to interacting with the large displays. Similar to our work in this chapter, Code Space [28] proposes arm extension as a form of an implicit mode switch for a multi-display environment to enhance the code review process.

Techniques have also been proposed that combine a mobile device with an AR HMD for spatial selection [136] or for visualization of high-dimensional datasets [227]. Büschel et al. [31] used a mobile phone with an AR HMD to evaluate pan and zoom techniques for 3D data spaces. They found that device movement and touch-based drag operations were most effective for unimanual interaction.

Around-Body Interaction

Conceptually, the area around the body has pericutaneous, peripersonal, and extrapersonal layers [58]. Each layer describes how we view ourselves in relation to the objects situated around us, and prior work has investigated aspects of these layers to expand the set of affordances the mobile phone can provide. For example, the space in front of the user has been imagined as containing hidden digital information that is viewed through the mobile phone's screen [280], or as a means to explore multi-layered panorama images [251].

Most relevant to this chapter is using the space around the body for input. Virtual Shelves [139] used spatial locations positioned around the user to trigger mobile phone shortcuts, and Chen et al. proposed a set of techniques that map in-air spatial locations (as well as body parts) to a set of gestures for information retrieval, storage, and actions [39]. Chen et al. conduct a preliminary study where they use the mobile phone's 3D position relative to the location of the face to classify the phone's position along different distance and orientation categories [40]. The study is a preliminary study consisting of only a single user. Our work classifies the extended and retracted states which depends on the distance and orientation of the phone relative to the user's head, while considering other influencing factors including the target location and the user's posture.

In the next section, we describe the design overview of our technique that ensures conflict-free interaction for mobile and spatial modes, while ensuring other design principles including user comfort and eyes-free operation during spatial manipulation. We then describe our prototype implementation, followed by the classification analysis and usability study.

4.2 DESIGN OVERVIEW

The primary goal of our interaction technique is to use an arm extension as an intuitive mode switch to support both a default mobile interaction mode as well as a rich spatial interaction mode when interacting within a SAR environment. The interactions supported for the spatial mode are: push content from smartphone to SAR, delete content from SAR, RST (rotate-scale-translate) manipulation of app windows in SAR, and capture content from

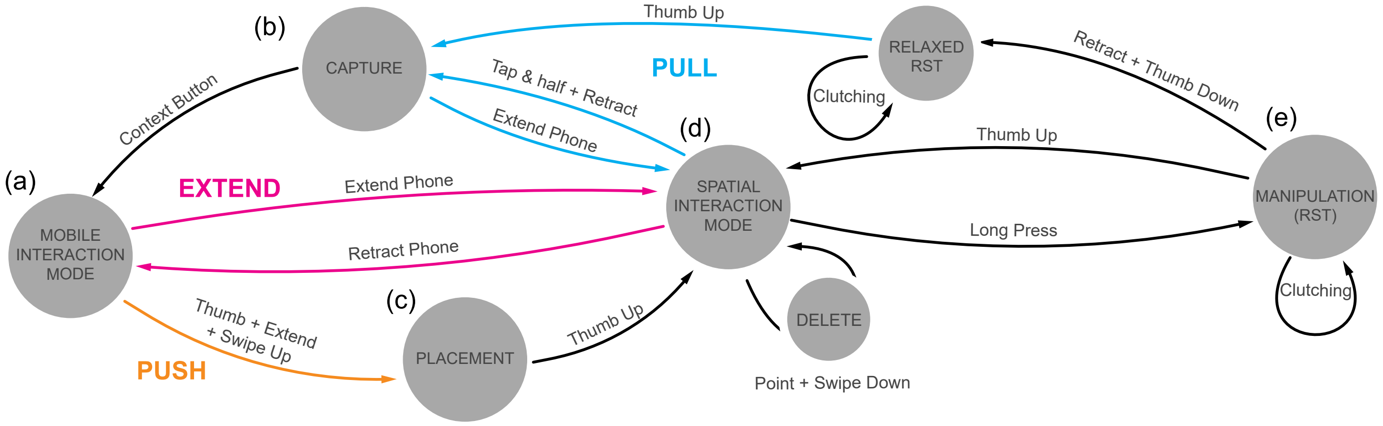


Figure 4.2: Interaction state diagram: (a) the native mobile phone application when arm is retracted; (b) a tap-and-a-half while retracting the mobile phone captures the spatial content that is in focus; (c) content is placed by holding the thumb down on an item (e.g. photo, app, etc), extending the arm, and flicking up; (d) extend arm to activate spatial interaction mode, removal of spatial content is achieved by pointing and flicking down on screen; and (e) holding the thumb on the screen while pointing enters manipulation mode where RST can be performed, retracting the arm in this state allows a relaxed posture.

SAR to perform synchronized content-specific manipulation between SAR and the smartphone.

Our design is aimed at achieving the following five design goals:

1. *Intuitive:* Transitioning between a native mobile phone application to spatial content should be easy to understand and discover.

2. *Conflict-free:* The method should avoid actions that conflict with existing system wide mobile phone input. For example, the smartphone supports different types of touch gestures, bezel swipes, force presses, and overloaded physical buttons, but all of them have designated default system-level or application-level functions and cannot be used to enable fast, low-friction mode switch to another spatial mode. The arm extension and retraction enables a conflict-free mode switch while being *intuitive*.

3. *Comfortable:* One problem with using the phone as a remote pointer when the arm is extended is that it leads to rapid arm fatigue (gorilla arm effect). We avoid extended periods of strain in our design by enabling a relaxed RST mode where the user can perform RST operations with a retracted hand while maintaining the conflict-free use.

4. *One-Handed Extended Use:* All extended hand interactions in our design work one-handed because it is difficult to perform interactions with two extended hands.

5. *Eyes-free Extended Use:* When interacting in extended mode, the interaction should not require the user to look at the phone screen because it may not

be easily visible and also because the user should be able to focus on the spatial content while manipulating it. Our design ensures this by using a combination of taps, long presses, swipes, and 3D displacement and rotation of the phone in the extended mode, all of which are eyes-free.

Interaction Technique

Figure 4.2 illustrates the action states and transitions in our interaction technique.

Extended and Retracted State (Fig. 4.1a, Fig. 4.2a,d)

The user extends their hand to interact with the spatial content. The system continually uses the 3D position of the phone relative to the user's head to determine if the phone is in the extended state or retracted state. As soon as the system detects that the user has transitioned from retracted to extended, the system enters the spatial interaction mode. The extend motion naturally becomes a pointing gesture to specify a spatial location to place, remove, or manipulate content. When the user brings their arm back to the retracted state, the system switches back to the smartphone interaction mode. To enable *comfort*, the exception to this rule is when the user wants to perform relaxed RST manipulation or content-specific manipulation. While the user is in the extended state, the user can perform specific gestures to continue to interact with the spatial content in the retracted state. We detail these later.

The notion of extending the hand vs. retracting is subjective and does not depend solely on the distance or orientation of the phone. Primarily, it depends on four factors: 1) *Target Location*: the targeted spatial location of interaction. For instance, the distances when the user extends the phone towards the floor, wall, or roof would be very different. 2) *User Posture*: There would be variations in how the hand is extended, depending on the user's posture, whether they are standing, sitting, or lying down. 3) *Distance of the projection surface*: The arm extension will also be impacted by the distance of the projection surface. For instance, the extension may be smaller if the wall is nearer and less than arm's length. 4) *Users*: Different users may extend their arm differently. For example, while some may perceive an arm extension to be a complete arm-stretch, others may opt for a slightly more relaxed version closer to their bodies. There may be different ways users respond in the above conditions. Further the distance of the phone relative to the head may also depend on users' arm lengths. Due to these factors, it is difficult to specify a simple threshold-based classification of extended vs retracted states or use heuristic based raycasting like in Langner et al. [135]. We therefore conduct a classification study as described in the next section.

Placement and Removal (Fig. 4.2c)

To distribute spatial content from the mobile phone into the spatial environment, the user holds their thumb on top of the application they wish to place in the environment. With the thumb held down, they extend their hand to switch into spatial interaction mode and flick their thumb up (Fig. 4.1b). This can be done *one-handed*. The location and orientation of the content is dependent on where the ray from the mobile phone points just before the flick occurs. This avoids any errors due to unintentional movements during the flick. Raycasting has been shown to have good performance for these types of tasks [87].

Removal of spatial content works in a complementary way. When in spatial interaction mode and pointing at a content item, swiping down on the phone screen removes it (Fig. 4.1c).

RST Manipulation (Fig. 4.2e, Fig. 4.3)

We consider four of our design principles when constructing the interactions around spatial content manipulation: *One-handed*: Any two-fingered gestures like pinching or rotating are not possible, *Intuitiveness*: RST interactions should not be hidden behind nested menus or a complicated interface, *eyes-free extended use*, and *Comfort*.

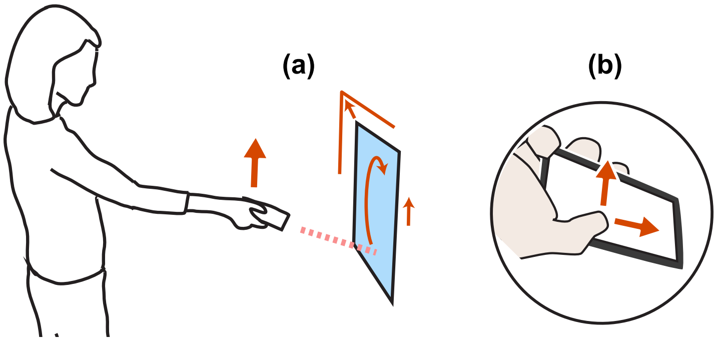


Figure 4.3: RST manipulation of spatial content using eyes-free touch for rotate and scale and raycasting for position.

To manipulate content, the user must extend their arm and point the mobile phone towards a spatial content item, then hold their thumb anywhere on the screen for dwell period of 200ms. After, the spatial content will be in a selected state and the user can relax their arm to a *comfortable* position.

Rotation is accomplished by moving the thumb along the x-axis of the mobile phone. The rotation occurs around the content's centroid where the rotation axis is the surface normal. Moving the thumb to the left will rotate counter-clockwise and to the right, clockwise. *Scaling* uses thumb movement along the y-axis of the mobile phone. This will cause a uniform scaling across all dimension, making it larger when pushing the thumb up, and smaller then pulling the thumb down. *Translation* uses raycasting, the content will automatically follow the ray and snaps to the intersecting spatial surface.

For comfort, the user can relax their arm during RST by retracting their hand while the thumb is down on the touchscreen. The system then enters the *Relaxed RST* mode and stays in it as long as the thumb does not lift from the screen for more than 1000ms (Figure 4.2e). This delay is needed to enable clutching for rotation and scaling. Figure 4.2 shows how the interaction remains *conflict-free*.

Capture for content-specific manipulation (Fig. 4.2b)

Capturing spatial content into the mobile phone enables more detailed manipulation, for example adjusting application-specific parameters of the content, such as a map location, or weather forecast type. This can be thought of as an extension to the content itself, an *intuitive* remote interface. To capture content, the user extends their hand, points toward the content, performs a tap-and-a-half (a tap immediately followed by a touch-down) on the screen, and then brings the phone back towards their body into a *comfortable* state. This opens a specialized application-specific interface corresponding to the spatial content. Exiting content capture mode uses a method *compatible* with standard mobile operating systems, like the contextual back-button or home screen gesture.

4.3 IMPLEMENTATION AND APPLICATIONS

We built a proof-of-concept system to enable applications that demonstrate our interaction technique in SAR. To eliminate confounds and simplify engineering, we use a commercial motion tracking system to track the user's head and the phone. Later, we describe how this system was used first to evaluate the feasibility of the extend gesture while gathering data to build a recognizer, and second, to evaluate the usability of our interaction technique.

SAR Environment

Our environment is a corner of a large room occupying approximately 4 × 4 meters of floor space (Fig. 4.4). Placed around the environment are five digital projectors, six Microsoft Kinect cameras (each connected to an IntelNUC Intel® Core™ i7-7567U PC), and a ten-camera Vicon motion tracking system (Vera/Bonita IR cameras). An instance of the Vicon Tracker 3.6.0 software running on a dedicated server handles real time tracking of a mobile phone and a person's head. The phone tracking object is a custom-printed phone case with seven 6.4mm spherical reflective markers and two 9.5mm ones. The head is tracked through a baseball cap with five markers attached to the visor and crown. All tracking is filtered using the One Euro Filter [36] ($f = 9.9$ and $\beta = 0.5$ for position, $f = 20$ and $\beta = 0.5$ for orientation).

The main server (Windows 10, Intel® Core™ i7-6850K) is connected to the Vicon server and IntelNUCs using a local intranet (LinkSys WRT3200ACM 10Gb router). All data processing and software powering the environment is computed and rendered using this main server. The server sends transformed projection-mapped content to the five projectors using two GeForce® GTX 1080 WINDFORCE OC 8G graphic cards at approximately 60 FPS.

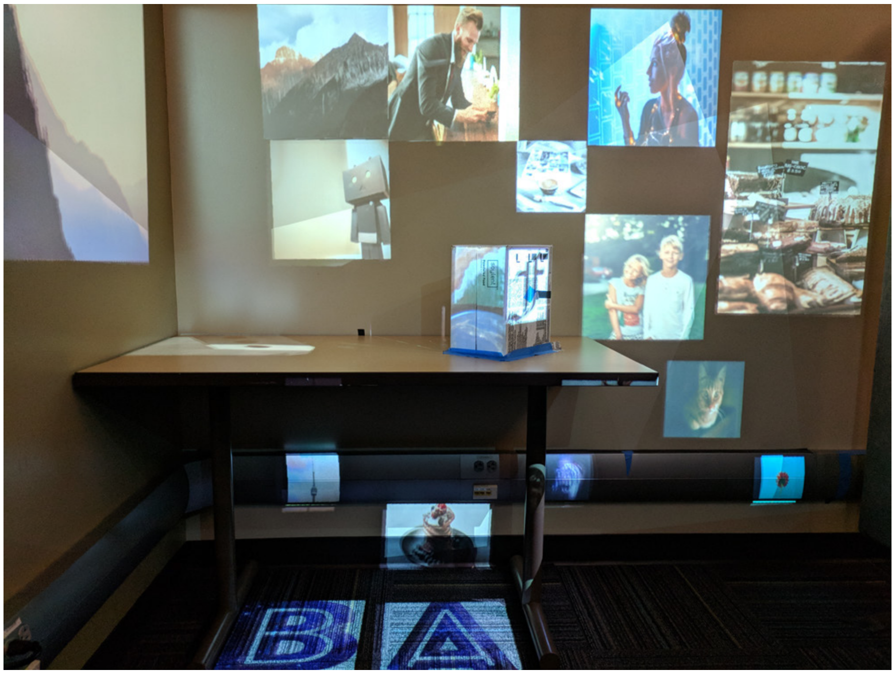


Figure 4.4: An example of spatially distributed content viewed in a SAR setup. Note how content can be displayed on any surface including walls, floor, furniture, and objects.

The software powering the environment uses Unity[1] for the rendering back-end. Projectors and Kinect cameras are calibrated using the RoomAlive toolkit [114]. The resulting 3D reconstruction of the room imported into Unity. Further calibration synchronized the 3D environment with the Vicon tracking system.

The mobile phone is a Google Pixel (5.0 inch display, 149 × 74 × 11mm with case) running Android 8.1. The complete environment allows for fast and accurate prototyping of various interaction techniques within a spatially enabled environment.

System Architecture

Our framework handles connections, event processing, and room rendering. Each connected mobile phone contains a spatial client running in the background that communicates with the native applications running on the device. The client handles phone localization, gesture recognition, and switching between personal mobile phone use to spatial interaction. All communication from a native application to its spatial content is handled through the client by a set of application programming interfaces (API). These sets of APIs provide an interface for mobile applications to create, delete, control, and manipulate associated spatial content.

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

The spatial server receives communication events from the client, manages the spatial content, and handles projection mapping. All connected projectors are managed by the RoomAlive Toolkit [114]. All spatial content is persisted inside the server where all logic for content layout, such as snapping to planar surfaces, are handled.

Demonstration Applications

We implemented three prototype applications using Unity for a modern mobile phone (Fig. 4.5).

PHOTOS APPLICATION (FIG. 4.5 TOP) To place a photo in the environment, the user touches a single photo in the application with their thumb. With the thumb on the photo, they extend their arm to activate spatial interaction mode, and flick their thumb forward to push the photo onto the surface the mobile phone is pointing toward. This can be repeated for multiple photos. Once a series of images have been placed, position, scale, and orientation can be determined through the manipulation interactions described above, or the other attributes (e.g. brightness) can be controlled by pulling in the photo, bringing up the spatial UI.

WEATHER APPLICATION (FIG. 4.5 MIDDLE) To place ambient weather information in the environment, the user touches a GUI component with their thumb, extends their arm, and then flicks their thumb forward to place the ambient display on one of the room's surfaces. The location, scale, and orientation can be manipulated similarly to the photos example. If the user needs finer control over aspects of the spatial content, they can capture the spatial UI through the pull gesture described previously.

MAPS APPLICATION (FIG. 4.5 BOTTOM) To place a map, the user touches the map bar on the bottom of the application, extends their arm, and flicks their thumb forward. The location, scale, and orientation can be adjusted through the methods stated previously. If the map is placed on the floor, it can create the illusion of walking long the route presented on the map.

4.4 STUDY 1: EXTENDED VS RETRACTED CLASSIFICATION

Our technique requires robust detection of whether the user is in the *extended* state or the non-extended *retracted* state when the user interacts with the touchscreen. Existing work [40] shows promise that the position and orientation of the phone with respect to the user's head can be obtained using inside-out tracking from the phone. One trivial approach to determining the state is to calculate the distance using ℓ_2-norm from the head to the mobile phone and use a simple threshold for delineation. However, as mentioned earlier, this approach would be unable to generalize for deviations in the mobile phones's target location, user posture, surface distance, and a user's

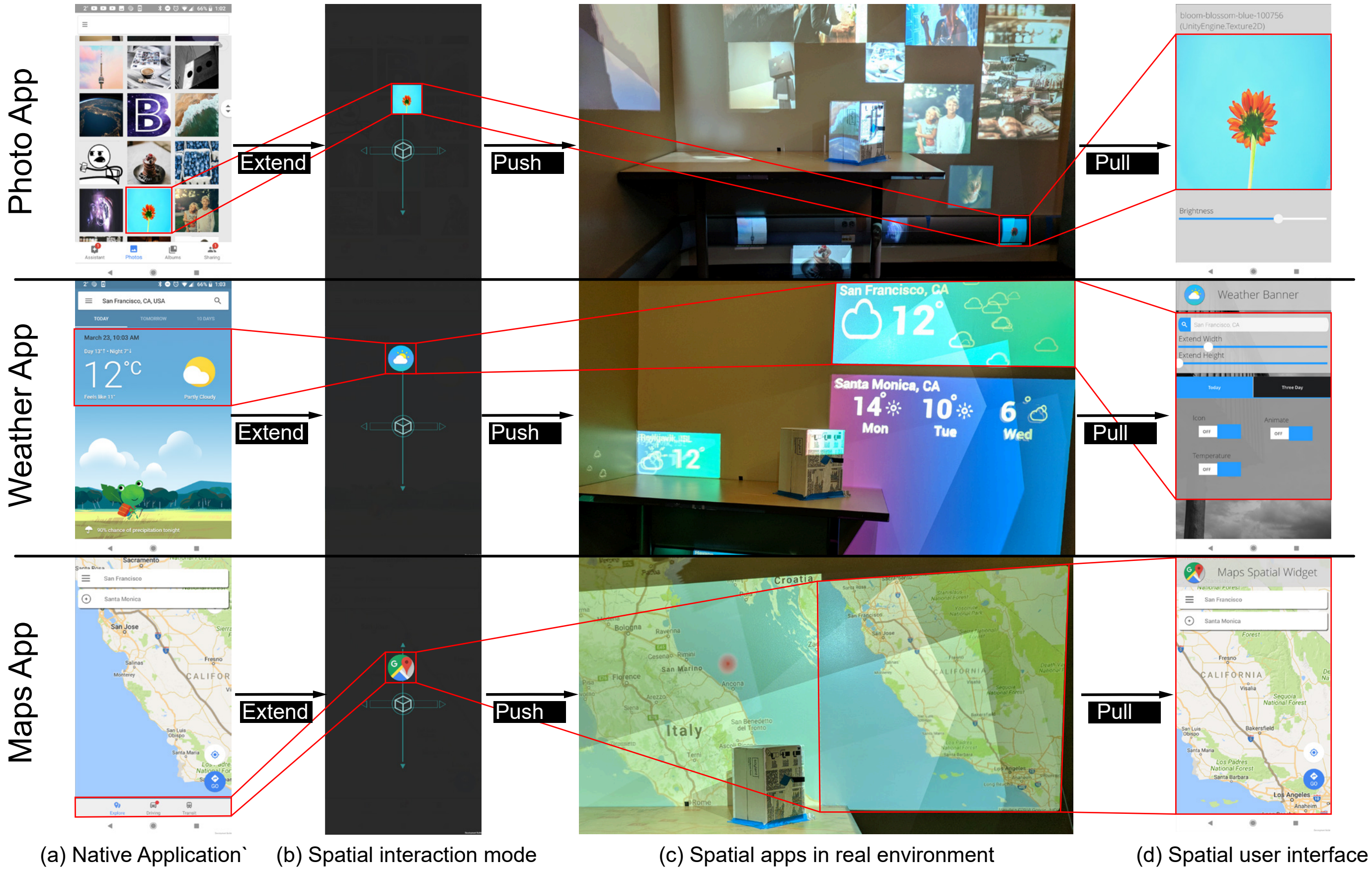


Figure 4.5: Three example scenarios created with our framework. (a) The native photo, weather, and maps application use the spatial APIs to enable elements of their interface for spatial use (highlighted in red). (b) Content can be pushed onto surfaces in the physical environment by extending the arm and flicking the thumb forward. (c) Spatial content existing within the physical environment and managed through the framework's spatial server. (d) Content already in the environment can be pulled into the mobile phone by using a tap & a half gesture which will bring up a customized spatial UI for detailed adjustments on the mobile phone.

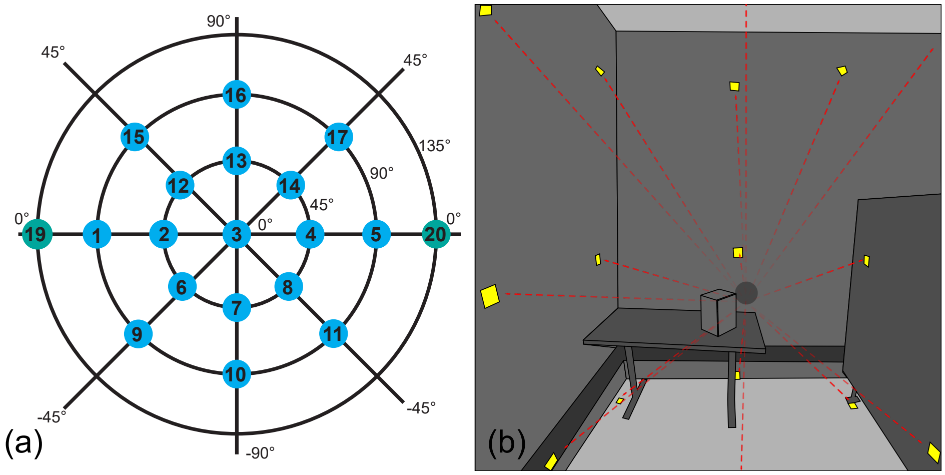


Figure 4.6: Target placement. (a) Depicts the mapping of targets onto a 2D projected sphere, where target 3 is the forward vector relative to the user's head. The far variant for sit and stand uses all the blue targets, and the near variant uses all targets up to 11. Supine-far uses a subset from 1-11 excluding 9 and 11; and supine-near uses a subset (1, 2, 4, 5, 10, 13, 16, 19, 20). (b) Illustration of the target placement for the sit-far configuration mapped onto a physical environment.

specific way of extending their hand and their arm length. To demonstrate feasibility of our interaction we need to demonstrate the feasibility of accurately classifying the *extended* and *retracted* states under the variations of these factors.

We conducted a study to collect data on multiple extend target locations (angles) across three different postures: standing, sitting, and laying down supine, two different surface distances: near and far, across 12 users. We trained a binary classifier on the collected data that consisted of smartphone's position and orientation relative to the head. We also used the user's height as an additional feature to investigate its effect on the classification. We now describe the experiment procedure and classification results.

Data Collection

We recruited 12 participants, ages 20 to 29, 3 female. All participants were right-handed. Most participants actively used a mobile device an average of 4.1 hours a day. Height varied within 158 cm to 187 cm and the length of their right shoulder to their index finger ranged from 66cm to 79cm. Participants received $15 for their time.

We collected data for six configurations consisting of posture state (sit, stand, and supine) and room state (near and far): sit-near, sit-far, stand-near, stand-far, supine-near, and supine-far. An office divider 168cm tall and 151cm long oriented perpendicular to one wall allowed us to simulate near and far surfaces.

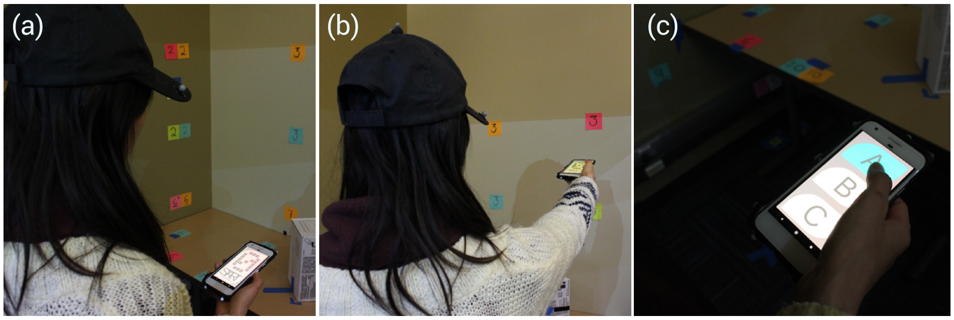


Figure 4.7: Study 1 trial task. (a) User is prompted to select a target upon which the user taps to confirm their retracted state. (b) User performs the extend gesture and flicks their thumb up on the screen. (c) User retracts the hand and performs a tap to confirm the retracted state and is then prompted to select three buttons ('A', 'B', and 'C') to simulate native phone usage until the next trial.

Physical targets were placed around the user with an associated number and color (Fig. 4.6). Targets were positioned relative to a canonical head location with angles determined by a laser pointer attached to a mobile phone with an orientation sensor. In each stand-far and sit-far configuration, targets were placed in the environment using 0°, 45°, and 90°offsets across both the x- and y- axes relative to their origin point, resulting in 17 directions (Fig. 4.6a *all blue targets*). In each stand-near and sit-near configuration, targets were generated with a similar approach, resulting in 11 directions (Fig. 4.6a: *blue targets 1-11*). Supine-far excluded targets 9 and 11, while supine-near used a subset of all 20 targets, resulting in 9 directions (see Fig. 4.6a).

The task in each trial was to extend, point towards a specified target, and retract back (Figure 4.7). At the beginning of a trial, the participant holds their phone in the non-extended *retracted* manner. They then receive a mobile phone prompt to extend and point to a specific target. Participant taps the screen and then extends their arm towards the target and swipes up. The participant then retracts the arm and taps again followed by a series of button presses to simulate phone usage before the next trial starts. The data is recorded at the time of the two taps and the swipe up gesture. Participants were asked to extend their arm naturally without overstraining their arms.

The order of the 6 configurations were counter-balanced using a balanced Latin square. For each configuration, the participant completed a short practice block of trials, then 3 blocks of measured trials consisting of all target positions in a random order. Participants were given breaks after each block to ensure minimal effect of fatigue on the data. Each session lasted approximately 70 minutes. In total, there were 12 participants × (17 [stand-far] + 17 [sit-far] + 11 [stand-far] + 11 [sit-far] + 9 [supine-far] + 9 [supine-near]) × 3 blocks = 2,664 trials that were used for classification.

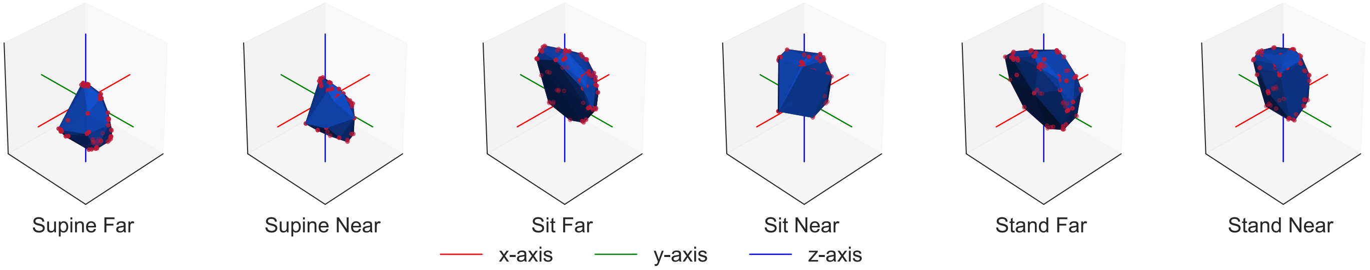


Figure 4.8: Three-dimensional volumes depicting the extend gesture point clouds. Origin is the head position and axes range from ±100cm .

Classification

Figure 4.8 shows the convex hull for the *extended* smartphone's relative position with respect to the head. It illustrates the diversity in the point clouds of the six configurations. We first conduct an analysis of how much of the data can be explained by using a single radius threshold value. The spherical volume that results from the radius delineates the space. We optimized a sphere fitting algorithm that minimises its cost function to find the optimal radius (x) through least squares [176]:

$$\underset{x}{\operatorname{argmin}} \sum_{q \in Q} x - \frac{q_n + q_e}{2}$$

Q is the set of datapoints containing the head to mobile phone distances, q_n is the distance in the *retracted* state, and q_e is the distance in the *extended* state. The resulting optimal radius come out to be 53.85cm with a classification accuracy of 82.9%. This shows that the optimal radius can delineate 82.9% of the *extended* and *retracted* data. Of course since the optimization is across the whole data set without splitting it out for individual test sets, whether this radius value generalizes well is an open question. However, it does indicate that a more advanced classifier that includes the relative position and orientation features might yield good generalizable performance. We trained a per-user random forest classifier [72] as well as a general leave-one-out cross-validation classifier for each of the six configurations.

PER-USER CLASSIFIERS We trained on two blocks of user data and tested on the third. We evaluated all three train-test combinations and averaged the results per user. The overall mean accuracy for all users came out to be 96%. A summary of the results can be viewed in Table 4.1. For the conditions stand-far, stand-near, and sit-far, the classifier shows near perfect accuracies. The sit-near condition is also high. However, the accuracy of both the supine-far and supine-near conditions are lower than the other conditions overall. This can be explained by how the participants held the phone while in a the supine posture, which deviated from both the sit and stand postures.

The user's height may influence the length of the hand extension gesture. We added the users' heights as a feature and redid the above analysis. Table 4.1 shows that while accuracy for the stand and sit conditions remain

Classifier		Stand Far	Stand Near	Sit Far	Sit Near	Supine Far	Supine Near
General	*M*	96.09	96.08	97.48	91.65	88.70	82.03
	SD	3.75	4.62	2.66	4.09	14.92	13.25
Per User: Height	*M*	**98.63**	**98.87**	**99.50**	**96.26**	**95.47**	**93.15**
	SD	2.13	1.01	0.56	2.18	4.31	4.73
Per User: No Height	*M*	98.46	98.61	99.45	96.11	93.20	90.35
	SD	2.27	1.18	0.61	1.83	5.03	5.07

Table 4.1: Three random forest classifiers trained on different variations of user data: *General* is trained on all data using cross-validation; *Per User: Height* is trained for each user using height as a feature; *Per User: No Height* is trained for each user without height. Overall accuracy is 96% (SD 4.5).

relatively unaffected, the accuracy in both supine conditions have improved. We conducted the McNemar's test [52] to compare the performance of the two classifiers for both the supine-near and supine-far conditions; the difference came out to be statistically significant ($p < 0.05$). Thus, including height as one of the features can increase the accuracy of the supine condition by a low but significant percentage for per-user classifiers. Overall, the results show that with user-specific classifiers, the extend gesture is a practical possibility.

GENERAL CLASSIFIER (LEAVE-ONE-OUT CROSS-VALIDATION) To evaluate the general predictive accuracy of the classifier, when there is no training data from the user, we conducted a 12-fold leave-one-out cross-validation where data from 11 users were used for training and the 12th was used for testing in 12 round robin rounds. The overall accuracy with a random forest classifier came out to be 92%. A summary of the results in Table 4.1 shows the accuracy per condition. The accuracy for stand-far, sit-far, and stand-near are good enough for practical use. However, the accuracies of sit-near and supine-far are lower. The accuracy of supine-near at 82% indicates the dependence of the user's specific way of handling a phone when laying on their back. Adding the height feature only added a marginally observable difference in this case and is therefore not reported.

SUMMARY Overall, our results show that simple heuristics are unable to account for the extend gesture's variance, and by utilizing the mobile phone's position and orientation, and the head to mobile phone distance of the user, a high degree of accuracy can be obtained (96%), thus demonstrating the feasibility of using arm extension as an intuitive mode switch.

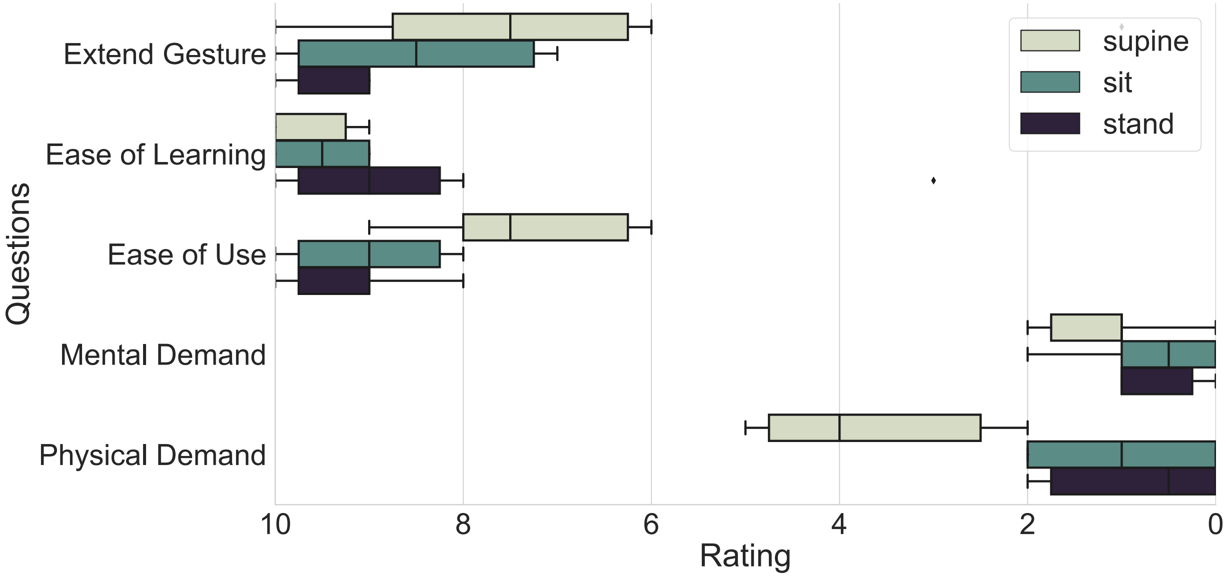


Figure 4.9: Boxplots showing usability study results.

4.5 STUDY 2: PILOT USABILITY EVALUATION

We evaluate the end-to-end usability of our interaction technique using the three applications we described above in a pilot study. We recruited 6 participants that did not participate in the previous study: ages 20 to 25, 1 male, all right handed, reported phone usage 3.6 hours per day on average. Remuneration was $10.

The protocol was as follows. First, the participant was briefly instructed on how to use the interaction technique, then they used the system for 5 minutes to familiarize themselves and practice the different actions. Next, they performed the different actions used by the interaction technique while assuming different postures: standing, sitting, and supine (laying down). After, they used the complete interaction technique in realistic usage scenarios enabled by the three prototype applications described above. Again, they completed each scenario while standing, sitting, and supine. At the end, they rated each posture condition on multiple measures, and participated in a closing interview. The posture condition order was counter-balanced.

Results

Ratings by posture are provided in Figure 4.9. Each uses a scale from 1 and 10, where 10 is a positive rating for *Extend Gesture*, *Ease of Learning*, and *Ease of Use*. For *Mental Demand* and *Physical Demand*, 0 indicates less demand.

Participants found the interactions easy to use (stand = 9.2, sit = 9, supine = 7.3); easy to learn (stand = 8.2, sit = 9.5, supine = 9.6), and thought they integrated well with the existing mobile phone ecosystem. The mental and physical demand were rated low for all postures (lower means less demand) except for supine which was rated higher then the others for physical demand (3.6). Overall, participants found the extend gesture intuitive to use for stand (9.3) and sit (8.5), while the gesture for supine was sometimes seen as cumbersome (6.8). Five participants stated that they would use spatial applications at home or office, but all were neutral on using them in a public space. All participants found laying down supine and using a mobile phone with a single hand sometimes difficult.

4.6 DISCUSSION AND FUTURE WORK

Our interaction technique is highly dependent on tracking a smartphone relative to the user's spatial location. In this section, we discuss current limitations with possible solutions and present compelling directions for future work.

Real World Tracking of a Mobile phone

Our current system uses absolute tracking provided by a Vicon motion tracking system to accurately track the mobile phone and the user's head position within an instrumented area. This was done to simplify prototyping and provide experimental control, so verifying that our techniques will work outside this kind of fixed tracking environment is currently an open question. However, recent advancements in 3D tracking that utilize a combination of an accelerometer and "inside-out" computer vision techniques [138, 180, 273], provide a robust experience for current generation mobile AR. Implementing and testing our interaction methods in these kinds of ad hoc tracking contexts remains a topic for future work.

Extended vs Retracted Classification

Our classification results demonstrate the feasibility of using arm extension as an intuitive mode switch gesture and provide the impetus for the next set of investigations in this space. There are multiple directions of future work pertaining to this classification problem. First, our results currently depend on the awareness of the configurations that the user is in. The user could set this up in the beginning depending on their most frequent use-case and switch it when their configuration changes. The implicit recognition of user posture and surface proximity is a good subject for future work. Second, while we demonstrate the feasibility of the extend gesture using robust 3D positions obtained from external tracking, further investigation is needed to ascertain that the 3D position obtained from inside-out tracking using a combination of 3D environment mapping, face tracking, and inertial measurement units provides a similar level of robustness. Third, we observed higher accuracies for per-user classifiers and more work needs to be done to investigate quick user calibrations or on-the-go personalization of the classifier model.

