

© 2015 Brett R. Jones

CONTENT CREATION FOR SEAMLESS AUGMENTED EXPERIENCES
WITH PROJECTION MAPPING

BY

BRETT R. JONES

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor David Forsyth, Chair
Professor Brian P. Bailey
Professor Derek Hoiem
Hrvoje Benko, Microsoft Research

ABSTRACT

This dissertation explores systems and methods for creating projection mapping content that seamlessly merges virtual and physical. Most virtual reality and augmented reality technologies rely on screens for display and interaction, where a mobile device or head mounted display mediates the user's experience. In contrast, projection mapping uses off-the-shelf video projectors to augment the appearance of physical objects, and with projection mapping there is no screen to mediate the experience. The physical world simply becomes the display. Projection mapping can provide users with a seamless augmented experience, where virtual and physical become indistinguishable in an apparently unmediated way.

Projection mapping is an old concept dating to Disney's 1969 Haunted Mansion. The core technical foundations were laid back in 1999 with UNC's Office of the Future and Shader Lamps projects. Since then, projectors have gotten brighter, higher resolution, and drastically decreased in price. Yet projection mapping has not crossed the chasm into mainstream use. The largest remaining challenge for projection mapping is that content creation is very difficult and time consuming. Content for projection mapping is still created via a tedious manual process by warping a 2D video file onto a 3D physical object using existing tools (e.g. Adobe Photoshop) which are not made for defining animated interactive effects on 3D object surfaces.

With existing tools, content must be created for each specific display object, and cannot be re-used across experiences. For each object the artist wants to animate, the artist must manually create a custom texture for that specific object, and warp the texture to the physical object. This limits projection mapped experiences to controlled environments and static scenes. If the artist wants to project onto a different object from the original, they must start from scratch creating custom content for that object. This manual content creation process is time consuming, expensive and doesn't scale.

This thesis explores new methods for creating projection mapping content. Our goal is to make projection mapping easier, cheaper and more scalable. We explore methods for adaptive projection mapping, which enables artists to create content once, and that content adapts based on the color and geometry of the display surface. Content can be created once, and re-used on any surface.

This thesis is composed of three proof-of-concept prototypes, exploring new methods for content creation for projection mapping. *IllumiRoom* expands video game content beyond the television screen and into the physical world using a standard video projector to surround a television with projected light. *IllumiRoom* works in any living room, the projected content dynamically adapts based on the color and geometry of

the room. *RoomAlive* expands on this idea, using multiple projectors to cover an entire living room in input/output pixels and dynamically adapts gaming experiences to fill an entire room. Finally, *Projectibles* focuses on the physical aspect of projection mapping. *Projectibles* optimizes the display surface color to increase the contrast and resolution of the overall experience, enabling artists to design the physical object along with the virtual content.

The proof-of-concept prototypes presented in this thesis are aimed at the not-to-distant future. The projects in this thesis are not theoretical concepts, but fully working prototype systems that demonstrate the practicality of projection mapping to create immersive experiences. It is the sincere hope of the author that these experiences quickly move of the lab and into the real world.

Note: I am currently in the midst of launching a startup, Luminous, with Rajinder Sodhi and Kevin Karsch. Whether we succeed or fail, it has been a heck of an awesome journey so far. See <http://brettrjones.com> and <http://luminous3d.com> for updates.

DEDICATION

To Mallory Casperson, without you this dissertation literally would not exist.

ACKNOWLEDGMENTS

The work presented in this dissertation was only made possible through extensive collaboration with some of the best researchers in the fields of computer vision, human computer interaction and interactive projection mapping. This thesis is shaped by the guidance of David Forsyth, who taught me to always attempt high impact research and to continually question everything. My entire PhD was also guided by Brian Bailey, who taught me to think like an HCI researcher and to attempt to write like one.

This thesis is the product of a unique research partnership with Rajinder Sodhi. Six years ago, we decided to pursue research in interactive projection mapping, despite our lack of knowledge and funding. Since then, we've worked closely together at UIUC, Walt Disney Imagineering (twice) and Microsoft Research. He is the intellectual foil and the jet fuel that makes our research possible.

I want to thank Hrvoje Benko, Andy Wilson and Eyal Ofek. When I started my PhD, one of my largest goals was to one day work with these pioneers of interactive projection mapping. *IllumiRoom* and *RoomAlive* were simply dreams come true. I can't even begin to describe how much I've learned from these guys. I've always been impressed with their high impact research that turns into high impact products, and hope to do so myself one day.

I also want to thank my collaborators on *RoomAlive*, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair Macintyre, Nikunj Raghuvanshi, and Lior Shapira. We had a lot of fun building a system of epic proportions, and I had an immense amount of fun during all of our visionary discussions on the future of projection mapping.

I've had the pleasure of working with Pulkit Budhiraja, who helped make *Projectibles* a reality. I know he will be a star researcher, and look forward to seeing the cool stuff he comes up with.

I would never have entered into this field without Mark Mine, Tom LaDuke and Charita Charter at Walt Disney Imagineering who showed me that magic is possible (with the right technology and the right artistic talent).

Finally, thank you to my wife, Mallory Casperson. Without her, this thesis literally would not exist. She advocated for its completion, dealt with its crunch-time, and inspired its results.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION	1
1.1 Seamless Augmented Experiences	2
1.2 Projection Mapping	2
1.3 Adaptive Projection Mapping	3
1.4 Co-Designing Physical and Virtual	4
1.5 Other Work	5
1.6 Forward to the Future	6
CHAPTER 2 BACKGROUND	7
2.1 Augmented Reality	7
2.2 Projection Mapping	7
CHAPTER 3 ILLUMIROOM: ADAPTIVE PROJECTION MAPPING AROUND A TV	12
3.1 Motivating Scenario	13
3.2 Background	13
3.3 IllumiRoom System	14
3.4 Peripheral Projected Illusions	16
3.5 Design Space	19
3.6 Triggers	21
3.7 End User Evaluation	22
3.8 Game Designer Feedback	25
3.9 Beyond Gaming	26
3.10 Discussion	27
CHAPTER 4 ROOMALIVE: WHOLE-ROOM ADAPTIVE PROJECTION MAPPING	28
4.1 Motivating Scenario	29
4.2 RoomAlive System	29
4.3 Example Experiences	37
4.4 Design Considerations	39
4.5 System Limitations	41
4.6 Summary	41
CHAPTER 5 PROJECTIBLES: OPTIMIZING SURFACE COLOR FOR PROJECTION	43
5.1 Background	45
5.2 Lighting Model	46
5.3 Problem Formulation	47
5.4 Splitting Static and Dynamic	50
5.5 Authoring Content	56
5.6 Results	57
5.7 Discussion	59
5.8 Summary	61

CHAPTER 6 SUMMARY AND CONCLUSION	62
6.1 Limitations and Future Work	62
6.2 Conclusion	64
REFERENCES	65

CHAPTER 1

INTRODUCTION

We are witnessing a second wave of virtual reality. We can now interact more naturally with the virtual world by moving our body (e.g. Nintendo Wiimote, Microsoft Kinect, and PlayStation Move). Coupled with new display technologies like the Oculus Rift head mounted display, we can feel more “in the game” than ever before. However, despite these advances, the game world is still distinctly separate from our real world. We may feel more present in the game’s world, but the game is not present in our world.

The burgeoning field of Augmented Reality attempts merge virtual and physical by inserting virtual elements into the real world. Augmented Reality provides an extra layer of information, allowing virtual elements to co-exist with the physical environment (e.g. inserting a virtual character onto a table-top). However, in most Augmented Reality systems, the existing physical environment remains untouched, leaving the experience heavily grounded in the existing physical environment, and therefore missing out on the sense of escapism provided by virtual reality systems.

The spectrum of these experiences are best explained by the Reality-Virtuality continuum introduced by Paul Milgrim [65], see Figure 1.1. The Reality-Virtuality spectrum goes from the existing physical environment (i.e. the real world) on the left side, to a completely virtual environment on the right side (e.g. a PC game). The middle left of the spectrum is *Augmented Reality*, where virtual objects are overlaid onto the physical environment. The middle right of the spectrum is the lesser known *Augmented Virtuality*, where elements of the real world are inserted into virtual environments; an example would be a real-world building re-created in a virtual environment like Second Life.

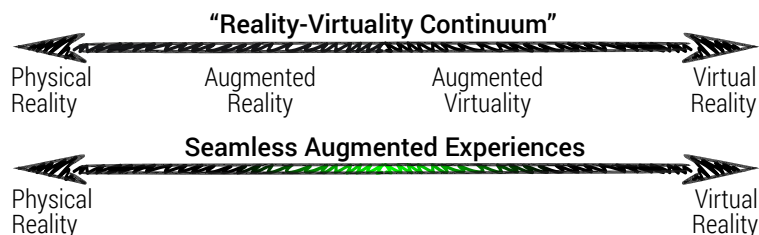


Figure 1.1: (top) The “Reality-Virtuality Continuum” introduced by Paul Milgrim [65]. (bottom) We explore *Seamless Augmented Experiences*, the middle of this spectrum where physical and virtual are indistinguishable, in an apparently unmediated way.