Extending the Interaction Space

The interaction vocabulary currently supports a subset of the interactions possible within an augmented environment (Fig. 4.2). A natural extension to explore would be the manipulation of grouped content, content snapping and layouts, and other higher level functionality whereby multiple objects can be manipulated at once.

In our technique design, we purposely created it to be usable across three common postures a user would frequently encounter. However, instead of our posture-invariant technique, it would be interesting to explicitly use these postures to control aspects of application state, changing how the technique functions based on the current posture. These posture-dependent techniques could be an interesting area for future work.

Direct Touch

Some participants found it difficult to perform the extend gesture while laying down supine (Fig 4.9). Comments indicate that they had trouble lifting the mobile phone away from their body and that they had a hard time holding onto the phone with a single hand when targets were beside them. Other smaller issues came about when targets where generally close in proximity overall. In our interaction space and system implementation, we refrained from using direct touch for nearby targets so we could focus on at-distance interaction, but investigating manipulation through direct touch would be the logical next step.

For our prototype environment, we utilized projection-based AR were multiple projectors were calibrated using the RoomAlive Toolkit [266]. The result of this calibration process can sometimes introduce artifacts that may reduce visual fidelity, such as projector misalignment. Some of these issues could be mitigated through better projector alignment techniques [221, 234] or laser projectors.

Two-Handed Interaction

We explicitly designed our technique for single-hand interaction, however there are two-handed mobile phone techniques, such as viewport pointing [20] and mid-air gestures [102], that have been used for similar types of object manipulation and selection. Previous work indicates raycasting from a phone held by a single hand has some advantages in a SAR environment compared to viewport pointing with two hands [87]. A head-to-head comparison between our one-handed method and two-handed techniques would be an interesting direction for future work.

4.7 SUMMARY

Pushing out and interacting with smartphone content in augmented reality is an increasingly relevant problem without any clear solutions so far. In this chapter, we proposed using the smartphone itself as the mediator of this interaction based on arm extension, a seamless and intuitive way for the phone to switch between the mobile interaction and spatial interaction modes, guiding the user's attention from the physical world to one augmented through SAR. Our interaction technique enables the user to push smartphone content to an external SAR environment, interact with the external content, rotate-scale-translate it, and pull the content back into the smartphone, all the while ensuring comfort, no conflict between the mobile and spatial interactions, and single-handed and eyes-free use in the spatial mode. To ensure feasibility of hand extension as mode switch, we evaluated the classification of extended and retracted states of the smartphone while varying user postures, surface distances, and target locations. Our results show that a random forest classifier can classify the extended and retracted states with 96% accuracy on average. A final usability study of the interaction space with three demonstrative applications found our interactions to be usable and intuitive.

5 BLENDING VIRTUAL ENVIRONMENTS WITH SITUATED PHYSICAL REALITY

Today's virtual reality (VR) systems such as the Oculus Rift, HTC Vive, and Windows Mixed Reality, aim to completely immerse the user in a virtual environment. However, such immersion comes at the cost of the user's awareness of their physical surroundings. Simple tasks, such as picking up small objects, moving within a physical space, or communicating with someone in the room become difficult if not impossible. Current systems render a 3D grid that appears whenever the user comes into close proximity to a predefined boundary. This is seen in both the Oculus Rift's "guardian" and the HTC Vive's "chaperone" systems. Equipped with a color camera, the HTC Vive offers a variant of the grid chaperone where an outline of the real world is rendered (composited) on top of the virtual environment (see Figure 5.1). Though these approaches help prevent unintended collisions, they employ a very simple 3D model of the room which is assumed to be static, with the floor clear of obstructions such as furniture, people and pets. We speculate that many VR users would often choose some other form of entertainment rather than clear their living room of obstacles.

In this chapter, we introduce a system that can exploit a realtime 3D reconstruction of the user's environment to enable the combination of real and virtual worlds that go beyond current state-of-the-art chaperone systems. With our system, the real world is embedded inside the virtual world as if the application rendered the physical world natively within the scene (Figure 5.2). The live reconstruction of the physical environment is obtained through two different approaches. In the first, we equip the physical environment with eight RGB-D cameras (Microsoft Kinect v2) that are positioned inside a physical environment to reconstruct a geometric representation of the world in real-time. In the second, we equip an RGB-D camera (Intel RealSense) directly onto the HMD and reconstruct a live view of the physical environment from the user's perspective.

Our system modifies the graphics rendering pipeline of existing VR titles that rely on OpenVR. Once references to the back buffer and z-buffer are acquired, multiple means of blending the real with the virtual are possible. For example, the player's couch may appear around them, correctly occluding virtual objects, allowing the player to safely take a seat during gameplay. Meanwhile, a non-player character (NPC) may correctly appear in front of the player's ottoman as it approaches. Since the rendering of the physical environment is dynamically updated, people or objects placed inside the environment will also appear inside the virtual scene. This allows for ad hoc manipulation of objects and for communicating with someone else in the room without removing the headset.

We make the following contributions:

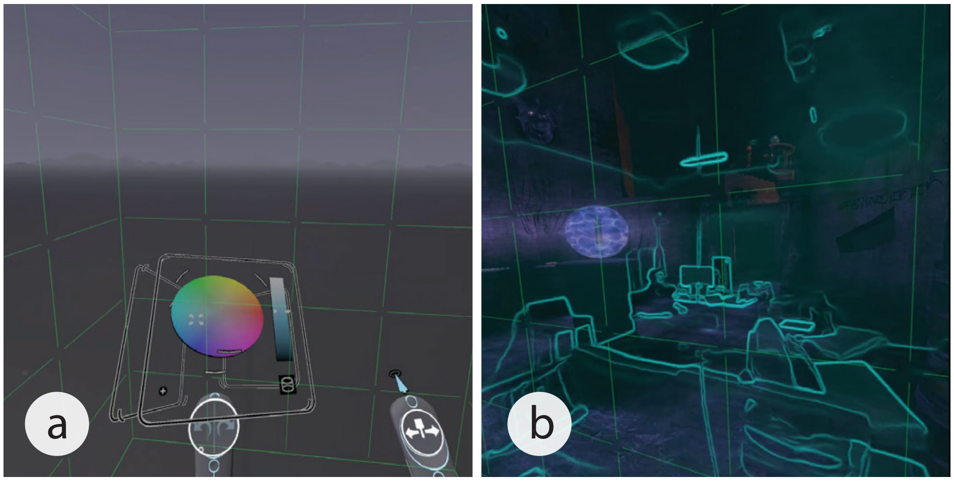


Figure 5.1: Current chaperone system implemented by HTC Vive: a) grid chaperone (*Tilt Brush*), and b) line overlay chaperone (*Waltz of the Wizard*).

- A new approach to blending the real world with the virtual world in VR, in which a 3D reconstruction of the real world is composited with the virtual world in the usual graphics rendering pipeline
- A prototype implementation that works with existing VR titles, without modification
- Several variations of the basic compositing technique that explore the interaction between real and virtual world geometry
- Demonstration of the approach with multiple hardware configurations: external and internal cameras, and external display (projection mapping) for spectators
- A user study that compares our system to state of the art chaperone techniques

5.1 RELATED WORK

One of the core ideas presented in this chapter lies in the merging of the real and the virtual. Milgram and Kishino describe various versions of this blending in their virtuality continuum [168]. Many extensions to this have been proposed, usually through the addition of axes orthogonal to the AR-VR continuum that provide new insights into aspects of human computer interaction [112, 154, 167]. A variation on this is Virtualized Reality [117], which involves creating a virtual copy of the real world. Our work pushes this further and explores the application of blending in the context of situated physical reality. Here we discuss work relating to the blending of VR environments in the context of interaction, communication, and object avoidance.

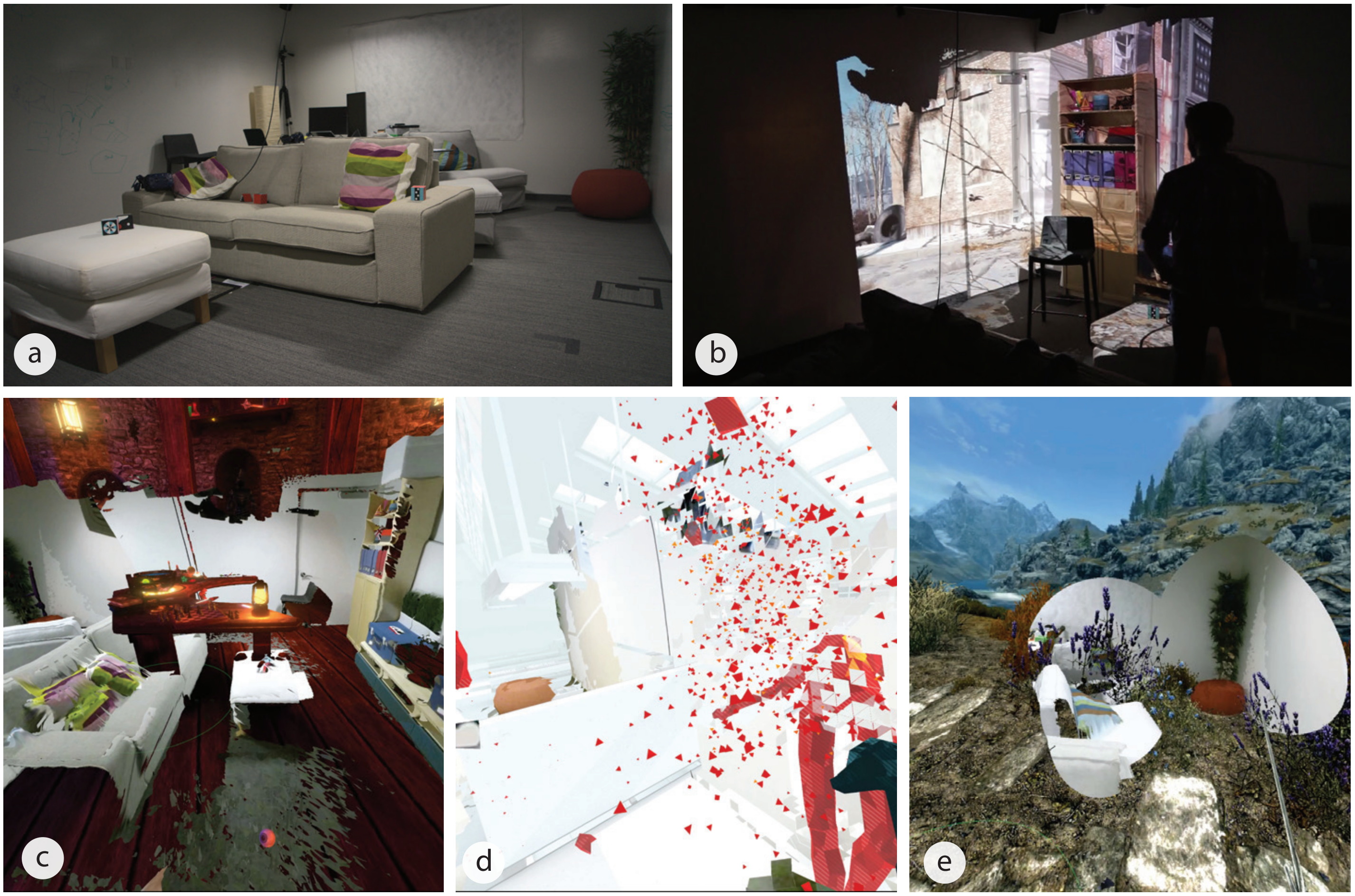


Figure 5.2: Our system blends the VR player's real world (a) with the virtual world. A real time 3D reconstruction of the player's room is integrated into the VR title's rendering pipeline to allow: b) co-located spectatorship through projection mapping (*Fallout 4 VR*), c) proper 3D hidden surface removal (*Waltz of the Wizard*), c) collision estimation (*SUPERHOT VR*), and d) a flashlight into reality using controllers as input (*Skyrim VR*).

In Chapter 2 we looked at a breadth of work relating to ways we can blend different realities together for novel effect. Here we look more precisely at specific work related to our research questions.

Blending the physical with the virtual

Blending virtual content with aspects of reality can come in many different forms. If a virtual item is blended inside a physical environment, we get augmented reality (AR). If a physical object is blended inside of a virtual environment, we get augmented virtuality (AV). These kinds of blending are common in mobile AR, and when combined with a real-time depth map, are effective at merging virtual and real content together [57].

In a virtual reality context, the representation of the physical world being blended can have an effect on how the blending takes place. For example, a simplified semantic representation of physical reality can be used as a blueprint to generate an approximated virtual environment that encapsulates a specific theme or aesthetic [229, 239]. Further, a reconstructed copy of physical reality can be used as a virtual proxy to the real thing. Allowing the user to bend reality into states that deviate from what is physically possible [143, 172].

Closely related to the themes in this chapter is the work by McGill et al. [161]. They identified several usability challenges when VR users interact with objects in the real world. The tasks they explored include having the VR user type on a physical keyboard, interact with small objects around them, and communicate with other people around them. They contribute a prototype that uses green screen compositing and simple background subtraction to blend a 2D video feed inside a virtual environment. They demonstrate multiple blending techniques, including the area around the user's hands and complete sections of a user's desk. In another study, they use a depth camera to segment video of people in the room, which is then composited into VR.

In contrast to previous work, we specifically investigate the system and techniques for taking a complete reconstruction of physical reality and merging that with a virtual environment for communication, awareness, and navigation.

5.2 BLENDING VIRTUAL REALITY WITH REALITY

VR systems are able to render highly detailed environments that allow users to explore vast worlds. While VR technology will continue to improve, providing ever more immersive experiences, problems relating to the isolation of the user from the real world will remain. We investigate these issues along three dimensions: 1) mitigating risk through increased awareness, 2) communication and spectatorship, and 3) physical manipulation.

Mitigating the Risk of Collision

Our system enables an enhanced chaperone system by integrating parts of the user's real world into the their virtual world in various ways (Section 5.4). When the blending is based on the distance to the user's head and controllers, a minimal chaperone is possible that only shows the physical environment when the user is at risk of collision with either a static or dynamic object. Current VR environments also use a chaperone system to help guide the user (Figure 5.1). However, current solutions require a sufficiently large play area (e.g. minimum area for the HTC Vive room-scale experience is 1.5m × 2m) where the perimeter of the space must be manually defined. While a grid is shown when the user is within a certain distance, any obstacles intruding or within its static and predefined border are not considered. In contrast, our chaperone is dynamic and does not need to be manually defined as the real world is used directly.

Outside Communication

A VR user is physically present but removed from their immediate context, making interpersonal engagement difficult to initiate. Our system provides a mechanism to merge dynamic objects into the virtual context of the user to help eliminate these kind of communication barriers. We use a method of background-subtraction to render salient objects such as a person crossing the room (section 5.4). Recent works have explored the use of vibro-tactile sensors to alert the VR user of a nearby presence [150], or use 2D blending techniques to bring a stenciled video of a person to the foreground [161]. In contrast, we explore communication under two contexts. First, from the perspective of the VR user, and second, from the perspective of a co-located person. Further, our approach has the advantage of rendering real people in the room as if they were part of the virtual environment, while not requiring the VR application developer to add special support.

Physical Manipulation

We tend to interact with the physical items around us. Research into the manipulation of real objects while in VR has largely come in the forms of haptics [4, 159], substitution [94, 231], or input [126, 161]. Supporting everyday physical interaction around the user has not been fully explored, for example the VR user could simultaneous grab a plate of snacks, a drink, move from their desk, and sit down on a nearby couch. Our system provides mechanisms for these kinds of interactions to take place. By bringing in part of the real world at appropriate moments, the user's virtual environment can become easier to use and be more enjoyable.

Overall, our system looks to address issues of safety, communication, and the physical manipulation of objects by merging the real world inside a virtual environment. This allows for proper 3D hidden surface removal,

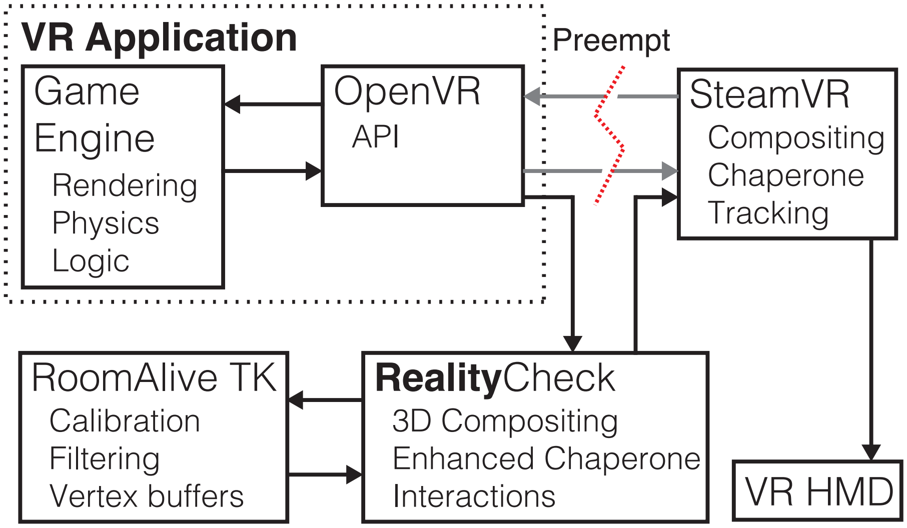


Figure 5.3: Overview of the 3D compositor system.

blending, interactions, and integration within a virtual environment. We demonstrate compositing techniques that are agnostic to the underlying game implementation, in fact, no development requirements or API integrations are required to use our system; and showcase their use in the transition between real and virtual worlds.

5.3 COMPOSITING REAL AND VIRTUAL WORLDS

In describing our system, we take as a starting point the problem of compositing a 3D model of the user's real physical environment into the rendered 3D graphics of a VR application. Such a 3D model may be collected from an array of depth cameras arranged around the user's space ("outside in"), or from a depth camera mounted on the user's head mounted display ("inside out"). In either case, we presume that the 3D model of the user's real physical environment has been calibrated to align with the native coordinate system of the VR system: in practice, this can be accomplished by a procedure in which the tracked VR controllers can be located in the reconstructed model of the room.

Given the 3D nature of both the virtual and real worlds, it is natural that a real object should appear in front of any virtual objects that are further away, and vice versa. In a traditional 3D graphics pipeline, occlusion of one object by another is accomplished during rendering by updating the *z-buffer*, a texture which records at each pixel location the distance of the nearest surface rendered thus far. In the rendering process, a given pixel's color is updated only if the currently rendered geometry falls at a point nearer than the value recorded in the z-buffer.

To perform our own rendering of real world geometry in a running VR title, we require access to its final rendered output for each eye, as well as the z-buffer, view and projection matrix used in the rendering process. Additionally, some modifications may make use of other information such as the poses of each VR controller.

Furthermore, we would like to demonstrate the broad applicability of our approach by using it with games that are popular with VR users today. To this end we developed a software framework (Figure 5.3) that allows the modification of the rendering process of an existing VR application without requiring the application's source code. It uses well-known techniques to replace or "hijack" calls to system APIs involved in rendering and VR compositing [173]. Such techniques are popular with the game hacking and modding community, and have also been a useful tool in building research prototypes on otherwise unmodified software systems [100, 215]. We later expand on these techniques in Chapter 7 for generalized specatorship of videogames in VR.

Exploiting OpenVR

OpenVR[1] is an API that VR applications use to communicate with SteamVR[2]. An application obtains HMD and controller pose from SteamVR via OpenVR calls, and provides SteamVR with final rendered frames for each eye by calling `IVRCompositor::Submit`. To add our own graphics to the VR's final rendered graphics, we replace OpenVR's dynamic link library (DLL) with a custom DLL which similarly implements OpenVR's `Submit` call but includes additional routines to render our real world geometry. This is done by modifying and recompiling OpenVR's open source.

The calls to OpenVR's `Submit` provide a convenient means of injecting our own code to render onto each eye's final output, but it does not provide access to the z-buffer. In a typical frame, the application will call `ID3D11DeviceContext::OMSetRenderTargets` many times throughout its rendering process. Some of these calls will be to set the color back-buffer and z-buffer that we require for our own rendering. To find the application's z-buffer, we first obtain the application's DirectX device by examining a texture passed to `Submit`, and then modify its C++ vtable to intercept all calls to `OMSetRenderTargets`.

Unfortunately there is no direct means to determine which of the render targets provided to `OMSetRenderTargets` is the final rendering and z-buffer we require. Furthermore, the application may use render targets in different ways: 1) each eye may be rendered to individual textures, 2) each eye may be rendered separately but to the same texture which is reused, and 3) both eyes may be rendered into the same texture, side by side (Figure 5.4). When each eye is rendered separately, there is similarly no direct way to determine whether a given render target corresponds to the left or right eye. When both

[1] https://github.com/ValveSoftware/openvr

[2] https://steamcommunity.com/steamvr

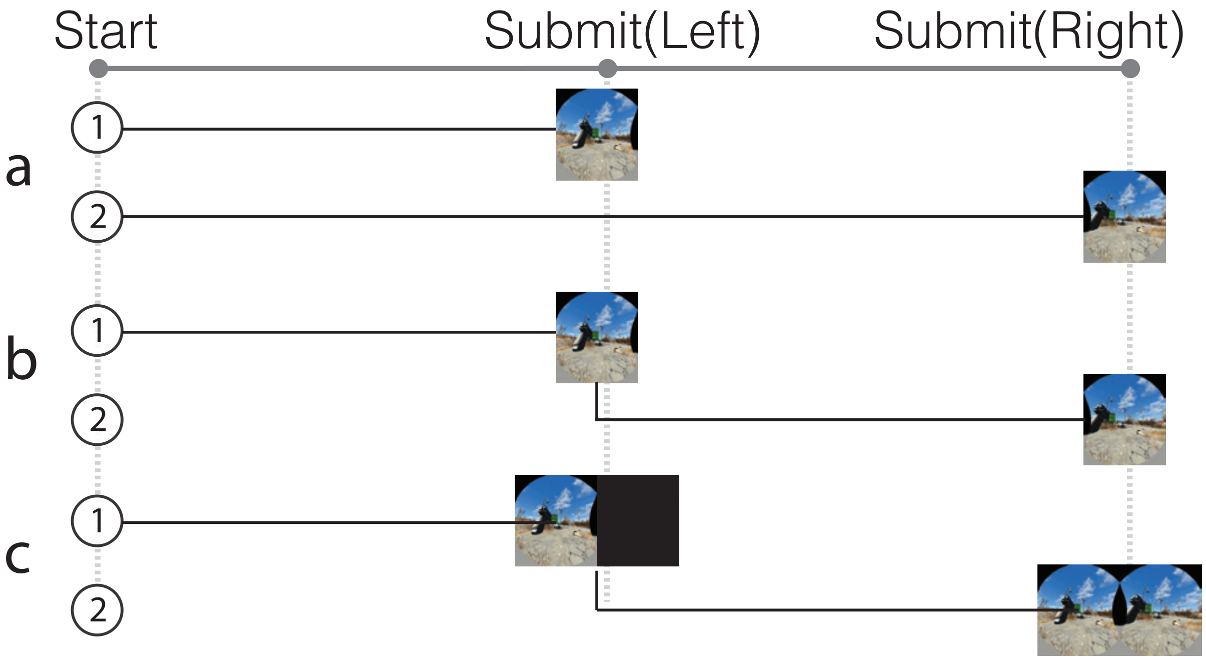


Figure 5.4: Three likely rendering pipelines used by VR applications. a) single eye textures, b) single texture shared between each eye, and c) stereo texture.

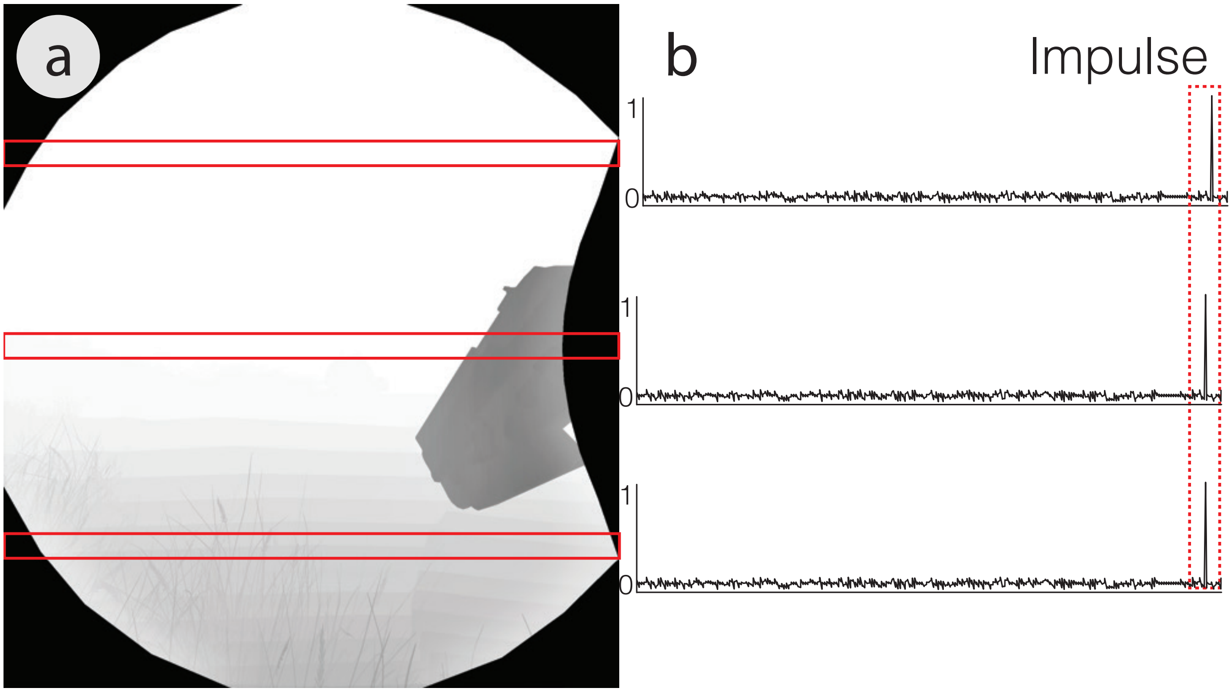


Figure 5.5: A compute shader analyzes the render target received through `OMSetRenderTargets`. a) the raster lines tested in the analysis, b) the impulse used in eye identification.

eyes are rendered into the same texture, there is no direct way to determine if both eyes have been rendered.

We employ a variety of heuristics and image processing techniques to resolve these ambiguities and find our render targets. For example, a compute shader is used to classify candidate z-buffers as left or right eye views based on the stencil's pattern (Figure 5.5). When both eyes are rendered into the same texture, symmetry of the image is a reliable indication of whether both eyes have been rendered. Once determined, references to the correct render targets are cached, and no further analysis is performed. In the case where the application renders eyes separately but reuses the render target, copies are made during rendering.

To perform our rendering, we also require the view and projection matrices used by the VR application in its own rendering of each eye. The view matrix is easily obtained from OpenVR, which provides the position and rotation of the user's HMD. This must be updated every frame to match the user's head pose in game. Obtaining the projection matrix is more difficult. OpenVR provides the projection matrix for each eye via a function call that takes the near and far plane values used by the application. Presently, we determine the application's near and far plane values empirically and retrieve each eye's projection matrix through OpenVR. With the view and projection matrices, we render the real world inside the virtual scene and make a submission on behalf of the game to SteamVR. This allows us to take advantage of advanced post-rendering techniques such as asynchronous reprojection and motion smoothing.

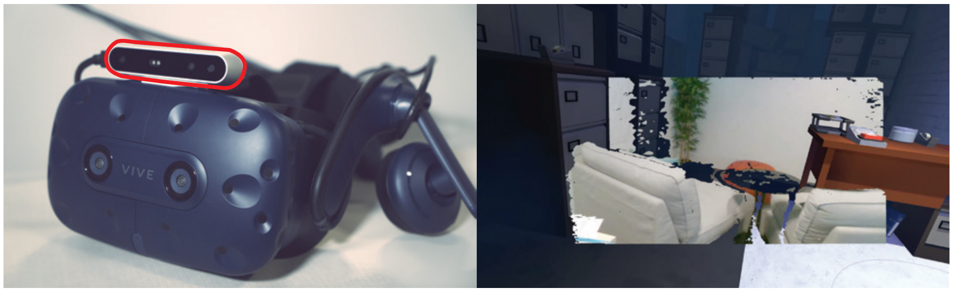


Figure 5.6: HTC Vive with Intel RealSense highlighted in red (left). The composited depth image with a virtual scene (right).

Data Acquisition and Calibration

The system is agnostic to what can be rendered on a submitted frame. In our implementation, we use the RoomAlive Toolkit [265] for both the room scale reconstruction and for the RealSense[3] head mounted camera (see Figure 5.6).

Our room scale deployment uses eight Kinect v2 depth cameras calibrated using the RoomAlive [265] calibration procedure, where five projectors display Gray codes onto the physical surfaces to resolve the poses and positions of

[3] https://realsense.intel.com/

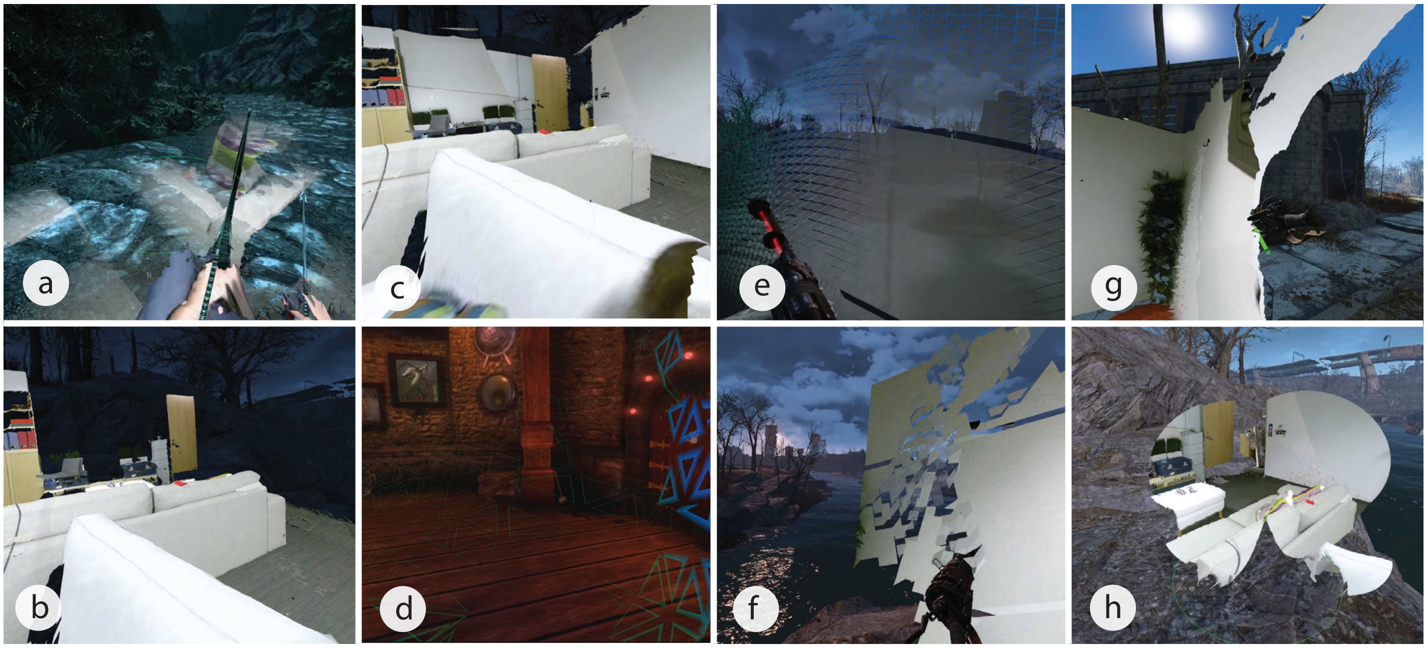


Figure 5.7: Techniques implemented with our system. a) alpha blending, b) salient objects (no Walls), c) full environment, d) texture abstraction (mesh outlines), e) polygon manipulation (floating polygons), f) collision estimation, g) mesh erasing, h) flashlight into reality.

all projectors and cameras. The depth and color data is compressed using RVL [266] and JPEG compression respectively and sent over a local Ethernet connection to the client. The RoomAlive RealSense server works similarly.

The compressed data is received by the RoomAlive client, in which the depth and colour data is decompressed and smoothed with a bilateral filter. The depth image is then converted into a DirectX vertex buffer for rendering.

5.4 COMPOSITING TECHNIQUES

We use the system described in Section 5.3 to create a variety of game-agnostic compositing techniques that demonstrate the flexibility of our approach across the categories of blending, texture and geometry manipulation, and interaction. We test these techniques within seven different VR titles available on Steam: Accounting, Waltz of the Wizard, SUPERHOT VR, Tilt Brush by Google, Blocks, Fallout 4 VR, and Skyrim VR.

Blending

We explore blending in the context of the full environment, only salient objects, and objects that are nearby.

FULL ENVIRONMENT BLENDING All available real world geometry is composited into the virtual environment (Figure 5.7c). While in a room,

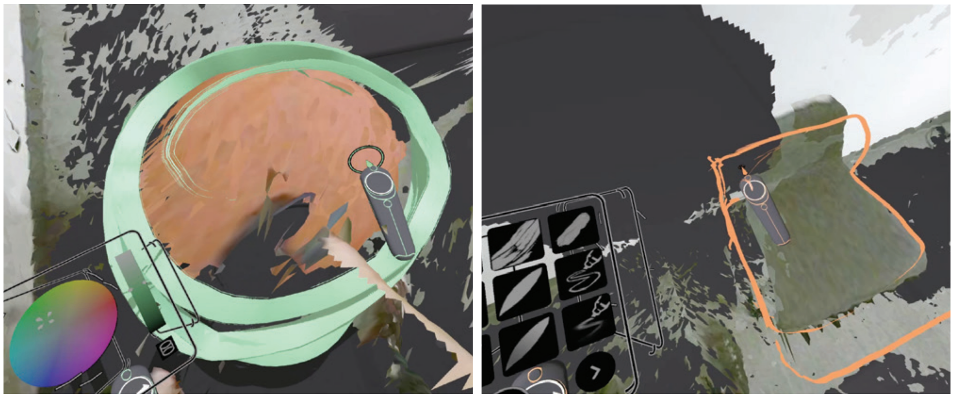


Figure 5.8: Real world guides: a) an ottoman being used to create a circle, and b) a chair being used as a reference (*Tilt Brush*).

walls and furniture will occlude everything outside the physical environment. Applications that are more productivity focused may benefit from this level of blending, such as *Tilt Brush* and *Blocks*. Users are able to navigate the real world while simultaneously allowed to design, draw, or build within it. This offers some interesting affordances, such as using a physical object as a reference while modeling a 3D object (Figure 5.8).

SALIENT OBJECT BLENDING These depict only the objects that are important to a room's composition, such as furniture, people, or pets (Figure 5.7b). This level of blending allows a user to freely move around the space with only a minimal amount of hidden surface removal on the virtual scene, and can be implemented through the removal of predefined objects (i.e. walls) or by showing only objects that have changed from one frame to the next (i.e. background subtraction). Small tasks, such as manipulating objects on a desk, eating food, or drinking can easily be performed.

PROXIMITY BLENDING Objects are selectively blended based on proximity to the user's head and hand positions (Figure 5.7a). This level of blending has minimal impact on the visual coherence of the virtual scene, since only portions of the physical world appears within a certain distance (1m). Proximity blending can act like an advanced chaperone system with the added benefit of also showing dynamic objects (e.g,. people, pets, chairs, etc.).

Texture

Changes to the underlying physical texture can be used to make the rendering of the real world more similar or different than the virtual world (Figure 5.7d).

COLOUR TRANSFER When compositing real world geometry into a virtual world, it may be desirable to light the real world to match that of the virtual world (see Figure 5.9). For example, if the player enters a dark cave, the

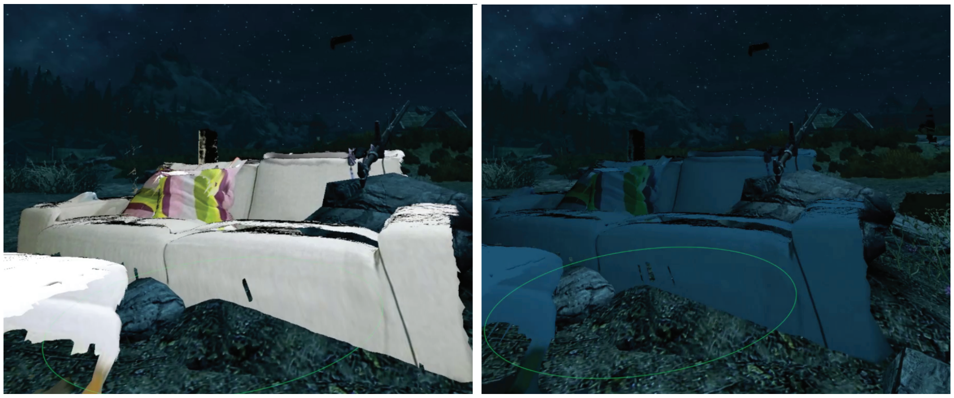


Figure 5.9: The real world environment before color transfer (left) and after the transfer (right) in the game *Skyrim VR*.

rendering of their couch should be similarly dark. We use an approach that is both fast and effective at modulating the colour of real world geometry to match that of the virtual rendering. We adapt the statistical methods by Reinhard et al. [212] using parallel reduction techniques on a compute shader.

Colour statistics are calculated in the CIEL*A*B* colour space. The method uses the global illumination (μ) and standard deviation (σ) from a source image (I_s) to transform a target image (I_t) to match the distributions found in each L*A*B* colour channel. Every pixel in I_t is scaled by a ratio between the standard deviation of the target (σ_t) and the standard deviation of the source (σ_s), giving:

$$I_t' = \frac{\sigma_t}{\sigma_s}(I_t - \mu_t) + \mu_s \tag{5.1}$$

This transformation is easily implemented on a compute shader.

ABSTRACTION Rendering the real world with a very different rendering style can make it clear to the user which parts of the world are real and which are virtual. Rendering the real world as a wireframe or with other stylistic effects (Figure 5.7d) can also allow the user see through real world objects, which may be important for gameplay.

Geometry

Manipulations of real world geometry may be useful when creating effects where the physical world appears to react to the virtual. Further, the abstraction of geometry can be used to incorporate artistic renderings of objects that approximate the original [94, 231].

We demonstrate a geometric effect based on the proximity of the user (Figure 5.7e). As the user moves around the space, the physical environment is reconstructed around them in real time. The user sees individual polygons float down and assemble at their feet, playing with their senses of reality.