1.1 Seamless Augmented Experiences

In this thesis, we explore how to create content for seamless augmented experiences, where virtual and physical seamlessly combine in an apparently unmediated way. As in other augmented reality systems, we will explore inserting virtual objects into the physical world, but we will also explore augmenting the physical world to match the virtual world. For instance, altering the physical appearance of an entire living room to thematically match a virtual experience appearing on a television. We will even explore co-designing physical and virtual to create high dynamic range, high resolution projected displays.

We will demonstrate systems for creating magical experiences in a un-mediated way. Most augmented reality experiences are mediated by a device screen, where users must view the experience through a mobile device screen, or through a head mounted display (HMD). In both of these cases, the user is cognizant of the device, and therefore is cognizant of the separation between physical reality and the virtual world. These screens can create additional issues with the user experience. Mobile devices have a small screen size which limits immersion, and must be held by the user which limits user input. Mobile devices and optically opaque (video see-through) HMDs can only display physical reality by first capturing a video feed of the world, processing it and displaying it. This leads to degrading physical reality to the resolution, brightness, contrast and frame-rate of the device, which can lead to nausea [56, 61]. Even transparent HMDs cannot seamlessly blend virtual and physical. Current transparent HMDs cannot display completely transparent, or completely opaque pixels, leading to ghosting of virtual objects. Finally, all HMDs require the user to wear a heavy device on their head for long periods of time, and have limited battery life.

Head mounted displays are definitely going to be a big part of our future. However, it will be some time before these user experience issues are resolved and the technology is truly ubiquitous. If we can invent a head mounted display that weighs zero pounds, has infinite battery life, zero latency, emits light fields, and has immense on-board processing, then we should use that instead of projection mapping. Until then, projection mapping is the best way to create un-mediated magical experiences.

1.2 Projection Mapping

Projection mapping uses off-the-shelf video projectors to change the appearance of physical objects. The appearance of any object, of any shape, can be augmented by controlling the amount of light incident on the surface from the projector. Previous research has established the core mathematical principles of accurately controlling the appearance of an object given the geometry and color of the display surface [10, 80, 1]. A large body of research attempts to compensate for the display surface color and geometry, to effectively eliminate the physical environment and accurately display a virtual view (e.g. to display a football match on the wall of a building [32]).

Most previous research that incorporates (instead of eliminates) the physical environment, uses manually

modeled content that is specific to the particular display object. For instance, in the original Shader Lamps project [80], a vase is texture mapped with a pattern that was manually created and mapped to the vase by an artist. First the object must be scanned, then an artist can author content specific to the display surface. If a different display object is used, content must be created from scratch for that object. This object specific content makes it time consuming, difficult and expensive to create projection mapped experiences. It also limits the potential applications to ones where there is time to manually model content.

In this thesis we will explore methods for content creation for seamless augmented experiences with projection mapping. Specifically, we will explore two main ideas: 1) dynamically adapting projection mapping content based on the color, geometry or semantics of the scene, and 2) co-designing physical and virtual when creating projection mapping content.

1.3 Adaptive Projection Mapping

A recent line of research is in interactive experiences with projection mapping. Projects like LightSpace [106] and Beamatron [107] combine virtual and physical through engaging interactive experiences. LightSpace enabled users to transfer files between displays by holding virtual files in their hand, and Beamatron enabled users to drive a virtual car over physical ramps or up a physical wall. These interactive experiences are made possible by real-time depth sensors. These projects use a projector-depth camera unit, or *procam* unit, that enables real-time scanning of the physical environment along with real-time display on physical objects. This enables interactive experiences that can adapt to any display surface. However, this previous work focused on inserting virtual objects into a physical environment, and did not adapt the appearance of the physical environment to match the virtual world. Additionally, previous work focused on investigating the core interaction principles, and did not explore *how* to create rich, high fidelity augmented experiences.

This thesis explores how to create content for rich augmented experiences where virtual and physical seamlessly combine, to the point that the user cannot distinguish the real world from the virtual world. As in other augmented reality systems, we will explore inserting virtual objects into the physical world, but we will also explore augmenting the physical world to match the virtual world. For instance, procedurally altering the physical appearance of an entire living room to thematically match a virtual experience appearing on a television. We will explore methods for dynamically controlling interactive objects with a surface constrained particle representation. We will also explore methods for analyzing a scene to detect basic scene attributes (e.g. floor, walls), and enable artist to author content using these attributes (e.g. only spawning enemy tanks on the floor of the room).

1.4 Co-Designing Physical and Virtual

Previous work has always assumed that the goal is to project onto an existing physical object. In doing so, we must compensate for the existing surface color via a process called radiometric compensation [32, 101, 20, 1, 14]. However, this process is limited by the brightness, contrast ratio, and color primaries of the projector, along with the material and color of the display surface. For instance, a bright red surface cannot be compensated to look bright green. This leads to visual artifacts with projection mapping, and can decrease the seamlessness of the augmented experience.

In Chapter 5, we will consider co-designing the physical world along with the virtual projected content. We don't assume a fixed, existing physical display surface, but instead investigate designing the physical object to better suit the desired augmented experience. By carefully controlling the color of the display surface, we can achieve higher resolution and higher dynamic range experiences [7, 9]. We combine static printed color and dynamic projected color, to create vibrant, animated displays that can take any contour and can be arranged to form 3D displays. We detail a novel process for taking an input high dynamic range video and splitting the video into a static printed component and a dynamic projected component. Artists simply mock up their desired experience on a computer, fiddle with some knobs in our interface, then assemble and view their augmented creation.

1.4.1 Contributions

This thesis contributes novel methods for creating content for seamless augmented experiences using projection mapping, through three research threads.

1. **A novel proof-of-concept system demonstrating adaptive projection mapping surrounding a television screen**

In Chapter 3, this thesis describes a novel proof-of-concept system, *IllumiRoom*, that augments the area surrounding a television screen with projected visualizations to enhance traditional gaming experiences. *IllumiRoom* investigates how adaptive projected visualizations in the periphery can negate, include, or augment the existing physical environment and complement the content displayed on the television screen. These adaptive visuals can change the appearance of the room, induce apparent motion, extend the field of view, and enable entirely new physical gaming experiences. *IllumiRoom* is entirely self-calibrating and is designed to work in any room. In this chapter, we present a detailed exploration of the design space of peripheral adaptive visualizations and we demonstrate ways to trigger and drive such illusions from gaming content. We also contribute specific feedback from two groups of target users (10 gamers and 15 game designers); providing insights for enhancing game experiences through peripheral adaptive visuals. *This work was done at Microsoft Research in collaboration with Hrvoje Benko, Andrew Wilson and Eyal Ofek [48].*

2. A novel proof-of-concept system expanding adaptive projection mapping to whole-room interactions with multiple procam units

In Chapter 4, this thesis describes *RoomAlive*, a proof-of-concept prototype that transforms any room into an immersive, augmented entertainment experience. RoomAlive enables new interactive projection mapping experiences that dynamically adapts content to any room. Users can touch, shoot, stomp, dodge and steer projected content that seamlessly co-exists with their existing physical environment. RoomAlive combines multiple projector-depth camera units through a scalable, distributed framework. The projector-depth camera units are individually auto-calibrating, self-localizing, and create a unified model of the room with no user intervention. In this chapter, we investigate the design space of gaming experiences that are possible with RoomAlive and explore methods for dynamically mapping content based on room layout and user position. We showcase four experience prototypes that demonstrate the novel interactive experiences that are possible with RoomAlive and discuss the design challenges of adapting any game to any room. *This work was done at Microsoft Research in a highly collaborative setting with Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair Macintyre, Nikunj Raghuvanshi, and Lior Shapira. My focus was on the automated calibration system, rendering pipeline, and general system building [52].*

3. A novel process for co-designing physical and virtual with projection mapping

In Chapter 5, this thesis describes *Projectibles*, which are high resolution, high dynamic range, animated, 3D, augmented worlds created by combining printed colors with projected light. Typically video projectors display images onto white screens, which can result in a washed out image. *Projectibles* algorithmically control the display surface color to increase the contrast and resolution. By combining a printed image with projected light, we can create animated, high resolution, high dynamic range visual experiences for video sequences. We present two algorithms for separating an input video sequence into a printed component and projected component, maximizing the combined contrast and resolution while minimizing any visual artifacts introduced from the decomposition. We present empirical measurements of real-world results of six example video sequences, subjective viewer feedback ratings, and we discuss the benefits and limitations of Projectibles. This is the first approach to combine a static display with a dynamic display for the display of video, and the first to optimize surface color for projection of video. *This work was done at the University of Illinois at Urbana-Champaign in collaboration with Rajinder Sodhi, Pulkit Budhiraja, Kevin Karsch, Brian Bailey, and David Forsyth.*

1.5 Other Work

Outside of the above dissertation projects, during my PhD I have also explored spatial input for multi-scale navigation on mobile devices [50], 3D remote collaboration using spatial input [91], and creativity support

tools [41]. My work with interactive projection mapping began with the project “*Build Your World and Play In It*”, which enabled users to physically construct a playset using wooden blocks, sand, or found objects, and then map interactive virtual objects onto the physical playset [49]. During my PhD I’ve co-developed a website with Rajinder Sodhi, <http://projection-mapping.org>, which is an online magazine and community resource for all things projection mapping. At the time of writing, we get over 110,000 visitors a year. Readers are encouraged to explore the site to learn more about the magical experiences that are possible with projection mapping.

1.6 Forward to the Future

I am currently launching a company, Luminous, with Rajinder Sodhi and Kevin Karsch. This thesis has taken a back seat to things like rapid prototyping our product, startup competitions, pitching to VCs, securing an NSF SBIR grant, setting up accounting/payroll, securing work visas, and generally flying by the seat of my pants. It has been a blast so far, and whether we succeed or fail, I have already learned a lot about turning research into products. To see where things go, visit <http://brettrjones.com> and <http://luminous3d.com>.

I look forward to the future...

CHAPTER 2

BACKGROUND

This dissertation amalgamates ideas from a wide variety of fields in human-computer interaction, computer vision and computer graphics. In this chapter, we focus on the most relevant previous work in augmented reality and projection mapping. We provide the history and high-level related work in this chapter, then provide project specific related work in the subsequent chapters.

2.1 Augmented Reality

This work draws on a long history of Augmented Reality research, starting with Ivan Sutherland’s creation of the first head mounted display in 1968 [93], which enabled virtual line drawings to appear in free-space thanks to an optically see-through HMD. Since the early Augmented Reality prototypes of Virtual Fixtures [83] and KARMA [26], we’ve seen advances in rendering, display, and tracking technologies that have led to heightened realism and feelings of immersion and presence [109].

As stated previously, most Augmented Reality systems focus on inserting virtual objects into the existing physical environment and on improving the tracking technology or rendering algorithms for a more plausible object insertion. In line with *Seamless Augmented Experiences*, a number of hand-held Augmented Reality projects have explored augmenting the physical environment by using non-photorealistic rendering (NPR) techniques [29, 19, 28, 35]; for instance, applying a “painterly” rendering to the physical environment. By augmenting and diminishing physical reality, the aim is to decrease the visual disparity between physical reality and the real-time rendered virtual objects, thereby increasing the user’s sense of immersion and presence. We are inspired by this approach, and explore a wide variety of ways to augment the existing physical environment to create a unified virtual/physical experience.

2.2 Projection Mapping

The history of projection mapping dates back earlier than most sources cite. The first documented instance of projection onto a non-flat surface dates back to the 1969 opening of the Haunted mansion ride in Disneyland. The next projection mapping instance comes in 1980, with the immersive film installation *Displacements* by Michael Naimark [67]. In this art installation a living room with two performers were filmed with a rotating



Figure 2.1: *The Grim Grinning Ghosts* of the Haunted Mansion ride (1969).

camera, then the camera was replaced with a projector. The result was rotating projection mapping (referred to at the time as “relief projection”).

In 1991, Disney revived their work with projection mapping, resulting in a patent entitled *Apparatus and method for projection upon a three-dimensional object* [66], essentially describing a system for digitally painting an image onto “a contoured, three-dimensional object”. Next, a patent by GE in 1994 describing “a system and method for precisely superimposing images of computer models in three-dimensional space to a corresponding physical object in physical space” [31].

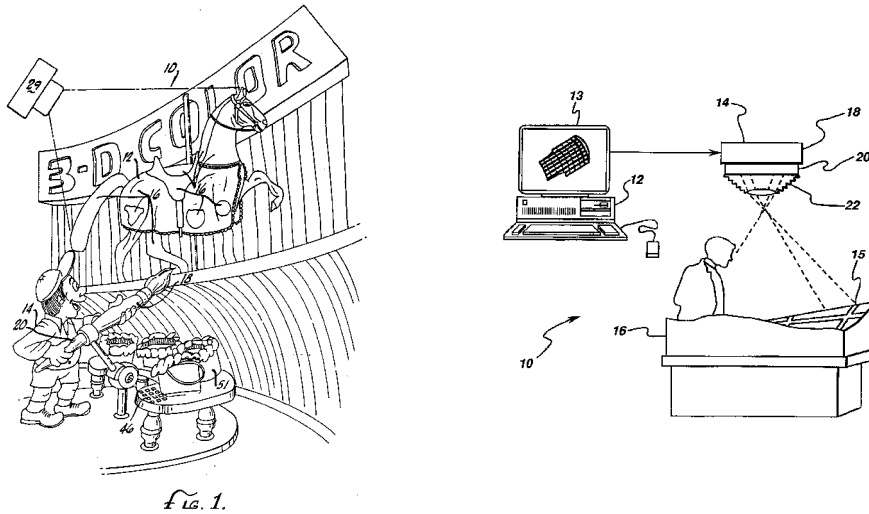


Figure 2.2: (left) Disney’s 1991 Patent [66]: “*Apparatus and method for projection upon a three-dimensional object*” (right) GE’s 1994 Patent [31] for “*a system and method for precisely superimposing images of computer models in three-dimensional space to a corresponding physical object in physical space.*”

In 1998, projection mapping was rigorously defined with *The Office of the Future* project by Ramesh Raskar, Grey Welch, Henry Fuchs and Deepak Bandyopadhyay at UNC Chapel Hill [81]. *The Office of the Future* envisioned a world where projectors could cover any surface, enabling tele-conferencing and visualization applications everywhere. *The Office of the Future* even presented a real-time, imperceptible structured light implementation to scan everyday objects for projection. Around the same, John Underkoffler explored interacting with projected widgets on a tabletop in *The Luminous Room* [97]. Also, the KidsRoom project

similarly envisioned a children’s room that used projection to educate and entertain [12]. Follow on work by Raskar established the mathematical foundations for augmenting an arbitrary surface with projection by compensating for the color and geometry of the underlying surface [80]. This work also coined the academic names “Spatial Augmented Reality” (SAR) and “projector based augmented reality”. Since this early work, a wealth of academic projects have refined the ideas of projection mapping as both a display and interactive medium.

2.2.1 Display Research

The field of projection mapping also shares its lineage with the development of large-tiled displays for visualization and simulation applications. One of the most popular examples, the *CAVE* (a recursive acronym: CAVE Automatic Virtual Environment) is a virtual environment where up to six walls of a room are covered in projected content, yielding a very immersive viewing experience [22]. To increase the display resolution, researchers turned to tiled-displays, using multiple projectors to eventually achieve eye-limiting resolution. Initially these displays needed to be manually calibrated by physically tweaking the projector location and adjusting optical blending systems [40]. This still left artifacts in the overlapping regions or regions in-between displays.

The solution was to pair a camera with the projectors, and digitally compensate the overlapping regions in software to achieve a seamlessly tiled display [108, 15]. We call this pairing of a projector and a camera, a *procam* unit (also referred to as a “smart projector” [8]). The basics of this process involve two steps, geometric and radiometric calibration. In geometric calibration, we determine a mapping from projector pixels to camera pixels, then fit this mapping to some model. For instance, if calibrating to a curved display we may use a parametric function for the geometric registration process. This enables us to correctly pre-warp the displayed content, such that it appears un-warped to the end user. In fact, the display can be optimized for a particular viewing position through a two pass rendering approach [80], making the content appear to “pop” out of the screen.

In radiometric calibration, each displayed pixel is compensated for the color of the underlying surface and to achieve the best possible image across the entire display [62]. This process is complicated because there are distinct physical limits on the brightness, contrast ratio, and color primaries of projectors. Also, projectors have many non-linear behaviors, such as a non-uniform brightness across the image, with the brightest spot at the center of the image. Previous work has investigated ways to decrease the appearance of these non-linear behaviors and to seamlessly tile multiple projectors together [63]. Besides tiled-displays, radiometric compensation enables projection onto non-white objects. For instance, it can be used to superimpose images on top of paintings [7]. Also it can be used to display video content on any surface in a living room, for instance, turning a plaid curtain in to a video screen [8, 32]. Chapter 5 covers previous work in radiometric compensation in more detail.

2.2.2 Interactive Projection Mapping

Interacting with projected mapped content began with the Dynamic Shader Lamps project [3], which enabled users to paint with light on top of everyday objects like a vase, using a stylus. A number of projects in the vein of *The Office of the Future* explored projection in office environments [82, 71, 57, 47] enabling users to interact with documents, pictures and other 2D media on flat surfaces (e.g. desks and walls). One of the key limiting factors with this early research, is that the 3D representation of the environment must be manually modeled or painstakingly measured, limiting the overall complexity of the surface. In this dissertation, we explore adaptive projection onto arbitrarily complex surfaces.

With the miniaturization of projector hardware, it became possible to pick up a projector to interact with the environment. Starting with Raskar’s iLamps [78], users could carry around a projector that dynamically adapts to the display surface. For instance, a user can receive visual instructions on how to change a printer cartridge, displayed directly on the printer. A number of hand-held interactive projects followed defining the design space of interaction metaphors for hand-held projection [18, 103, 102, 104, 77]. In this work, we focus on fixed projectors but the core idea of adaptive projection mapping could be extended to mobile systems.

The field of projection mapping has been greatly impacted by the recent proliferation of cheap, accurate, real-time depth sensors. Starting with early depth sensor prototypes, Wilson et al. explored [105] real-time simulated physical interactions with projection mapped objects. For instance, using a real-time depth sensor to enable a person to kick a virtual car. Fueled by the introduction of the Microsoft Kinect sensor, numerous projects have explored various interaction possibilities with real-time depth data including tele-presence applications [6], mobile form-factors [37], office applications [106], desktop computing [42] and gesture guidance [90].