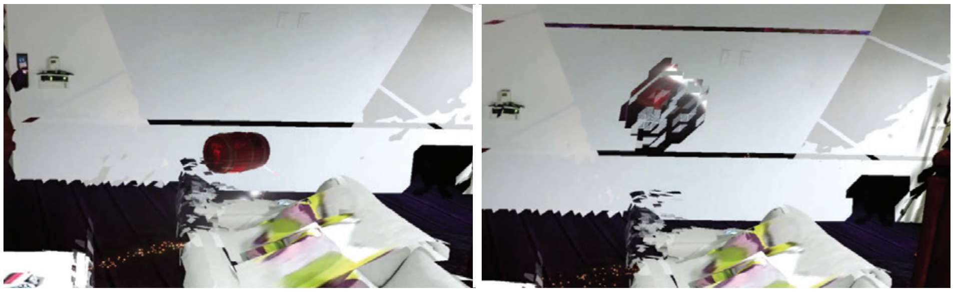


Figure 5.10: A before (left) and after (right) shot of a barrel colliding with a composited wall (*Waltz of the Wizard*).

IN-GAME COLLISION ESTIMATION Interaction between the physical world and in-application content is also possible. We use the system implementation in Section 5.4 to detect intersections between the real world and the virtual scene (Figure 5.7f). For example, when the user shoots an arrow or throws a barrel, the real world will react by breaking apart around the point of impact (Figure 5.10). These kinds of interactions make the real world seem alive and part of the virtual world.

We estimate collisions between the application and the composited live mesh data by comparing the real world depth of a rendered pixel against the corresponding z-buffer point. Z-buffer values are normalized by the application's projection matrix and may be converted back to world coordinate depths by inverting part of the projection matrix:

$$z' = \frac{nf}{f - zf + zn} \tag{5.2}$$

where n and f are near and far plane values, respectively. With the z-buffer projected into the system's coordinate space, collision with the application's geometry is approximated by computing the distance between the depth buffer's point to the corresponding point in the live mesh data. The rendering process may use this distance to appropriately modify the rendering of the real world geometry. For example, it may move the real world geometry out of the way.

Interactions

Giving the user full control over aspects of blending, object saliency, or recoloring may be useful when insufficient information is available to infer correct parameters or when greater control over compositing is desired.

Aspects of the real world can be dynamically shown and hidden to the user as they are inside their virtual world, but there are times when giving control over what objects persist and what do not is important. For example, a user may always want to know where their computer desk is within the virtual world. Our system realizes this by allowing users to bring in or remove the real world by drawing over them with their controllers, giving them some granularity of control over what is seen and what is not (Figure 5.7g)

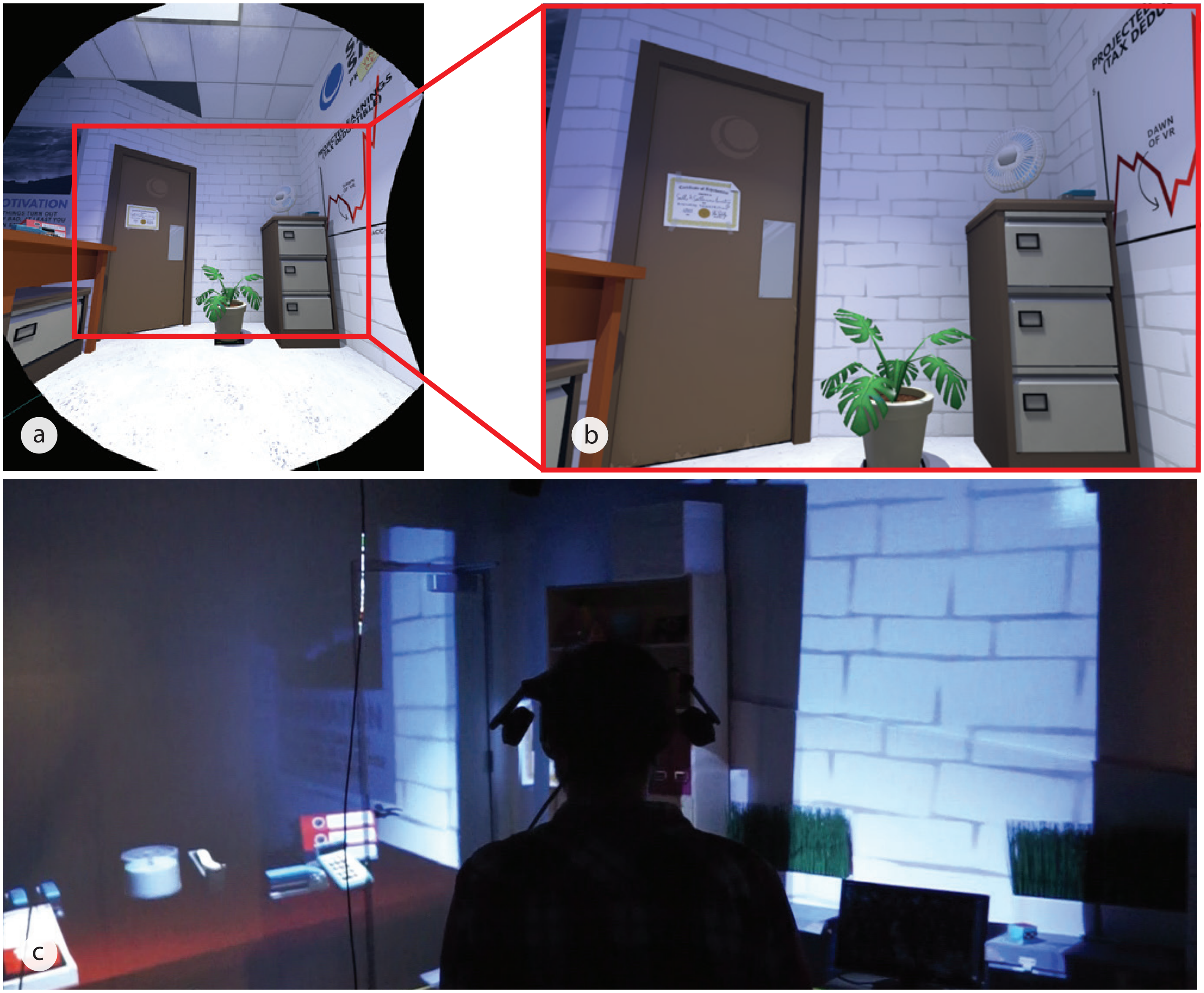


Figure 5.11: The HMD user's in-game view projection mapped onto physical surfaces from a co-located spectator's point of view. a) the left eye view, b) the companion window view, and c) the projection mapped content.

A flashlight into reality (Figure 5.7h) can be used as portal into the real world. A user moves the controller, like a flashlight, to reveal sections of the real world. All geometry in the real world that is contained within the solid angle emitted from the controller is revealed to the user.

Spectatorship

As part of our system, we implemented a projection mapping system where a view from the game is projected onto real physical surfaces from the perspective of a co-located viewer (Figure 5.11). For immersion, it is important that the mapping from the game to the physical surface is spatially stable, where a point in the game corresponds to a single point on the surface during rotation. This is accomplished by obtaining the rotation matrix from the HMD, the head position of the co-located user, and an adjusted field-of-view (FOV) for the projected content.

The HMD's orientation is retrieved from OpenVR. The head position of the co-located user is determined by a mean shift method [66] on the aggregated skeletal data from the eight Kinect sensors in the room.

During a VR session the game displays a companion window on the user's desktop (Figure 5.11b). The window is a subregion of the left-eye texture sent to the HMD (Figure 5.11a). Since the companion window is free of stencil marks, this is used as the projection mapped content. The FoV for the projection mapped content is then a ratio between the width of the companion window and the width of the left eye texture, ensuring that points in the game correspond to points in the real world.

5.5 USER EVALUATION

This within-subjects experiment evaluates immersion, safety, physical manipulation, and communication between our proposed 3D compositing techniques and the Vive's built in chaperone system. We chose three techniques outlined in Section 5.4 that are representative points along a continuum of blending (i.e. little to all of reality): 1) full reality (FULL) where all of physical reality is blended, 2) salient objects (SALIENT) where the furniture and dynamic objects are retained, and 3) proximity blending (PROXIMITY) where only a portion of reality around the user is retained.

Each blending level is assigned a virtual environment, physical space, and task in order to replicate realistic VR scenarios. The baseline for comparison is Vive's chaperone grid (GRID) and line overlay (LINE) where the outline of objects are rendered on top of the scene. Ordering of the tasks were counterbalanced with a Latin square. In summary: three blending levels (PROXIMITY, SALIENT, and FULL) and two baselines (GRID and LINES).

We recruited 12 participants, ages 28 to 40, 2 female. All participants were right-handed, and most of them use a VR devices a few times a year. The study lasted for approximately 60 minutes: 15 minutes per blending level and an additional 15 minutes of surveys.

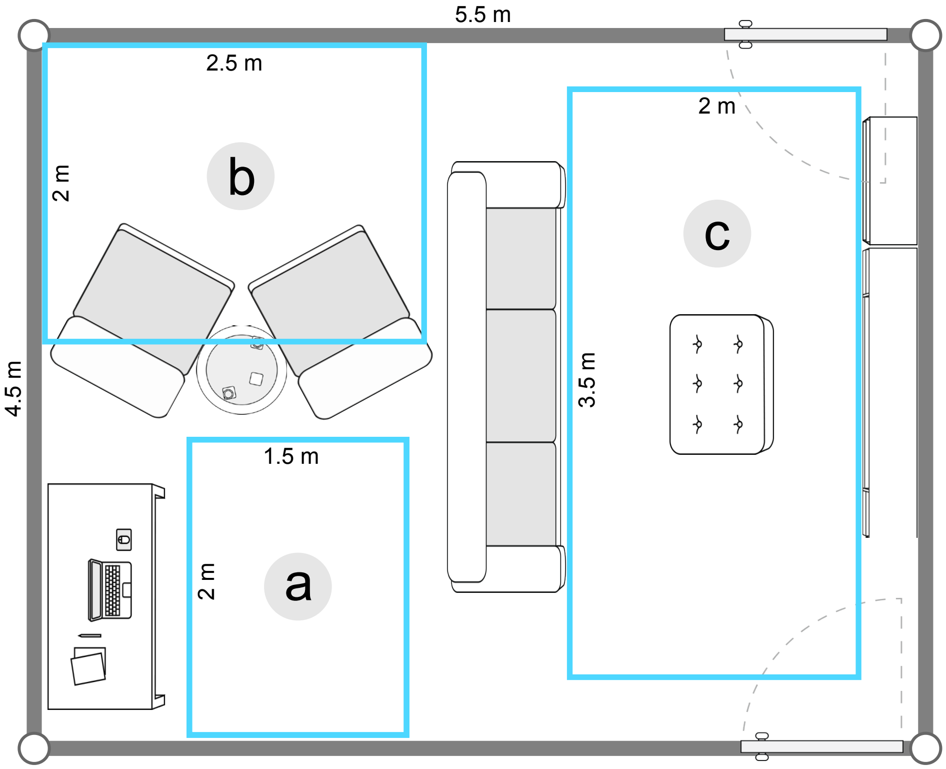


Figure 5.12: Room layout used in user study. a) office space, b) recreational area, and c) living room area.

The RealityCheck system described in Section 5.3 is used in a room environment that is approximately 4.5m × 5.5m × 2.5m (Figure 5.12). The room is split into three sections to replicate common VR configurations. Dimensions were determined by examining Steam's annual hardware surveys. The first room-scale environment (Figure 5.12a) replicates a home office containing an area of 1.5m × 2m. The factors FULL and GRID are compared. The second replicates a medium recreational area (Figure 5.12b) containing two sofa chairs and a small table, room-scale dimensions are 2.5m × 2m. The factors PROXIMITY and GRID are compared. The third area replicates a large living space (Figure 5.12c), with a full couch, foot stool, cabinets, and TV stand. Room-scale dimensions are 3.5m × 2m. The factors SALIENT and LINES are compared.

Task and Procedure

We asked participants to play a VR application within each of the three mock environments outlined in previous section. We chose a set of VR videogames based on their popularity and their required mobility during gameplay. In the mock office environment, the task was to use *Tilt Brush*, a 3D drawing application, to replicate a stool that was located just outside the chaperoned-off area. Participants were free to move within the allotted area but not outside it. In the mock recreation room, the task was to play the game *Skyrim VR* by Bethesda Studios and travel from Riverwood to Whiterun, fictional towns within the game. They were asked to switch between a standing and sitting posture every minute. In the mock living space, the task was to explore the world of *Waltz of the Wizard*, an exploration game set inside a fictional wizard's home. At one minute intervals, the participant was asked to pick up a block on the couch and place it on the ottoman in the center of the chaperoned area.

At the beginning of the experiment, participants filled out a short survey regarding their current VR usage. At the end of each VR session, a short 7-point preference questionnaire (7 = preferred) asked participants to reflect on their experience and the assigned tasks in the categories of safety, physical manipulations, communication, and their transition between the real and virtual worlds. A six part SUS [258] questionnaire followed. Finally, a post experiment survey was conducted asking about their overall experience. A researcher was in the room with the participant for the duration of the study and all instructions were verbally communicated.

Results

Aligned Rank Transform (ART) [272] and post-hoc t-tests with FDR [10] corrections are used for all non-parametric data. An overview of the results are outlined in Figure 5.13. On average, our system saw an increase in average scores across all game titles in each of the categories. A significant effect on Transitions ($F_{5,55} = 4.83$, $p < .001$), Physical Manipulation ($F_{5,55} = 4.98$, $p < .001$), and Safety ($F_{5,55} = 5.11$, $p < .001$) are observed. There is no effect on Communication.

On Saftey, a post-hoc test shows FULL (5.90) is perceived safer then GRID (3.81) for *Tilt Brush*. On the transitions between the physical and virtual world, FULL for *Tilt Brush* (5.54) is perceived easier then GRID (3.54) for *Skyrim VR*. Finally, on the physical manipulation of real world objects, FULL for *Tilt Brush* is perceived easier then GRID for *Skyrim VR*. No other significance is reported ($p > .082$).

The mean SUS scores are higher for our system across all games when compared with the baseline chaperones. An ART analysis shows a significant effect ($F_{5,380} = 6.62$, $p < .001$). Post-hoc t-tests suggest PROXIMITY (4.76) for *Skyrim VR* is perceived as more immersive than GRID (3.47) for *Tilt Brush*. The immersiveness of SALIENT (4.93) for *Waltz the Wizard* is perceived greater than both LINES (4.93) for *Waltz of the Wizard* and GRID (3.47) for *Tilt Brush*. There is no significant result for FULL (4.27) on immersion.

Discussion

We found compelling differences between the three different blending levels and the baseline chaperone. Among all three game titles, Tilt Brush has the

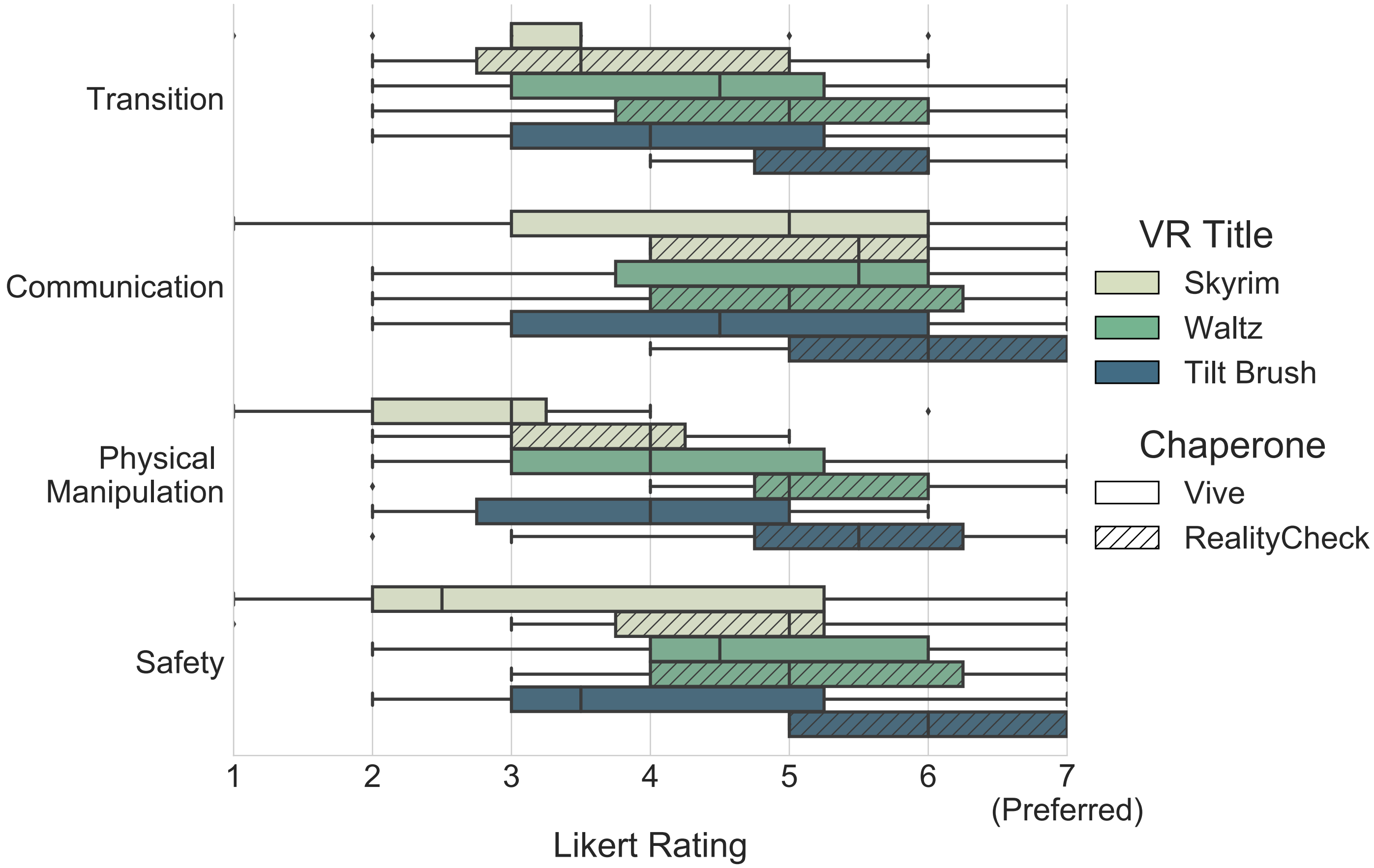


Figure 5.13: Preference ratings across all participants. A comparison between the Vive's chaperone system (no hatch) and the RealityCheck system (hatch) across three VR titles.

largest reported difference across the four questions posed to the user. This is likely due to the contrast between the environment in game and the physical environment merged with it. Surprisingly, there is no significant results for communication, though the mean scores are higher then the baseline.

Participant responses to the the grid and line overlay chaperone showed a generally negative sentiment, stating that grid chaperone "breaks the sense of virtual reality" (P11) and "makes the world less immersive" (P4). On the line overlay participants stated that it "does not work very well for me" (P3) and it "made depth perception slightly more confusing" (P11). In contrast to this, P6 and P8 stated that they preferred the line overlay when compared with the blending techniques. However, none of the participants preferred the grid chaperone and stated that it felt like a "virtual cave" (P5).

On the use of real world blending, participants generally thought is was useful. Stating that it "felt very immersive" and "[s]itting down was definitely more comfortable" (P5). Surprisingly, many participants expressed that the blending techniques seemed more immersive to them over the baseline chaperones, stating it "felt more like a virtual reality" (P12). Though the blending was sometimes seen as problematic when participants were not able to differentiate between virtual objects and real ones. One participant stated, "there was a bookshelf from a real world, which I thought was a VR bookshelf".

5.6 LIMITATIONS

While our system provides new ways to render content within a virtual world, there are still several challenges related to performance, visual fidelity, and OpenVR integration.

Today's VR systems typically aim to render each eye at 90Hz. The 3D compositing and spectatorship components of our system use the GPU extensively, making this framerate goal more difficult to achieve. Ultimate framerate will be a function of the rendering demands of the VR title, the complexity of the our system shaders, the number of cameras employed in the system, and, when spectatorship is enabled, the number of projectors. For example, without performing visibility based culling, each projector must render all cameras.

The SteamVR title Accounting, for example, easily renders at 90Hz (less than 11.1ms rendering time) with one camera on modern hardware. Our most complex configuration of eight Kinect v2 cameras and five projectors with spectatorship enabled, renders at 15ms, which causes SteamVR to render at 45Hz on Skyrim VR. SteamVR then doubles this framerate with its "motion smoothing" feature, whereby every other frame is synthesized by interpolation. The overall subjective experience is still good.

The visual fidelity of the system depends on the type of depth camera being used, as well as their calibration and alignment. Incorporating advanced filtering and reconstruction algorithms could help [54, 56, 244], however latency and frame rate need to be considered for a real-time system. The combination of highly accurate but sparse LIDAR sensors with highly dense depth cameras could be an alternative approach to increasing overall quality. Another approach would be to rely on depth cameras to perform coarse hidden surface removal, while otherwise relying on head mounted RGB cameras, such as those available on the HTC Vive, to render the color texture of the real world.

With the room scale deployment of cameras, latency of the acquisition of depth and color data has no discernible impact on the ultimate rendering when the scene is largely composed of stationary objects such as furniture. Meanwhile, the head-mounted version introduces a noticeable latency in rendering of the real world.

Our system sits on top of OpenVR, which allows us to modify the application's rendering. However, retrieving the application's near and far planes in a generalizable manner remains unsolved. Extending the OpenVR API to include the submission of the near and far planes could be one solution to alleviate this.

5.7 DISCUSSION AND FUTURE WORK

Our system aims to merge reality with a virtual environment for the purposes of safety, communication, and interaction. Our system works with existing VR applications and enables new ways to engage with the real world while inside the virtual world. Our system is a platform and concept that will enable interesting directions for future work.

HAPTICS AND GEOMETRIC MAPPINGS Our current implementation of our system focuses on manipulating the real world geometry to react to changes in the virtual world. Further enhancements may be possible by considering manipulating the virtual geometry as well. Methods that use raw mesh data [67] and semantic scene understanding [235] might be used to modify the virtual world to match the real in a meaningful way. Further, techniques such as change blindness or saccades [242] could be used to make these changes imperceptible to the user. With methods to align the real and virtual, passive haptics could be used to enhance the experience. For example, aligning a in-game wall to match a real physical wall or positioning a couch to align with its in-game counterpart.

Pushing this concept further, aspects of the real world could deviate from reality to match the style and tone of the game. Methods like in RealitySkin [229] could be used to create visually compelling scenes or create alternatives to the "Home" application currently used to launch applications.

TREATING THE GAME AS A DEPTH CAMERA Currently, we extract the depth and RGB data from the game for use in our compositing techniques. It may be instructive to think of this data as produced by RGB-D "camera". Computer vision research suggests many uses of depth cameras, such as in SLAM [180] or using it for in-game object detection and segmentation [211].

These methods could be used for extended types of interaction. For example, by reconstructing the game world through SLAM techniques or by extracting in-game avatars to superimpose onto a collocated person in the room.

Recommendation

Our system relies on the abilty to obtain the game's color back buffer, z-buffer and view and projection matrices through recompiling the OpenVR DLL and vtable injection on DirectX. Future versions of such APIs could make these components more readily available, encouraging researchers and developers to create novel experiences that expand the state of the art in VR.

5.8 SUMMARY

In this chapter we presented a system that builds on top of the current VR rendering pipeline. We demonstrate the capabilities of our system through a number of techniques that integrate the real world inside a virtual scene. A user study further demonstrated its ability to enhance current VR environments. We believe this approach enables applications in safety and awareness as well as creating more meaningful VR experiences. We see this work as the first step towards allowing the seamless transition between two realities.

6

AUGMENTING A WEARABLE AUGMENTED REALITY DISPLAY WITH AN ACTUATED HEAD-MOUNTED PROJECTOR

Augmented reality (AR) has the potential to truly merge digital and physical worlds. Typically, an optical see-through head-mounted display (HMD) is used to composite virtual content into the surrounding environment [130]. While effective in many ways, it also has a limited field of view and suffers from vergence-accommodation conflicts. Further, the user experience is isolating, since the virtual environment is only visible to the HMD user. This makes collaboration and communication with external users difficult.

An alternative to creating AR with an HMD is Spatial Augmented Reality (SAR) [18], which uses projected light to directly augment physical surfaces. SAR can be used with an optical see-through HMD to alleviate some limitations, such as simulating an expanded field of view [11] and improving perceptual depth cues [18]. Another possible way to improve on AR HMD experiences is with cross-device systems that combine many conventional displays and devices with an AR HMD, like smartphones, smartwatches, and large displays. This has been used to enable external communication [228], expand the capabilities of devices [74], and enhance interaction with 3D virtual objects [170]. However, both approaches limit user mobility and do not allow for ad hoc serendipitous collaborations with external users. SAR typically requires multiple external projectors installed and carefully calibrated to a specific environment, and cross-device systems require specialized software and experiences are constrained by the physical properties of the device.

In this chapter, we introduce a concept called *Augmented Augmented Reality* (AAR), the combination of a see-through wearable AR display with an actuated head-mounted projector. AAR can be expressed in a concise design space, where potential user roles and projector roles intersect. Using this space, we explore how AR interfaces can be extended and combined, enabling new ways to view, manipulate, and share AR content. For example, the HMD user experience can be enhanced by using the projector to augment their view with peripheral information, such as simulating a heads-up GUI (Figure 6.1b). Or, the projector can share AR with external users, for instance, a view-dependent rendering on a nearby wall so a bystander can see into the HMD user's virtual world (Figure 6.1c). A combined HMD and projector can enable new communication opportunities, such as an impromptu presentation with projected slides for an external audience and private notes for the HMD user (Figure 6.1d). The system can even transition the HMD user from virtual to real worlds, and when they remove the HMD, the projector could persist a portion of the virtual content (Figure 6.2).

To operationalize the AAR concept, we built an actuated pico projector system mounted on a HoloLens AR HMD. An important aspect to the system

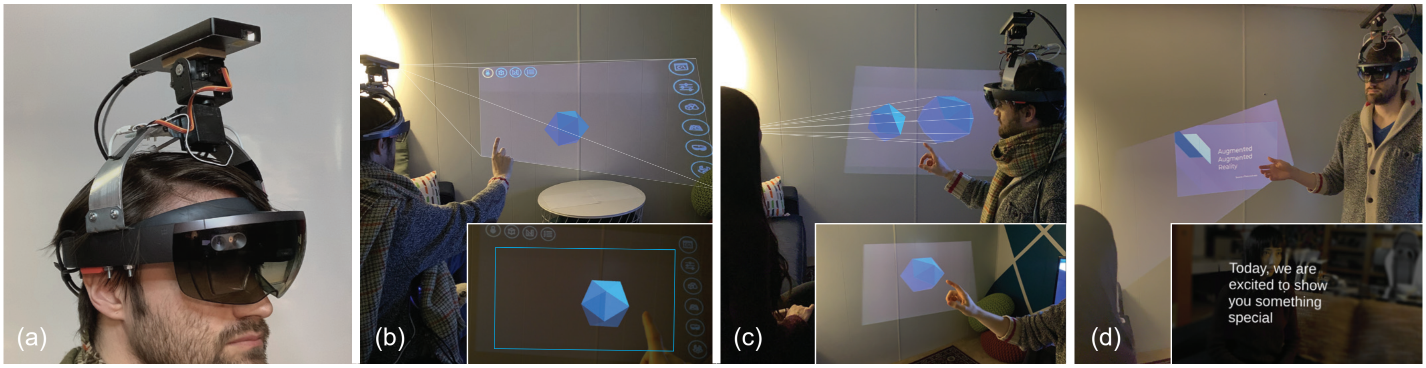

(a)
(b)
(c)
(d)
Augmented Augmented Reality
Today, we are excited to show you something special

Figure 6.1: System overview and use cases: (a) a pico projector on a servo-controlled pan-tilt head is mounted on a Microsoft Hololens V1 AR HMD and when calibrated, precise control of the projected image relative to the HMD or the realtime scan of world geometry is possible; (b) the projector can display interactive content beside the optical display, such as a toolbar; (c) the projector can let bystanders "peek" into the HMD user's virtual world; (d) the projector can show public content while keeping private information in the HMD, such as during a presentation.

is how the projector, pan-tilt geometry, and HMD are calibrated: for this, we developed a novel variation of the structure from motion (SfM) pipeline that utilizes a dense correspondence map between all camera and projector view combinations. Applications access the calibrated projector and HMD through a AAR software toolkit. This gives developers high level control over projector roles relative to the HMD, the user, and the physical environment, and defines content behaviour across the AR HMD and projected SAR.

In summary, we make the following contributions:

- Concept and applications for combining a wearable actuated projector with an AR HMD;
- An automated calibration process to build a geometric representation of a head-mounted projector pan-tilt structure;
- An open source toolkit to develop AAR applications.

6.1 RELATED WORK

In Chapter 2, we explored aspects of HMDs and hybrid displays. Here we look at specific work related to SAR and OST HMDs to explore how they fit within the context of augmented augmented reality.

Augmented Reality Wearable Projector Systems

Using projectors to augment real-world surface geometry was first explored in Shader Lamps, later conceptualized as SAR [209]. Since then, several projects have explored SAR in complex multi-projector arrangements [85, 114, 207] and

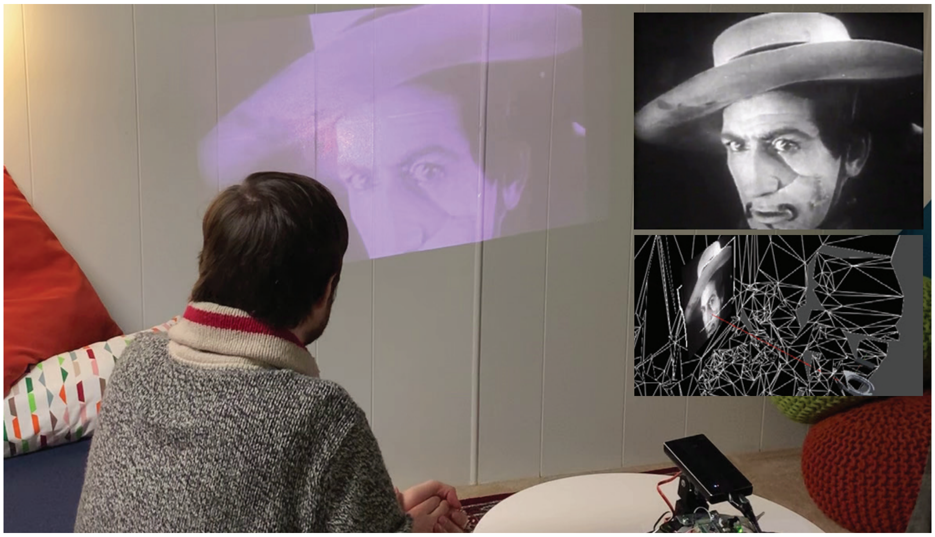


Figure 6.2: AR persistence example application. The projector acts as an ad hoc display when the user takes off the HMD, enabling them to continue watching a video started in AR.

steerable projector systems [126, 201]. Beamatron [267] explored a steerable SAR environment where the projector unit was statically fixed to a location on a ceiling. They demonstrated a broad set of applications that explore SAR under this context. Most relevant, mounting a projector directly on the body has been shown to provide new opportunities for on-body [171] and context driven interaction [160, 268]. OmniTouch [83], a shoulder-worn depth-sensing and projection system, can transform an everyday surface into an interactive space, focusing solely on the the user wearing the device.

Of the many possible on-body mounting locations, on or near the head has been of particular interest. A fixed, front facing head-mounted projector can be used to directly augment the physical environment to reproduce the effect of wearing an optical see-through AR HMD [99, 123]. This has benefits. Scape [98] showed how such "head-mounted projective displays (HMPD)" can enable multi-user collaborative AR. Krum et al. found this approach allowed for more natural depth cues [132], while Kade et al. demonstrated entertainment applications like a shooting game [116]. Genç et al. showed a head-mounted projected image of static and dynamic content is effective when the user is in motion [70].

Our work in this chapter builds on these concepts and extends them by actuating a head-mounted projector and using it with an optical see-through AR HMD. Different from SixthSense [171] and OmniTouch [83], we enlarge the SAR display space and enable interactions which can be independent to the user's location with the actuated projector. By combining a steerable projector with an AR HMD, we can explore the design space between an HMD and external users more broadly and under different location contexts. This contrasts with other work in steerable displays that are fixed, large, and limited to a single location [267]. Our approach can augment the surface geometry for the user, while also enabling external ad hoc collaborations with outside observers.

Public and Private Context Sharing

The use of public and private displays to share content between users has been explored thoroughly in previous works [30]. Augmented Surfaces [214] introduced the concept of hyperdragging which allows co-located users to share private content with a shared public space. Code Space [28] investigates cross-device content sharing with a large public display in the context of code reviews, and MeetAlive [63] explores multi-device sharing in a SAR equipped meeting environments. Sharing can also be accomplished using collaborative augmented reality [15, 198], where interactive experiences are shared among multiple co-located or remote users wearing AR HMDs.

EMMIE [32] uses an AR HMD combined with external displays to merge private and publicly viewable content. Elements of this were further explored in Focus+Context screens [6]. Machuca et al. outlined some design considerations when blending 3D content between a handheld device and a public screen [147]. Serrano et al. [228] explored the combination of an AR HMD within a distributed display environment. Rukzio and Holleis explored a design space that spans a mobile phone and a public projector [219]. Our work builds on previous explorations in context sharing by exploring the asymmetric duality provided by an AR HMD and SAR display and how the user fits within it.

Hybrid AR Displays

Researchers have investigated ways to overcome the limitations of current generation AR HMDs [148, 165] by using sparse peripheral displays, adding LED arrays surrounding AR HMD [75, 276], or by combining a SAR type environments with AR HMDs [11]. Combining a SAR environment with an AR HMD offers extra information [282] and can improve the visual effects [152] inside the AR world. It also provides some useful affordances since the the user's view and the environment are independent, which allows enhanced material rendering [79], shared multi-user experience [125], and fixed environments for an expanded field of view [11].

Closely related to our work is FoveAR [11] which combines a single fixed ceiling-mounted projector SAR environment with an AR-HMD as an extended peripheral display. They demonstrate their approach through a set of four experiences: 3D model animation, wide-angle immersive simulation, 3D life-size telepresence, and an AR shooter game. Each utilize the capabilities of both the projector and the AR HMD. Our work also utilizes a projector and AR display, but the projector is directly attached to the HMD and can be freely repositioned into different viewing configurations.

While these works combine an AR HMD with a fixed ceiling-mounted projector, none have investigated a compact inside-out actuated projector display combined with an AR HMD, nor do they fully explore design considerations for both the HMD user and other external users who could also benefit from projected AR content.

Summary

Our work builds and significantly extends previous concepts with a re-imagined and more comprehensive exploration, applications, and technical solutions. For example, the mobility and flexibility of a head-mounted steerable projector provides a larger set of experiences not possible in fixed projector environments, and our design space spans the asymmetric duality between the HMD user and external observers.

6.2 AUGMENTED AUGMENTED REALITY

An AR display and a SAR environment both have advantages and disadvantages in how they augment the environment and how the user interacts with the virtual content. The AR display can produce high-quality 3D holograms, but is limited to a fixed focus plane with a smaller field-of-view. SAR is able to produce realistic depth cues for surface-mapped 2D content, but 3D content is limited to a single view-dependent perspective. One goal of AAR is to create a setting where a strength of one AR device can offset a weakness in the other. For the HMD user, the two displays can work together to create an enhanced AR experience. In addition, considering the projected display as public, the two displays can provide a dynamic environment in which the HMD user can communicate their virtual environment with external observers. This provides opportunities for new AR modalities for communication. In both cases, the projector is essentially “augmenting” augmented reality.

Design Space

To generate and describe different types of AAR experiences, we developed a concise two-dimensional design space (Figure 6.3). It captures the two important factors: *who* is benefiting from AAR (the "User Role") and *how* the actuated projector is used with respect to the AR HMD view and physical environment (the "Projector Role"). Our design space is complementary to the technically-focused design considerations provided in FoveAR [11]. They present a set of techniques to render an AR HMD with a single projector for only the HMD user. For example, demonstrating how to surface shade textures into the environment with overlayed 3D holograms in the HMD.

USER ROLE DIMENSION *(HMD User, External User, Both Users)*
The projector can be used to improve the experience for the user wearing the AR HMD, one or more “external” users who are standing near the HMD user, or in some cases both types of users can benefit simultaneously.

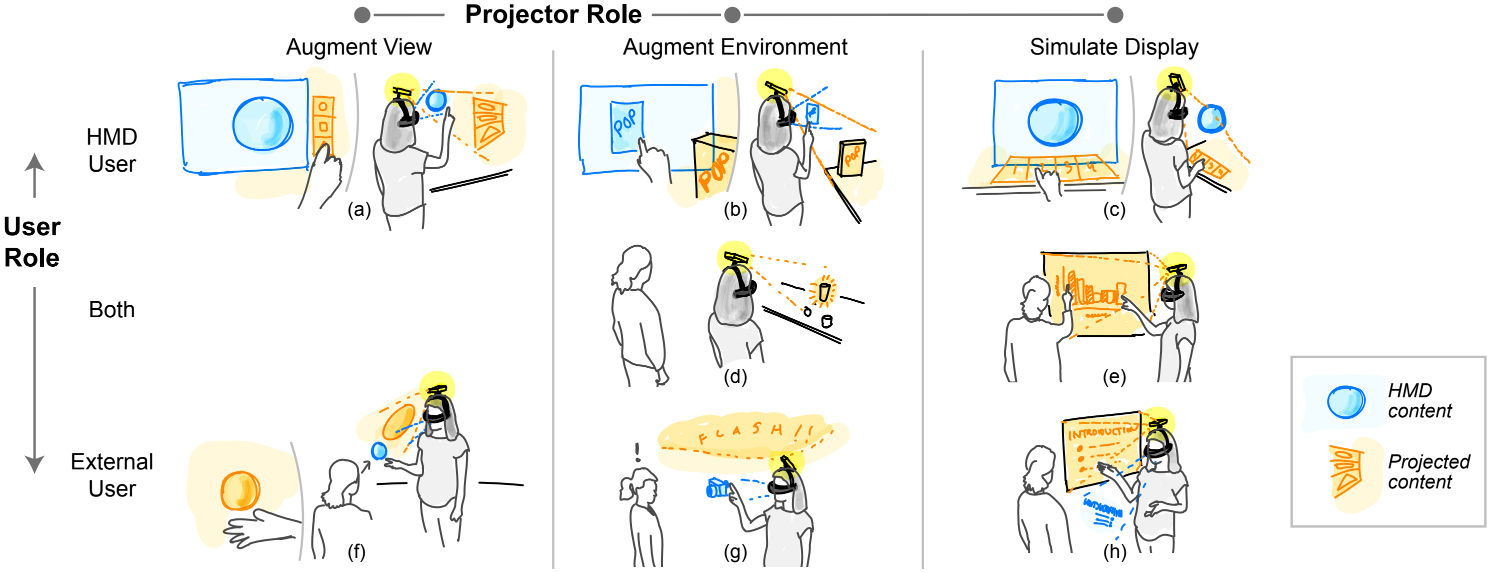


Figure 6.3: AAR design space considering user role and projector role: (a) HMD user's view augmented with heads-up GUI; (b) environment object augmented to show physical preview of a product design; (c) simulating a GUI display on a nearby surface; (d) highlighting an object in the environment to communicate with external user; (e) collaborating on a simulated whiteboard display; (f) augmenting an external user's view with a view-dependent rendering of an object to create the illusion that it is placed in front of the HMD user; (g) augmenting the environment with a camera flash so external users are aware a photo was taken; (h) using a nearby wall to display a slide-show presentation.