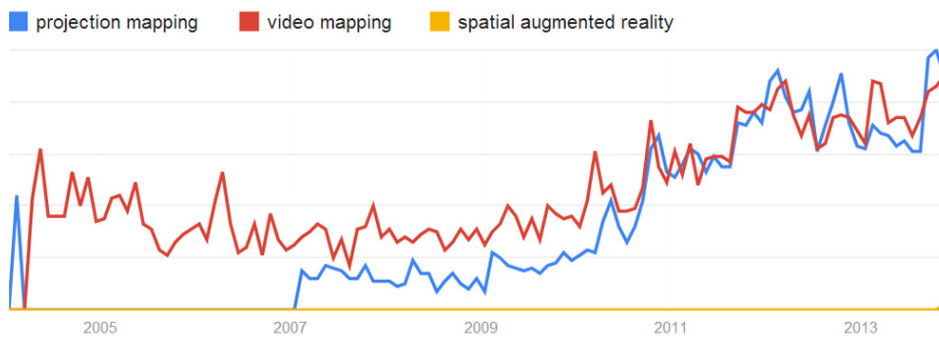


Figure 2.3: Artists and advertisers have flocked to the terms “projection mapping” and “video mapping.” (data from Google Trends 01/18/2014).

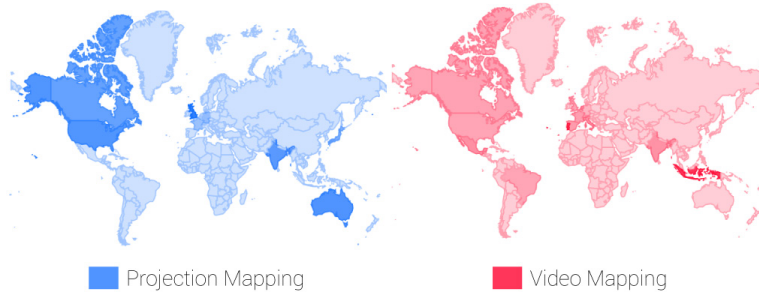


Figure 2.4: “Projection mapping” is more popular in the USA and “video mapping” is more popular in Europe and Asia (data from Google Trends 01/18/2014).

2.2.3 Art, Advertising & Entertainment

In the past five years, projection mapping has exploded into the consciousness of artists and advertisers everywhere. It is here that the term “projection mapping” (and the alternative “video mapping”) began to win popular acceptance (see Figure 2.3), with “projection mapping” more popular in the USA and “video mapping” more popular in Europe and Asia (see Figure 2.4). One aid to projection mapping’s popularity is the practice of *Son et Lumire* (sound and light) shows in Europe, where special lighting effects are projected onto the facade of a building. Some notably popular viral videos were Sony’s real-time projection mapping of a living room [64], Urbanscreen’s building mapping [98], BotNDolly’s robotic projection mapping [13], and deadmau5’s building mapping in London [23]. With the spread of projection mapping, a number of commercial software applications now support warping video content onto non-flat objects [51] through the use of manual alignment tool (like a mesh warp).

This thesis builds on this previous work, through three projects that explore novel systems and methods for content creation for projection mapping, with the goal of creating seamless augmented experiences.

CHAPTER 3

ILLUMIROOM: ADAPTIVE PROJECTION MAPPING AROUND A TV

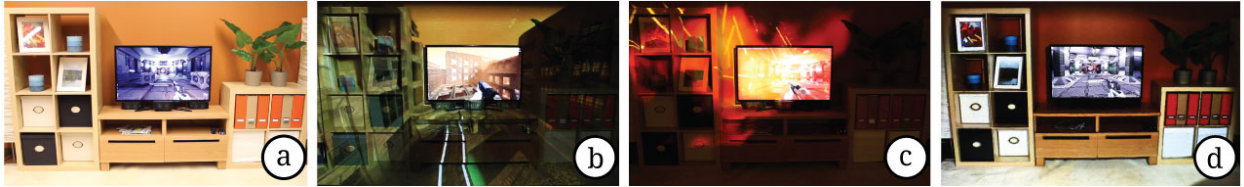


Figure 3.1: IllumiRoom is a proof-of-concept system that augments the physical environment surrounding a television to enhance interactive experiences. We explore the design space of projected visualizations which can negate, include or augment the surrounding physical environment. (a) With a 3D scan of the physical environment we can (b) directly extend the FOV of the game, (c) selectively render scene elements, (d) augment the appearance of the physical environment (here as a cartoon).

In the first project, we present IllumiRoom, a proof-of-concept system consisting of a projector and a depth sensor that covers a wide area surrounding a television screen. The projected visuals enhance the viewing experience and blur the boundary between the on-screen content and the surrounding room. Thus exploring the concept of *Seamless Augmented Experiences* within a context of living room gaming. We demonstrate how projected visualizations in the periphery can negate, include or augment the physical environment, and thus enhance the content displayed on the television screen. We call such visualizations peripheral projected illusions.

Similar to Focus+Context displays [5], the television provides a traditional, high resolution gaming experience and the projector provides low resolution information for the user’s peripheral vision. In contrast to previous work, we do not use flat, white projection screens, but instead adapt the projection to the

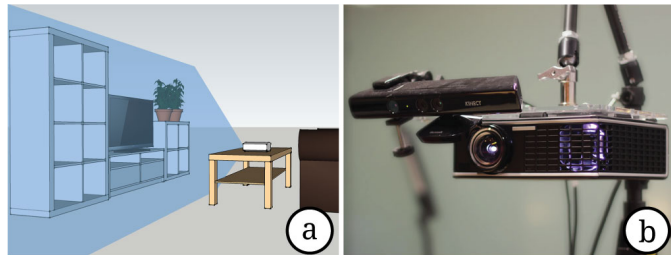


Figure 3.2: (a) Our vision for a productized IllumiRoom system includes an ultra-wide field of view device that sits on a coffee table and projects content in the area surrounding the television. (b) Our current proof-of-concept prototype uses an off-the-shelf projector and a Kinect sensor.

existing living room environment. We are the first project to consider the geometry and appearance of the surrounding room and use that information to create novel, interactive visual experiences.

In this chapter, we explore the design space of peripheral projected illusions and demonstrate eleven example illusions. For instance, we can directly extend the game content into the living room by displaying a wide field-of-view rendering of the game on the surrounding physical environment with radiometric compensation (Figure 3.1b). We can also selectively introduce game elements into the living room; e.g. only displaying weapons fire and explosions in a first person shooter (Figure 3.1c). Or, we can augment the living room to match the theme or mood of the game; e.g. the room can transform into a ‘cartoon world’ by super-saturating the colors and adding black silhouette edges. We can also enable physical-virtual gaming experiences by using the 3D data of the living room; e.g. a grenade can roll out of the television, then bounce off the coffee table, and roll on the floor of the room. All of these effects adapt to the color and geometry of the living room, and are design to work in any room.

A note: This work was done at Microsoft Research in collaboration with Hrvoje Benko, Andrew Wilson and Eyal Ofek [48].

3.1 Motivating Scenario

To illustrate the idea behind the IllumiRoom system, imagine sitting down in your living room to play a video game on your television. When the game starts, the room magically transforms to look like a cartoon, matching the shading in the video game. The colors of the room become super-saturated and cartoon edges appear on your bookshelves. You come across an enemy in the game, and suddenly a streaking bullet flies towards your character and then out of the television. The enemy throws a grenade towards you. The grenade rolls out of the television, bounces off your coffee table and explodes in your living room, throwing shrapnel across your bookshelf. The entire living room appears to shake as you take damage from the explosion. This is just one of many interaction scenarios made possible with the peripheral projected illusions described in this chapter.

3.2 Background

The configuration of a small, high resolution display within a large, surrounding projection is consistent with the setup of human visual system. Compared to the center of gaze, peripheral vision is highly sensitive to motion, but is relatively poor at distinguishing color and shape [74]. This is due to the lower density of receptors and the changing distribution of receptor types outside the fovea. In IllumiRoom, the smaller display serves the high resolution capability of the fovea, while the projector serves the low resolution, motion sensitive periphery. As peripheral vision is insensitive to color and highly sensitive to motion, many of the illusions are tailored to peripheral vision. For instance, abstract patterns can be used to induce optical flow

and give an impression of motion.

The characteristics of a display can have a large impact on user performance, especially while gaming. A larger display has a wider field of view, can cause the user to feel more immersed and more present in the experience, and is more likely to induce simulator sickness [44, 94]. Unfortunately, large displays can be expensive. To balance the costs of a high resolution, larger display with the benefits of a wider field of view, Baudisch et al. proposed Focus+Context displays. Focus+Context displays have a high resolution computer monitor (focus) surrounded by a lower resolution projection screen (context). This configuration takes advantage of the lower optical resolution in peripheral vision [39]. We refer the reader to [21] for a more comprehensive review of Focus+Context techniques.

The Philips Ambilight television is an example of a commercially successful Focus+Context display. The Ambilight TV has a series of color LED strips surrounding its edges. The color of the LEDs is dynamically changed to match the television content. In a follow-up research project, Philips created a prototype with multiple LED strips located around the room that change colors to adapt to gaming or movies [100]. Our system advances this idea to high-resolution projected content which is dynamically adapted to the appearance and geometry of the room.

Most similar to our work, a few research projects have explored automatically extending video beyond the television frame for Focus+Context displays. The Infinity-By-Nine project [70] displayed extrapolated video content on three projector screens surrounding a television, in a small CAVE [22] setup. Aides et al. described a system that creates an ultra-wide foveated video from a normal input video sequence [46].

As in these previous works, the IllumiRoom system can be used for panoramic movies and television, and could be extended to use extrapolated content from standard video sources. Rather than relying on separate projection screens, IllumiRoom adapts the projections to the user’s existing living room, and demonstrates a large range of possible peripheral projected illusions.

3.3 IllumiRoom System

Our vision for a fully developed IllumiRoom system includes an ultra-wide field of view device sitting on the user’s coffee table, projecting over a large area surrounding the television (see Figure 3.2a). The device would be connected wirelessly to a next generation gaming console as a secondary display. The ultra-wide field of view could be obtained with an ultra-short throw projector, or by coupling a standard projector with a spherical/parabolic mirror (as the peripheral content does not need a uniform pixel aspect ratio). The room geometry could be acquired with a depth sensor or a structured light scan (which would require only a web camera, similar to the setups in [8, 49]). The acquired living room geometry is used during calibration and to drive illusions that respond to living room furniture in a physically realistic manner (similar to the ideas in [6, 107]).

Our current proof-of-concept prototype uses a commodity wide field of view projector (InFocus IN126ST)

and a Microsoft Kinect for Windows sensor (see Figure 3.2b). The prototype is limited by the field of view of the Kinect and projector. Therefore, the system is mounted above and behind the user’s head, as they are seated on a couch in front of the television.

The Kinect sensor captures the color and geometry of the scene, and the system renders the illusions using the acquired depth map. Careful calibration of the system is required in order for the illusions to tightly match the on-screen content and the physical environment. The calibration of the IllumiRoom system is fully automatic, determining the relative pose of the projector to the depth sensor and the position of the television in the projector image. Therefore, setup only requires that the projector and depth camera are placed such that they cover the area surrounding the TV.

The automatic calibration projects Gray code sequences [4] to establish dense correspondences between the projector and the Kinect’s color camera. Using the default parameters for the Kinect, we transform these correspondences into 3D points. We then solve for the intrinsic and extrinsic parameters of the projector. As there is no depth data for specular surfaces such as that of shiny, black televisions, we recover the position of the TV using a 2D homography, with virtual markers displayed in each corner of the screen. It is important to note that the calibration need only happen once. Thus the Kinect camera could be used to calibrate the projector and then returned to the TV stand for gaming.

The acquired living room geometry is used during calibration and to drive illusions that respond to living room furniture in a physically realistic manner (similar to the ideas in [6, 107]). IllumiRoom could easily be extended to continuously capture the geometry, enabling ad hoc changes to the living room such as moving furniture and people.

3.3.1 Radiometric Compensation

IllumiRoom projects on the existing living room furniture, and therefore cannot rely on the presence of a white, flat projection surface. Using radiometric compensation [7, 9, 80] the projected light may be modulated to achieve a desired color, given knowledge of the existing surface color and geometry. However, this process is limited by the brightness, dynamic range and color primaries of the projector, as well as with the existing surface color, material, and geometry. Accordingly, many desired surface colors are unachievable. For example, a solid red object in our physical environment cannot be made to appear green or blue.

Therefore, all of the illusions use the fact that they occur in peripheral vision. Illusions that may not seem realistic on close examination may be quite effective in the user’s peripheral vision. Most illusions do not attempt to accurately reproduce a scene, but aim to introduce motion in the periphery. Also, subtly modifying the existing surface color may be more effective than replacing the color entirely.

Illusions that modify the surface color and not the apparent geometry of the room are independent of viewing position, and thus can be viewed by multiple users from any position within the room. Illusions

that add to or modify the geometry of the room are inherently dependent on viewing position and are most effective when viewed from a fixed point within the room. Generally, as the room’s furniture becomes more complex, effects which virtually modify the geometry are more sensitive to viewing position.

3.4 Peripheral Projected Illusions

We envision the IllumiRoom system supporting a great variety of peripheral projected illusions, limited only by the imagination of game designers. We have prototyped illusions that we believe represent the primary dimensions of the design space (discussed in the next section). This represents an initial survey of the design space and is not exhaustive. We outline the techniques used to create the illusions, and describe the illusions that were implemented. We present the illusions as distinct points in the design space. However, the illusions can be combined in the IllumiRoom system. While some illusions are mutually exclusive (F+C Full & F+C Edges), almost all illusions can be combined.

3.4.1 Focus+Context

The most obvious way to increase the user’s sense of immersion is to extend the content from the television screen out into the room, replacing the physical reality with the game’s reality. We call this Focus+Context Full (F+C Full) (see Figure 3.3a). This is essentially the Focus+Context approach [5], but using a non-flat, non-white projection surface, with radiometric compensation. As stated above, the ability to compensate for the existing surface color is limited, and so this effect relies on the fact that the user will directly attend to the television screen rather than the periphery.

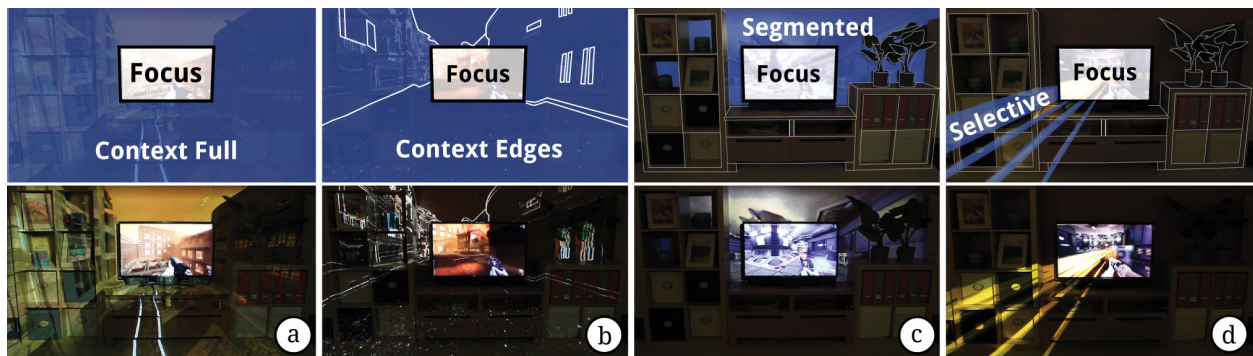


Figure 3.3: Projecting wider field of view game content around the TV can increase immersion. (a) *F+C Full*: The full game content is projected with radiometric compensation onto the surrounding furniture. (b) *F+C Edges*: A high contrast version where only the edges in the game content are projected. (c) *F+C Segmented*: Game content is projected only onto the flat, rear wall. (d) *F+C Selective*: Selective game content (e.g. bullets) bleed out of the TV.

The next illusion, Focus+Context Edges (F+C Edges), provides a maximum contrast representation of the peripheral content by displaying black-and-white edge information (Figure 3.3b). This illusion is more

robust to ambient light in the room, and increases peripheral optical flow.

Focus+Context Segmented (F+C Seg) extends the game only onto the rear wall surrounding the television (Figure 3.3c). While the furniture in the room may not be ideal for projection, most televisions are at least partially surrounded by a flat, light colored wall which provides an ideal, view independent projection surface. This rear wall could be found by a recursive RANSAC plane fitting procedure [73], seeded with the location of the television. Our prototype currently relies on a manually specified image mask identifying the rear wall.

While the previous illusions bleed all game information into the physical environment, there are times when a more subtle approach may be appropriate. For example, when watching a horror movie, an atmosphere of immersion and suspense is created by the absence of the villain, and then their sudden appearance. When all information is present all the time, there is no opportunity for suspense or surprise. With Focus+Context Selective (F+C Sel) only certain game elements escape the television (Figure 3.3d). For instance, a first-person shooter might bleed only weapons fire or explosions out of the television. Or, markers representing other characters or key items in the game may be selectively displayed, thus increasing their emphasis.

These illusions increase immersion, induce apparent motion, and provide additional information about the game content. The Focus+Context illusions all require access to the video game’s rendering process.

3.4.2 Peripheral Flow

If we do not have access to the rendering process, apparent motion may still be induced through peripheral visual flow, as long as we have information about the motion of the game’s camera. In Grid (Figure 3.4a), an infinite grid moves with the motion in the game. When the user moves or looks left or right, the grid moves in the opposite direction. When the user moves forwards or backwards, the grid zooms in or out.

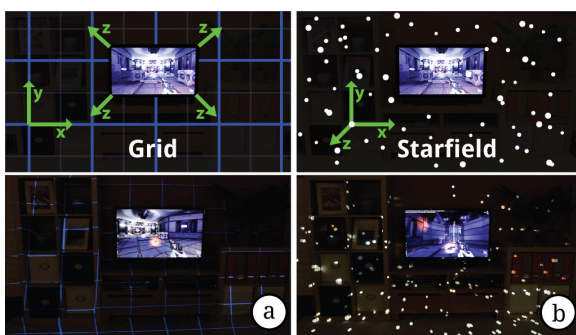


Figure 3.4: Apparent motion may be induced without access to game rendering by displaying abstract peripheral visual flow. (a) An infinite *Grid* moves with the game’s camera. (b) A 3D *Starfield* of spheres moves with the game’s camera.



Figure 3.5: *Appearance*: The appearance of the physical environment may be augmented (a), to match the theme of the game. (b) Cartoon super-saturated colors and silhouette edges (c) Black and white (de-saturated) colors.