There is a dichotomy between the HMD user and external user, as their ability to engage with virtual content is asymmetrical. Clearly, the HMD user has more affordance in the range of actions they can execute on virtual objects and how they view objects across AR views in the HMD and from the projector. For the external user, what content they can see and how they might interact with it is likely to be determined by the HMD user who is present and the dominant actor in the virtual scene. There are exceptions to the requirement of a primary HMD user, for example the HMD could be set on a table so the projector acts like a steerable projection mapped display.

PROJECTOR ROLE DIMENSION *(Augment View, Augment Environment, Simulate Display)*
The interplay between the AR HMD and projector display can be thought of in terms of assistive modalities, where the projector aids the HMD display or the HMD display aids the projector. Previous work has focused on the former, such as FoveAR [11] and occlusion shadows to artificially increase AR HMD contrast [17]. In our space, the projector's role can be expressed as one of three ways in which it renders content.

The projector can be used to *augment the view* of the HMD user or in more limited cases, the external user. Using view dependent rendering with the projector frustum near the HMD view frustum, the projected image can create the illusion of peripheral content on or around the HMD view. For example, it could be used to create the illusion of an extended AR HMD for heads-up GUI (Figure 6.3a). The projector can also augment the external user's view, for example creating a view-dependent rendering on a nearby wall such that they can see a location-matched 3D view into the HMD user's virtual world for the purpose of collaboration (Figure 6.3f).

The projector can *augment the environment* by enabling a steerable surface-mapped SAR or ambient lighting effects. For example, real objects in the environment can be texture mapped to support HMD AR tasks (Figure 6.3b), or ambient lighting effects, like a spotlight, can highlight specific physical objects or locations (Figure 6.3d), or a simulated bright flash can provide feedback when the HMD user captures a photo of the environment (Figure 6.3g). These can support either user, or both users, depending on the context. For example, the spotlight could be to direct the HMD user to a specific object in support of their HMD AR task, or the spotlight can be a way for the HMD user to communicate a spatial location or object to an external user. A surface mapped object like a cereal box could be solely for the HMD user to support their primary activity, or to show a design to an external user for collaboration.

The projector can also *simulate a flat digital display*, whether rectangular on a wall, or mapped to a nearby surface like a table or floor. A simulated display can benefit the HMD user, such as creating a touch GUI on a table to manipulate HMD AR content (Figure 6.3c). For an external user, a simulated display can be projected on a nearby wall, for example so the HMD user can project a presentation on a nearby wall while they consult speaking notes

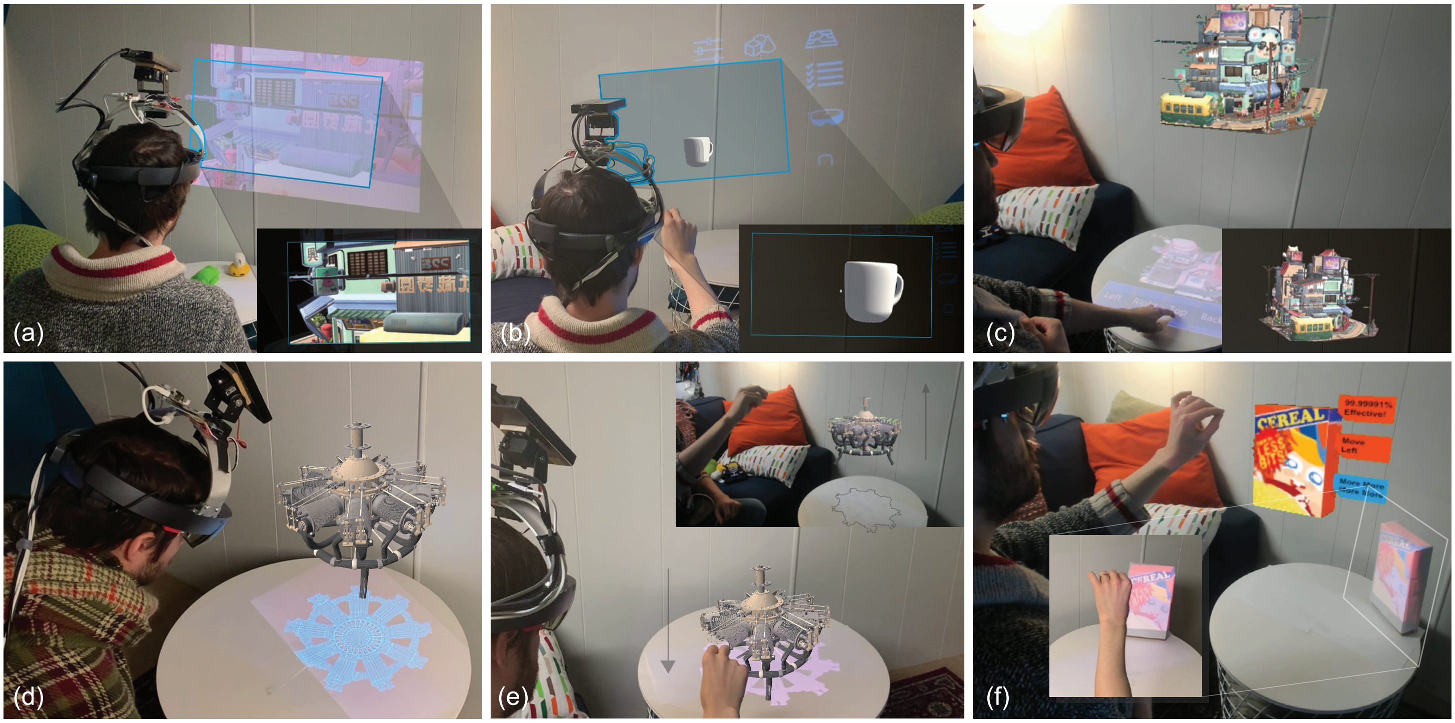


Figure 6.4: Enhanced AR applications: (a) steerable expanded FoV; (b) GUI heads-Up secondary AR display; (c) surface mapped GUI for peripheral interaction; (d) simulated display in physical environment to show orthographic projection of CAD model; (e) simulating virtual object shadows in physical environment; (f) augmenting a physical box in the environment with a surface mapped texture.

rendered only in the HMD (Figure 6.3h), or both the HMD user and external user could jointly collaborate on a projected whiteboard (Figure 6.3e).

6.3 USAGE SCENARIOS AND APPLICATIONS

AAR can be used in a diverse set of applications to elevate the experiences of the HMD user and external collaborators around them. We explore three general categories, with demos that span our design space (Chapter 6.2) to demonstrate the range of experiences possible.

Enhanced AR

The two displays create opportunities to expand the utility and visual quality of AR for the HMD user.

Steerable Expanded HMD Field of View — Similar to FoveAR [11], if the virtual scene is too large to fit within the AR HMD's field of view, the projector can function to artificially widen the user's effective view of the scene (Figure 6.4a). With an actuated projector, utility is further expanded by allowing the user to switch focus to particulars parts of the object, or by locking onto a target object, continually rendering it in the periphery even as they move around. *(Design space: Augment view, HMD user).*

Secondary Heads-Up Display — A view-dependent render of virtual objects floating around the viewing frustum of the AR HMD can be used to create a heads-up display for the HMD user without interfering with any content within the AR display. For example, a GUI can be created, such as a toolbar, menu, or clipboard, that can expand the user's ability to work (Figure 6.4b). Another use can be for peripheral awareness. For example, if an object of interest is outside the user's current field of view, an arrow can be used to point in the direction of the object, guiding the user to find it. *(Design space: Augment view, HMD user).*

Secondary Environment Display — A secondary display can be rendered onto nearby surfaces to provide new utility for the HMD user. For example, a GUI can be rendered on a nearby table that allows the user to interact with virtual content while maintaining their focus on a object they are working with (Figure 6.4c). Another use case is to provide an alternative perspective on a virtual object. For example, an orthographic projection of a plane engine can be rendered so that its schematics can be projected onto a table in front of an engineer. Because the projector is attached to their head, they can move in closer and study its finer details (Figure 6.4d). *(Design space: Simulate display, HMD user).*

Simulate Physical Phenomenon in Environment — A simulated shadow of a virtual object can be projected onto a physically realistic location in the environment. For example, when an engineer is examining a virtual 3D model, an inverted shadow cast on the nearby table or wall could communicate its physical height off the surface below (Figure 6.4e). The projector can simulate the flash of a camera when the HMD user captures a "photo" of the physical environment. This adds additional meaningful feedback and increases realism. Both of these physical phenomenon provide some benefit to the external user as well. Object shadows can provide ambient awareness to external users, communicating that the HMD user is editing some type of object. A flash effect lets any external users nearby know their image may have been captured. *(Design space: Augment environment, Both users).*

Physical Object Augmentation — The projector can directly augment a physical object in the space around the user. For example, if a graphic designer is iterating on a product box design, the projector can surface map the virtual box onto a physical prop in the real environment, giving them an idea of what the final product will physically look like (Figure 6.4f). This would also benefit external users as well, enabling them to monitor progress or critique design choices. *(Design space: Augment environment, HMD user).*

Sharing AR

The HMD user can utilize the projector to explicitly share virtual content with the external users around them, enabling new forms of interaction and collaboration.

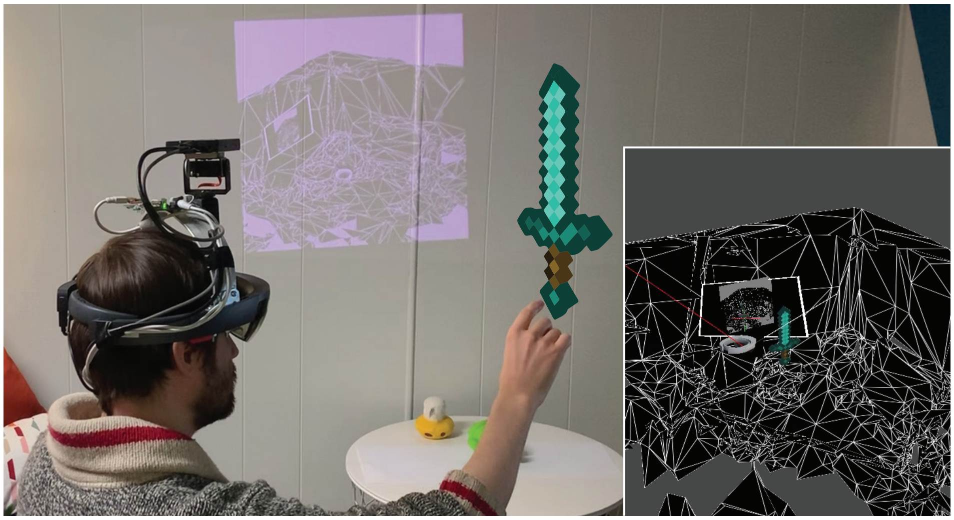


Figure 6.5: Sharing AR example: an external user viewing the virtual world of the HMD user on a simulated display created by the projector.

Explicit Content Sharing — The projector can be used to display curated content for external users. A simple example is when the HMD user and external user jointly interact with a simulated display, like a whiteboard brainstorm or card sorting task. During collaboration, the HMD user may be focusing entirely on the shared projected display. *(Design space: Simulate display, Both users).* A more interesting example is when the HMD user focuses on AR content while external users focus on the projected display. For example, during a meeting, a slideshow presentation could be rendered on an adjacent wall for external users to view, while the HMD user presents using speaker notes rendered in the HMD (Figure 6.1d). *(Design space: Simulate display, External user).*

Virtual World Camera — The HMD user can share a 2D rendering of the virtual world as viewed from an arbitrary virtual camera position. This would be a simulated display projected on an available surface near the HMD user, enabling external users to see the virtual world as they would from a typical desktop display. The camera could even be controlled by the external user, allowing them to explore the HMD user's virtual environment. (Figure 6.5) *(Design space: Simulate display, External user).*

Window to Virtual World — When the HMD user is working with a virtual 3D object, they may need to show an external user what it looks like or how they are interacting with it. This is made possible by projecting a view onto a wall opposite of the external user and then rendering the scene from their perspective. This will create an illusion where the scene objects appear to be at the correct location and have the same shape and size relative to the HMD user (Figure 6.1c). *(Design space: Augment view, External user).*

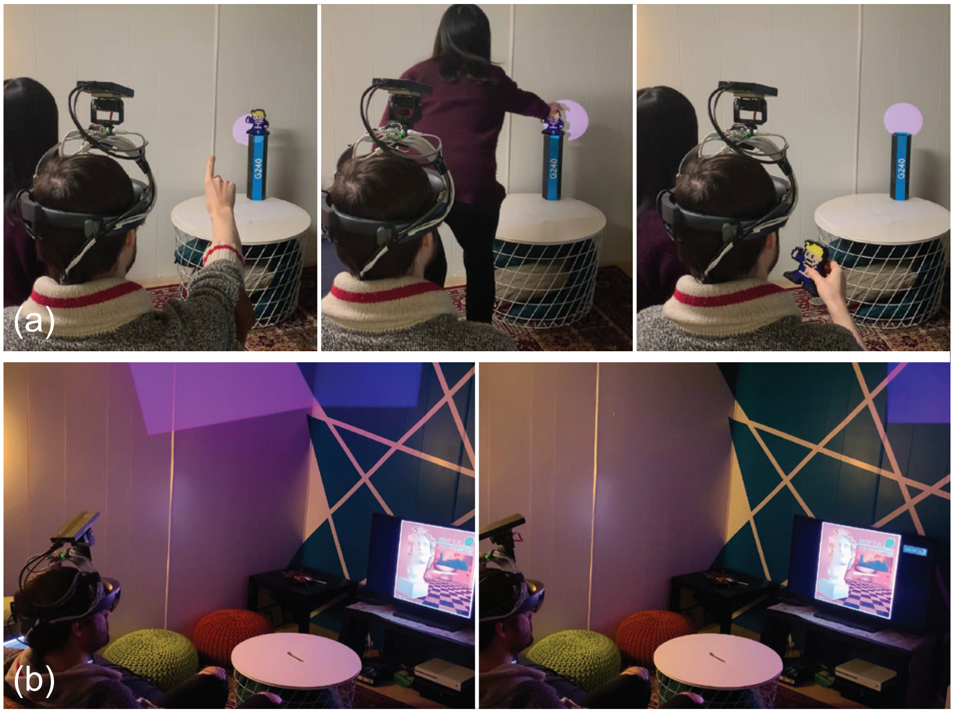


Figure 6.6: Ambient display examples: (a) spotlight as a contextual environment display; (b) mood-lights as ambient environment lighting

Ambient Display

There are circumstances where rendering scene objects or projection mapping is not needed, but an external light source may be desirable for notification, awareness, or navigation. We explore these across two related cases.

Contextual Environment Display — The projector can be used as a controllable spotlight, directed by the HMD user to highlight objects or locations in their environment. For example, the HMD user could pin an object with light, keeping track of where it is, or they could highlight an object to direct external users to it (Figure 6.6a). *(Design space: Augment environment, Both users).*

Ambient Environment Lighting — The projector can act as a generalized source of light. Reflecting it off a ceiling could add illumination to the surround environment or artificially adjust its colour temperature. The projector can also be used to enhance multimedia experiences by producing ambient RGB lighting effects. For example, it could be used to enhance a music listening experience or to elevate PC gaming sessions during a livestream (Figure 6.6b). *(Design space: Augment environment, Both users).*

Persistent AR

Its likely that the HMD user may wish to remove the HMD from time-to-time to take a break, have a snack, or other real world tasks. Current HMDs have no self-contained way to transition from a virtual AR task to one in the real world. The projector can enable such a transition for limited, but potentially useful, interaction and awareness of the virtual world. When the HMD user removes the headset, the projector can create an ad hoc inside-out SAR environment. For example, if the HMD user is watching a video in the virtual world, then removes the HMD to place it on a table, the projector can automatically transition the video to a simulated display on an adjacent wall (Figure 6.2). In this example, the HMD user becomes an external user in a unique "external users only" AAR usage context. *(Design space: Simulate display, External user).*

6.4 ACTUATED PROJECTOR FOR AN AR HMD

Prototyping the AAR design space with real applications requires precise control over the projector's movement relative to the AR HMD. We present a novel approach to calibrate the positions and offsets for the projector and pan-tilt structure.

Hardware

A Celluon PicoBit laser projector [199] (Fig. 6.7a) is mounted onto two linked servo motors that form the pan-tilt mechanism (Fig. 6.7b). An Arduino Pro Mini ATmega328 acts as their controller (Fig. 6.7c). Both the Arduino and pan-tilt mechanism are attached to a custom aluminum mounting bracket which is bolted onto a Hololens V1 [165] (Fig. 6.7d).

The projector has a resolution of 1280 × 720 pixels with a brightness of 63 ANSI lumens. The laser projection module is infinite focus, which eliminates the need to manually adjust the projector's focal plane. Considering a left-handed coordinate frame, the Hololens points along positive Z, the top servo tilts the projector along its X-axis, and the bottom servo pans the projector along its Y-axis. A Kuman [133] 17 Kg high torque 270° motor is used for the bottom servo and a DFRobot DSS-M15 180° motor is used for the top. Both servos are hard-limited in range to ensure no damage to the Arduino or projector can occur. The bottom servo is limited to a range between 15° and 255°. The top servo is limited to a range between 30° and 125°. When the servos are set to 135° and 90° respectively, we consider the projector to be in its default position, pointing forward along the Z-axis with the Hololens.

Automatic Calibration

In order to enable the range of experiences outlined in the usage scenarios, a one-time calibration is required for the hardware. Steerable projector systems

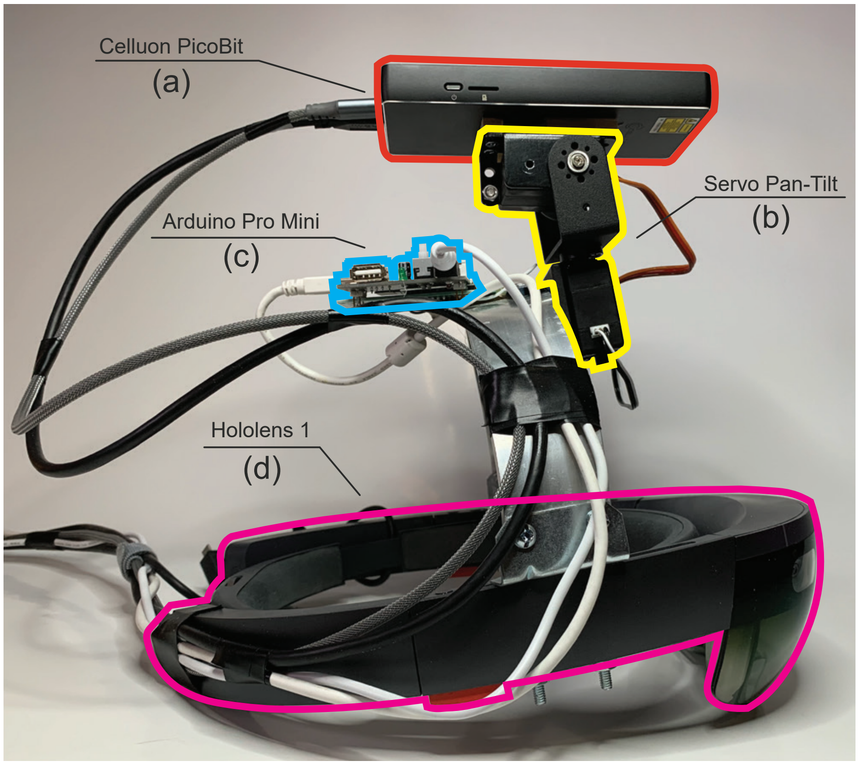


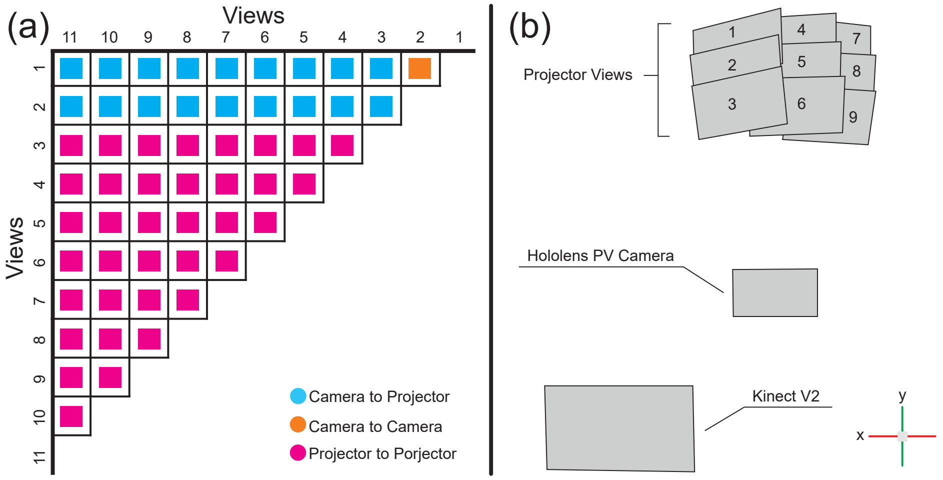
Figure 6.7: Actuated projector HMD: (a) Celluon PicoBit laser projector; (b) Kuman and DFRobot pan-tilt servos; (c) Arduino Pro Mini; (d) Microsoft Hololens V1.

have previously discussed calibration techniques for projector-camera units through the physical repositioning of a checkerboard in an environment [186, 267]. However, these current approaches are labour-intensive and the internal geometric structure of the actuators are not fully captured. Another concern is the coupling between the projector and the HMD device, where the projector's world pose changes with the transformation of the HMD.

We frame projector and camera pose reconstruction as a Structure from Motion (SfM) [177] problem, which is commonly used in large scale computer vision problems, like photogrammetry. The reconstructed poses are used in an optimization routine over a Denavit-Hartenberg (DH) [51] parameterization of the servos' kinematic chain, finding their axes of rotation and offsets. The resulting geometric relationships can be used for precise movement control.

An SfM pipeline typically has four phases: (1) data acquisition, (2) feature point detection, (3) putative point correspondence matching, and (4) pose and point cloud reconstruction [177]. We adapt this pipeline for an expanded set of correspondences required by our pose reconstruction problem that utilizes projector-projector, projector-camera, and camera-camera correspondence pairs.

Data acquisition — Previous work has used structured light to create dense camera-projector maps using Gray codes [114]. In contrast, we adapt the



Views	11	10	9	8	7	6	5	4	3	2	1
1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟧	
2	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		
3	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪			
4	🟪	🟪	🟪	🟪	🟪	🟪	🟪				
5	🟪	🟪	🟪	🟪	🟪	🟪					
6	🟪	🟪	🟪	🟪	🟪						
7	🟪	🟪	🟪	🟪							
8	🟪	🟪	🟪								
9	🟪	🟪									
10	🟪										
11											

● Camera to Projector
● Camera to Camera
● Projector to Porjector

Figure 6.8: View-pair correlation matrix and view frustums: (a) point correspondence pairs between the camera views (1 and 2) and projector views (3-11); (b) reconstructed view frustums for the projector, Kinect, and Hololens viewed in the direction of the Z-axis.

approach from Yamazaki et al. [279] which combines Gray codes with phase-shifting sinusoidal codes to increase the sub-pixel accuracy of the resulting maps. During data aquistion, the projector is repositioned 9 times using the pan-tilt servos, moving in increments of 10° for the X-axis and 15° for the Y-axis to create a 3 × 3 grid of projector views (Fig. 6.8b). For each of these views, the built in Photo-Video (PV) camera from the Hololens V1 and a Kinect V2 camera is used to capture the projected structured light. We use a Kinect camera to capture a wider FoV of the scene, which allows larger projector movement during acquisition. However, we do not use any depth data in our pipeline, any other wide FoV RGB cameras would also work. The total number of views is 11, (9 repositioned projector views and 2 stationary cameras views), resulting in 18 camera-projector pairs with a total of 1134 captured images across all structured light sequences.

Putative Correspondences — Calibrating with structured light does not require feature point detection to build view correspondences. Instead a dense point-to-point correspondence between the camera and projector can be achieved by decoding the captured structured light during data acquisition. By utilizing both a forward mapping (a pixel-point in the image to sub-pixel in the projector) and a reverse mapping (pixel-point in projector to camera pixel), a complete set of putative correspondence pairs can be created for all 11 represented views. The complete set of view-pair regions is 55, composed of 18 camera-projector, 1 camera-camera, and 36 projector-projector putative correspondence pairs (Fig. 6.8a).

Structure from Motion — Using the complete set of correspondence pairs, we extend the open source implementation of OpenMVG [178] to account for the different view pair regions discussed above. We solve for all 11 views using sequential SfM [177] with AContrario RANSAC [174]. The reconstruction process solves for: (1) the extrinsic (i.e. poses) and intrinsic parameters of the views, and (2) the 3D point cloud of the environment (Fig. 6.8b). A 25mm × 25mm checkerboard is used to solve the scale ambiguity of the resulting reconstruction; no repositioning is required. In actuality, any known point-to-point distance in the scene could be used instead; the checkerboard is used for convenience, it is not a requirement for our calibration. The final RMSE is 0.51mm on 2.3 million residuals taking 131 seconds.

Optimization Solver to Recover Pan-Tilt Geometry

We take the 9 transformation matrices representing the views of the projector when actuated, and solve for the kinematic chain and rotation axes of the servo motors. A Denavit–Hartenberg (DH) parameter representation is used to represent this structure relative to the projector's frustum (F). Each rotation axis is represented by a rotation matrix (R_X and R_Y) that is parameterized by a single rotation value θ and ϕ in radians (Fig. 6.9a). A DH parameter is 4×4 transformation matrix defined by a translational offset and rotational displacement along a single axis (e.g. the DH parameter for the transformation along the z-axis is $Z = \{z^t,\ z^\theta\}$ where z^t represents translational offset and z^θ rotation around the z-axis). In this way, a homogeneous point ($\hat{x}$) in the projector's coordinate space (F) can be transformed to the coordinate space of the base servo (B) by

$$\hat{x}_B = R_Y^\phi Z_2 X_2 R_X^\theta Y_1 Z_1 \hat{x}_F \tag{6.1}$$

where the unknown parameters are: (1) the rotations around the x- and y-axis (R_X^θ and R_Y^ϕ), (2) the DH parameters (Z_1 and Y_1) describing the link between the projector's view to the x-axis, and (3) the DH parameters (X_2 and Z_2) describing the link between the x- and y-axes, which is the base servo. The knowns are the point observations ($\mathcal{X}_{obs}$) and the 3×3 grid of projector view transformations.

Further, if we consider the relationship between the projector's centre transformation matrix (T_5) with the other 8 surrounding transformation matrices (T_i) in the 3×3 grid of projector views (Fig. 6.8b), we can use a variation of equation 6.1 to relate a homogeneous world point ($\hat{x}$) observed from the centre projector view to any other projector view through the following equality constraint:

$$R_Y^\phi Z_2 X_2 R_X^\theta Y_1 Z_1 T_i \hat{x} = Z_2 X_2 Y_1 Z_1 T_5 \hat{x} \tag{6.2}$$

where we consider the centre view (T_5) to be the calibrated view and default projector transformation.

With the equality outlined in equation 6.2, a cost function is constructed to solve for the unknowns, enumerated as: $\Phi = \{z_1^t, z_1^\theta, y_1^t, y_1^\theta, x_2^t, x_2^\theta, z_2^t, z_2^\theta, \theta, \phi\}$. The cost function contains two parts, one describing the constraint for the servo rotating around the x-axis ($f_1^{(i)}(\hat{x})$) and one describing the constraint for the servo rotating around the y-axis ($f_2^{(i)}(\hat{x})$), where both are parameterized by the view transformation i:

$$f_1^{(i)}(\hat{x}) = Y_1 Z_1 T_5 \hat{x} - R_x^\theta(i) Y_1 Z_1 T_i \hat{x} \tag{6.3}$$

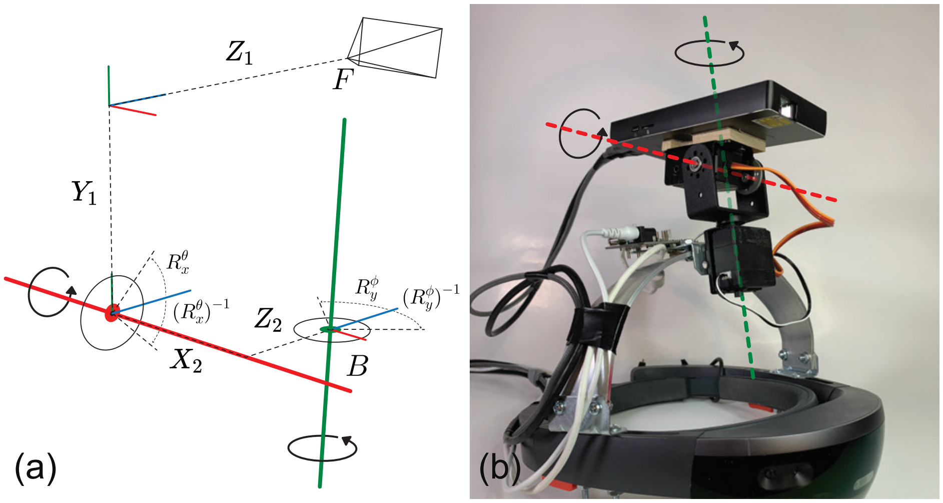


Figure 6.9: Projector mount geometry: (a) DH parameters and pan-tilt axes forming a kinematic chain from projector (F) to base servo (B). (b) depiction of the pan-tilt axes.

$$f_2^{(i)}(\hat{x}) = Z_2 X_2 Y_1 Z_1 T_5 \hat{x} - R_y^\phi(i) Z_2 X_2 R_x^\theta(i) Y_1 Z_1 T_i \hat{x} \tag{6.4}$$

Given that the rotations around the x- and y-axes are symmetrical, we need to determine whether the inverse of the rotation matrix is appropriate based on what view transformation is used within the equality constraint. We define a function $R_x^\theta(i)$ and $R_y^\phi(i)$ to provide the correct rotation matrix based on whether the transformation is from the top, bottom, left, or right column of the 3 × 3 grid of projector views.

$$R_x^\theta(i) = \begin{cases} R_x^\theta & i \in T_{1,*} \\ (R_x^\theta)^{-1} & i \in T_{3,*} \\ I & \text{otherwise} \end{cases} \qquad R_y^\phi(i) = \begin{cases} R_y^\phi & i \in T_{*,1} \\ (R_y^\phi)^{-1} & i \in T_{*,3} \\ I & \text{otherwise} \end{cases} \tag{6.5}$$

If we consider $\mathbf{x} \in \mathcal{X}_{obs}$ an n-dimensional vector of homogeneous world point observations and we utilize equations 6.3, 6.4, and 6.5, we can minimize the following through a non-linear least square Levenberg-Marquardt [137, 155] trust region [33] method with Cauchy loss:

$$\underset{\Phi}{\operatorname{argmin}} \frac{1}{2} \left[f_1^{(1)}(\mathbf{x}), ..., f_1^{(9)}(\mathbf{x}), f_2^{(1)}(\mathbf{x}), ..., f_2^{(9)}(\mathbf{x})\right]^{\top} {}^{2} \tag{6.6}$$

The resulting solution is used to reconstruct the projector's centre view (T_5) and the servos' kinematic chain relative to the Hololens PV camera. A final step rectifies all reconstructed transforms to the origin point of the Hololens device. We use these final transforms in our software toolkit.

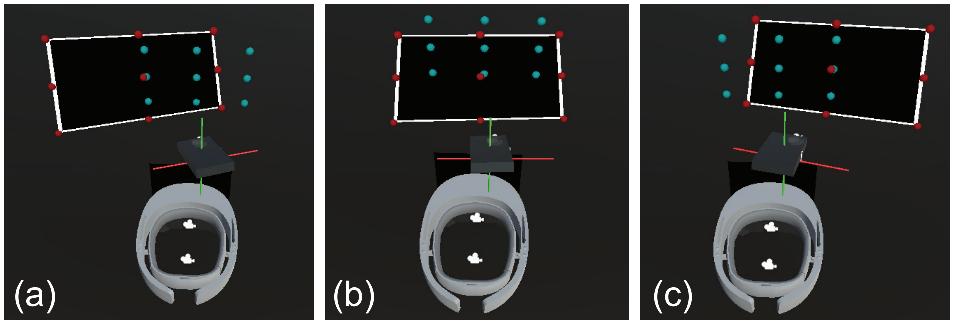


Figure 6.10: Movement of the projected image (red dots) to locations around the HMD (blue dots) in Unity3D: (a) left; (b) centre; (c) right.

6.5 AAR SOFTWARE TOOLKIT

To assist in the creation and design of AAR experiences, we developed an open source software toolkit[1] for Unity3D [257]. Our toolkit works in conjunction with the Microsoft Mixed Reality Toolkit (MRTK) and provides a streamlined development experience for creating and iterating on AAR design concepts.

Toolkit functionality is divided into two parts: (1) a native C++ plugin encapsulating the servo, projector, and calibration controllers; (2) a C# unity package that interfaces with our native plugin and provides high-level APIs for AAR services. Each service is associated with a Unity Prefab Asset, encapsulating 3D object information, editor metadata, and the toolkit's C# scripting components. These can be easily drag-and-dropped into an active scene. The toolkit is able to simulate all the functionality of the real hardware, enabling development without the need for a physical device.

Projector Control — This provides interfaces to manipulate the projector and servo hardware, giving developers high-level control over where the projector is pointing in the virtual environment. This can be to locations relative to the frustum of the Hololens (Fig. 6.10), or can be to specific points in the world coordinate system. The toolkit uses the servos' calibrated axes to calculate all necessary rotations for both software simulation and hardware control.

Spatial Awareness — A unified 3D model of the environment is provided by the Hololens. This is represented inside Unity as a mesh object from which ray intersections and object collisions are possible. Similar to RoomAlive [114], our toolkit reconstructs the planes in the scene and provides them with meaningful semantic names (e.g. floor, wall, ceiling, and unknown) based on their surface normal, size, and position.

Rendering Spatial Content

We built a rendering engine that is able to handle different configurations of the AR display and projector.

[1] https://github.com/exii-uw/AARToolkit

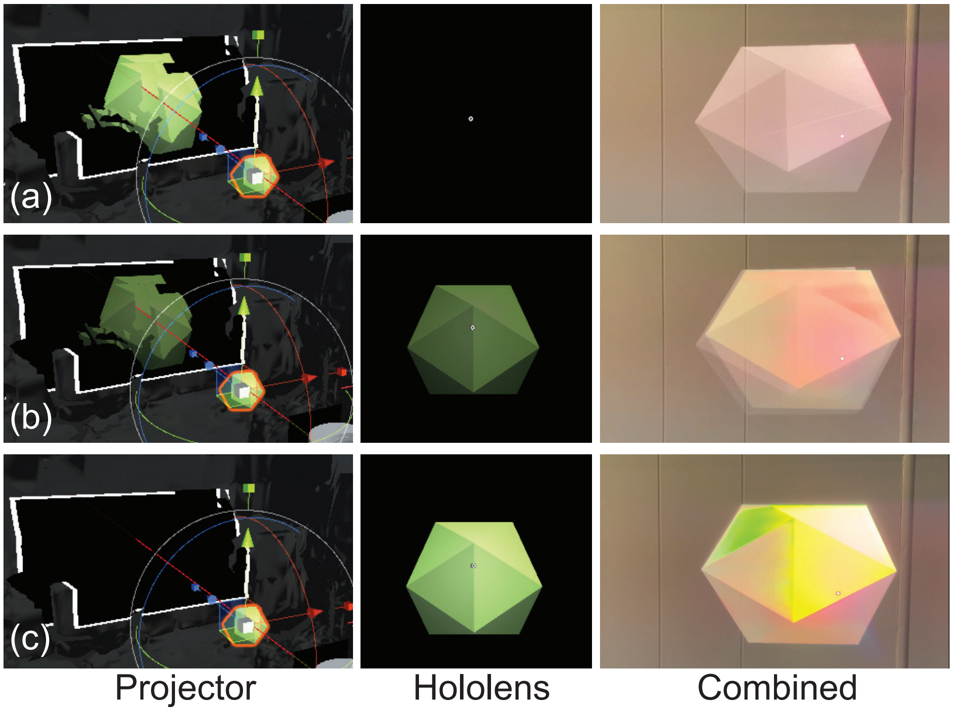


Figure 6.11: Blending a virtual object between the projector and Hololens at: (a) 100% and 0%; (b) 50% and 50%; (c) 0% and 100%.

Rendering as an External Display — Static textures, shaders, or videos are rendered and displayed in the real-world environment. This is useful for situations where projection mapping is not needed, such as ambient lighting or making a spotlight. With surface mapping enabled, the view can be used as a secondary display, showing contextual information or an interface to interact with, all rendered onto a physical surface in the environment. We developed a custom material object to abstract all possible combinations.

Rendering for the HMD User — The projector can be treated as a camera in the virtual environment, enabling it to render virtual scene objects from multiple different viewing locations. When rendering for the HMD user, the developer must decide how much of each object is visible in each display. This is realized through a custom shader post-processing stack that can be accessed through a single C# script attached to the object being blended. The developer can individually assign how much the object is visible in each display, allowing detailed control (see Fig. 6.11).

Rendering for External Users — Our toolkit gives developers access to specific projection mapping functionality, allowing a perspectively correct view from any external location.

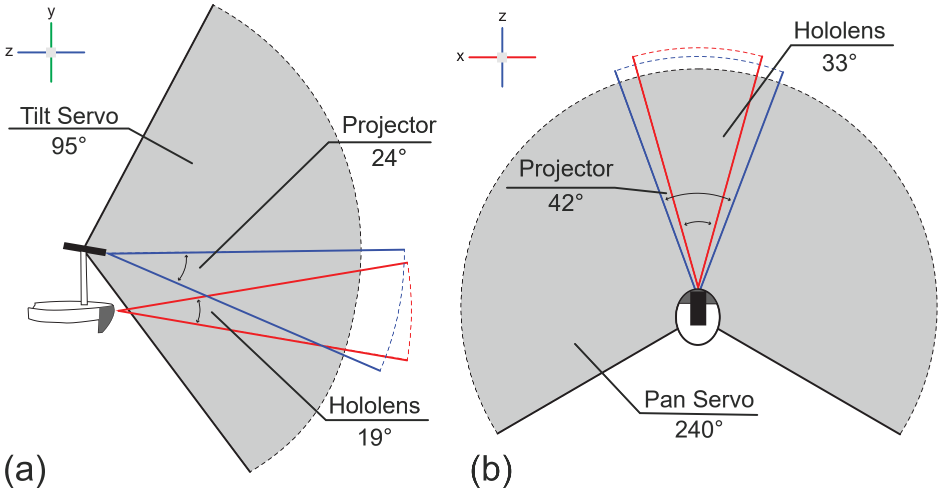


Figure 6.12: AR display and project field of view (FoV) with pan-tilt servo range: a) vertical FoV and tilt range, b) horizontal FoV and pan range.

Performance and Quality Analysis

In this section, we report on toolkit performance for the rendering pipeline and servo control. Note the calibration process is only for initial hardware setup, it has no impact on runtime performance.

Field of view — The vertical FoV of the projector and Hololens is 24° and 19° (Fig. 6.12a), while the horizontal FoV is 42° and 33° (Fig. 6.12b).

Servo control — The servos are able to relocate the projector within a range of 95° vertically and 240° horizontally. There is a 74ms latency from the moment an action occurs in the toolkit to the physical movement of the servo. The servo takes 180ms for every 60° rotated.

Rendering performance and latency — The toolkit updates at a frame rate of 60 FPS. The point-to-point latency from the moment the toolkit serves a rendered frame to the moment the projector displays that same frame is 116ms. This latency results from the projector's hardware and firmware design.

6.6 DEVELOPER TOOLKIT STUDY

The goals of this open ended remote study were to verify whether developers understood the concept of AAR and whether the toolkit provided adequate tools to create AAR experiences. Over 7 days, invited AR developers familiarized themselves with the AAR concept and toolkit, and created their own AAR applications. They did not have access to our hardware, but the toolkit simulates the HMD and projector inside Unity3D.

We recruited 6 developers through social media websites: ages 20 to 38, 2 male and 1 female. Three became unresponsive after initial onboarding, and so were dropped from the study. All reported experience with AR and VR development, and all are familiar with Unity and the Microsoft Mixed Reality

Toolkit (MRTK). Over 7 days, the participants spent 10, 3, and 6 hours using the toolkit, totalling 19 hours altogether. Each received a $50 gift card for their time.

Our study included 3 stages: (1) tutorial and pre-experiment questionnaire, (2) independent development and (3) post-experiment questionnaire.

On day 1, participants completed a demographic and development experience survey, then participated in an one-hour live-streaming tutorial with a question and answer period. During the tutorial, the concept of AAR was introduced and the hardware was demonstrated to ensure that participants were familiar with the structure of the actuated projector and Hololens. This was followed by an introduction to our toolkit, including software environment, installation, and functions.

On days 2 to 7, the participants developed an AAR application at their own pace. They used Slack to ask questions and to comment on their progress.

After the seventh day, each participant submitted their Unity3D project and completed a questionnaire about how they worked with the toolkit, what applications they created, and their thoughts when designing for AAR.

Results

Although our study was small in terms of participants, we believe results from even a few expert developers over this longer usage period are reasonable as a first validation of the AAR concept and toolkit.

AAR APPLICATIONS Each participant was tasked to create an AAR application. P1 created a virtual gallery where the HMD user can see descriptive text of gallery objects projected on the wall with the projector. They describe it as: "a 'virtual gallery' wherein the AR headset wearer acts as a docent and can view a script which can prompt them with information on what art piece is being viewed, either for lecture or other use." P2 explored the use of AAR for human perception experiments. They continue to state the "The external observer's task would be to judge which plate has a larger portion of food." P3 created a card game in which the HMD and external user can both view their cards through their respective displays. The projector's direction would be changed according to external user's viewpoint. They described it as: "this scene ties into AAR since both an HMD user and an external user are being part of an AR experience."

FEEDBACK AND DISCUSSION Participants generally enjoyed working with the toolkit. P1 said that "It was straightforward to work with the library and functionality ... I understood the concepts being presented, and I can see a lot of potential use for this type of setup." P3 commented "It's an interesting and unique concept with its own set of UX considerations to think about. I think it could be really useful in providing AR experiences to large environments with small groups of people."

Most of the issues and suggested improvements are related to our toolkit or the MRTK directly. P1 said "The tooling was not straightforward to set up at first, better documentation will help." P3 echoed this, "I found the documentation a little confusing at first." P1 suggested that "Simplification of which camera mode and which layers can be viewed will help with this process." P3 commented on the need for MRTK profiles, suggesting "a lot of the project settings/configuration [can] be automated, just like the MRTK." For the spatial awareness feature, they suggested additional features like a plane destruction function. They continued to suggest future support for other HMDs and multi-HMD and multi-projector setups.

Overall, participant comments and applications suggest they all understood the concept of AAR. Without knowing our proposed usage scenarios and applications, participants developed similar ideas. The virtual gallery (P1) was similar to our explicit content sharing applications and the card game (P3) was an extension of the window to a virtual world usage scenario. This showed how practical and essential AAR can be. All had a positive experience and agreed that our toolkit is useful: *"excited by the possibilities"* [P1], *"a lot of potential"* [P2], and *"unique concept"* [P3].

6.7 DISCUSSION

Our AAR concept is highly dependent on hardware capabilities and usability. In this section, we enumerate current limitations with possible solutions, and present the most compelling future enhancements.

AAR Hardware and Usability

Our proof-of-concept prototype was adequate for demonstrating applications, but there remain aspects that could be refined.

Projector Latency and Refresh Rate — The PicoBit projector scan process is 60-hertz interlaced with no persistence, and its input to output latency is reported as 116ms. This can cause notable image lag during movement, even when the projector is fixed and not moving relative to the HMD. This problem can also be observed in commercial HMDs, where the rendering pipeline, latency, and refresh rate of the HMD can cause the image to lag, creating a mismatch between the in-game state and what the user can see. This could be partially migrated by using a 90 or 120 Hz projector with low latency output which will help minimize mismatch and improve the overall experience.

Servo Accuracy — We actuate the projector with servos commonly used by model and electronic hobbyists. While sufficient for small robots, we found their movement sometimes inconsistent, and lacking some precision. This is especially noticeable when projecting over a large distance (>5m): if the servo is off by 1°, that equates to 9cm displacement of projected content. To help alleviate this, our calibration process recovers the "real" degrees moved, and scales output in the toolkit accordingly. Further improvements could be achieved with higher-accuracy actuators.

Projected Image Stability — Related to latency and servo accuracy is projected image stability. Currently, the toolkit knows where the HMD is relative to world geometry, which allows the calibrated pan-tilt mechanism to compensate for any head movement in the world and can keep the projected image steady for different positions, movements, and angles. However, tracking and environment scanning is only as good as what the HMD device provides, and more critically, the servos are limited by their rotation speed. Projector output latency further compounds this. In most cases, instability is not very perceptible, such as when augmenting the view of the HMD user where the projected image is anchored to the HMD frame of reference. In other cases, like surface mapping a 3D object or projecting onto a distant display for an external user, projector instability can be noticeable. Although this was acceptable for our usage, faster and more accurate servos, along with improved refresh rates and low latency projectors, could further improve image stability and the user experience.

Weight and Ergonomics — The original Hololens V1 is notorious for being uncomfortable to wear for long periods of time. It weighs 579 grams which is ideally distributed around the crown of the head. Our mounting bracket, servos, and pico projector add an additional 585 grams, resulting in just more than 1kg total. It is not to the point where a pulley counter balance is needed, but it can cause noticeable discomfort for periods longer than 10 or 20 minutes. Improvements in weight distribution and minimization of the hardware through OEM LBS projection engines [164] and custom circuit designs will reduce the weight and improve comfort.

Projector Eye Safety — Our PicoBit is a laser beam scanning (LBS) projector [78], which has a potential safety issue due to the "IEC-60825-1: Class 3R Laser" classification: prolonged direct eye contact into the beam can be harmful [220]. In practice, staring into projectors using other technologies, like LCoS and DLP, should generally be avoided too. Methods have been proposed to automatically block projected light from entering people's eyes [121], and these could be incorporated into our system. A key advantage for LBS in an ad hoc SAR setting such as ours is infinite focus, this ensures that the image is crisp no matter where it is pointed within an environment.

System Extensions

We enumerate some possible future extensions.

External user tracking — Tracking external users in the space around the HMD user is a potentially exciting avenue for future work. The expanded utility could open up interesting new experiences for AAR interaction, like adding motion parallax to a view-dependent projection or creating new immersive gaming experiences.

Improved Toolkit Integration — The toolkit is specific to a Hololens V1 with a head-mounted projector. Extending it to work with other AR HMDs is one direction for future work that will broaden its generalizability.

Future Work

In its current form, the AAR hardware and toolkit allow a user to freely move around a physical space and augment their environment through either the projector, AR display, or both. Here, we discuss two topics for future work.

Hardware minimization — As outlined above, there are many directions to improve the hardware implementation. These include using custom projection engines with low latency output, to smaller and faster servo motors. Minimizing and improving hardware could reveal new interaction modalities and alternative mounting locations on the HMD.

User-centric Studies — AAR specific input and interaction could be further extended and investigated, and the user-centric impacts of the system pipeline could be further explored. For example, specialized interaction techniques could be created and evaluated in experiments for critical tasks like pointing and selection. Studies could investigate the perceived user affordances of imagery when presented on the projector compared to the HMD. Another avenue of exploration would be a study that investigates specific asymmetric interactions between an external and HMD user during collocated collaborative tasks.

6.8 SUMMARY

We presented the concept of Augmented Augmented Reality for a wearable augmented reality HMD and an actuated head-mounted projector. We constructed a working hardware and software system, calibrated through a modified structure from motion algorithm and a novel optimization solver to reconstruct the kinematic chain and rotation axes of the actuators. Our Unity3D toolkit encapsulates a set of high-level functionality for the iteration of AAR experiences. We hope our work inspires more investigations into combining different AR devices in new ways, and the pursuit of ever more immersive experiences that can still remain grounded in our physical and social world.

7 SPECTATORSHIP OF VIDEOGAME LIVE STREAMS IN VIRTUAL REALITY

Videogame live-streaming has become a popular pastime for both the streamers producing content and for the spectators consuming it [80, 232]. Web-streaming services, like Twitch [107] and YouTube Gaming [105], provide a platform for not only distribution of this video content but also a way for audiences to engage with the streamers and each other.

The typical stream consists of a primary game view, containing the actual gameplay footage, and a composited picture-in-picture feed of the streamer captured through an external front-facing camera. All this footage is acquired through an external application, like Open Broadcast Software (OBS) [5], that duplicates the rendered videogame frame, encodes it, and then transports it to a streaming media server for distribution. The final content can then be viewed on various devices such as a desktop computer, mobile phone, or television screen. In this current structure, the role of the spectator is asymmetric to that of the streamer: the spectator's primary role is to passively watch the streamer with an optional and minimal chat interface for shared discussion. However, there is a growing trend of adding interactive elements into the stream for spectators. These are typically composited animations and graphics that react to specific keywords in the chat, but these can also consist of more complex arrangements where the spectator is given the ability to invoke an action directly within a predefined virtual environment [241].

Despite the maturity of this medium, the rise of virtual reality (VR) has remained a challenge for both streamers and their spectators in subtly different ways. For streamers playing VR games, they have the challenging task of communicating what they are doing. One popular solution for non-VR spectators is to swap the streamer's first-person headset view for a third-person perspective using software like *LIV* [104]. However, the effectiveness and benefit of this simple approach remain unclear [60]. For spectators watching a videogame stream in VR, they are relegated to using a virtual theatre-like environment with a *big* screen [13]. These environments are effective at creating social spaces [45, 140] but they are incapable of taking advantage of the 3D environment of the videogame in any meaningful way.

In this chapter, we introduce a system that expands the capabilities of videogame specatorship by incorporating interactive 3D and VR elements into existing videogame streams. We realize this by intercepting the depth data exposed by low-level graphics rendering pipelines and utilize that data to dynamically build a reconstruction of the videogame environment at runtime. This reconstructed 3D environment enables new visual and interactive capabilities for the spectator. For example, the spectator can view the streamer's actions from inside the game environment with full control over their position and vantage point. We demonstrate the flexibility of our approach through a

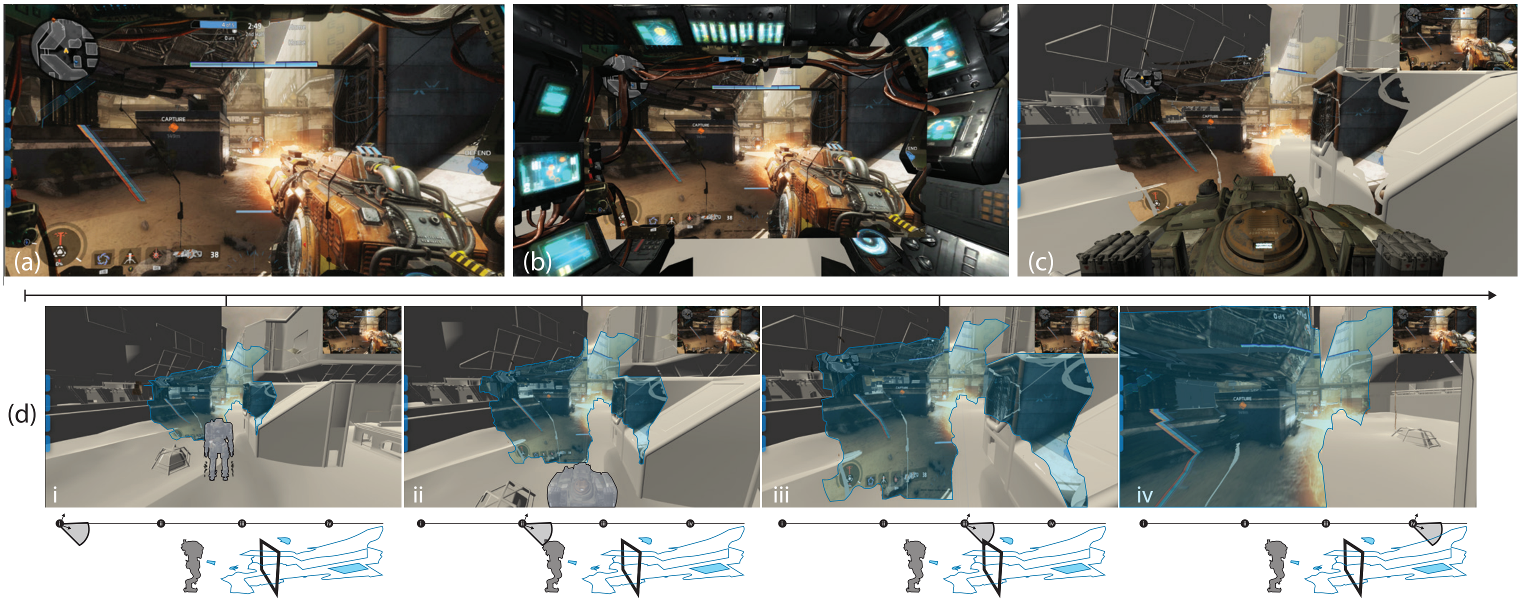


Figure 7.1: Enhanced videogame livestreaming examples for the game Titanfall 2: (a) a single RGB frame captured from the game; (b) a 3D projection of the frame inside the diegetic environment of the titan; (c) a 3D composite of the frame with a low fidelity environment model; (d) a series of frames captured as the spectator translates from behind the titan into the 3D projection.

design space spanning "screen space," "volumetric space," and "world space" for conventional 2D displays and 3D VR (Figure 7.1), and through a user study which shows that our approach is preferred to existing techniques.

In summary, we make the following contributions:

- A new streaming paradigm that leverages available 3D data from realtime gameplay to enable new ways for people to experience videogame streams;
- An end-to-end live-streaming system that demonstrates the approach is technically feasible, scalable, and generalizable;
- A remote study showing the effectiveness of our approach for 2D displays and VR.

7.1 RELATED WORK

In Chapter 2 we investigated a broad set of related work having to do with videogame spectatorship. These types of spectating experience can be broadly categorized as either collocated or remote. Here we look specifically at the system and techniques used to enhance spectatorship for remote and collocated audiences.

Systems that Enhance Spectatorship

Research has investigated ways in which an external non-VR user can view what another VR user is doing while in an immersive virtual environment. Silhouette Games [129] explores this through a mirror metaphor by compositing the mirror reflection of the VR user inside the videogame world for external viewers. ShareVR [76] uses Spatial Augmented Reality (SAR) to communicate what the players in VR are doing with the other collocated players in same room-scale experience. TransceiVR [254] explored communication between a VR and external user in the context of productivity applications. RealityCheck [84] used a reconstruction of the VR player's physical environment for communication with external users. Though our work builds off of the insights explored in these works, we specifically look at the inverse problem: spectating non-VR games remotely in VR and using a desktop.

Directly augmenting a head-mounted display (HMD) has been used for external communication across AR and VR. This has been explored through the direct placement of touchscreen displays onto the HMD [77] and through the attachment of small actuated pico projectors [88, 110, 262]. All of these systems specifically focus on how to bring context outside the virtual environment so external users can observe and interact without needing to be inside the same virtual space.

In contrast to exploring external non-VR spectatorship of VR users, is to spectate them while *in* VR. This has largely focused on live music concerts [118, 120] and live theatre [90]. Yakura and Goto looked at the individual audience member and their affective experience while inside a virtual concert event with others [278]. They proposed a machine-learning approach to synthesize audience movement when virtual concert attendance is minimal. Investigating multi-user collocated VR, Herscher et al. proposed a system and design hypotheses for enabling collective VR experiences for large theatre productions [93].

While there has been significant exploration of viewing VR users and for evaluating VR spectatorship experiences, little work has explored ways in which we can enhance current non-VR videogames for spectators in VR. The existing approaches are relegated to applications like BigScreen [13] and AlSpaceVR [1] that give the user a virtual place in which to watch different kinds of media on a 2D screen. In contrast, we explore a system and its uses for enhancing spectatorship for existing non-VR videogames.

7.2 ENHANCED VIDEOGAME SPECTATORSHIP

There are inherent differences between spectators and streamers in a livestreaming system, as the primary role of the streamer is to entertain their spectating audience and for the spectator to watch. What they watch is typically a 2D live video feed, where a front facing camera view of the streamer is overlaid on top of the main videogame content in a picture-in-picture arrangement. Additional graphical information is commonly composited into this arrangement

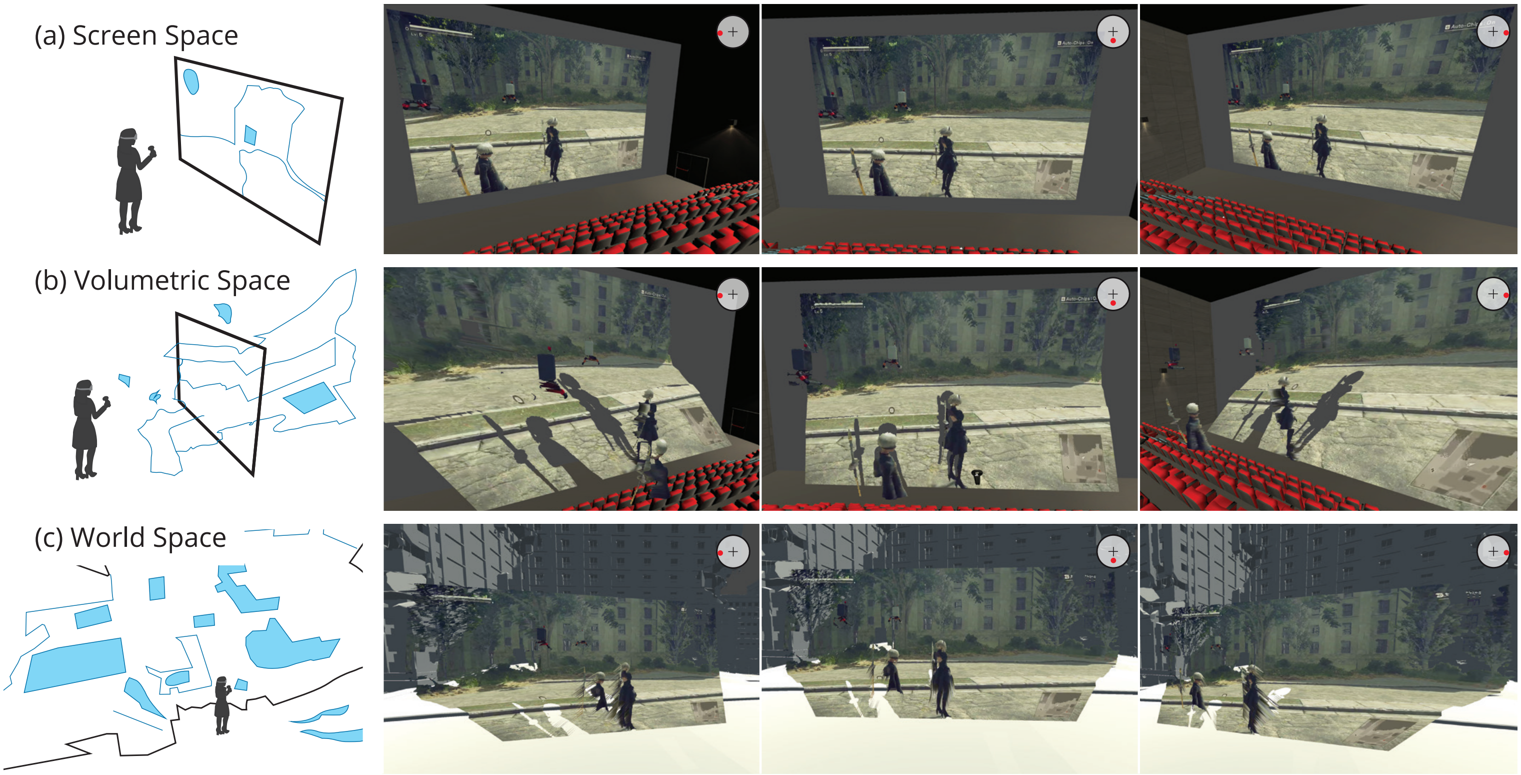


Figure 7.2: Spectator experience levels: (a) screen space view uses the 2D video frames from the stream; (b) volumetric view uses the depth data to provide a 3D effect with limited locomotion and interaction; (c) the world space view uses both the depth data and low-fidelity models from the game to create an environment that maximizes the spectator's locomotion and interaction capabilities.

to provide the spectators with information about the stream and to notify them about events. If we were to imagine an optimal form of videogame specatorship, the spectator would be immersed right into the videogame environment side-by-side with the streamer, where they could choose a vantage point, interact with the game world and the streamer, and be able to share their experience with other spectators in the real game space. This would require spectators to have access to a perfect realtime 3D reconstruction of the entire videogame environment. A simplified version of this is similar to the spectatorship modes common in competitive multiplayer videogames like Fortnite [68].

Though this would provide the best possible experience for the spectator, there are real-world implementation problems that make this practically impossible. Such an approach would require direct access to videogame assets, would need to provide tools for direct and indirect communication between streamers and spectators, and would need to handle both ambient and active modes of spectating. Considering this, it is important to identify trade-offs between immersion, agency, fidelity, and interaction to make real-world applications for enhanced videogame spectatorship effective and feasible.

Design Space

We consider the trade-offs associated with possible enhancements across two technical dimensions: (1) the *medium* used by the spectator, and (2) the amount of videogame data needed to enable an experience. We explore these dimensions in three discrete *immersion levels*: screen, volumetric, and world. These represent increasing amounts of videogame data to produce spectator experiences that vary in the amount of agency and control they have within the spectating system. Each of these levels can be generalized to two broad categories of mediums used by the spectator: 2D display (i.e. a desktop computer) and 3D immersion (i.e. VR). Figure 7.2 illustrates each level conceptually and with screen captures from our system.

SCREEN The screen space level can be considered the canonical 2D live streaming experience. On desktop, the output of the game is displayed on a flat 2D display, which is similar to existing methods used by Twitch [107] and YouTube [105]. Alternatively, the spectator could watch in VR on a large virtual cinema style screen. This is equivalent to existing experiences provided through applications like BigScreen [13].

VOLUMETRIC The volumetric space projects the incoming game data into a 3D environment to reconstruct parts of the game world for the spectator. This act of projection transforms the stream from the space of the screen into a separate virtual world that encapsulates it. Now, both the spectator and the stream occupy the same virtual space, where the spectator can act on the stream independently of the streamer producing it. This arrangement opens up new opportunities for spectating with additional interactive elements designed to take advantage of the virtual space containing the spectators and stream data. For example, setting the user inside a diegetic room where the projected videogame data is composited within it, or by allowing them to shoot orbs at the reconstructed geometry of the stream and have it react to the spectator's actions.

Conceptually, we can think of this shared environment as a liminal space that sits in between both the physical environment of the spectator and the virtual environment of the videogame. This gives a designer the freedom to think of this space as being separate from the videogame environment, where there is no narrative connection. Alternatively, it might be desirable to create deliberate connections between the space the spectator is in and the videogame environment. These diegetic spaces could be used to advance the story in interesting and novel ways outside the primary narrative.

WORLD The world space combines the 3D volumetric space with the positional and rotational information from the streaming game viewport. This provides not only the geometry from the game, but also where in the game this geometry is located. When combined together, new experiences can be created that place the spectator inside an approximation of the game being streamed. This gives the spectator the most agency over what they can do in the context of the videogame stream. For example, they now have the choice to follow along with the streamer as they play, or detach from the streamer to explore the areas around them.

An alternative to reconstructing the videogame environment at runtime is to utilize a low-fidelity 3D model of the videogame environment and composite the runtime 3D view on top of it. This type of configuration requires extra environmental information that is outside the current stream, but will also give the spectator extra context as to where they are in the videogame world and will effectively fill in information that could be lost when relying only on runtime reconstruction. One advantage of this is when multiple streamers are playing on the same map in a competitive battle royal or esports setting. A designer could tag specific spectator vantage points into the 3D environment to enable curated view such as a top-down view of all the players within the environment.

Each immersion level has advantages and disadvantages for the spectator. Later, we will evaluate each of these levels in a remote study to examine how they affect the viewer experience, however first we discuss the system infrastructure and technologies that enable these experiences.

7.3 SYSTEM ARCHITECTURE

Current live streaming pipelines can be broken into three broad phases: data acquisition, content distribution, and client-side playback. We make modifications to each of these in order to create a streaming architecture that is capable of extracting and transporting the additional data we use for our reconstruction and visualizations of 3D videogame streams. This allows us

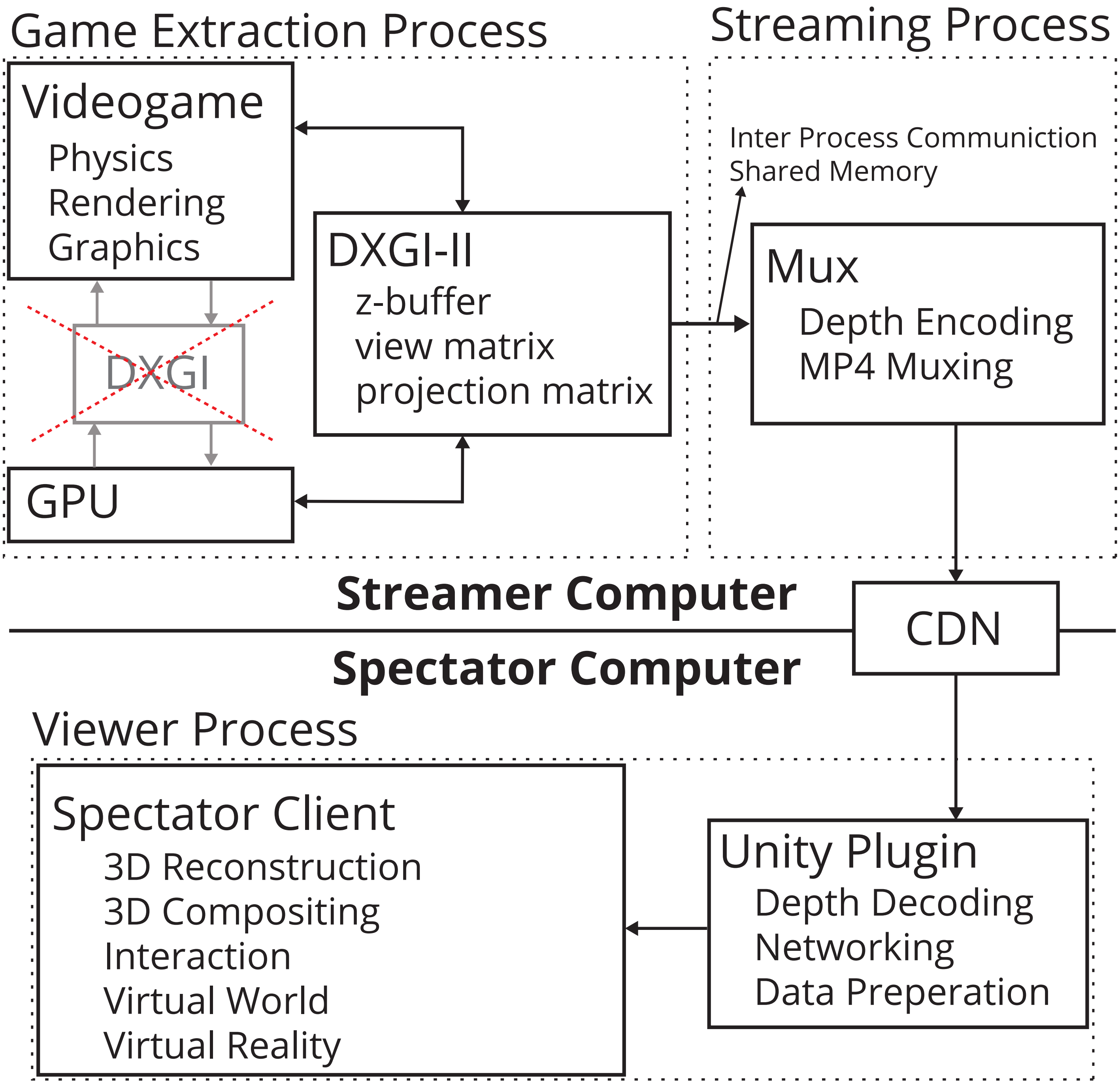


Figure 7.3: Overview of the system architecture.

to extract and transport the RGB and depth frame as well as the view and projection matrix of a game running on a Windows PC using DirectX 11. An overview of our architecture is in Figure 7.3. We describe each phase in detail next.

Data Acquisition

We take as a starting point the problem of extracting data from the rendering pipeline of a videogame. In Chapter 5, we showed how the OpenVR DLL (Dynamic Link Library) can be exploited to extract the z-buffer from a VR game [84]. We used a method that "hijacks" specific API calls, which is a general approach used in software analysis and reverse engineering [101]. However, this is limited to only OpenVR and fails to generalize to other games that do not bind to this specific protocol. Further, it is not clear how other videogame data, like the view or projection matrices, can be extracted through this higher-level technique.

To overcome this, we utilize the general idea of hijacking a DLL and extend it to work directly with the graphics API layer, bypassing higher level APIs like OpenVR. This allows us to directly intercept graphics data passed from the game to the GPU. We accomplish this be wrapping all DirectX 11 Graphics Interface (DXGI) definitions for all of `IDXGISwapChain`, `ID3D11Device`, and `ID3D11DeviceContext` interface classes, forcing the videogame process to link to our implementation of these APIs. This is visualized in Figure 7.3 as `DXGI-II`, and is an approach used within the game modding community through tools like Special K and Reshade that adjust stylistic aspects of the rendering pipeline [46, 179].

With a backdoor into the videogame rendering pipeline, we are able to build methods that extract the data we need to enable our novel spectator experiences. The data consists of RGB textures, depth textures, as well as view and projection matrices. Together, a single frame is composed of all four data types. We explain how each of these are extracted and bundled into a single frame next.

TEXTURE EXTRACTION Extracting the RGB texture can be obtained by copying the backbuffer associated with the graphic device swap chain before `IDXGISwapChain::Present` is called. However, extracting the z-buffer texture is more involved.

A fully featured videogame uses many different z-buffers during a render pass. Typically, a z-buffer is used to ensure sufficient object culling, but it is also used for other post-processing passes, like screen space ambient occlusion (SSAO) [8]. In a videogame scene, every virtual camera will produce a z-buffer. This includes not only the players' view, but any other view into the scene. For example, it is common for an in-game world map to be rendered using actual environment geometry from a separate camera pass.

Since we are interested in the player's view of the game environment, we search for the z-buffer that corresponds to the primary RGB texture of the main game view. We accomplish this by directly utilizing and expanding on the injection system presented within the Reshade post-processing framework [179] by analyzing incoming data during calls to `D3D11DeviceContext::Draw`. During each draw call, pointers to associated depth textures are cached along with simple statistics, like the number of draw calls, vertices rendered, and the texture dimensions. This is then used to choose a z-buffer that best corresponds to our target RGB texture. Alternatively, since we retain all the pointers to the z-buffers rendered during a frame, they can be displayed to the streamer through a simple interface overlaid on top of the game view. If the selection heuristics are wrong, the desired z-buffer can be manually selected. Note that this is typically a one time task at the start of a gameplay session or level.

MATRIX DATA EXTRACTION Getting both the view and projection matrices is a far more challenging task as there is no direct way to extract this data without source code access. To overcome this, we employ two approaches: shader reflections and constant buffer (`cbuffer`) analysis.

The shader reflection approach is composed of three steps: (1) parse the shader byte code; (2) look for shader variables that contain typical view and projection matrix naming conventions like `view` or `proj`; and (3) store an index and offset into the constant buffer containing the matrix data for fast recall later. During runtime, the index and offset for the constant buffer is used to copy the matrix data associated with that particular frame.

The second approach requires analysis of the constant buffer during runtime. For each frame, the system analyses incoming constant buffer data passing through `D3D11DeviceContext::VSSetConstantBuffers`. For each of these constant buffers, the underlying raw information is extracted and specific signatures associated with a view or projection matrix are searched for. For the view matrix, we use a set of heuristics to ensure the transformation is well formed. This includes checking whether it has a valid determinate, whether its rotation matrix is well formed, and whether the translation vector is within reasonable bounds. For the projection matrix, a similar approach is used that looks for common signatures found within the data. This includes the proper placement of coefficients, well formed focal length values, and reasonable near and far planes. If the data passes these checks, the index and offsets corresponding to each matrix are stored.

The collection of candidates is then visually presented to the streamer during initial setup, in which they are able to manually override the default selection to ensure the correct view and projection matrices are selected for streaming.

Data Preparation and Distribution

Once the data is extracted from the game, each frame will contain an RGB and z-buffer texture along with the view and projection matrices. Before the data can be distributed to remote spectators, it first needs to be transformed into a format with a sufficiently small memory profile suitable for streaming over the internet. We do this in a four step process: (1) get the raw uncompressed frame; (2) encode the RGB texture data using a H.264 video codec; (3) encode the depth data using a special codec; and (4) package all the data inside a customized MPEG-4 container.

TRANSFERRING DATA BETWEEN PROCESSES The streamer pipeline is composed of two processes. The first co-opts the videogame process and is responsible for the data extraction discussed above. The second process runs separately on the desktop and is tasked with preparing data for streaming. This is visualized by the line leading out of the *Game Extraction Process* into the *Streaming Process* in Figure 7.3. Keeping these separate has advantages. First it ensures that any issues in the data preparation process does not affect the streamer's gameplay. Second, it reduces computational overhead which can impact game performance. To transfer the data from one process to the other, we utilize an inter-process communication (IPC) bridge and shared virtual memory. This provides an efficient means to transfer data between the videogame process and the process used for stream preparation.

ENCODING RGB TEXTURES Methods to encode RGB textures are well understood as specific standards have been developed [146]. We use a lossy H.264 [263] codec for all encoding and decoding of colour texture data.

Encoding Depth Textures

The extracted depth texture (z-buffer) is typically composed of 32-bit pixels, where 24 bits represent the distances of objects in camera space and the remaining 8 bits are used as a stencil. Unlike RGB textures, there is no established method to efficiently encode depth data for streaming. Previous work has proposed methods that repurpose existing encoding technology to transform depth data into a suitable format for compression. However, these approaches are expensive to run [192] or have explicit assumptions on pixel bitness [144]. To overcome these limitations, we use a depth encoding technique that transforms the depth data into a double-helix colour space that makes it error-tolerant when compressed using the standard H.264 codec[1].

Data Multiplexing

Multiplexing video data typically consists of first encoding the images, audio, subtitles, and other data into an appropriate representation, and then placing the encoded data into a media container with metadata to describe the content. This step is visualized as the `Mux` container in Figure 7.3. For on-demand video and livestreaming, there are a number of media container formats that

[1] This refers to a set of works that are being actively researched and are not made publicly available as of the writing of this thesis. Look for new work post thesis titled: *A Double-Helix Colour-Space Transformation for Error-Tolerant Streaming of Depth Data in Video Codecs.*

are typically used, such as Webm [205], HLS [2], and MPEG-4 [106]. We use the MPEG-4 (.mp4) family of formats due to its flexibility and extensibility.

Unlike video files that contain only a video and audio track, our stream contains five data types: RGB, depth, view and projection matrices, and audio. This requires multiplexing more data than what a media container typically handles.

ENCAPSULATING VIDEOGAME DATA An MPEG-4 container file is composed of boxes called Atoms [252]. These define what type of data is contained within the the MPEG-4 container and how a media server should prepare that data for transport when playing files remotely. Each type of data is contained within an atom called a `trak`. This could be video, audio, subtitles, or something else. Associated with the `trak` atom are handlers (`hdlr`) that describe how the data within a `trak` atom is structured. This can include the type of encoding method used, framerate, and other metadata. This is then used by a media player to properly decode and transform the data for playback.

We define four MPEG-4 `trak`s that contain unique specifiers based on their data type. Two of these are dedicated to video content. The `trak` containing RGB video data data uses default MPEG-4 atoms. However, the `trak` containing the depth data needs to be identified during decoding in order to convert it from double-helix colour space back into depth space using the specialized depth encoding method described earlier.

The last two `trak`s contain the view and projection matrix data. These each consist of a compressed array of 64 bytes, representing the 4×4 matrix. We define two additional handler types, one for the view matrix (`vmtn`) and one of the projection matrix (`pmtn`). During the parsing and decoding process, we intercept these data packets in order to process the view and projection matrix separately from the video data. An advantage of encapsulating the view and projection matrices inside the MPEG-4 is that it guarantees synchronization between all data with little extra overhead.