Similarly, Starfield (Figure 3.4b) creates peripheral flow by moving 3D points to match the movement

of the game’s camera. Points are randomly generated in the virtual space between the TV and the user, according to the position and orientation of the game’s camera.

3.4.3 Color Augmentation

The physical environment may be augmented to match the theme or mood of the game. For instance, room colors may be super-saturated to resemble a cartoon look by simply projecting the color of the surface back onto itself [9], and then drawing silhouette edges black (Figure 3.5). Similarly colors may be desaturated to create a black-and-white, “film noir” look. We leave more complex texture replacement techniques (i.e., procedural texturing) for future work.

3.4.4 Texture Displacement

Imagine an explosion in a video game or film that is so powerful that it shakes your entire room. IllumiRoom can use projected light to modify the appearance of objects in the room, making them appear to move or change size, or distort in more extreme ways. Due to the projector’s limited ability to compensate for surface color, these illusions are most effective if the displacements are small and short-lived. Radial Wobble (Figure 3.6) distorts the apparent room geometry in a ‘force field’ ripple effect. The illusion displaces the original surface texture in an expanding, radial sinusoidal wave.



Figure 3.6: *Radial Wobble*: To create a ‘force field’ ripple effect, the physical surface texture is displaced with a radial expanding sinusoidal pattern.

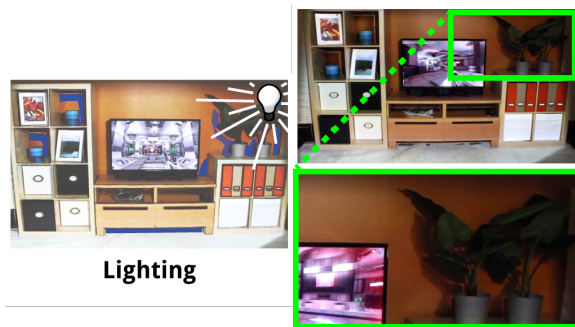


Figure 3.7: *Lighting*: The lighting and shadows of the physical environment are matched to lighting in the virtual environment.

3.4.5 Lighting

With full control over the illumination, the lighting in the physical environment may be altered [80, 79]. Room lighting can change based on the mood or theme in the game. For example, the look of a scene set in space might be achieved by illuminating with point light sources and harsh shadows. The Lighting effect attempts to match the lighting conditions of the physical environment with the virtual environment

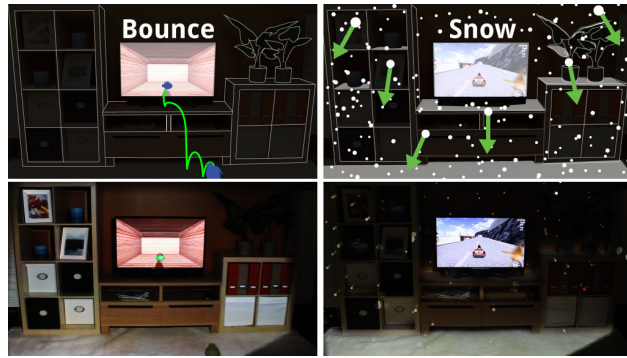


Figure 3.8: Virtual objects can realistically interact with physical objects. (a) In *Bounce* a grenade rolls out of the TV and bounces around the living room. (b) In *Snow*, falling snow moves according to the user’s movement in the game and interacts with the physical environment.

(Figure 3.7). Our prototype implements soft shadows and a single point-light source, matching the closest point-light source in the game.

3.4.6 Physical Interactions

In Focus+Context Selective elements from the game (e.g., bullets) break the screen barrier and fly into the physical environment; however, there are no physical interactions with the environment. *Bounce* envisions an element in the game leaving the virtual environment and interacting with physical reality. In our prototype, a grenade rolls out of the television, and then bounces and rolls around the physical environment according to a physics simulation using the room geometry (Figure 3.8).

In *Snow*, falling snow interacts with the physical environment, briefly collecting on surfaces in the room. Similar to *Starfield*, the snow moves according to the movement of the user in the virtual environment, allowing the user to walk or drive through snow (Figure 3.8). Our prototype simulates the accumulation of snow by turning surfaces with an upward facing normal gradually white.

3.5 Design Space

The range of possible peripheral projected illusions is as great as the range of visual effects in video games or film. In order to better understand the full range of illusions, we identify the primary dimensions of the design space. Illusions are defined by three primary factors: the goals of the illusion, the connection or separation from physical reality and the level of abstraction from the game content.

3.5.1 Illusion Goals

Peripheral projected illusions can enhance gaming experiences in a variety of ways. All of the proposed illusions attempt to increase the user’s sense of immersion by increasing the FOV and surrounding the user

	Immersion	Presence	Apparent Motion	Theme	Content	Information	Reality Connection	Reality Distortion
F+C Full	✓	✓	✓		✓			
F+C Edges	✓	✓	✓		✓			
F+C Segmented	✓	✓	✓		✓			
F+C Selective	✓	✓	✓		✓			
Grid	✓	✓	✓					
Starfield	✓	✓	✓					
Appearance	✓	✓		✓		✓	✓	
Radial Wobble	✓	✓	✓	✓	✓	✓	✓	✓
Lighting	✓	✓	✓	✓		✓	✓	
Bounce	✓	✓			✓	✓		
Snow	✓	✓	✓			✓		

Figure 3.9: IllumiRoom illusions have a variety of goals: increasing immersion and presence, inducing apparent motion, providing additional information or content, creating a sense of atmosphere or theme and supporting entirely new physical-virtual game mechanics.

with content. All of the illusions can increase the user’s sense of presence, making the user feel ‘in the game.’ More specifically, the illusions can induce apparent motion, provide additional information and content, create a sense of atmosphere or theme and support entirely new physical-virtual game mechanics. These goals are not mutually exclusive, and a single illusion may support multiple goals (see Figure 3.9).

3.5.2 Connection to Reality

IllumiRoom enables designers to explore the entire spectrum of the reality-virtuality continuum [65]; by negating, including, or augmenting the surrounding physical environment. Designers can negate the physical environment, making the living room disappear; thereby immersing the user in an entirely virtual environment (e.g., F+C Full, F+C Edges, Grid, Starfield).

Or the game can be grounded in reality, by including the physical environment and selectively introducing virtual objects into physical reality (e.g., F+C Seg, F+C Sel, Bounce, Snow).

Finally, IllumiRoom enables a unique middle ground, where the physical surroundings are augmented to match the virtual environment, creating a mixed reality where it is not clear what is real or virtual (e.g., Appearance, Lighting, Radial Wobble).

3.5.3 Abstraction

The illusions can be dependent or independent of the underlying game content. For instance, if we extend the exact game content outside of the TV, the illusion is entirely dependent on the game content (e.g., F+C

	Controller Input	Controller Rumble	Optical Flow	Source (Network Msgs)	Source (Rendering)	Unified (Ideal)
F+C Full					●	●
F+C Edges					●	●
F+C Segmented					●	●
F+C Selective					●	●
Grid	○		●	●		●
Starfield	○		●	●		●
Appearance						●
Radial Wobble	○	○		●		●
Lighting				○		●
Bounce				○		●
Snow	○			○		●

Figure 3.10: Ideally the IllumiRoom system is unified with the game content. However, existing game content can be used with the prototype through a variety of input sources. Here we show the different ways of connecting game content to the illusions.

Full, F+C Edges, F+C Seg, F+C Sel, Bounce). However, if we display an abstract visualization, such as a moving grid, then the peripheral illusion is independent of the game content (e.g., Grid, Starfield, Snow, Appearance, Lighting, and Radial Wobble).

3.6 Triggers

The illusions can be connected with game content through a variety of means. Ideally, IllumiRoom would be directly integrated into a next generation console and new games would be designed for IllumiRoom from the ground up. We envision an API that enables triggering illusions, changing surface appearance, controlling room lighting, inserting objects into the physical environment, etc. In our prototype, we also demonstrate how to connect existing game content to drive illusions. In order to make the system work with numerous games, the IllumiRoom system is a separate component from the game examples, and networking messages are sent to share information. We present a variety of methods for modifying existing content. Figure 3.10 enumerates the input sources that can be used for each illusion.

If the source code of the game is available, then the game can be easily modified to trigger effects. In the current prototype, the game sends networking messages to the IllumiRoom system in order to toggle illusions on and off. The position and orientation of the player (or often, equivalently, graphics camera) in the virtual world may be sent to drive the Peripheral Flow illusions. With access to source-code the game

can support Focus+Context illusions by conducting a second rendering pass using a wider field of view. In our prototype this information is sent to the IllumiRoom system through a texture residing in shared memory.

Source-code is not available for most existing games. Fortunately, many of the illusions can be controlled by other means. For example, the Peripheral Flow illusions need only some measurement of the player/camera movement. An approximation can be obtained by real-time image analysis of the rendered game (e.g., optical flow). In our prototype, any game’s image may be captured using a video capture card (DeckLink Intensity Pro), and subsequently analyzed by optical flow techniques to drive the illusions.

A significant amount of game state may be inferred solely from the user’s input through the controller. Given knowledge of the game and the controller mapping, controller input may be mapped directly onto illusions. For instance, if the game is a first-person shooter and the player presses the ‘Right Trigger’ button on the Xbox controller, then the player must be shooting. The Radial Wobble illusion may then be activated, for example. In our prototype, input from the Xbox controller and vibrotactile output sent to the controller from the game can drive illusions. This works with any game. We envision users scripting and sharing ‘mods’ of their favorite games to work with the IllumiRoom system.