7.4 SPECTATOR VIEWER

To view and interact with the live streaming content, we built a prototype spectator player that is capable of playing our modified MPEG-4 formatted stream from a remote media server. This is visualized as the `Spectator Client` in Figure 7.3. The streaming data can be rendered as either a 2D video, a 3D projection, or a 3D reconstruction of the environment being streamed. The rendering can additionally be targeted for a desktop or VR experience.

We use Unity 2019.4 LTS to implement the viewer application. This allows us to compose 3D objects and build out a user experience inside a game engine-like environment. However, all streaming and reconstruction functionality is contained in separate C++ libraries integrated into Unity through a plugin. This loose coupling means other editors or game engines could be used in the future.

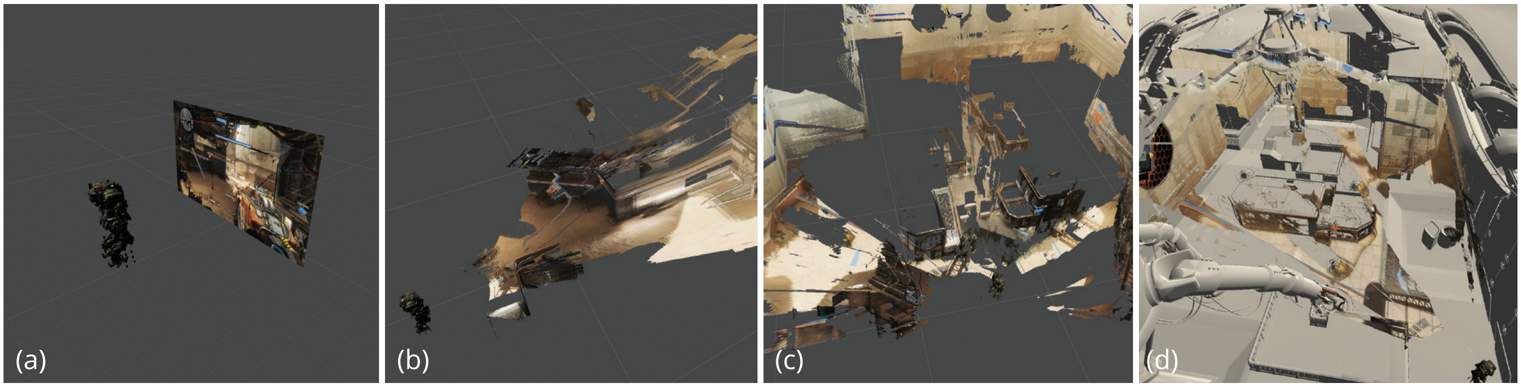


Figure 7.4: Visual representations: (a) screen, (b) volumetric; (c) reconstruction; and (d) world composite with low fidelity environment model. All frames captured from Titanfall 2.

Playback Engine

We connect to a remote media source through a custom media player with an API for media control and to access the raw decoded frames. This reads our enhanced MPEG-4 file either locally or from a uniform resource locator (URL). We use FFMpeg [62] for reading packets with a custom extension to delegate incoming `AVFrames` to specific routines for processing based on their underlying data and `hdlr` types embedded in metadata.

The RGB and depth video data types are decoded using the H.264 codec. For depth, additional decoding using the depth colour transformation method recovers the high-quality z-buffer texture. The two matrix data types are decoded using the LZ4 [47] compression algorithm. Together, all the decompressed data is composed into a single `DataFrame` in our library, then accessed on demand by any calling application. Within Unity, the `DataFrame` is processed and rendered to create the desired spectator experience.

Environment and Reconstruction

The data packaged by the playback engine allows us to create different visual representations of the video stream for the spectator. An overview of these can be seen in Figure 7.4.

A 2D representation of the stream is comparable to typical video streaming experiences seen on websites such as Twitch or Youtube. This type of video can be viewed on a desktop computer or can be viewed within a VR theatre-like environment. This kind of experience directly relates to the *screen space* immersion level discussed in our design space from Chapter 7.2 (Figure 7.4a).

By utilizing the depth data associated with the frame, a 3D projection of the current view can be generated (Figure 7.4b). The 3D projection is created using a single perspective into the video game environment based on the projection matrix extracted from the game. The reconstructed geometry of the view has a one-to-one correspondence with the geometry in the videogame. This corresponds to the *volumetric space* immersion level in our design space.

Utilizing all the data contained in the 3D video frame, a reconstruction of the videogame environment is possible (Figure 7.4c). This corresponds to the *world space* immersion level in our design space. We accomplish this by utilizing the view matrix data with the depth data. We assign each 3D projection a specific position and rotation within a 3D coordinate system. Each frame is then added to the previous, building up a static rendering of the environment as seen in the videogame stream. This is similar to what simultaneous location and mapping (SLAM) algorithms do to build up 3D representations of a physical space.

Further visualizations are possible by combining both the 3D video data with a low-fidelity model of the environment taken directly from the videogame (Figure 7.4d). The low-fidelity model is used as a backdrop from which the 3D projected mesh is composited directly on top of. This gives the spectator further context as to how the videogame environment is structured during a livestream. The model could be directly extracted from the game as an array of vertex buffers or extracted offline through data mining techniques.

The spectator can then be given differing levels of agency in how they interact and move around the environment. This can range from no control for the simple 2D video case to complete 6DoF control over their viewport in the reconstruction case. When full 6DoF control is given, the spectator is able to move around the environment to either follow the streamer or go and explore past reconstructed video frames.

Extensions and Enhancements

Additional experiences are possible by providing ways for the spectator to interact directly with the 3D projected videogame geometry. The viewer application allows the spectator to play along with the streamer by allowing the spectator to shoot orbs into the scene (Figure 7.5a). The orbs interact with the reconstructed videogame frame by causing an area of effect at the point of intersection, making the mesh glow brightly. Additionally, the spectator can add waypoints to the videogame environment that are decoupled from the streamer's current view (Figure 7.5b). The waypoints can be composted directly within the current view from the streamer or used to indicate to other spectators where points of interest are located in either the current frame or past ones. These extensions and enhancements are implemented in our viewer, but not tested in our user study.

7.5 USER STUDY

The goal of this study is to evaluate how differing levels of immersion of a videogame livestream can affect the experience of the viewer who watches it. We explore these effects across across two mediums: *desktop* and *VR*.

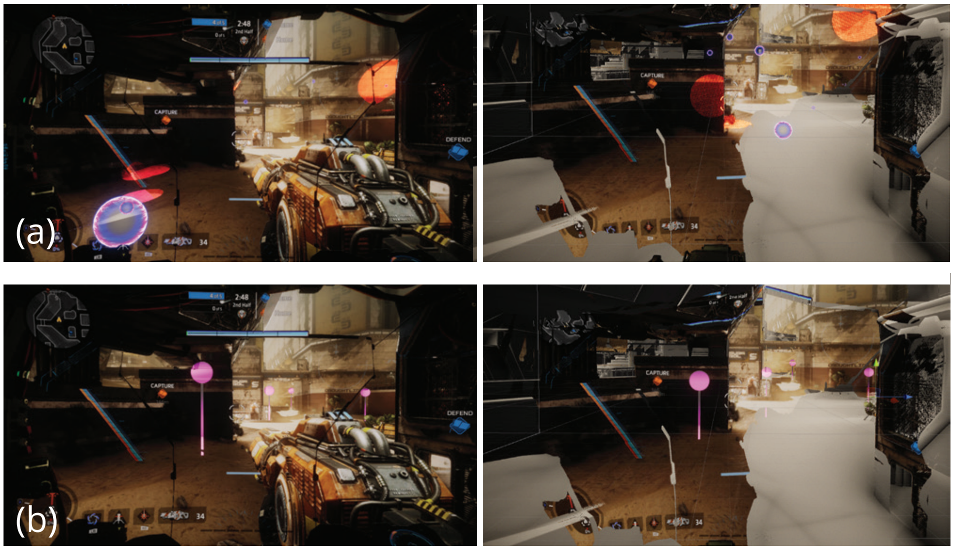


Figure 7.5: Spectator interaction with streaming geometry: (a) orbs shot into scene interact with geometry; (b) waypoint markers are placed to notify other spectators.

Levels of immersion differ in both the agency the user has while spectating the videogame stream and the amount of 3D data used in the experience. We evaluate the three levels of our design space: *screen*, *volumetric*, and *world*. At the lowest level is *screen* which consists of only a 2D RGB video feed of the videogame stream. The next level is *volumetric* which projects the videogame view into 3D space. At the highest level is *world* which utilizes the 3D videogame projection with with a low-fidelity environment to geometrically composite them into a unified experience.

Videogames were chosen to be representative of common gameplay genres. These consist of: *Titanfall 2*, a first-person shooter (FPS); *NieR Automata*, a third-person action role-playing game (RPG); and *Homeworld: Desert of Kharak*, a top-down real-time strategy game (RTS). An overview of the *world* immersion level in *VR* for each *videogame* type can be viewed in Figure 7.6.

Participants

We recruited 18 participants, ages 16 to 36, of which 2 were female and 16 male. Participants were recruited through online social media outlets including Reddit, Facebook, and Twitter. Each participant received $15 USD for successful completion of the study. Each participants used their own VR headset tethered to a desktop gaming PC. This included: 14 Oculus Quest 2 and 4 Oculus Rift. Internet speed across all participants averaged 133.9 Mbps ($\sigma = 206.6$). Geographical representation included participants from 2 continents: Europe and North America.

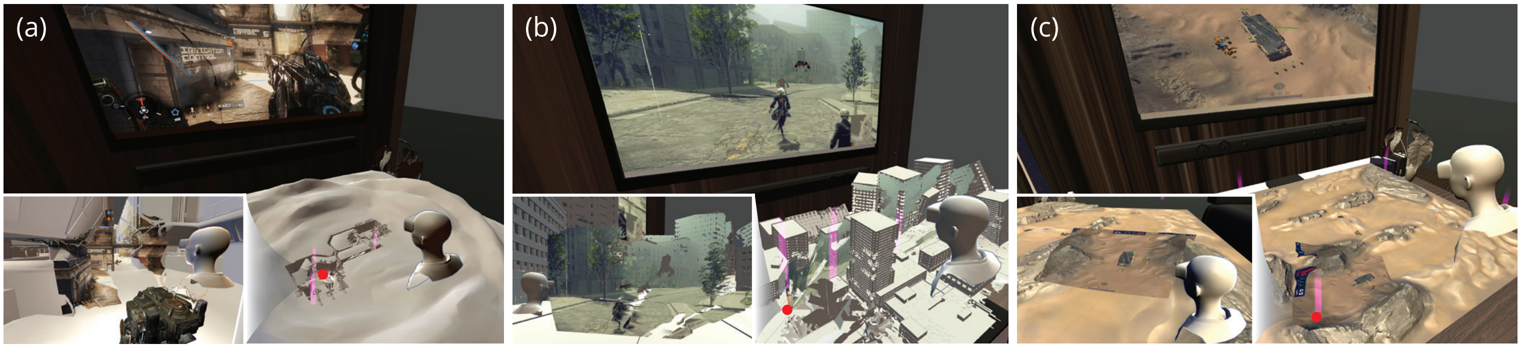


Figure 7.6: WORLD immersion for VR across each VIDEOGAME type: (a) Titanfall 2; (b) NieR:Automata; and (c) Homeworld: Desert of Kharak. The spectator can watch from above or teleport into the scene below, demonstrated by the picture-in-picture view.

Apparatus

A modified version of our spectator viewer (Section 7.4) is used with the participant's own gaming desktop computer and VR headset. Their computer needed to have at least an Intel i7 or AMD Ryzen 9 CPU, and at least an Nvidia GTX 1070 or AMD Radeon RX 580 GPU. We required a "tethered" VR headset to ensure consistent graphic fidelity across all participants. No headset was used in a standalone mode.

The spectatorship software accessed each 3D stream through a global content distribution network (CDN) provided through Amazon Web Services (AWS). Endpoints were distributed across all major continents, ensuring low latency and high bandwidth access to each of the 3D video files for the entire participant pool.

Procedure

For each participant, the study started with a 15 minute onboarding session to outline the experiment procedure and the participant's responsibilities. Then the participant used our spectator viewer to view a series of 45 to 60 second 3D streams of the game in different immersion and medium conditions. They watched 18 streams in total: 3 different immersion levels in 2 different mediums, each with 3 videogames. For desktop, the participant viewed the streams on their computer monitor and interacted using mouse and keyboard input. For VR, they watched the streams using a VR HMD with all interaction using the standard handheld controllers. The pacing and completion of each viewing was self-directed by the participant. Breaks in between were encouraged.

After completing a each stream, there was a survey with 6 preference questions. When all streams were complete, a final questionnaire captured final thoughts on their experiences across all conditions.

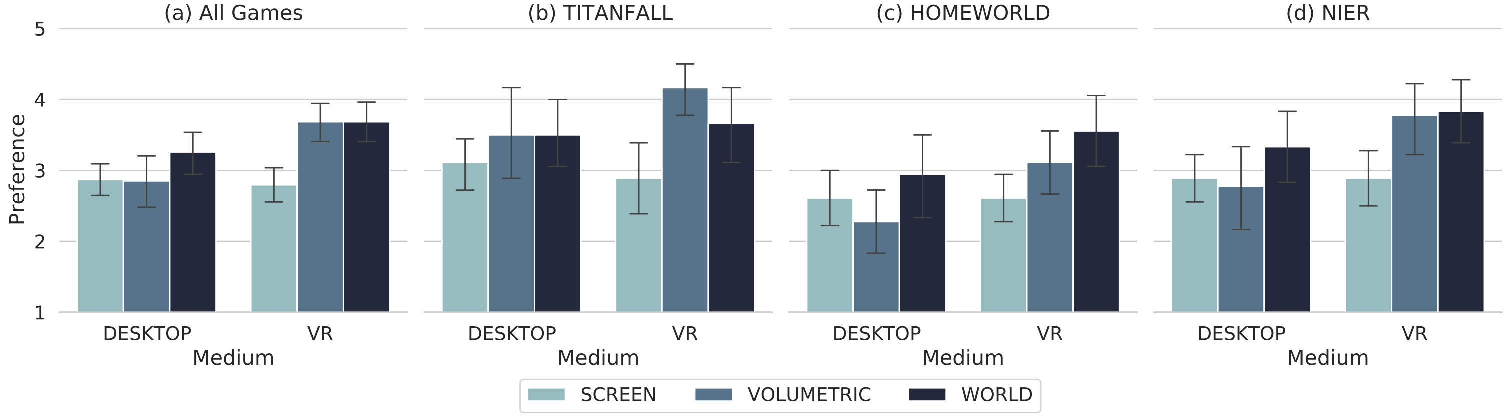


Figure 7.7: Overall preference ratings by (a) IMMERSION and MEDIUM and (b, c, d) VIDEOGAME type (error bars 95% CI).

Overall, the study was approximately 90 minutes: 15 minute onboarding, 60 minutes for stream evaluations, and 15 minutes for the closing questionnaire. The study had to be completed within 3 days from the onboarding interview.

Design

This is a within subjects design with two primary independent variables: MEDIUM with 2 levels (VR, DESKTOP); and IMMERSION with 3 levels (SCREEN, VOLUMETRIC, WORLD). VIDEOGAME, which consists of three levels (TITANFALL, NEIR, HOMEWORLD), form secondary independent variables. Each combination of MEDIUM and IMMERSION were repeated 3 times, one for each of the VIDEOGAME types. The combination of MEDIUM and IMMERSION were counter balanced using a Latin square. A random task order was used for VIDEOGAME.

The primary measures consisted of two subjective ratings asking if participants felt like they were immersed inside the videogame, and about their overall preference. Another, composite metric introduced by Venkatesh [260] was used to evaluate *perceived enjoyment*. This uses 4 separate questions to measure how much enjoyment the participant felt while watching the stream [232]. The composite metric was verified through factor analysis, verifying that each question contributed to the same measure ($\lambda = [3.47, 0.23, 0.16, 0.13]$). All measures are on a 5-point interval scale.

In summary: 2 MEDIUM × 3 IMMERSION × 3 VIDEOGAME = 18 data points per question per participant.

Results

Aligned Rank Transform (ART) [272] and post hoc pairwise ART-C [59] tests with Holm correction were used for all non-parametric preference measures. Figure 7.7 provides an overview of the results.

OVERALL PREFERENCE Across both mediums, participants preferred volumetric and world experiences over the baseline screen experience. There is a main effect of IMMERSION on overall user preference ($F_{2,304} = 15.3$, $p < 0.001$). Post hoc tests show that VOLUMETRIC ($\mu = 3.3$, $\sigma = 1.3$) and WORLD ($\mu = 3.5$, $\sigma = 1.1$) are both preferred over SCREEN ($\mu = 2.8$, $\sigma = 0.8$) irrespective of MEDIUM (all $p < 0.001$).

For medium type, participants preferred VR over desktop. There is a main effect of MEDIUM on overall user preference ($F_{1,305} = 11.2$, $p < 0.001$). A post hoc test shows VR ($\mu = 3.4$, $\sigma = 1.1$) is preferred over DESKTOP ($\mu = 3$, $\sigma = 1.2$) ($p < 0.001$).

For videogame type, participants preferred both third person NieR and first person Titanfall over Homeworld, the top down strategy game. There is a main effect of VIDEOGAME on overall user preference ($F_{2,304} = 12.7$, $p < 0.001$) Post hoc tests show that NIER ($\mu = 3.3$, $\sigma = 1.1$) and TITANFALL ($\mu = 3.5$, $\sigma = 1.2$) are preferred to HOMEWORLD ($\mu = 2.9$, $\sigma = 1.1$) (all $p < 0.005$). There is no significant difference between NIER and TITANFALL ($p = 0.09$).

Overall, participants preferred the world immersion level across both desktop and VR. There is an interaction between IMMERSION and MEDIUM on overall user preference ($F_{2,289} = 6.4$, $p < 0.002$). For VR, post hoc tests found that VOLUMETRIC ($\mu = 3.7$, $\sigma = 1.0$) and WORLD ($\mu = 3.7$, $\sigma = 1.1$) are preferred over SCREEN ($\mu = 2.8$, $\sigma = 0.9$) (all $p < 0.001$). No effect is reported between VOLUMETRIC and WORLD ($p = 1$). For DESKTOP, post hoc tests found WORLD ($\mu = 3.3$, $\sigma = 1.2$) to be preferred over SCREEN ($\mu = 2.9$, $\sigma = 0.9$) ($p < 0.045$). No other differences were found between any of the other IMMERSION types for DESKTOP (all $p > 0.4$).

FEELING IMMERSED INSIDE THE VIDEOGAME Participants felt more inside the videogame for both the volumetric and world immersion levels when compared with the baseline screen experience. There is a main effect of IMMERSION on the participant's affective experience of of being present within the videogame with the streamer ($F_{2,304} = 18.7$, $p < 0.001$). Post hoc tests show that both VOLUMETRIC ($\mu = 3.1$, $\sigma = 1.4$) and WORLD ($\mu = 3.2$, $\sigma = 1.3$) are more aligned with feeling inside the videogame than the baseline SCREEN ($\mu = 2.3$, $\sigma = 1.1$) (all $p < 0.001$). There is no significant difference between VOLUMETRIC and WORLD ($p = 0.61$).

For medium, participants felt more inside the videogame for VR when compared with desktop. There is a main effect of MEDIUM on the user's affectual experience of being inside the videogame ($F_{1,305} = 4.8$, $p < 0.03$). A post hoc test shows that participants felt more inside the videogame for VR ($\mu = 3$, $\sigma = 1.3$) when compared with DESKTOP ($\mu = 2.7$, $\sigma = 1.3$) ($p < 0.03$).

Overall, we found the world immersion level to be the most effective at evoking feelings of being in the game regardless of medium type. There is an interaction between IMMERSION and MEDIUM on overall feelings of being inside the game with the streamer ($F_{2,289} = 7.6$, $p < 0.001$). For VR, post hoc tests show that participants felt more inside the game for VOLUMETRIC ($\mu = 3.3$, $\sigma = 1.2$) and WORLD ($\mu = 3.6$, $\sigma = 1.2$) when compared with SCREEN ($\mu = 2.2$, $\sigma = 1.1$) (all $p < 0.001$). No other differences are observed between VOLUMETRIC and

WORLD ($p = 0.46$). For DESKTOP, post hoc tests show that WORLD ($\mu = 3.1$, $\sigma = 1.4$) felt more inside the game then SCREEN ($\mu = 2.4$, $\sigma = 1.1$) ($p < 0.003$). No other differences are observed for DESKTOP (all $p > 0.15$).

PERCEIVED ENJOYMENT Participants reported the most enjoyment from both the world and volumetric immersion levels over the baseline screen experience. There is a main effect of IMMERSION on perceived enjoyment ($F_{2,304} = 11.67$, $p < 0.001$). Post hoc test show that WORLD ($\mu = 3.4$, $\sigma = 1.1$) and VOLUMETRIC ($\mu = 3.2$, $\sigma = 1.3$) are preceived as being more enjoyable when compared with SCREEN ($\mu = 2.8$, $\sigma = 0.9$) (all $p < 0.001$). There is no significant difference between WORLD and VOLUMETRIC ($p = 0.19$).

Participants enjoyed the videogame experiences more in VR than they did on desktop. There is a significant effect of MEDIUM on perceived enjoyment ($F_{1,305} = 14.24$, $p < 0.001$). A post hoc test shows that VR ($\mu = 3.3$, $\sigma = 1.1$) is perceived more enjoyable when compared with DESKTOP ($\mu = 2.9$, $\sigma = 1.15$) ($p < 0.001$).

Overall, participants perceived the world immersion level as being the most enjoyable regardless of medium type. There is an interaction between IMMERSION and MEDIUM for perceived enjoyment ($F_{2,301} = 3.7$, $p < 0.03$). For VR, post hoc tests show that WORLD ($\mu = 3.6$, $\sigma = 1.0$) and VOLUMETRIC ($\mu = 3.6$, $\sigma = 1.1$) are perceived more enjoyable then SCREEN ($\mu = 2.8$, $\sigma = 1.0$) (all $p < 0.001$). There is no significant difference between WORLD and VOLUMETRIC ($p = 0.98$). For DESKTOP, post hoc tests show that WORLD ($\mu = 3.6$, $\sigma = 1.0$) is perceived as more enjoyable then SCREEN ($\mu = 2.8$, $\sigma = 1.0$) ($p < 0.045$). No other differences are observed (all $p > 0.44$).

7.6 DISCUSSION

We found compelling differences between the medium and immersion types in how they affected participant sentiments towards specific visualizations of the videogame streams. Overall, participants found watching 3D videogame streams to be beneficial to their overall enjoyment and immersion when compared to a 2D stream of the same videogame content regardless of medium.

The difference between a 2D and 3D stream was more apparent in VR than on desktop. Participants stated that the *"watching experience [was] greatly improved by the 3D reconstruction"* [P4], *"felt like [they were] in a 3D cinema"* [P2], and that it made them *"feel like [they were] playing along"* [P15] when spectating in VR. In contrast, viewing the 3D reconstruction on desktop was mixed. This is reflected in the lack of differences between immersion levels for desktop and in the participant's individual comments. Some did not see *"any value in adding false depth"* [P3] or thought that it did not provide *"any benefit on a monitor"* [P1] screen. However, some other participants felt it continued to make them feel *"like [they were] there with the streamer"* [P8] and that it was able to provide *"additional context to the game being played"* [P15], even when viewing on a desktop screen.

Some participants commented on the inherent limitations of the live-streaming system. Due to how we capture depth data from the videogame, sections of the scene will be occluded by objects directly in front of the camera. These are known as depth shadows. In total, 5 participants directly or indirectly made comments about these depth reconstruction artifacts. For example, they noted that the cutout from the *"gun"* [P14] in Titanfall 2 or the *"shadow"* [P3] created by 2B in NieR:Automata could sometimes be distracting. Other participants commented directly on the quality of 3D reconstruction, stating that the geometry could be *"spiky"* [P4] and that the image would become more distorted around complex geometry like trees [P2].

We reported an overall preference for Titanfall 2 and NieR:Automata over the videogame Homeworld: Deserts of Kharak. This may be due to two intrinsic qualities that Homeworld has that the other two videogames do not. The first of these being that it is a real-time strategy game (RTS) and the other is in how the virtual environment is rendered through the game's camera. Some participants stated their general dislike for RTS games in general, where they felt bored as they did not *"care about the subject"* [P8] matter presented to them. Other participants commented on the general 'flatness' of the scene due to the camera vantage point, stating that *"everything looks flat"* [P5] when viewing the 3D reconstruction and that the *"perspective and distance [made it] too hard to tell what the player's doing"* [P1]. The 'flatness' some observed is the result of positioning the camera very far from the game geometry, making the depth effect less pronounced. However, in contrast to this sentiment, some participants explicitly stated that Homeworld was their *"favorite way to view a stream"* [a]nd felt that it created a type of *"2.5D game"* [P17] experience.

Many participants suggested future directions and extensions to the system. Stating how this could work in an esports setting (4 participants) or having the ability to dynamically 'switch' between views would be beneficial (3 participants). In particular, [P3] suggested using the 3D reconstruction of the videogame *"as a replay environment [where you] could sort of pause and rewind, and move the camera to check out details"* [P3] in a dynamic setting.

A number of participants stated that they felt 'in' the game with the streamer when watching in 3D (8 participants). Commenting how it *"felt like I was in the game right behind the player"* [P12], *"felt like I was part of the battle"* [P10], and how the characters seemed *"larger than life [where] the action seemed to be particularly clearer and real as a result of the level of depth"* [P17]. This sentiment is also reflected in our reported results, where the overall effect of the 3D reconstruction had an impact on participant's feelings of being there with the streamer.

Across all videogame types, the volumetric 3D rendering experience had the most pronounced effect inside of VR with the exception of Titanfall 2, which saw a moderately positive increase on desktop as well. This could be due to the diegetic environment of the titan when rendering the 3D volumetric stream. In this scenario, the spectator view was actually inside the titan which may have contributed to the feelings of being more immersed in the experience. However, there was no interaction effect between immersion, medium, and videogame ($p = 0.89$) so no definite conclusions can be made.

Our study took place across two continents. This gave us the opportunity to test our infrastructure and system at scale. There are trade offs to this, one being that we did not have precise control over what equipment the participants use or network bandwidth and latency. However, we gained valuable insight as to extent and feasibility of deploying such a system in the wild. For the most part, participants did not report many issues related to network connectivity or reconstruction. Participants that did report issues found they were typically resolved once the CDN network cached packets closer to their physical location. As reported previously, a few participants commented on the reconstruction being "spiky." This can occur at times when the graphics shaders do not detect depth discontinuities properly in the depth buffer, which can result in geometry being generated where it should not be. Another possible explanation could be due to how the depth codec reconstructed the scene. At lower bandwidths, it would have to reconstruct more lost depth data which can affect visual fidelity.

Limitations and Future Work

While our system and infrastructure is adequate for the study we conducted, there are areas that could be refined.

DEPTH SHADOWS We capture the depth buffer directly from the videogame we stream. The advantage of this is that it gives us an exact replica of the geometry as it was rendered. However, the data from anything occluded during rendering will be lost causing “depth shadows.” As mentioned in our discussion, some participants commented on this. One possible solution is to use an array of virtual cameras in the game view to generate light field video [29]. However, this would require extra rendering passes per virtual camera in the array, which could affect the frame rate of the videogame. Another approach to consider is inpainting via neural irradiance fields to fill in missing geometry and pixels [275]. Both of these are interesting directions for future work.

REMOTE STUDY We conducted a distributed study across two continents with 18 participants. Though a remote study has its advantages like sampling from a wider participant pool with different setups and configurations, there are disadvantages in level of control over variables such as bandwidth and equipment, and the amount of supervision that can be reasonably given. Though we did attempt to normalize these variables across participants, they are harder to control when compared with an in-lab study.

VIDEOGAME VIGNETTES We sampled a set of videogames that were representative of three prominent game genres. The vignettes were prerecorded and streamed to participants in real-time. Our goal was to simulate a livestream on a technical level in a condensed form suitable for a within-subjects study to enable direct comparisons. However, this does not capture how an immersive

3D streaming experience might affect the bidirectional relationship and social dynamics between streamers and spectators. Conducting a more narrow study using the world immersion level with one game and one medium would be an interesting direction for future work.

7.7 SUMMARY

We presented a system and study that demonstrated the feasibility of capturing, encoding, transporting, and rendering immersive 3D streams for spectators to view on desktop or VR. A distributed study demonstrated our approach at scale, and the results show that immersive 3D streams enhance the overall spectator experience. In the future we plan to explore how our system can enhance the streamer to spectator relationship and how our system can be adapted to virtual tubing (VTubing) to leverage depth data and immersion for communication and entertainment.

8 CONCLUSION

In this thesis, we started out dissecting the high-level conceptual aspects of the virtual and then proceeded to explore the methods, systems, and techniques required to guide user interaction and attention between different states of reality. We moved through the Virtuality Continuum [169], from the physical, to the augmented, to augmented virtuality, and finally to the virtual. We interrogated the inbetweenness of these states, and explored them through the subjective viewpoint of the user in order to fullfill our high-level research objectives:

> *Investigate the systems, techniques, and interactions that guide the user's subjective experiences, awareness, and tasks between different states of reality.*

To achieve this, we looked at pointing and interaction techniques in SAR, the compositing and reconstruction of physical spaces for VR, a hybrid AR HMD with an actuated projector for blending realities, and the spectatorship of real-time videogame reconstructions in VR. What began in 1938 by Antonin Artaud who first described the theatre and its spectators as partaking in a type of virtual reality, so too does this thesis end in its exploration of the virtual with the simple act of spectatorship—a reconstructed videogame environment presented in a virtual world in a virtual theatre.

From the real to the virtual, and theatre to theatre; in this final chapter we provide a summary of work, discuss opportunities for future research, and make final conclusions.

Summary

The chapters of this thesis each explore a specific aspect of the Virtuality Continuum [169] in order to investigate the systems and techniques needed to enable new modes of interaction (Figure 1.2 and 1.1).

In chapter 3 (*AR*), we examined fundamental characteristics of device-based interaction in SAR: pointing at surface mapped targets. Our results show how the simplicity and speed of raycasting results in excellent performance for many situations, and how surprisingly versatile a simple method like directly tapping the phone to a target can be in many situations. Our results for our implementation of the viewport pointing method is mixed. However, having a reconstructed virtualized reality view did appear promise in certain circumstances, such as the overlaying of extra information on top of the environment. In the ad hoc realistic SAR setting of Experiment 1, the viewport could approach raycasting performance, but was never significantly better in the tested tasks. In the controlled and more restricted setting of Experiment 2, the viewport method outperformed raycasting for distant targets that were facing the user. Our conclusion is that each method has beneficial characteristics, and that depending on the expected SAR usage context, a hybrid method or mode-switching technique to switch between methods could be the best solution.

In chapter 4 $\boxed{R \longleftrightarrow AR}$, we proposed using the smartphone as a mediator to interact with virtual content that can transition from the real space of the smartphone to the augmented environment of SAR. We based this on an arm extension technique to seamlessly and intuitively transition the phone between mobile interaction and spatial interaction modes, guiding the user's attention from the physical world to one augmented through SAR. Our interaction technique enables the user to push smartphone content from their physical reality to an external SAR environment, interact with the external content, rotate-scale-translate it, and pull content back into the smartphone, all the while ensuring comfort, no conflict between the mobile and spatial interactions, and preserving single-handed eyes-free use in the spatial mode. To ensure feasibility of hand extension as mode switch, we evaluated the classification of extended and retracted states of the smartphone while varying user postures, surface distances, and target locations. Our results show that a random forest classifier can classify the extended and retracted states with a 96% accuracy on average. A final usability study of the interaction space with three demonstrative applications found interactions to be usable and intuitive.

In chapter 5 $\boxed{VR \longrightarrow R}$, we presented a system that builds on top of current VR rendering pipelines in order to extract the depth buffer from the videogame to enable 3D compositing with virtual environments. We demonstrated the capabilities of our system through a number of techniques that integrate the real world inside a virtual scene. We further demonstrated the utility of our methods by enabling collocated spectatorship of a user in VR through a view-dependent projection mapping of the virtual scene. A user study demonstrated that our system can enhance current VR environments by guiding user awareness between physical reality and virtual environments. This approach enables applications in safety and awareness as well as creating more meaningful VR experiences.

In chapter 6 $\boxed{AR \longleftrightarrow AR}$, we presented the concept of Augmented Augmented Reality for a wearable augmented reality HMD and an actuated head-mounted projector. We constructed a working hardware and software system, calibrated through a modified structure from motion algorithm and a novel optimization solver to reconstruct the kinematic chain and rotation axes of the actuators. Our Unity3D toolkit encapsulates a set of high-level functionality for the iteration of AAR experiences, and a developer study demonstrated its usability and general appeal for creating AAR applications. We then utilized this to propose a design space with demonstrations for how an AR HMD and projector can enhance the experience of the user using the device and non-augmented external viewers.

In chapter 7 $\boxed{VR \longleftrightarrow VR}$, we presented a system and study that demonstrated the feasibility of capturing, encoding, transporting, and rendering immersive 3D streams for spectators to view on desktop or VR. We explore how this extra data can be used to create reconstructions of the videogame scene, allowing users to be inside the environment with streamer. A distributed study demonstrated our approach at scale, and the results show that immersive 3D streams enhance the overall spectator experience.

Future Research

Based on the work presented in previous chapters, we believe there are many opportunities for future work in the areas of livestreaming and spectatorship, virtual tubing, rendering techniques, and additional studies. Here we discuss the most promising directions for future work.

using a semantic mapping of physical spaces for just-in-time opportunistic mapping on virtual environments In Chapter 5, we explored ideas around how a reconstruction of reality can be used within a videogame context to transition the user's attention between the virtual environment of the videogame to their physical environment. Even though we demonstrated the effectiveness of this approach for awareness, there remains an inherent misalignment between the virtual space the user sees and the physical space the user is acting within. This can occur when the 3D reconstruction of the physical world does not line up with the in-game geometry of the virtual environment. For example, a physical and virtual wall may not align with each other or the physical rendering of a couch may misalign with an interactive sofa in the videogame. Aspects of this problem have been explored within a limited capacity. For example, re-skinning the physical environment with virtual proxies has been explored in the context of rooms [231] and in large space at a lower alignment fidelity [239]. The human vision system, such as eye gaze and saccades, have also been used as means to change aspects of the virtual environment for infinite walking [242] and as a way to change elements directly in the scene [157]. Combinations of these approaches could be investigated to create an environment which opportunistically maps portions of the physical world onto segments of the virtual environment. Such a system would then be able to transform the virtual environment such that the user could interact with the virtual objects in that space through mapped physical proxies.

using light-fields to overcome depth shadows and discontinuities One of the limitations we reported in Chapter 7 revolved around the depth shadows and discontinuities left over from the rendering data we collected from the videogame. During our study, multiple participants reported or commented on how these type of artifacts can diminish their enjoyment when watching a videogame livestream. Unfortunately, the depth shadows and discontinuities are a direct result of how the game renders the scene, as the depth buffer captures the geometry facing the camera but nothing behind it. One possible solution to overcome this is to render the game using light fields.

Light fields are a promising direction of work as they capture more of the surrounding environment, which includes occluded region data. Current research has looked at light fields in the context of in situ real life [29, 166], but not in the context of videogame or virtual environments. However, there are clear limitations, especially when considering the number of required rendering passes needed to produce the data used for reconstruction. Other avenues of research worth investigating are hybrid systems, which delegate rendering between local and remote compute clusters [162], or deep learning approaches that utilize neural irradiance fields [274].

SPECTATORSHIP OF VIRTUALIZED PHYSICAL REALITY In Chapter 7, we proposed a system that enables spectators of videogame livestreams to be inside the game with the streamer. We demonstrated the feasibility of this by using the depth data generated during the rendering pass of the videogame to create a 3D view of the videogame scene. This was used as a way for the spectator to move around separately from the streamer's view. This could be considered a virtualisation of a virtual space, or an incomplete copy of an original. In earlier work, we explored this conceptually as a virtualized reality [117]. We used aspects of this in Chapter 3 for the viewport technique, in which the user interacts with the virtual content through a virtualized version of reality. We also used this more explicitly in Chapter 5, which utilized a complete live reconstruction of reality. One aspect we did not explore is the combination of these two things together. Such a combination would produce a framework in which a spectator would be able to move around a virtualized version of physical reality, uncoupled by singular points of view or constrained by cameras.

Existing work has explored aspects of this under the context of telepresence [127, 143] and remote collaboration [253]. JackIn Airsoft [128] investigated a limited form of this in the context of airsoft, but not specifically for the spectator experience or what interactions are meaningful. The challenge is to design a system that handles the many different modes of interaction that occur in a immersive environment like this and how to best present them to the spectators watching.

AN AUTOMATED MOBILE SPATIAL AUGMENTED REALITY DEVICE FOR AD HOC EVERYWHERE INTERACTION In Chapter 6, we explored the combination of SAR with an OST HMD in order to enhance the capabilities of both devices. Specifically, we utilized the projector as a way to expand the augmentation capabilities of the HMD and to communicate internal AR content with an outside observer. In both cases, the use of SAR proved to be beneficial for collaboration and communication. These benefits can also be seen in other works, such as RoomAlive [114], IllumiRoom [113], and Code Space [28]. These type of systems could be considered *outside-in* SAR, as all the cameras and projectors are statically mounted onto the walls or ceilings of a small room. In contrast, our augmented HMD could be considered *inside-out* SAR, as the camera and projector are mobile. Expanding this to encapsulate multiple projectors and cameras would be an interesting direction for future work. Such a system could be capable of providing 360° AR coverage, as well as dynamic and responsive interactive content for users. The challenge is to design a system that is robust to calibration and one that can easily reposition itself while maintaining interactive multi-user functionality.

Final Word

In the text above, we explored a range of systems, techniques, and methods that focus on specific aspects of virtuality and the utility that it can provide to the users of those systems. We show how these kinds of explorations can provide deeper insight into the design and exploration of these types of systems, and that there are specific challenges to system design when bridging aspects of the real and the virtual. We believe that our studies, systems, and implementations are valuable to designers, researchers, and companies that seek to further expand on the approaches needed to holistically bridge interactions, communication, and users across disparate parts of virtuality. We hope that our work will further motivate these types of explorations and help empower practitioners in making informed decisions when thinking about user experiences across realities.

BIBLIOGRAPHY

[1] AltspaceVR. *AltspaceVR | Be there, together*. https://altvr.com/. (Accessed on 05/06/2021). 2021.

[2] Apple. *HTTP Live Streaming (HLS) - Apple Developer*. https://developer.apple.com/streaming/. (Accessed on 09/01/2021). Sept. 2021.

[3] Antonin Artaud. *The Theater and its double*. Vol. 53. 9. 2013, pp. 1689–1699. ISBN: 9788578110796. DOI: 10.1017/CBO9781107415324.004. URL: http://katarze.mysteria.cz/artaud/theatre_its_double.pdf.

[4] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. "Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. New York, New York, USA: ACM Press, 2016, pp. 1968–1979. ISBN: 9781450333627. DOI: 10.1145/2858036.2858226. URL: http://dl.acm.org/citation.cfm?doid=2858036.2858226.

[5] Hugh Bailey. *Open Broadcaster Software | OBS*. https://obsproject.com/. (Accessed on 09/03/2021). Sept. 2021.

[6] 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*. New York, New York, USA: ACM Press, 2001, p. 31. ISBN: 158113438X. DOI: 10.1145/502348.502354. URL: http://portal.acm.org/citation.cfm?doid=502348.502354.

[7] Jean Baudrillard. *Simulacra and simulation*. University of Michigan press, 1994.

[8] Louis Bavoil and Miguel Sainz. "Screen space ambient occlusion." In: *NVIDIA developer information: http://developers. nvidia. com* 6 (2008).

[9] Michel Beaudouin-Lafon et al. "Multisurface Interaction in the WILD Room." In: *Computer* 45.4 (2012), pp. 48–56. ISSN: 0018-9162. DOI: 10.1109/MC.2012.110. URL: http://ieeexplore.ieee.org/document/6171141/.

[10] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995), pp. 289–300. ISSN: 00359246. URL: http://www.jstor.org/stable/2346101.

[11] Hrvoje Benko, Eyal Ofek, Feng Zheng, and Andrew D. Wilson. "FoveAR: Combining an Optically See-Through Near-Eye Display with Spatial Augmented Reality Projections." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*. New York, New York, USA: ACM Press, 2015, pp. 129–135. ISBN: 9781450337793. DOI: 10.1145/2807442.2807493. URL: http://doi.acm.org/10.1145/2807442.2807493.

[12] Hrvoje Benko, Andrew D. Wilson, and Federico Zannier. "Dyadic Projected Spatial Augmented Reality." In: *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST)* (2014), pp. 645–655. DOI: 10.1145/2642918.2647402. URL: http://dl.acm.org/citation.cfm?doid=2642918.2647402.

[13] Bigscreen. *Bigscreen*. https://www.bigscreenvr.com/. (Accessed on 05/02/2021). 2021.

[14] Mark Billinghurst, Adrian Clark, and Gun Lee. "A Survey of Augmented Reality." In: *Foundations and Trends® in Human–Computer Interaction* 8.2-3 (2015), pp. 73–272. ISSN: 1551-3955. DOI: 10.1561/1100000049.

[15] Mark Billinghurst and Hirokazu Kato. "Collaborative augmented reality." In: *Communications of the ACM* 45.7 (2002), pp. 64–70.

[16] Mark Billinghurst, Suzanne Weghorst, and Tom Furness. "Wearable computers for three dimensional CSCW." In: *Digest of Papers. First International Symposium on Wearable Computers*. IEEE Comput. Soc, 1997, pp. 39–46. ISBN: 0-8186-8192-6. DOI: 10.1109/ISWC.1997.629917.

[17] Oliver Bimber and Bernd Frohlich. "Occlusion shadows: using projected light to generate realistic occlusion effects for view-dependent optical see-through displays." In: *Proceedings. International Symposium on Mixed and Augmented Reality*. IEEE Comput. Soc, 2002, pp. 186–319. ISBN: 0-7695-1781-1. DOI: 10.1109/ISMAR.2002.1115088. URL: http://ieeexplore.ieee.org/document/1115088/.

[18] Oliver Bimber and Ramesh Raskar. *Spatial Augmented Reality Merging Real and Virtual Worlds*. Vol. 6. 2005, pp. 83–92. ISBN: 1568812302. DOI: 10.1260/147807708784640126. arXiv: arXiv:1011.1669v3.

[19] Richard A. Bolt. ""Put-that-there": Voice and gesture at the graphics interface." In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques - SIGGRAPH '80*. New York, New York, USA: ACM Press, 1980, pp. 262–270. ISBN: 0897910214. DOI: 10.1145/800250.807503.

[20] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. "Touch Projector: Mobile Interaction through Video." In: *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. CHI '10. New York, New York, USA: ACM Press, 2010, p. 2287. ISBN: 9781605589299. DOI: 10.1145/1753326.1753671. URL: http://doi.acm.org/10.1145/1753326.1753671.

[21] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. "Touch projector: mobile interaction through video." In: *SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. New York, New York, USA: ACM Press, 2010, pp. 2287–2296. ISBN: 9781605589299. DOI: 10.1145/1753326.1753671. URL: http://portal.acm.org/citation.cfm?doid=1753326.1753671.

[22] Sebastian Boring, Marko Jurmu, and Andreas Butz. "Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays." In: *Conference of the Australian Computer-Human Interaction Special Interest Group* (2009), pp. 161 –168. DOI: 10.1145/1738826.1738853.

[23] James Boritz and Kellogg S Booth. "A study of interactive 3D point location in a computer simulated virtual environment." In: *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '97* (1997), pp. 181–187. DOI: 10.1145/261135.261168. URL: http://portal.acm.org/citation.cfm?doid=261135.261168.

[24] Nick Bostrom. "Are we living in a computer simulation?" In: *The Philosophical Quarterly* 53.211 (2003), pp. 243–255.

[25] Doug A Bowman and Larry F Hodges. "An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments." In: *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*. New York, New York, USA: ACM Press, 1997, 35–ff. ISBN: 0897918843. DOI: 10.1145/253284.253301. URL: http://portal.acm.org/citation.cfm?doid=253284.253301.

[26] Doug A Bowman, Donald B Johnson, and Larry F Hodges. "Testbed evaluation of virtual environment interaction techniques." In: *Presence: Teleoperators and Virtual Environments* 10.1 (1999), pp. 26–33. ISSN: 1054-7460. DOI: 10.1145/323663.323667. URL: http://dl.acm.org/citation.cfm?id=323663.323667.

[27] George EP Box and David R Cox. "An analysis of transformations." In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1964), pp. 211–252.

[28] Andrew Bragdon, Rob DeLine, Ken Hinckley, and Meredith Ringel Morris. "Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings." In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*. Vol. 16. 4. New York, New York, USA: ACM Press, 2011, p. 212. ISBN: 9781450308717. DOI: 10.1145/2076354.2076393. URL: http://dl.acm.org/citation.cfm?doid=2076354.2076393.

[29] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. "Immersive light field video with a layered mesh representation." In: *ACM Transactions on Graphics* 39.4 (2020), p. 15. ISSN: 0730-0301. DOI: 10.1145/3386569.3392485. URL: https://doi.org/10.1145/3386569.3392485.

[30] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. "Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. New York, New York, USA: ACM Press, 2019, pp. 1–28. ISBN: 9781450359702. DOI: 10.1145/3290605.3300792. URL: http://dl.acm.org/citation.cfm?doid=3290605.3300792.

[31] Wolfgang Büschel, Annett Mitschick, Thomas Meyer, and Raimund Dachselt. "Investigating Smartphone-based Pan and Zoom in 3D Data Spaces in Augmented Reality." In: *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '19*. Vol. 19. New York, New York, USA: ACM Press, 2019, pp. 1–13. ISBN: 9781450368254. DOI: 10.1145/3338286.3340113. URL: https://doi.org/10.1145/3338286.3340113http://dl.acm.org/citation.cfm?doid=3338286.3340113.

[32] Andreas Butz, T. Hollerer, Steven Feiner, B. MacIntyre, and Clifford Beshers. "Enveloping Users and Computers in a Collaborative 3D Augmented Reality." In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. IEEE Comput. Soc, pp. 35–44. ISBN: 0-7695-0359-4. DOI: 10.1109/IWAR.1999.803804. URL: http://ieeexplore.ieee.org/document/803804/.

[33] Gerald A Byrd, Richard H and Schnabel, Robert B and Shultz. "A trust region algorithm for nonlinearly constrained optimization." In: *SIAM Journal on Numerical Analysis* 24 (1987), pp. 1152–1170.

[34] Jeffrey Cashion, Chadwick Wingrave, and Joseph J. Laviola. "Dense and dynamic 3D selection for game-based virtual environments." In: *IEEE Transactions on Visualization and Computer Graphics* 18.4 (2012), pp. 634–642. ISSN: 10772626. DOI: 10.1109/TVCG.2012.40.

[35] Jeffrey Cashion, Chadwick Wingrave, and Joseph J Laviola. "Optimal 3D selection technique assignment using real-time contextual analysis." In: *IEEE Symposium on 3D User Interface 2013, 3DUI 2013 - Proceedings*. 2013, pp. 107–110. ISBN: 9781467360975. DOI: 10.1109/3DUI.2013.6550205.

[36] Géry Casiez, Nicolas Roussel, and Daniel Vogel. "1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems." In: *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI '12* (2012), p. 2527. DOI: 10.1145/2207676.2208639. URL: http://dl.acm.org/citation.cfm?doid=2207676.2208639.

[37] David J. Chalmers. "The Virtual and the Real." In: *Disputatio* 9.46 (2017), pp. 309–352. ISSN: 0873-626X. DOI: 10.1515/disp-2017-0009.

[38] Liwei Chan and Kouta Minamizawa. "FrontFace: Facilitating communication between HMD users and outsiders using front-facing-screen HMDs." In: *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2017, pp. 1–5. ISBN: 9781450350754. DOI: 10.1145/3098279.3098548. URL: https://dl.acm.org/doi/10.1145/3098279.3098548.

[39] Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, Sebastian Boring, and Saul Greenberg. "Extending a mobile device's interaction space through body-centric interaction." In: *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12*. New York, New York, USA: ACM Press, 2012, p. 151. ISBN: 9781450311052. DOI: 10.1145/2371574.2371599. URL: http://dl.acm.org/citation.cfm?doid=2371574.2371599.

[40] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott Hudson. "Around-Body Interaction: Sensing & Interaction Techniques for Proprioception-Enhanced Input with Mobile Devices." In: *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services - MobileHCI '14*. New York, New York, USA: ACM Press, 2014, pp. 287–290. ISBN: 9781450330046. DOI: 10.1145/2628363.2628402. URL: http://dx.doi.org/10.1145/2628363.2628402.

[41] Lung-Pan Cheng, Li Chang, Sebastian Marwecki, and Patrick Baudisch. "iTurk: Turning Passive Haptics into Active Haptics by Making Users Reconfigure Props in Virtual Reality." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 89.

[42] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew Wilson. "Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. 2017, pp. 3718–3728. ISBN: 9781450346559. DOI: 10.1145/3025453.3025753. URL: http://dl.acm.org/citation.cfm?doid=3025453.3025753.

[43] Lung-Pan Cheng, Thijs Roumen, Hannes Rantzsch, Sven Köhler, Patrick Schmidt, Robert Kovacs, Johannes Jasper, Jonas Kemper, and Patrick Baudisch. "TurkDeck: Physical Virtual Reality Based on People." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15* (2015), pp. 417–426. DOI: 10.1145/2807442.2807463.

[44] Gifford Cheung and Jeff Huang. "Starcraft from the Stands: Understanding the Game Spectator." In: *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, 2011, p. 763. ISBN: 9781450302289. DOI: 10.1145/1978942.1979053.

[45] Elizabeth F Churchill, David N Snowdon, and Alan J Munro. *Collaborative virtual environments: digital places and spaces for interaction*. Springer Science & Business Media, 2012.

[46] Andon M. Coleman. *The Complete Guide to SK | Special K - The Official Wiki*. https://wiki.special-k.info/. (Accessed on 05/11/2021). 2021.

[47] Yann Collet. *LZ4 - Extremely fast compression*. https://lz4.github.io/lz4/. (Accessed on 09/04/2021). Sept. 2021.

[48] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A. DeFanti. "Surround-screen projection-based virtual reality: The design and implementation of the CAVE." In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques - SIGGRAPH '93*. New York, New York, USA: ACM Press, 1993, pp. 135–142. ISBN: 0897916018. DOI: 10.1145/166117.166134.

[49] Nguyen-Thong Dang. "A Survey and Classification of 3D Pointing Techniques." In: *2007 IEEE International Conference on Research, Innovation and Vision for the Future*. IEEE, 2007, pp. 71–80. ISBN: 1-4244-0694-3. DOI: 10.1109/RIVF.2007.369138. URL: http://ieeexplore.ieee.org/document/4223055/.

[50] Ben Delaney. *Sex Drugs and Tessellation: The truth about virtual reality as revealed in the pages of CybrEdge Journal*. 2014, pp. 179–184. ISBN: 9781500893293.

[51] J. Denavit and R.S. Hartenberg. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices." In: *Journal of Applied Mechanics* 22.2 (1955), pp. 215–221.

[52] Thomas G. Dietterich. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms." In: *Neural Computation* 10.7 (1998), pp. 1895–1923. ISSN: 0899-7667. DOI: 10.1162/089976698300017197. URL: http://www.mitpressjournals.org/doi/10.1162/089976698300017197.

[53] Andrew Dolce, Joshua Nasman, and Barbara Cutler. "ARmy: A study of multi-user interaction in spatially augmented games." In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 43–50. ISBN: 978-1-4673-1612-5. DOI: 10.1109/CVPRW.2012.6239198. URL: http://ieeexplore.ieee.org/document/6239198/.

[54] Mingsong Dou et al. "Fusion4D: Real-time Performance Capture of Challenging Scenes." In: *ACM Transactions on Graphics* 35.4 (2016), pp. 1–13. ISSN: 07300301. DOI: 10.1145/2897824.2925969. URL: http://dl.acm.org/citation.cfm?doid=2897824.2925969.

[55] John Downs, Frank Vetere, Steve Howard, Steve Loughnan, and Wally Smith. "Audience Experience in Social Videogaming: Effects of Turn Expectation and Game Physicality." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 3473–3482. ISBN: 9781450324731. DOI: 10.1145/2556288.2556965.

[56] Ruofei Du, Ming Chuang, Wayne Chang, Hugues Hoppe, and Amitabh Varshney. "Montage4D: Interactive Seamless Fusion of Multiview Video Textures." In: *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '18*. Vol. 11. New York, New York, USA: ACM Press, 2018, pp. 1–11. ISBN: 9781450357050. DOI: 10.1145/3190834.3190843. URL: https://doi.org/10.1145/3190834.3190843.

[57] Ruofei Du et al. "DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality." In: *UIST 2020 - Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Virtual Event, 2020, pp. 829–843. ISBN: 9781450375146. DOI: 10.1145/3379337.3415881. URL: http://dx.doi.org/10.1145/3379337.3415881.

[58] Lorin J. Elias and Deborah M. Saucier. "Neuropsychology : clinical and experimental foundations." In: (2006), p. 531.

[59] Lisa A Elkin, Matthew Kay, James J Higgins, and Jacob O Wobbrock. "An aligned rank transform procedure for multifactor contrast tests." In: *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '21)*. 2021. DOI: 10.5281/zenodo.594511. URL: https://dx.doi.org/10.5281/zenodo.594511.

[60] Katharina Emmerich, Andrey Krekhov, Sebastian Cmentowski, and Jens Krueger. "Streaming VR Games to the Broad Audience: A Comparison of the First-Person and Third-Person Perspectives." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14. ISBN: 9781450380966. DOI: 10.1145/3411764.3445515. arXiv: 2101.04449.

[61] *Ergonomics of human-system interaction – Part 411: Evaluation methods for the design of physical input devices*. Standard. Geneva, CH: International Organization for Standardization, May 2012.

[62] FFmpeg. *FFmpeg*. https://www.ffmpeg.org/. (Accessed on 08/06/2021). 2021.

[63] Andreas Rene Fender, Hrvoje Benko, and Andy Wilson. "MeetAlive: Room-Scale Omni-Directional Display System for Multi-User Content and Control Sharing." In: *Proceedings of the Interactive Surfaces and Spaces on ZZZ - ISS '17*. New York, New York, USA: ACM Press, 2017, pp. 106–115. ISBN: 9781450346917. DOI: 10.1145/3132272.3134117. URL: http://dl.acm.org/citation.cfm?doid=3132272.3134117.

[64] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. "Aperture based selection for immersive virtual environments." In: *Proceedings of the 9th annual ACM symposium on User interface software and technology - UIST '96*. New York, New York, USA: ACM Press, 1996, pp. 95–96. ISBN: 0897917987. DOI: 10.1145/237091.237105.

[65] Henry Fuchs, Andrei State, Etta D. Pisano, William F. Garrett, Gentaro Hirota, Mark Livingston, Mary C. Whitton, and Stephen M. Pizer. "Towards performing ultrasound-guided needle biopsies from within a head-mounted display." In: *Visualization in Biomedical Computing*. Ed. by Karl Heinz Höhne and Ron Kikinis. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 591–600. ISBN: 978-3-540-70739-4.

[66] Keinosuke Fukunaga and L. Hostetler. "The estimation of the gradient of a density function, with applications in pattern recognition." In: *IEEE Transactions on Information Theory* 21.1 (1975), pp. 32–40. ISSN: 0018-9448. DOI: 10.1109/TIT.1975.1055330.

[67] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. "FLARE: Fast layout for augmented reality applications." In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2014, pp. 207–212. ISBN: 978-1-4799-6184-9. DOI: 10.1109/ISMAR.2014.6948429.

[68] Epic Games. *Fortnite | Free-to-Play Cross-Platform Game - Fortnite*. https://www.epicgames.com/fortnite/en-US/home. (Accessed on 08/04/2021). 2021.

[69] Resolution Games. *Acron: Attack of the Squirrels!* https://www.resolutiongames.com/acron. (Accessed on 09/06/2021). 2019.

[70] Çağlar Genç, Shoaib Soomro, Yalçın Duyan, Selim Ölçer, Fuat Balcı, Hakan Ürey, and Oğuzhan Özcan. "Head Mounted Projection Display & Visual Attention: Visual attentional processing of head referenced static and dynamic displays while in motion and standing." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016, pp. 1538–1547. ISBN: 9781450333627. DOI: 10.1145/2858036.2858449. URL: https://dl.acm.org/doi/10.1145/2858036.2858449.

[71] Renaud Gervais, Jérémy Frey, and Martin Hachet. "Pointing in Spatial Augmented Reality from 2D Pointing Devices." In: ed. by Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler. Vol. 9299. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 381–389. ISBN: 978-3-319-22722-1. DOI: 10.1007/978-3-319-22723-8_30. URL: http://link.springer.com/10.1007/978-3-319-22723-8.

[72] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." In: *Machine Learning* 63.1 (2006), pp. 3–42. ISSN: 0885-6125. DOI: 10.1007/s10994-006-6226-1.

[73] René Glas. "Vicarious play: Engaging the viewer in Let's Play videos." In: *Empedocles: European Journal for The Philosophy of Communication* 5 (2015), pp. 81–86.

[74] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. "MultiFi: Multi-Fidelity Interaction with Displays On and Around the Body." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. Vol. 2015-April. New York, New York, USA: ACM Press, 2015, pp. 3933–3942. ISBN: 9781450331456. DOI: 10.1145/2702123.2702331. URL: http://dl.acm.org/citation.cfm?doid=2702123.2702331.

[75] Uwe Gruenefeld, Tim Claudius Stratmann, Lars Prädel, and Wilko Heuten. "MonoculAR: A radial light display to point towards out-of-view objects on augmented reality devices." In: *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI '18*. New York, New York, USA: ACM Press, 2018, pp. 16–22. ISBN: 9781450359412. DOI: 10.1145/3236112.3236115. URL: http://dl.acm.org/citation.cfm?doid=3236112.3236115.

[76] Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. "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: ACM, 2017, pp. 4021–4033. ISBN: 9781450346559. DOI: 10.1145/3025453.3025683. URL: https://dl.acm.org/doi/10.1145/3025453.3025683.

[77] Jan Gugenheimer, Evgeny Stemasov, Harpreet Sareen, and Enrico Rukzio. "FaceDisplay: Towards Asymmetric Multi-User Interaction for Nomadic Virtual Reality." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173628. URL: https://dl.acm.org/doi/10.1145/3173574.3173628.

[78] Karl Guttag. *Celluon Laser Beam Scanning Projector Technical Analysis – Part 1 – Karl Guttag on Technology*. https://www.kguttag.com/2015/06/01/celluon-laser-beam-scanning-projector-part-1/. (Accessed on 05/01/2020). 2020.

[79] Takumi Hamasaki, Yuta Itoh, Yuichi Hiroi, Daisuke Iwai, and Maki Sugimoto. "HySAR: Hybrid Material Rendering by an Optical See-Through Head-Mounted Display with Spatial Augmented Reality Projection." In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (2018), pp. 1457–1466. ISSN: 1077-2626. DOI: 10.1109/TVCG.2018.2793659. URL: https://ieeexplore.ieee.org/document/8260968/.

[80] 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*. New York, New York, USA: ACM Press, 2014, pp. 1315–1324. ISBN: 9781450324731. DOI: 10.1145/2556288.2557048. URL: http://dl.acm.org/citation.cfm?doid=2556288.2557048.

[81] Chris Hand. "A Survey of 3D Interaction Techniques." In: *Computer Graphics Forum* 16.5 (Dec. 1997), pp. 269–281. ISSN: 01677055. DOI: 10.1111/1467-8659.00194. URL: http://doi.wiley.com/10.1111/1467-8659.00194.

[82] Robert Hardy and Enrico Rukzio. "Touch & interact." In: *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services - MobileHCI '08*. New York, New York, USA: ACM Press, 2008, p. 245. ISBN: 9781595939524. DOI: 10.1145/1409240.1409267. URL: http://portal.acm.org/citation.cfm?doid=1409240.1409267.

[83] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. "OmniTouch: Wearable Multitouch Interaction Everywhere." In: *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. New York, New York, USA: ACM Press, 2011, p. 441. ISBN: 9781450307161. DOI: 10.1145/2047196.2047255. URL: http://dl.acm.org/citation.cfm?doid=2047196.2047255.

[84] Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D. Wilson. "RealityCheck: Blending Virtual Environments with Situated Physical Reality." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. New York, New York, USA: ACM Press, 2019, pp. 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300577. URL: http://dl.acm.org/citation.cfm?doid=3290605.3300577.

[85] Jeremy Hartmann, Hemant Bhaskar Surale, Aakar Gupta, and Daniel Vogel. "Using Conformity to Probe Interaction Challenges in XR Collaboration." In: *Workshop on Novel Interaction Techniques for Collaboration in VR. CHI'18* (2018).

[86] Jeremy Hartmann and Daniel Vogel. "An Evaluation of Mobile Phone Pointing in Spatial Augmented Reality." In: *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–6. ISBN: 9781450356213. DOI: 10.1145/3170427.3188535. URL: http://dl.acm.org/citation.cfm?doid=3170427.3188535.

[87] Jeremy Hartmann and Daniel Vogel. "An Evaluation of Mobile Phone Pointing in Spatial Augmented Reality." In: *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. Vol. 2018-April. New York, New York, USA: ACM Press, 2018, pp. 1–6. ISBN: 9781450356213. DOI: 10.1145/3170427.3188535. URL: http://dl.acm.org/citation.cfm?doid=3170427.3188535.

[88] 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*. New York, NY, USA: ACM, 2020, pp. 445–458. ISBN: 9781450375146. DOI: 10.1145/3379337.3415849. URL: https://dl.acm.org/doi/10.1145/3379337.3415849.

[89] Susan Hayward. *Cinema studies: the key concepts*. 2006, p. 586. ISBN: 0415367816.

[90] Linjia He, Hongsong Li, Tong Xue, Deyuan Sun, Shoulun Zhu, and Gangyi Ding. "Am I in the theater? Usability Study of Live Performance Based Virtual Reality." In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. New York, NY, USA: ACM, 2018, pp. 1–11. ISBN: 9781450360869. DOI: 10.1145/3281505.3281508.

[91] David Heaney. *HoloLens 2's Real Field Of View Revealed - UploadVR*. https://uploadvr.com/hololens-2-field-of-view/. (Accessed on 10/08/2021). 2019.

[92] Martin Heidegger. *Being and Time*. Trans. by John Macquarrie and Edward Robinson. Original, 1927. New York, NY: Harper & Row, 2011.

[93] Sebastian Herscher, Connor DeFanti, Nicholas Gregory Vitovitch, Corinne Brenner, Haijun Xia, Kris Layng, and Ken Perlin. "CAVRN: An exploration and evaluation of a collective audience virtual reality nexus experience." In: *UIST 2019 - Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2019, pp. 1137–1150. ISBN: 9781450368162. DOI: 10.1145/3332165.3347929. URL: https://dl.acm.org/doi/10.1145/3332165.3347929.

[94] Anuruddha Hettiarachchi and Daniel Wigdor. "Annexing Reality: Enabling Opportunistic Use of Everyday Objects as Tangible Proxies in Augmented Reality." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. New York, New York, USA: ACM Press, 2016, pp. 1957–1967. ISBN: 9781450333627. DOI: 10.1145/2858036.2858134. URL: http://dl.acm.org/citation.cfm?doid=2858036.2858134.

[95] Juan David Hincapié-Ramos, Kasim Ozacar, Pourang P. Irani, and Yoshifumi Kitamura. "GyroWand: IMU-based Raycasting for Augmented Reality Head-Mounted Displays." In: *Proceedings of the 3rd ACM Symposium on Spatial User Interaction - SUI '15*. August. New York, New York, USA: ACM Press, 2015, pp. 89–98. ISBN: 9781450337038. DOI: 10.1145/2788940.2788947. URL: http://dl.acm.org/citation.cfm?doid=2788940.2788947.

[96] Nicholas P. Holmes and Charles Spence. "The body schema and multisensory representation(s) of peripersonal space." In: *Cognitive Processing* 5.2 (2004), pp. 94–105. ISSN: 1612-4782. DOI: 10.1007/s10339-004-0013-3. URL: http://link.springer.com/10.1007/s10339-004-0013-3.

[97] Marcella Horrigan-Kelly, Michelle Millar, and Maura Dowling. "Understanding the Key Tenets of Heidegger's Philosophy for Interpretive Phenomenological Research." In: *International Journal of Qualitative Methods* 15.1 (2016). ISSN: 16094069. DOI: 10.1177/1609406916680634. URL: https://journals.sagepub.com/doi/full/10.1177/1609406916680634.

[98] Hong Hua, L.D. Brown, and Chunyu Gao. "Scape: Supporting Stereoscopic Collaboration in Augmented and Projective Environments." In: *IEEE Computer Graphics and Applications* 24.1 (2004), pp. 66–75. ISSN: 0272-1716. DOI: 10.1109/MCG.2004.1255811. URL: http://ieeexplore.ieee.org/document/1255811/.

[99] Hong Hua, Axelle Girardot, Chunyu Gao, and Jannick P Rolland. "Engineering of head-mounted projective displays." In: *Applied Optics* 39.22 (2000), p. 3814. ISSN: 0003-6935. DOI: 10.1364/AO.39.003814. URL: https://www.osapublishing.org/abstract.cfm?URI=ao-39-22-3814.

[100] Galen Hunt and Doug Brubacher. "Detours: Binary Interception of Win32 Functions." In: *Proceedings of the 3rd Conference on USENIX Windows NT Symposium - Volume 3*. WINSYM'99. Seattle, Washington: USENIX Association, 1999, pp. 14–14. URL: http://dl.acm.org/citation.cfm?id=1268427.1268441.

[101] Galen Hunt and Doug Brubacher. "Detours: Binary interception of Win32 functions." In: *3rd USENIX Windows NT Symposium* (1999). URL: http://research.microsoft.com/sn/detours.

[102] Wolfgang Hürst and Casper Van Wezel. "Gesture-based interaction via finger tracking for mobile augmented reality." In: *Multimedia Tools and Applications* 62.1 (2013), pp. 233–258. ISSN: 13807501. DOI: 10.1007/s11042-011-0983-y.

[103] Hikaru Ibayashi, Yuta Sugiura, Daisuke Sakamoto, Natsuki Miyata, Mitsunori Tada, Takashi Okuma, Takeshi Kurata, Masaaki Mochimaru, and Takeo Igarashi. "Dollhouse VR: A Multi-view, Multi-user Collaborative Design Workspace with VR Technology." In: *SIGGRAPH Asia 2015 Emerging Technologies*. New York, NY, USA: ACM, 2015, pp. 1–2. ISBN: 9781450339254. DOI: 10.1145/2818466.2818480. URL: https://dl.acm.org/doi/10.1145/2818466.2818480.

[104] LIV Inc. *LIV | Your VR capture toolbox*. https://www.liv.tv/. (Accessed on 05/02/2021). 2021.

[105] Youtube Inc. *YouTube Gaming*. https://www.youtube.com/gaming. (Accessed on 05/04/2021). 2021.

[106] *Information technology — Coding of audio-visual objects — Part 11: Scene description and application engine*. Standard. Geneva, CH: International Organization for Standardization, Nov. 2015.

[107] Twitch Interactive. *Twitch*. https://www.twitch.tv/. (Accessed on 05/04/2021). 2021.

[108] Richard H. Jacoby, Mark Ferneau, and Jim Humphries. "Gestural interaction in a virtual environment." In: *Stereoscopic Displays and Virtual Reality Systems* 2177.April 1994 (1994), pp. 355–364. DOI: 10.1117/12.173892.

[109] J. Jankowski and M. Hachet. "Advances in interaction with 3D environments." In: *Computer Graphics Forum* 34.1 (2015), pp. 152–190. ISSN: 14678659. DOI: 10.1111/cgf.12466.

[110] Pascal Jansen, Fabian Fischbach, Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. "ShARe: Enabling Co-Located Asymmetric Multi-User Interaction for Augmented Reality Head-Mounted Displays." In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2020, pp. 459–471. ISBN: 9781450375146. DOI: 10.1145/3379337.3415843. URL: http://dx.doi.org/10.1145/3379337.3415843.

[111] Jeroen Jansz and Lonneke Martens. "Gaming at a LAN event: the social context of playing video games." In: *New Media & Society* 7.3 (2005), pp. 333–355. ISSN: 1461-4448. DOI: 10.1177/1461444805052280. URL: http://journals.sagepub.com/doi/10.1177/1461444805052280.

[112] Seokhee Jeon and Seungmoon Choi. "Haptic Augmented Reality: Taxonomy and an Example of Stiffness Modulation." In: *Presence: Teleoperators and Virtual Environments* 18.5 (2009), pp. 387–408. ISSN: 1054-7460. DOI: 10.1162/pres.18.5.387. URL: http://www.mitpressjournals.org/doi/10.1162/pres.18.5.387.

[113] Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. "IllumiRoom: Peripheral Projected Illusions for Interactive Experiences." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. New York, New York, USA: ACM Press, 2013, p. 869. ISBN: 9781450318990. DOI: 10.1145/2470654.2466112. URL: http://dl.acm.org/citation.cfm?doid=2470654.2466112.

[114] Brett Jones, Lior Shapira, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, and Nikunj Raghuvanshi. "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* (2014), pp. 637–644. DOI: 10.1145/2642918.2647383. URL: http://dl.acm.org/citation.cfm?id=2647383.

[115] Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. "A Comparison of Ray Pointing Techniques for Very Large Displays." In: *Proceedings of Graphics Interface 2010*. GI '10. Ottawa, Ontario, Canada: Canadian Information Processing Society, 2010, pp. 269–276. ISBN: 978-1-56881-712-5. URL: http://dl.acm.org/citation.cfm?id=1839214.1839261.

[116] Daniel Kade, Kaan Akşit, Hakan Ürey, and Oğuzhan Özcan. "Head-mounted mixed reality projection display for games production and entertainment." In: *Personal and Ubiquitous Computing* 19.3-4 (2015), pp. 509–521. ISSN: 1617-4909. DOI: 10.1007/s00779-015-0847-y. URL: http://link.springer.com/10.1007/s00779-015-0847-y.

[117] Takeo Kanade, Peter Rander, and P.J. Narayanan. "Virtualized reality: constructing virtual worlds from real scenes." In: *IEEE Multimedia* 4.1 (1997), pp. 34–47. ISSN: 1070986X. DOI: 10.1109/93.580394.

[118] Tatsuyoshi Kaneko, Hiroyuki Tarumi, Keiya Kataoka, Yuki Kubochi, Daiki Yamashita, Tomoki Nakai, and Ryota Yamaguchi. "Supporting the sense of unity between remote audiences in VR-based remote live music support system KSA2." In: *Proceedings - 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2018*. 2019, pp. 124–127. ISBN: 9781538692691. DOI: 10.1109/AIVR.2018.00025.

[119] Dennis L. Kappen, Pejman Mirza-Babaei, Jens Johannsmeier, Daniel Buckstein, James Robb, and Lennart E. Nacke. "Engaged By Boos and Cheers: The Effect of Co-Located Game Audiences on Social Player Experience." In: *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*. New York, NY, USA: ACM, 2014, pp. 151–160. ISBN: 9781450330145. DOI: 10.1145/2658537.2658687.

[120] Shunichi Kasahara and Jun Rekimoto. "JackIn: Integrating first-person view with out-of-body vision generation for human-human augmentation." In: *ACM International Conference Proceeding Series*. Association for Computing Machinery, 2014. ISBN: 9781450327619. DOI: 10.1145/2582051.2582097.

[121] Bonifaz Kaufmann and Martin Hitz. "Eye-Shield: Protecting Bystanders from Being Blinded by Mobile Projectors." In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS '11. Kobe, Japan: Association for Computing Machinery, 2011, 31–34. ISBN: 9781450308717. DOI: 10.1145/2076354.2076359. URL: https://doi.org/10.1145/2076354.2076359.

[122] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira, and Chedy Raïssi. "Watch me Playing, I am a Professional: a First Study on Video Game Live Streaming." In: *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*. New York, New York, USA: ACM Press, 2012, p. 1181. ISBN: 9781450312301. DOI: 10.1145/2187980.2188259.

[123] Ryugo Kijima and Takeo Ojika. "Transition between virtual environment and workstation environment with projective head mounted display." In: *Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality*. IEEE Comput. Soc. Press, 1997, pp. 130–137. ISBN: 0-8186-7843-7. DOI: 10.1109/VRAIS.1997.583062. URL: http://ieeexplore.ieee.org/document/583062/.

[124] Daehwa Kim, Keunwoo Park, and Geehyuk Lee. "OddEyeCam: A Sensing Technique for Body-Centric Peephole Interaction Using WFoV RGB and NFoV Depth Cameras." In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2020, pp. 85–97. ISBN: 9781450375146. DOI: 10.1145/3379337.3415889. URL: http://dx.doi.org/10.1145/3379337.3415889.

[125] Pascal Knierim, Markus Funk, Thomas Kosch, Anton Fedosov, Tamara Müller, Benjamin Schopf, Marc Weise, and Albrecht Schmidt. "UbiBeam++: Augmenting interactive projection with head-mounted displays." In: *Proceedings of the 9th Nordic Conference on Human-Computer Interaction - NordiCHI '16*. Vol. 23-27-Octo. New York, New York, USA: ACM Press, 2016, pp. 1–6. ISBN: 9781450347631. DOI: 10.1145/2971485.2996747. URL: http://dl.acm.org/citation.cfm?doid=2971485.2996747.

[126] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. "Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–9. ISBN: 9781450356206. DOI: 10.1145/3173574.3173919. URL: https://doi.org/10.1145/3173574.3173919.

[127] Ryohei Komiyama, Takashi Miyaki, and Jun Rekimoto. "JackIn Space: Designing a Seamless Transition Between First and Third Person View for Effective Telepresence Collaborations Ryohei." In: *Proceedings of the 8th Augmented Human International Conference on - AH '17*. New York, New York, USA: ACM Press, 2017, pp. 1–9. ISBN: 9781450348355. DOI: 10.1145/3041164.3041183. URL: http://dl.acm.org/citation.cfm?doid=3041164.3041183.

[128] Michinari Kono, Takashi Miyaki, and Jun Rekimoto. "JackIn Airsoft: Localization and view sharing for strategic sports." In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*. Vol. Part F1319. Association for Computing Machinery, 2017. ISBN: 9781450355483. DOI: 10.1145/3139131.3139161.

[129] Andrey Krekhov, Daniel Preuß, Sebastian Cmentowski, and Jens Krüger. "Silhouette Games: An Interactive One-Way Mirror Approach to Watching Players in VR." In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. New York, NY, USA, 2020. ISBN: 9781450380744. DOI: 10.1145/3410404.3414247. URL: http://dx.doi.org/10.1145/3410404.3414247.

[130] Bernard C. Kress. *Optical Architectures for Augmented-, Virtual-, and Mixed-Reality Headsets*. Bellingham, Washington: SPIE PRESS, 2020, p. 270. ISBN: 9781510634336.

[131] Bernard C. Kress. "Waveguide Combiners." In: *Optical Architectures for Augmented-, Virtual-, and Mixed-Reality Headsets*. Bellingham, Washington: SPIE PRESS, 2020. Chap. 14, pp. 127–150. ISBN: 9781510634336.

[132] David M Krum, Evan A Suma, and Mark Bolas. "Augmented reality using personal projection and retroreflection." In: *Personal and Ubiquitous Computing* 16.1 (2012), pp. 17–26. ISSN: 1617-4909. DOI: 10.1007/s00779-011-0374-4. URL: http://link.springer.com/10.1007/s00779-011-0374-4.

[133] *Kuman 17Kg 270 Degree Metal Gear Digital Servo with U Bracket & Side mount for RC Robot Helicopter Airplane Car Boat KY72-1,Servos*. http://www.kumantech.com/kuman-17kg-270-degree-metal-gear-digital-servo-with-u-bracket-amp-side-mount-for-rc-robot-helicopter-airplane-car-boat-ky72-1_p0398.html. (Accessed on 04/12/2020). 2020.

[134] Jérémy Lacoche, Nico Pallamin, Thomas Boggini, and Jérôme Royan. "Improved Redirection with Distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments." In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology - VRST '17*. New York, New York, USA: ACM Press, 2017, pp. 1–9. ISBN: 9781450355483. DOI: 10.1145/3139131.3139142. URL: https://doi.org/10.1145/3139131.3139142.

[135] Ricardo Langner, Ulrich von Zadow, Tom Horak, Annett Mitschick, and Raimund Dachselt. "Content Sharing Between Spatially-Aware Mobile Phones and Large Vertical Displays Supporting Collaborative Work." In: *Collaboration Meets Interactive Spaces*. Cham: Springer International Publishing, 2016, pp. 75–96. DOI: 10.1007/978-3-319-45853-3_5.

[136] Chi-Jung Lee and Hung-Kuo Chu. "Dual-MR: Interaction with Mixed Reality Using Smartphones." In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology - VRST '18*. Vol. 18. New York, New York, USA: ACM Press, 2018, pp. 1–2. ISBN: 9781450360869. DOI: 10.1145/3281505.3281618. URL: https://doi.org/10.1145/3281505.3281618.

[137] Kenneth Levenberg. "A method for the solution of certain non-linear problems in least squares." In: *Quarterly of Applied Mathematics* 2.2 (1944), pp. 164–168. ISSN: 0033-569X. DOI: 10.1090/qam/10666. URL: http://www.ams.org/qam/1944-02-02/S0033-569X-1944-10666-0/.