For our prototype, games are run on the same PC as the IllumiRoom system. GlovePIE¹ intercepts and sends game controller input to the IllumiRoom system by networking messages. Vibrotactile output at the controller is captured by using an Xbox controller modified with an Arduino.

3.7 End User Evaluation

We evaluated eleven points in the design space of peripheral projected illusions through a user study. We were primarily interested in determining user perceptions along different dimensions of the design space. Just as there is no ‘best visual effect’ for film or video games, we were not expecting to identify a ‘best illusion’. Rather, we sought the strengths and weaknesses of each technique. We were also interested in the user’s overall comfort level during the illusions. Each user interacted with the illusions, answered a Simulator Sickness Questionnaire (SSQ) [56], and performed a card sorting task where they rated and ranked each illusion along a variety of dimensions.

We recruited 10 participants (ages 20-30, 2 female). In order to ensure that participants could focus on the gaming experience and not the control scheme, we required that participants be familiar playing first-person shooter games with an Xbox controller. The user study lasted approximately one hour and participants received \$10 for their participation. Audio and video of the sessions were recorded for the purposes of extracting quotations.

In our experiments, the user sat on a couch 8 ft. from a 40 diagonal television displaying 720p content. A projector was mounted above and slightly behind the user’s head, 11 ft. from the television and 5 ft. above

¹<http://glovepie.org/glovepie.php>

the ground, at a resolution of 1280x800, casting a 130 inch diagonal image (measured within the plane of the television). The television viewing angle was 20° and the projector’s viewing angle was 45°. These distances were chosen to maximize the projected area, within the limits of the working range of the Kinect sensor.

3.7.1 Game Content

Each participant interacted with eleven illusions paired with matching game content. We paired game content with the illusions in order to maximize the effectiveness of the experience. The majority of the illusions were paired with an open-source first-person shooter (Red Eclipse²). This created a rich, interactive experience, enabled by access to source code. The Snow illusion was paired with a racing game with a snow level (SuperTuxKart³), triggered with controller input. Lighting was illustrated by a Unity3D race car example⁴, which uses a few clear light sources. The light positions were sent by network messages. The Bounce example was paired with a custom DirectX application in order to perfectly synchronize the content between the physical and virtual worlds. This example was not interactive, but served to demonstrate the basic concept of virtual to physical interactions for users. See the accompanying video for the exact content.

3.7.2 Procedure

Participants interacted with the eleven illusions with paired video game content in a randomized order. Participants were first introduced to each game without any peripheral projected illusions so that they could learn the control scheme and game objectives. Then, the illusion was revealed and gameplay continued. Immediately following the last illusion, participants filled out a Simulator Sickness Questionnaire (SSQ) evaluating their overall comfort level.

Participants then engaged in a card sorting task, where they rated the illusions on various dimensions. Cards with a picture representing each illusion were placed on a coffee table in front of the user to act as a visual reminder of the illusion. The participants rated the ‘overall satisfaction’ of the illusion on a 5 point Likert scale, and then ranked its ‘overall satisfaction’ (without ties). This procedure was designed to spur careful critique and comparisons, not to identify a ‘best illusion’. Participants then rated the illusions on a 5 point Likert scale in response to the following prompts: “It was fun”, “I felt sick”, “I felt like I was moving”, “I felt like I was in the game world”, and “The game and the physical world felt like separate spaces.” As this task was completed after viewing all illusions, this enabled the users to mentally compare and contrast the illusions. Users did not express any concern regarding their ability to recollect the illusions.

²<http://redeclipse.net>

³<http://supertuxkart.sourceforge.net>

⁴<http://unity3d.com>



Figure 3.11: Mean rating for each illusion evaluated in the user study. Participants rated ‘overall satisfaction’ and responded to five questions on a five point Likert scale from ‘strongly disagree’ to ‘strongly agree’.

3.7.3 Results & Feedback

Participants responded very positively to the illusions and the concept of IllumiRoom. As expected, there was not one ‘winning’ illusion, but a diverse set of illusions that have strengths on different dimensions of the design space. For a summary of participant responses see Figure 3.11.

The results from the SSQ give a coarse indication of potential issues with simulator sickness. The mean value of Total SSQ (14.96), was within normal ranges for playing stereoscopic video games [85] or interacting with immersive simulators [56]. To keep the user study a reasonable length we did not compare SSQ scores across illusions. Future work could explicitly compare individual SSQ scores.

Participants found the Focus+Context techniques some of the most useful illusions for game play, because they naturally provide additional gaming information. Simply by extending the field of view of the game, these illusions increase the user’s sense of motion and presence (“I was in the game”). For instance, one user commented that F+C Selective was “very useful for finding items, it kinda lets you see just pieces of the game that are important to know around you, and do spatial orientation.” However, the F+C illusions can also make the user overwhelmed or more prone to sickness. For instance, one user commented that with F+C Full, “There was just a lot going on. To see everything going on around you it was like really cool but especially in as fast paced a game as that, it was just too much happening at once.” F+C Edges only made that problem worse, by emphasizing high contrast edges; thereby receiving the lowest satisfaction scores. The F+C Selective and F+C Segmented illusions seem to represent a middle ground between immersion, adding content information and balancing user comfort. As one user said about F+C Selective, “Out of all of these that would be the most useful and immersive without being distracting.”

Of the Peripheral Flow illusions, Grid seemed to be more successful at imparting a sense of apparent motion; possibly due to the more coherent, 2D nature of the visual flow that it created. One user commented about Starfield, “I like the idea of being able to feel your motion in the game, but the random placement of points made it less consistent.” Users were split as to whether the Peripheral Flow illusions were more or less distracting than the F+C illusions. Multiple users commented that they had a sense of motion, but

missed the contextual information of the F+C illusions.

Appearance worked well at creating a sense of atmosphere for the game. As one user put it “The appearance is awesome. Like that could just be an effect in someone’s living room all the time. That is just super cool.” However, the effect by itself was deemed not as interesting as it would be if it were combined with other effects. One user stated “If you are going to project something, you might as well project something that is directly relevant to the game.”

Radial Wobble was another ‘magical’ effect, that “kinda made you feel like you are physically doing something in [the game].” However, some users said the effect was off-putting, because “all your stuff is shaking in the real world.” In the evaluation, Radial Wobble was triggered in the first-person shooter every time a weapon was fired. As one user stated, “It just seemed like one that you should save for really impactful moments. So I just wouldn’t want it all the time.”

Participants generally liked the play between the physical and virtual worlds in the Bounce, Snow and Lighting illusions. For example, Lighting served to create a better sense of immersion and presence for some users “I think more so than any of the others, that one made me feel like I was in the game world.” For the Bounce illusion, due to the rough nature of the paired DirectX sample application users had to imagine what it would be like to incorporate this effect into a real game. Multiple users referenced the grenade indicator in Infinity Ward’s Call of Duty, “Most of the time I don’t see the grenades at all...They always kill me, so that was a good one.” The Snow illusion created a sense of atmosphere and movement for users; however most users did not notice the physical interactions with the geometry. This is because the snow quickly ‘melted’ after impact. One user commented that they “really wanted it to pile up more. I found it to be melting and if it was piling up it would be an indication of how long I was playing and that would be cool.” However, these illusions (particularly Lighting) easily revealed even the smallest mismatch between the game world and the real world, which hampered the illusions for the participants. One user commented that a better use for Lighting would be in illustrating the difference in illumination when the player moves between indoors and outdoors in an open world game (e.g., Grand Theft Auto).

3.8 Game Designer Feedback

In addition to gamers, we also elicited informal feedback about our illusions from game designers (N=15). From the previous user study, we grasped how gamers would react to the illusions, but we wanted to see what content creators thought about the design space. We showed the illusions to three groups, each composed of five game designers or game artists. Each participant had more than 5 years of experience in the game industry.

While individual game designers had different preferences, we extracted the commonalities amongst designers. Generally, the game designers thought that F+C Full was impressive at first, but “would probably not hold up to the long term play.” They thought the most promising illusions were ones that altered the

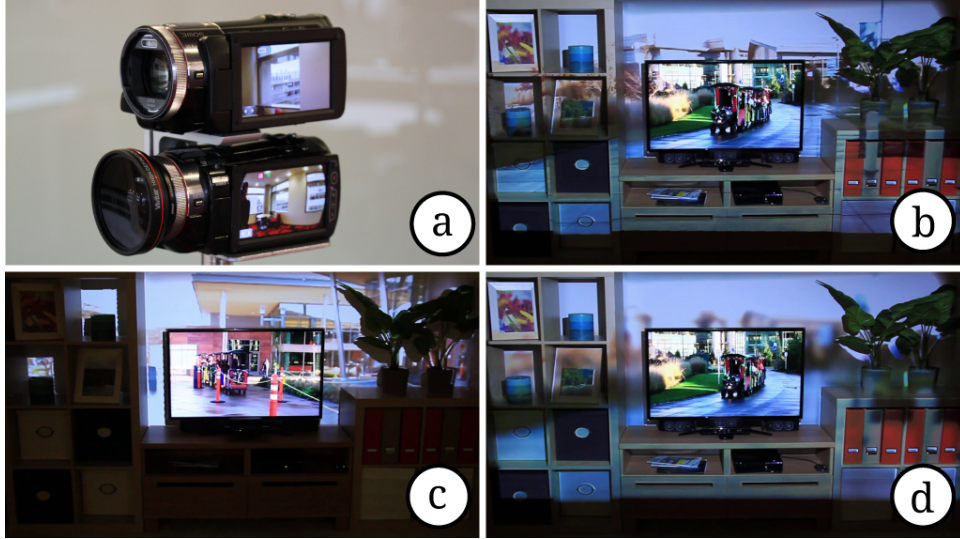


Figure 3.12: IllumiRoom can also be used for panoramic video. (a) A custom dual camera rig for capturing F+C Video (a). The panoramic video content can be displayed as (b) F+C Full, (c) F+C Segmented. (d) Peripheral content can be de-emphasized by blurring (shown with F+C Full).

mood (Appearance), altered the players sense of reality (Radial Wobble), or ones that selectively showed game content without becoming overwhelming (F+C Selective, F+C Segmented, Bounce). They expressed concern over illusions that need a tightly, fine-tuned connection with the game (Lighting); as they were worried about destroying the illusion for the user if the illusion was not perfectly matched to the game content. Therefore, they saw the merits of Snow and Starfield, which have much looser connections with the source content. They all suggested games where IllumiRoom could be applied (e.g., Forza, Skyrim, Halo, Portal) and enthusiastically suggested that they would like to have a setup like this at home.