[138] Ce Li, Chunyu Xie, Baochang Zhang, Chen Chen, and Jungong Han. "Deep Fisher discriminant learning for mobile hand gesture recognition." In: *Pattern Recognition* 77 (2018), pp. 276–288. DOI: 10.1016/j.patcog.2017.12.023.

[139] Frank Chun Yat Li, David Dearman, and Khai N Truong. “Virtual Shelves: Interactions with Orientation Aware Devices.” In: *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST ’09*. New York, New York, USA: ACM Press, 2009, p. 125. ISBN: 9781605587455. DOI: 10.1145/1622176.1622200. URL: http://portal.acm.org/citation.cfm?doid=1622176.1622200.

[140] Jie Li, Yiping Kong, Thomas Röggla, Francesca De Simone, Swamy Ananthanarayan, Huib de Ridder, Abdallah El Ali, and Pablo Cesar. “Measuring and Understanding Photo Sharing Experiences in Social Virtual Reality.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, pp. 1–14. ISBN: 9781450359702. DOI: 10.1145/3290605.3300897. URL: https://dl.acm.org/doi/10.1145/3290605.3300897.

[141] Jiandong Liang and Mark Green. “JDCAD: A highly interactive 3D modeling system.” In: *Computers & Graphics* 18.4 (1994), pp. 499–506. ISSN: 00978493. DOI: 10.1016/0097-8493(94)90062-0.

[142] Natan Linder. “LuminAR: a compact and kinetic projected augmented reality interface.” In: May 2003 (2011). URL: http://dspace.mit.edu/handle/1721.1/69803.

[143] 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*. New York, New York, USA: ACM Press, 2018, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173703.

[144] Yunpeng Liu, Stephan Beck, Renfang Wang, Jin Li, Huixia Xu, Shijie Yao, Xiaopeng Tong, and Bernd Froehlich. “Hybrid Lossless-Lossy Compression for Real-Time Depth-Sensor Streams in 3D Telepresence Applications.” In: *Advances in Multimedia Information Processing – PCM 2015*. Vol. 9314. Cham: Springer International Publishing, 2015, pp. 442–452. DOI: 10.1007/978-3-319-24075-6_43.

[145] Pedro Lopes, Ricardo Jota, and Joaquim A Jorge. “Augmenting touch interaction through acoustic sensing.” In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS ’11*. New York, New York, USA: ACM Press, 2011, p. 53. ISBN: 9781450308717. DOI: 10.1145/2076354.2076364. URL: http://dl.acm.org/citation.cfm?doid=2076354.2076364.

[146] Moving Picture Experts Group (MPEG). *MPEG – The Moving Picture Experts Group*. https://www.mpegstandards.org/. (Accessed on 05/13/2021). 2021.

[147] Mayra Donaji Barrera Machuca, Winyu Chinthammit, Yi Yang, and Henry Duh. “3D mobile interactions for public displays.” English. In: *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications on - SA ’14*. New York, New York, USA: ACM Press, 2014, pp. 1–4. ISBN: 9781450318914. DOI: 10.1145/2669062.2669074. URL: http://dl.acm.org/citation.cfm?doid=2669062.2669074.

[148] MagicLeap. *Spatial Computing for Enterprise | Magic Leap*. https://www.magicleap.com/. (Accessed on 02/10/2020). 2020.

[149] Christian Mai, Sarah Aragon Bartsch, and Lea Rieger. "Evaluating Shared Surfaces for Co-Located Mixed-Presence Collaboration." In: *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA: ACM, 2018, pp. 1–5. ISBN: 9781450365949. DOI: 10.1145/3282894.3282910. URL: https://doi.org/10.1145/3282894.3282910.

[150] Christian Mai, Mariam Hassib, and Ceenu George. *Like Elephants Do: Sensing Bystanders During HMD Usage*. Tech. rep. 2017.

[151] Christian Mai, Lukas Rambold, and Mohamed Khamis. "TransparentHMD: Revealing the HMD User's Face to Bystanders Introduction and Related Work." In: *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA: ACM, 2017, pp. 515–520. ISBN: 9781450353786. DOI: 10.1145/3152832.3157813. URL: https://dl.acm.org/doi/10.1145/3152832.3157813.

[152] Andrew Maimone, Xubo Yang, Nate Dierk, Andrei State, Mingsong Dou, and Henry Fuchs. "General-purpose telepresence with head-worn optical see-through displays and projector-based lighting." In: *2013 IEEE Virtual Reality (VR)*. IEEE, 2013, pp. 23–26. ISBN: 978-1-4673-4796-9. DOI: 10.1109/VR.2013.6549352. URL: http://ieeexplore.ieee.org/document/6549352/.

[153] Steve Mann. "Mediated Reality." In: *Linux J.* 1999.59es (Mar. 1999), 5–es. ISSN: 1075-3583.

[154] Steve Mann. "Mediated Reality." In: *Linux J.* 1999.59es (Mar. 1999). ISSN: 1075-3583. URL: http://dl.acm.org/citation.cfm?id=327697.327702.

[155] Donald W Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441. ISSN: 0368-4245. DOI: 10.1137/0111030. URL: http://www.jstor.org/stable/2098941.

[156] Sebastian Marwecki, Maximilian Brehm, Lukas Wagner, Lung-Pan Cheng, Florian 'Floyd' Mueller, and Patrick Baudisch. "VirtualSpace - Overloading Physical Space with Multiple Virtual Reality Users." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–10. ISBN: 9781450356206. DOI: 10.1145/3173574.3173815. URL: https://doi.org/10.1145/3173574.3173815.

[157] Sebastian Marwecki, Andrew D Wilson, Eyal Ofek, Mar Gonzalez Franco, and Christian Holz. "Mise-Unseen: Using Eye-Tracking to Hide Virtual Reality Scene Changes in Plain Sight." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2019, pp. 777–789. ISBN: 9781450368162. DOI: 10.1145/3332165.3347919. URL: https://dl.acm.org/doi/10.1145/3332165.3347919.

[158] Bernhard Maurer, Ilhan Aslan, Martin Wuchse, Katja Neureiter, and Manfred Tscheligi. "Gaze-Based Onlooker Integration: Exploring the In-Between of Active Player and Passive Spectator in Co-Located Gaming." In: *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. New York, NY, USA: ACM, 2015, pp. 163–173. ISBN: 9781450334662. DOI: 10.1145/2793107.2793126.

[159] John C. McClelland, Robert J Teather, and Audrey Girouard. "Haptobend: Shape-Changing Passive Haptic Feedback in Virtual Reality." In: *Proceedings of the 5th Symposium on Spatial User Interaction - SUI '17*. New York, New York, USA: ACM Press, 2017, pp. 82–90. ISBN: 9781450354868. DOI: 10.1145/3131277.3132179.

[160] Daniel C. McFarlane and Steven M. Wilder. "Interactive dirt: Increasing mobile work performance with a wearable projector-camera system." In: *Proceedings of the 11th international conference on Ubiquitous computing*. New York, NY, USA: ACM, 2009, pp. 205–214. ISBN: 9781605584317. DOI: 10.1145/1620545.1620577. URL: https://dl.acm.org/doi/10.1145/1620545.1620577.

[161] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. "A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. May. New York, NY, USA: ACM, 2015, pp. 2143–2152. ISBN: 9781450331456. DOI: 10.1145/2702123.2702382. URL: https://dl.acm.org/doi/10.1145/2702123.2702382.

[162] Morgan McGuire, Mike Mara, Derek Nowrouzezahrai, and David Luebke. "Real-time global illumination using precomputed light field probes." In: *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. New York, NY, USA: ACM, 2017, pp. 1–11. ISBN: 9781450348867. DOI: 10.1145/3023368.3023378. URL: http://dx.doi.org/10.1145/3023368.3023378.

[163] Metaverse. *HOME - Metaverse*. https://mvs.org/. (Accessed on 02/26/2021).

[164] *MicroVision Technology*. https://www.microvision.com/technology/. (Accessed on 05/01/2020). 2020.

[165] Microsoft. *Microsoft HoloLens | Mixed Reality Technology for Business*. https://www.microsoft.com/en-us/hololens. (Accessed on 02/10/2020). 2020.

[166] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines." In: *ACM Transactions on Graphics* 38.4 (2019). ISSN: 15577368. DOI: 10.1145/3306346.3322980. URL: https://doi.org/10.1145/3306346.3322980..

[167] Paul Milgram, Herman Colquhoun, et al. "A taxonomy of real and virtual world display integration." In: *Mixed reality: Merging real and virtual worlds* 1.1999 (1999), pp. 1–26.

[168] Paul Milgram and Fumio Kishino. "A Taxonomy of mixed reality visual displays." In: *IEICE Transactions on Information and Systems* 77.12 (1994), pp. 1321–1329.

[169] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. "Augmented reality: a class of displays on the reality-virtuality continuum." In: *Telemanipulator and Telepresence Technologies*. Ed. by Hari Das. Vol. 2351. 1995, pp. 282–292. DOI: 10.1117/12.197321.

[170] Alexandre Millette and Michael J. McGuffin. "DualCAD: Integrating Augmented Reality with a Desktop GUI and Smartphone Interaction." In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. IEEE, 2016, pp. 21–26. ISBN: 978-1-5090-3740-7. DOI: 10.1109/ISMAR-Adjunct.2016.0030. URL: http://ieeexplore.ieee.org/document/7836451/.

[171] Pranav Mistry and Pattie Maes. "SixthSense: A Wearable Gestural Interface." In: *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation*. SIGGRAPH ASIA '09. Yokohama, Japan: Association for Computing Machinery, 2009, p. 85. ISBN: 9781605588780. DOI: 10.1145/1665137.1665204. URL: https://doi.org/10.1145/1665137.1665204.

[172] Takashi Miyaki and Jun Rekimoto. "LiDARMAN: Reprogramming reality with egocentric laser depth scanning." In: *ACM SIGGRAPH 2016 Emerging Technologies, SIGGRAPH 2016*. Association for Computing Machinery, Inc, 2016. ISBN: 9781450343725. DOI: 10.1145/2929464.2929481.

[173] Alex Mohr and Michael Gleicher. "HijackGL: reconstructing from streams for stylized rendering." In: *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*. 2002, 13–ff.

[174] Lionel Moisan, Pierre Moulon, and Pascal Monasse. "Automatic homographic registration of a pair of images, with a contrario elimination of outliers." In: *Image Processing On Line* 2 (2012), pp. 56–73.

[175] David Molyneaux, Shahram Izadi, David Kim, Otmar Hilliges, Steve Hodges, Xiang Cao, Alex Butler, and Hans Gellersen. "Interactive Environment-aware Handheld Projectors for Pervasive Computing Spaces." In: *Proceedings of the 10th International Conference on Pervasive Computing*. Newcastle, UK: Springer-Verlag, 2012, pp. 197–215. DOI: 10.1007/978-3-642-31205-2_13. URL: http://link.springer.com/10.1007/978-3-642-31205-2_13.

[176] Jorge J Moré. *The Levenberg-Marquardt algorithm: implementation and theory*. Springer, 1978, pp. 105–116. URL: https://link.springer.com/content/pdf/10.1007/BFb0067700.pdf.

[177] Pierre Moulon, Pascal Monasse, and Renaud Marlet. "Adaptive Structure from Motion with a Contrario Model Estimation." In: *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*. Springer Berlin Heidelberg, 2012, pp. 257–270. ISBN: 9783642374463. DOI: 10.1007/978-3-642-37447-0_20.

[178] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. "Openmvg: Open multiple view geometry." In: *International Workshop on Reproducible Research in Pattern Recognition*. Springer. 2016, pp. 60–74.

[179] Patrick Mours. *Reshade*. https://reshade.me/. (Accessed on 09/01/2021). Sept. 2021.

[180] Raul Mur-Artal and Juan D. Tardos. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras." In: *arXiv* October (2016). arXiv: 1610.06475.

[181] Atsuo Murata and Hirokazu Iwase. "Extending Fitts' law to a three-dimensional pointing task." In: *Human Movement Science* 20 (2001), pp. 791–805.

[182] Brad A Myers, Choon Hong Peck, Jeffrey Nichols, Dave Kong, and Robert Miller. "Interacting at a Distance Using Semantic Snarfing." In: *LNCS*. Vol. 2201. Springer-Verlag, 2001, pp. 305–314. DOI: 10.1007/3-540-45427-6_26. URL: http://link.springer.com/10.1007/3-540-45427-6_26.

[183] N. Navab, A. Bani-Kashemi, and M. Mitschke. "Merging visible and invisible: two Camera-Augmented Mobile C-arm (CAMC) applications." In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. IEEE Comput. Soc, 1999, pp. 134–141. ISBN: 0-7695-0359-4. DOI: 10.1109/IWAR.1999.803814. URL: http://ieeexplore.ieee.org/document/803814/.

[184] James Newman. *Videogames*. Routledge, 2004, p. 192.

[185] Casey Newton. *Is Facebook cornering the VR market? - The Verge*. https://www.theverge.com/2021/6/16/22537795/is-facebook-cornering-the-vr-market. (Accessed on 09/26/2021). 7.

[186] P M Ngan and R J Valkenburg. *Calibrating a pan-tilt camera head*. Tech. rep. November. 2015, pp. 2–7. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.6.3733&rep=rep1&type=pdf.

[187] Oculus. *Oculus Rift: Step Into the Game by Oculus — Kickstarter*. https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game. (Accessed on 02/19/2021). 2013.

[188] Oculus. *Oculus*. https://www.oculus.com/. 2017.

[189] Peter Ondruska, Pushmeet Kohli, and Shahram Izadi. "MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones." In: *IEEE Transactions on Visualization and Computer Graphics* 21.11 (2015), pp. 1251–1258. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2459902. URL: http://ieeexplore.ieee.org/document/7165662/.

[190] Patrick Oswald, Jordi Tost, and Reto Wettach. "The real augmented reality." In: *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology - ACE '14*. New York, New York, USA: ACM Press, 2014, pp. 1–4. ISBN: 9781450329453. DOI: 10.1145/2663806.2663853.

[191] Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. "TractorBeam: Seamless integration of local and remote pointing for tabletop displays." In: *Proceedings - Graphics Interface*. 2005, pp. 33–40.

[192] Fabrizio Pece, Jan Kautz, and Tim Weyrich. "Adapting standard video codecs for depth streaming." In: *Joint Virtual Reality Conference of EGVE 2011 - The 17th Eurographics Symposium on Virtual Environments, EuroVR 2011 - The 8th EuroVR (INTUITION) Conference*. 2011, pp. 59–66. ISBN: 9783905674330. DOI: 10.2312/EGVE/JVRC11/059-066.

[193] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton. "Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments." In: *Proceedings - IEEE Virtual Reality*. IEEE, 2010, pp. 35–38. ISBN: 9781424462582. DOI: 10.1109/VR.2010.5444816. arXiv: NIHMS150003. URL: http://ieeexplore.ieee.org/document/5444816/.

[194] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. "Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment." In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*. New York, New York, USA: ACM Press, 2016, pp. 1714–1723. ISBN: 9781450335928. DOI: 10.1145/2818048.2819965. URL: http://dl.acm.org/citation.cfm?doid=2818048.2819965.

[195] Julian Petford, Miguel A. Nacenta, and Carl Gutwin. "Pointing All Around You: Selection Performance of Mouse and Ray-Cast Pointing in Full-Coverage Displays." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 533:1–533:14. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174107. URL: http://doi.acm.org/10.1145/3173574.3174107.

[196] Julian Petford, Miguel A Nacenta, and Carl Gutwin. "Pointing All Around You: Selection Performance of Mouse and Ray-Cast Pointing in Full-Coverage Displays." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–14. ISBN: 9781450356206. DOI:

10.1145/3173574.3174107. URL: http://dl.acm.org/citation.cfm?doid=3173574.3174107.

[197] Julian Petford, Miguel A Nacenta, Carl Gutwin, Joseph Eremondi, and Cody Ede. "The ASPECTA Toolkit: Affordable Full Coverage Displays." In: *Proceedings of the 5th ACM International Symposium on Pervasive Displays - PerDis '16*. New York, New York, USA: ACM Press, 2016, pp. 87–105. ISBN: 9781450343664. DOI: 10.1145/2914920.2915006. URL: http://dl.acm.org/citation.cfm?doid=2914920.2915006.

[198] Danakorn Nincarean Eh Phon, Mohamad Bilal Ali, and Noor Dayana Abd Halim. "Collaborative augmented reality in education: A review." In: *2014 International Conference on Teaching and Learning in Computing and Engineering*. IEEE. 2014, pp. 78–83.

[199] *PicoBit – Celluon Inc.* https://celluon.com/picobit/. (Accessed on 04/12/2020). 2020.

[200] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. "Image plane interaction techniques in 3D immersive environments." In: *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*. New York, New York, USA: ACM Press, 1997, 39–ff. ISBN: 0897918843. DOI: 10.1145/253284.253303.

[201] Claudio Pinhanez. "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 2201. Springer Verlag, 2001, pp. 315–331. ISBN: 3540426140. DOI: 10.1007/3-540-45427-6_27. URL: http://link.springer.com/10.1007/3-540-45427-6_27.

[202] Bridget Poetker. *A Brief History of Augmented Reality (+Future Trends & Impact)*. https://learn.g2.com/history-of-augmented-reality. (Accessed on 03/15/2021).

[203] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques." In: *Computer Graphics Forum* 17.3 (1998), pp. 41–52. DOI: 10.1111/1467-8659.00252.

[204] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. "The go-go interaction technique: non-linear mapping for direct manipulation in VR." In: *Proceedings of the 9th annual ACM symposium on User interface software and technology - UIST '96*. New York, New York, USA: ACM Press, 1996, pp. 79–80. ISBN: 0897917987. DOI: 10.1145/237091.237102.

[205] WebM Project. *The WebM Project | Welcome to the WebM Project*. https://www.webmproject.org/. (Accessed on 09/01/2021). Sept. 2021.

[206] Jana Rambusch, Anna - Sofia Alklind Taylor, and Tarja Susi. "A pre-study on spectatorship in eSports." In: *Spectating Play. 13TH ANNUAL GAME RESEARCH LAB SPRING SEMINAR*. April. 2017, pp. 24–25.

[207] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. "The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays." In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*. SIGGRAPH '98. New York, New York, USA: ACM Press, 1998, pp. 179–188. ISBN: 0897919998. DOI: 10.1145/280814.280861. URL: http://portal.acm.org/citation.cfm?doid=280814.280861.

[208] Ramesh Raskar, Greg Welch, and Henry Fuchs. *Spatially Augmented Reality*. Natick, MA, USA: A. K. Peters, Ltd., 1998, pp. 63–72. ISBN: 1-56881-098-9. URL: http://dl.acm.org/citation.cfm?id=322690.322696.

[209] Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. "Shader Lamps: Animating Real Objects With Image-Based Illumination." In: *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*. 2001, pp. 89–102. ISBN: 3211837094. DOI: 10.1007/978-3-7091-6242-2_9.

[210] Marguerite Reardon. *Augmented reality comes to mobile phones*. https://www.cnet.com/news/augmented-reality-comes-to-mobile-phones/. (Accessed on 02/26/2021). Sept. 2010.

[211] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 779–788. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.91. arXiv: 1506.02640. URL: http://ieeexplore.ieee.org/document/7780460/.

[212] Erik Reinhard, M. Adhikhmin, Bruce Gooch, and Peter Shirley. "Color transfer between images." In: *IEEE Computer Graphics and Applications* 21.4 (2001), pp. 34–41. ISSN: 02721716. DOI: 10.1109/38.946629. URL: http://ieeexplore.ieee.org/document/946629/.

[213] Jun Rekimoto. "Transvision: A hand-held augmented reality system for collaborative design." In: *Proc. Virtual Systems and Multimedia* December (1996), pp. 85–90. URL: https://www.researchgate.net/publication/228929153.

[214] Jun Rekimoto and Masanori Saitoh. "Augmented surfaces: a spatially continuous work space for hybrid computing environments." In: *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*. New York, New York, USA: ACM Press, 1999, pp. 378–385. ISBN: 0201485591. DOI: 10.1145/302979.303113. URL: http://portal.acm.org/citation.cfm?doid=302979.303113.

[215] Jeffrey M. Richter. *Programming Applications for Microsoft Windows with Cdrom*. 4th. Redmond, WA, USA: Microsoft Press, 1999. ISBN: 1572319968.

[216] Michael Rohs and Antti Oulasvirta. "Target acquisition with camera phones when used as magic lenses." In: *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*. New York, New York, USA: ACM Press, 2008, p. 1409. ISBN: 9781605580111. DOI: 10.1145/1357054.1357275. URL: http://portal.acm.org/citation.cfm?doid=1357054.1357275.

[217] Michael Rohs, Antti Oulasvirta, and Tiia Suomalainen. "Interaction with Magic Lenses: Real-World Validation of a Fitts' Law Model." In: *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, 2011, p. 2725. ISBN: 9781450302289. DOI: 10.1145/1978942.1979343.

[218] Joan Sol Roo and Martin Hachet. "One Reality: Augmenting How the Physical World is Experienced by combining Multiple Mixed Reality Modalities." In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17*. New York, New York, USA: ACM Press, 2017, pp. 787–795. ISBN: 9781450349819. DOI: 10.1145/3126594.3126638. URL: http://dl.acm.org/citation.cfm?doid=3126594.3126638.

[219] Enrico Rukzio and Paul Holleis. "Projector Phone Interactions: Design Space and Survey." In: *Workshop on coupled display visual interfaces at AVI* April (2016). URL: https://www.researchgate.net/publication/228370421.

[220] "Safety of laser products." In: *IEC TR60825-3*. International Electrotechnical Commission, 2008. Chap. Part 3: Guidance for laser displays and shows. URL: https://webstore.iec.ch/publication/3598.

[221] Behzad Sajadi and Aditi Majumder. "Autocalibration of Multiprojector CAVE-Like Immersive Environments." In: *IEEE Transactions on Visualization and Computer Graphics* 18.3 (2012), pp. 381–393. ISSN: 1077-2626. DOI: 10.1109/TVCG.2011.271. URL: http://ieeexplore.ieee.org/document/6060818/.

[222] Jean-Paul Sartre. *Being and Nothingness*. Éditions Gallimard, 1943, p. 628. ISBN: 0671867806.

[223] Vlad Savov. *Microsoft announces Windows Holographic with HoloLens headset - The Verge*. https://www.theverge.com/2015/1/21/7867593/microsoft-announces-windows-holographic. (Accessed on 02/19/2021). 2015.

[224] Dominik Schmidt, Julian Seifert, Enrico Rukzio, and Hans Gellersen. "A cross-device interaction style for mobiles and surfaces." In: *Proceedings of the Designing Interactive Systems Conference on - DIS '12*. New York, New York, USA: ACM Press, 2012, p. 318. ISBN: 9781450312103. DOI: 10.1145/2317956.2318005. URL: http://dl.acm.org/citation.cfm?doid=2317956.2318005.

[225] J Seifert, A Bayer, and E Rukzio. *PointerPhone: Using mobile phones for direct pointing interactions with remote displays*. English. 2013. DOI: 10.1007/978-3-642-40477-1_2.

[226] J Seifert, D Schneider, and E Rukzio. *Extending mobile interfaces with external screens*. English. Ulm University, Ulm, Germany, 2013. DOI: 10.1007/978-3-642-40480-1_50.

[227] Mickael Sereno, Lonni Besançon, and Tobias Isenberg. "Supporting Volumetric Data Visualization and Analysis by Combining Augmented Reality Visuals with Multi-Touch Input." In: (2019), pp. 16–18. DOI: 10.2312/eurp.20191136. URL: https://hal.inria.fr/hal-02123904.

[228] Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. "Gluey: Developing a Head-Worn Display Interface to Unify the Interaction Experience in Distributed Display Environments." In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15*. New York, New York, USA: ACM Press, 2015, pp. 161–171. ISBN: 9781450336529. DOI: 10.1145/2785830.2785838. URL: http://dx.doi.org/10.1145/2785830.2785838.

[229] Lior Shapira and Daniel Freedman. "Reality Skins: Creating Immersive and Tactile Virtual Environments." In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2016, pp. 115–124. ISBN: 978-1-5090-3641-7. DOI: 10.1109/ISMAR.2016.23. URL: http://ieeexplore.ieee.org/document/7781774/.

[230] Katie A Siek, Yvonne Rogers, and Kay H Connelly. "Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 3585 LNCS. 2005, pp. 267–280. ISBN: 3540289437. DOI: 10.1007/11555261_24.

[231] Adalberto L Simeone. "Substitutional reality: Towards a research agenda." In: *2015 IEEE 1st Workshop on Everyday Virtual Reality (WEVR)*. IEEE, 2015, pp. 19–22. ISBN: 978-1-4799-1725-9. DOI: 10.1109/WEVR.2015.7151690. URL: http://ieeexplore.ieee.org/document/7151690/.

[232] Max Sjöblom and Juho Hamari. "Why do people watch others play video games? An empirical study on the motivations of Twitch users." In: *Computers in Human Behavior* 75 (2017), pp. 985–996. ISSN: 07475632. DOI: 10.1016/j.chb.2016.10.019.

[233] David Woodruff Smith. "Phenomenology." In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford University, 2018.

[234] Ross T. Smith, Guy Webber, Maki Sugimoto, Michael Marner, and Bruce H. Thomas. "Automatic Sub-pixel Projector Calibration." In: *ITE Transactions on Media Technology and Applications* 1.3 (2013), pp. 204–213. ISSN: 2186-7364. DOI: 10.3169/mta.1.204. URL: http://jlc.jst.go.jp/DN/JST.JSTAGE/mta/1.204.

[235] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. "Semantic Scene Completion from a Single Depth Image." In: (2016). arXiv: 1611.08974. URL: http://arxiv.org/abs/1611.08974.

[236] Maximilian Speicher, Brian D Hall, and Michael Nebeling. "What is Mixed Reality?" In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Vol. 15. New York, NY, USA: ACM, 2019, pp. 1–15. ISBN: 9781450359702. DOI: 10.1145/3290605.3300767. URL: https://doi.org/10.1145/3290605.3300767.

[237] Martin Spindler, Wolfgang Büschel, and Raimund Dachselt. "Use Your Head: Tangible Windows for 3D Information Spaces in a Tabletop Environment." In: *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces* (2012), pp. 245–254. DOI: 10.1145/2396636.2396674. URL: http://doi.acm.org/10.1145/2396636.2396674.

[238] Martin Spindler and Raimund Dachselt. "PaperLens: Advanced Magic Lens Interaction Above the Tabletop." In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09*. New York, New York, USA: ACM Press, 2009, p. 1. ISBN: 9781605587332. DOI: 10.1145/1731903.1731948. URL: http://portal.acm.org/citation.cfm?doid=1731903.1731948.

[239] Misha Sra, Sergio Garrido-Jurado, and Chris Schmandt. "Procedurally generated virtual reality from 3D reconstructed physical space." In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology - VRST '16*. New York, New York, USA: ACM Press, 2016, pp. 191–200. ISBN: 9781450344913. DOI: 10.1145/2993369.2993372.

[240] Aaron Stafford, Wayne Piekarski, and Bruce Thomas. "Implementation of god-like interaction techniques for supporting collaboration between outdoor AR and indoor tabletop users." In: *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2006, pp. 165–172. ISBN: 1-4244-0650-1. DOI: 10.1109/ISMAR.2006.297809. URL: http://ieeexplore.ieee.org/document/4079271/.

[241] Bijan Stephen. *CodeMiko will see you now - The Verge*. https://www.theverge.com/22370260/codemiko-twitch-interview-stream-technician. (Accessed on 05/02/2021). 2021.

[242] Qi Sun, Arie Kaufman, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan Mcguire, and David Luebke. "Towards Virtual Reality Infinite Walking: Dynamic Saccadic Redirection." In: *ACM Transactions on Graphics* 37.4 (2018), pp. 1–13. ISSN: 07300301. DOI: 10.1145/3197517.3201294. URL: http://dl.acm.org/citation.cfm?doid=3197517.3201294.

[243] Qi Sun, Li-Yi Wei, and Arie Kaufman. "Mapping virtual and physical reality." In: *ACM Transactions on Graphics* 35.4 (2016), pp. 1–12. ISSN: 07300301. DOI: 10.1145/2897824.2925883. URL: http://dx.doi.org/10.1145/2897824.2925883.

[244] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. "Robust reconstruction of indoor scenes." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 5556–5565. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7299195.

[245] Ivan E Sutherland. "The Ultimate Display." In: *Proceedings of the IFIP Congress* (1965), pp. 506–508.

[246] Ivan E Sutherland. "A head-mounted three dimensional display." In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I)*. New York, New York, USA: ACM Press, 1968, p. 757. DOI: 10.1145/1476589.1476686.

[247] Z. Szalavári, D. Schmalstieg, A. Fuhrmann, and M. Gervautz. ""Studierstube": An environment for collaboration in augmented reality." In: *Virtual Reality* 3.1 (1998), pp. 37–48. ISSN: 14349957. DOI: 10.1007/BF01409796. URL: http://link.springer.com/10.1007/BF01409796.

[248] Robert J. Teather and Wolfgang Stuerzlinger. "Target pointing in 3D user interfaces." In: *CEUR Workshop Proceedings* 588 (2010), pp. 20–21. ISSN: 16130073. URL: http://ceur-ws.org/Vol-588/107.pdf.

[249] Robert J Teather and Wolfgang Stuerzlinger. "Pointing at 3D targets in a stereo head-tracked virtual environment." In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2011, pp. 87–94. ISBN: 978-1-4577-0063-7. DOI: 10.1109/3DUI.2011.5759222. URL: http://ieeexplore.ieee.org/document/5759222/.

[250] 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*. Vol. 2017-May. New York, NY, USA: ACM, 2017, pp. 1558–1570. ISBN: 9781450346559. DOI: 10.1145/3025453.3025813.

[251] Shan-Yuan Teng, Mu-Hsuan Chen, and Yung-Ta Lin. "Way Out: A Multi-Layer Panorama Mobile Game Using Around-Body Interactions." In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 230–233. ISBN: 9781450346566. DOI: 10.1145/3027063.3048410. URL: https://doi.org/10.1145/3027063.3048410.

[252] Emmanuel Thomas. *The 'MP4' Registration Authority*. https://mp4ra.org/. (Accessed on 05/18/2021).

[253] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. "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*. New York, NY, USA: ACM, 2019, pp. 161–174. ISBN: 9781450368162. DOI: 10.1145/3332165.3347872. URL: https://dl.acm.org/doi/10.1145/3332165.3347872.

[254] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Bjoern Hartmann. “TransceiVR: Bridging Asymmetrical Communication Between VR Users and External Collaborators.” In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2020, pp. 182–195. ISBN: 9781450375146. DOI: 10.1145/3379337.3415827. URL: http://dx.doi.org/10.1145/3379337.3415827.

[255] Wolfgang Tschauko. *VR Giants on Steam*. https://store.steampowered.com/app/1124160/VR_Giants/. (Accessed on 09/06/2021). Sept. 2021.

[256] John Underkoffler, Brygg Ullmer, and Hiroshi Ishii. “Emancipated Pixels: Real-World Graphics In The Luminous Room.” In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*. New York, New York, USA: ACM Press, 1999, pp. 385–392. ISBN: 0201485605. DOI: 10.1145/311535.311593. URL: http://dl.acm.org/citation.cfm?id=311535.311593.

[257] *Unity Real-Time Development Platform | 3D, 2D VR & AR Visualizations*. https://unity.com/. (Accessed on 04/14/2020). 2020.

[258] Martin Usoh, Ernest Catena, Sima Arman, and Mel Slater. “Using Presence Questionnaires in Reality.” In: *Presence: Teleoperators and Virtual Environments* 9.5 (2000), pp. 497–503. ISSN: 1054-7460. DOI: 10.1162/105474600566989. URL: http://www.mitpressjournals.org/doi/10.1162/105474600566989.

[259] Sandeep Vaishnavi, Jesse Calhoun, and Anjan Chatterjee. “Binding personal and peripersonal space: evidence from tactile extinction.” In: *Journal of Cognitive Neuroscience* 13.2 (2001), pp. 181–189.

[260] Viswanath Venkatesh. “Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model.” In: *Information systems research* 11.4 (2000), pp. 342–365.

[261] Daniel Vogel and Ravin Balakrishnan. “Distant freehand pointing and clicking on very large, high resolution displays.” In: *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology*. 2005, pp. 33–42. ISBN: 159593023X. DOI: 10.1145/1095034.1095041.

[262] Chiu-Hsuan Wang, Seraphina Yong, Hsin-Yu Chen, Yuan-Syun Ye, and Liwei Chan. “HMD Light: Sharing In-VR Experience via Head-Mounted Projector for Asymmetric Interaction.” In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2020, pp. 472–486. ISBN: 9781450375146. DOI: 10.1145/3379337.3415847. URL: http://dx.doi.org/10.1145/3379337.3415847.

[263] Thomas Wiegand, G.J. Sullivan, G. Bjontegaard, and Ajay Luthra. "Overview of the H.264/AVC Video Coding Standard." In: *IEEE Transactions on Circuits and Systems for Video Technology* 13.7 (2003), pp. 560–576. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2003.815165. URL: http://ieeexplore.ieee.org/document/1218189/.

[264] Andrew D. Wilson and Hrvoje Benko. "Combining multiple depth cameras and projectors for interactions on, above and between surfaces." In: *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10*. Figure 2. New York, New York, USA: ACM Press, 2010, p. 273. ISBN: 9781450302715. DOI: 10.1145/1866029.1866073. URL: http://portal.acm.org/citation.cfm?doid=1866029.1866073.

[265] Andrew D Wilson and Hrvoje Benko. "Projected Augmented Reality with the RoomAlive Toolkit." In: *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*. ACM. 2016, pp. 517–520.

[266] Andrew D. Wilson and Hrvoje Benko. "Holograms without Headsets: Projected Augmented Reality with the RoomAlive Toolkit." In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17*. New York, New York, USA: ACM Press, 2017, pp. 425–428. ISBN: 9781450346566. DOI: 10.1145/3027063.3050433.

[267] Andrew Wilson, Hrvoje Benko, Shahram Izadi, and Otmar Hilliges. "Steerable Augmented Reality with the Beamatron." 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. 413. ISBN: 9781450315807. DOI: 10.1145/2380116.2380169.

[268] Christian Winkler, Markus Löchtefeld, David Dobbelstein, Antonio Krüger, and Enrico Rukzio. "SurfacePhone: a mobile projection device for single- and multiuser everywhere tabletop interaction." English. In: *CHI '14 Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (2014), pp. 3513–3522. DOI: 10.1145/2556288.2557075. URL: http://dl.acm.org/citation.cfm?id=2556288.2557075.

[269] Christian Winkler, Julian Seifert, David Dobbelstein, and Enrico Rukzio. "Pervasive Information through Constant Personal Projection: The Ambient Mobile Pervasive Display (AMP-D)." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 4117–4126. ISBN: 9781450324731. DOI: 10.1145/2556288.2557365. URL: https://dl.acm.org/doi/10.1145/2556288.2557365.

[270] Wallace Witkowski. *Videogames are a bigger industry than movies and North American sports combined, thanks to the pandemic - MarketWatch*. https://web.archive.org/web/20210905093408/https://www.marketwatch.com/story/videogames-are-a-bigger-industry-

than-sports-and-movies-combined-thanks-to-the-pandemic-11608654990. (Accessed on 10/08/2021). 2021.

[271] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. “The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. 2011, pp. 143–146.

[272] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. “The aligned rank transform for nonparametric factorial analyses using only anova procedures.” In: *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI ’11*. New York, New York, USA: ACM Press, 2011, p. 143. ISBN: 9781450302289. DOI: 10.1145/1978942.1978963. URL: http://faculty.washington.edu/wobbrock/art/.

[273] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. “Gesture Recognition with a 3-D Accelerometer.” In: *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing*. Vol. 5585. 2009, pp. 25–38. DOI: 10.1007/978-3-642-02830-4_4.

[274] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. “Space-time Neural Irradiance Fields for Free-Viewpoint Video.” In: (2020). arXiv: 2011.12950. URL: https://video-nerf.github.iohttp://arxiv.org/abs/2011.12950.

[275] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. “Space-time Neural Irradiance Fields for Free-Viewpoint Video.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9421–9431.

[276] R Xiao, S Hudson, and C Harrison. “CapCam: Enabling quick, ad-hoc, position-tracked interactions between devices.” English. In: *11th Annual ACM International Conference on Interactive Surfaces and Spaces, ISS 2016*. Association for Computing Machinery, Inc, 2016, pp. 169–178. ISBN: 9781450342483 (ISBN). DOI: 10.1145/2992154.2992182.

[277] Shihui Xu, Bo Yang, Boyang Liu, Kelvin Cheng, Soh Masuko, and Jiro Tanaka. “Sharing Augmented Reality Experience Between HMD and Non-HMD User.” In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11570 LNCS. Springer International Publishing, 2019, pp. 187–202. DOI: 10.1007/978-3-030-22649-7_16. URL: http://dx.doi.org/10.1007/978-3-030-22649-7_16.

[278] Hiromu Yakura and Masataka Goto. “Enhancing Participation Experience in VR Live Concerts by Improving Motions of Virtual Audience Avatars.” In: *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2020, pp. 555–565. ISBN: 978-1-7281-8508-8. DOI: 10.1109/ISMAR50242.2020.00083. URL: https://skybox.xyz/https://ieeexplore.ieee.org/document/9284728/.

[279] Shuntaro Yamazaki, Masaaki Mochimaru, and Takeo Kanade. "Simultaneous self-calibration of a projector and a camera using structured light." In: *CVPR 2011 WORKSHOPS*. IEEE, 2011, pp. 60–67. ISBN: 978-1-4577-0529-8. DOI: 10.1109/CVPRW.2011.5981781. URL: http://ieeexplore.ieee.org/document/5981781/.

[280] Ka Ping Yee. "Peephole Displays: Pen interaction on spatially aware handheld computers." In: *Conference on Human Factors in Computing Systems - Proceedings*. 2003, pp. 1–8.

[281] Yiwei Zhao and Sean Follmer. "A Functional Optimization Based Approach for Continuous 3D Retargeted Touch of Arbitrary, Complex Boundaries in Haptic Virtual Reality." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–12. ISBN: 9781450356206. DOI: 10.1145/3173574.3174118. URL: https://doi.org/10.1145/3173574.3174118.

[282] Jianlong Zhou, Ivan Lee, Bruce Thomas, Roland Menassa, Anthony Farrant, and Andrew Sansome. "In-Situ Support for Automotive Manufacturing Using Spatial Augmented Reality." In: *International Journal of Virtual Reality* 11.1 (2012), pp. 33–41. ISSN: 1081-1451. DOI: 10.20870/IJVR.2012.11.1.2835. URL: https://ijvr.eu/article/view/2835.