3.9 Beyond Gaming

All our examples so far have focused on computer games; however, IllumiRoom can also be used to enhance the viewing experience of film and television. In such applications, the user is not in direct interactive control of the experience, but the dual nature of the content can create some interesting cinematic effects. For example, imagine watching a sporting event (e.g., football game) where the ‘focus’ video is tightly held on the players capturing their actions, while the ‘surround’ video is immersing the viewer with live panoramic views of the stands and the fans at the game. Or imagine a music video where the focus and surround video show different footage, but mimic the camera motion (e.g., pan with exactly the same speed and direction).

To begin experimenting with such video experiences, we built a custom dual camera rig, with a narrow FOV ‘focus’ camera and a wide FOV ‘surround’ camera (Figure 3.12). This panoramic content can be displayed around the television using F+C Full or F+C Segmented (with radiometric compensation). Additionally, it can be blurred to decrease the emphasis on the peripheral content. Future video content could be filmed

with such a rig, or existing video content could be extrapolated (e.g. [46, 70]). Our existing experience with recording and watching such dual feed videos in IllumiRoom leads us to believe that a whole new set of cinematic effects could be produced by creatively combining the two video feeds.

3.10 Discussion

While we explored eleven points in the design space, there are many peripheral projected illusions left to be explored. We only scratched the surface of what is possible with a shared physics representation between the virtual and physical environments in Bounce. It is not clear what it means if a grenade falls out of a game and explodes in your living room; particularly, if the virtual camera is moving around and the physical environment is static. Furthermore, as some illusions distort reality, future work should explore how far users are willing to push their concept of reality, in their own living room.

Another promising direction is to further explore illusions specifically for video content (e.g. TV shows or full length feature films). Authoring such content is a challenge, but it could enable unique viewing experiences. Can a grenade from the latest Bond film explode in your living room? How would such content be authored? It would be important to investigate how the movie director should deal with the fixed nature of a film and the randomness imbued by the system adapting to the user's living room.

Additionally, our methods for connecting content and illusions are not exhaustive. It should be possible to use other cues (e.g., audio) to trigger illusions. For instance, a loud explosion in a film could trigger a Radial Wobble illusion. Simple techniques such as these could enable users to 'mod' existing films and games, and share the scripts. Future work could explore this design process from a standpoint of collaborative design or crowd-sourcing. Finally, before the IllumiRoom system can be in every living room, the final form factor, cost and computational requirements of the system must be determined.

CHAPTER 4

ROOMALIVE: WHOLE-ROOM ADAPTIVE PROJECTION MAPPING

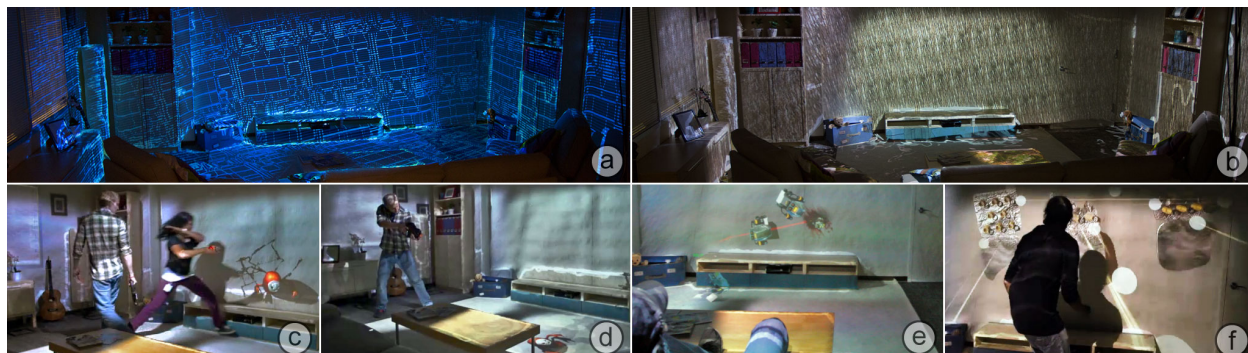


Figure 4.1: RoomAlive is a proof-of-concept system using multiple projector-depth camera units to transform any room into an immersive, augmented, magical gaming experience. RoomAlive captures a 3D scan of the room and adapts the gaming content to stuff in the room. It can transform a room into any environment imaginable including: (a) a Tron inspired game room, or (b) a swampy river. Users can play Whack-A-Mole and physically (c) whack or (d) shoot moles popping out of their floor, walls and couch. (e) Users can control a character running around their room and shooting enemy robots, by using a video game controller. (f) Users can play an adventure game in their living room, avoiding deadly blow dart traps with physical body movement. (All of the images in this chapter were captured live; showing the real-time working prototype).

In this chapter, we present a systems contribution by demonstrating a novel distributed system that enables a unique set of augmented reality experiences in any room. Our proof-of-concept prototype, RoomAlive, transforms any room into an immersive, augmented entertainment experience and enables users to naturally interact with augmented content in their everyday physical environment. RoomAlive builds on a wide array of research fields. Although each individual component of RoomAlive could be improved, the unified system demonstrates the immersive, augmented gaming experiences that might be possible when games can adapt to our room

The basic building block of RoomAlive is a combination of a projector and a depth camera (Figure 2), or procam unit [72]. By tiling and overlapping multiple procam units, RoomAlive is able to cover the room’s walls and furniture with input/output pixels. RoomAlive tracks users’ movements and dynamically adapts the gaming content to the room. Users can touch, shoot, dodge and steer virtual content that seamlessly co-exists with the existing physical environment.

RoomAlive captures and analyses a unified 3D model of the appearance and geometry of the room,

identifying planar surfaces like walls and the floor. This is used to adapt the augmented content to the particular room, for instance spawning enemy robots only on the floor of the room. The content also reacts to the user’s movement. RoomAlive uses a distributed framework for tracking body movement and touch detection using optical-flow based particle tracking [6, 42], and pointing using an infrared gun [49].

To demonstrate the scalability of our prototype system, six procam units were used to cover the floor, walls and furniture of a large living room. We explore the design space of whole-room, augmented interactive experiences and the authoring process for these experiences. Through our exploration, we showcase four example experiences that demonstrate the rich and immersive experiences that are possible with RoomAlive. Finally, we discuss design guidelines and system limitations for future game designers who wish to create adaptive interactive projection mapped games.

A note: this work was done at Microsoft Research in a highly collaborative setting with Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair Macintyre, Nikunj Raghuvanshi, and Lior Shapira. My focus was on the automated calibration system, and general system building for the rendering pipeline [52].

4.1 Motivating Scenario

To illustrate the idea behind the RoomAlive system, imagine playing a video game in your living room that doesn’t happen inside a television, but instead happens in your living room, all around you. When the game starts, the living room magically transforms into an ancient castle, your walls turn to stone, and flaming torches pop out of the walls casting flickering shadows onto your furniture. Out of the corner of your eye you see a glowing idol magically appear on your couch. You creep towards the idol. Suddenly, a trap opens on the wall next to you, exposing blow darts ready to fire. You leap out of the way, only to land on the floor face-to-face with a giant cockroach. You quickly get up and jump on the roach, sending bug guts splatting across your living room. You reach the idol successfully, and a scoreboard drops down showing that you just scored the best time for the adventure course. This is just one of many interaction scenarios that are made possible by RoomAlive.

4.2 RoomAlive System

The RoomAlive system is comprised of multiple projector-depth camera units, or procam units. Each unit contains a depth-camera (which contains an RGB camera, IR camera and IR emitter), a commodity wide field-of-view projector, and a computer. A single unit can be used in isolation to re-create a set-up like IllumiRoom [48], or multiple units can be combined to canvas an entire living room in I/O pixels. Our current proof-of-concept prototype demonstrates the RoomAlive concept in a large living room (18 ft x 12 ft), with six procam units overlapping to cover three walls, the floor and all the furniture in the room, see



Figure 4.2: Each procam unit consists of a commodity wide field-of-view projector, depth camera and computer.

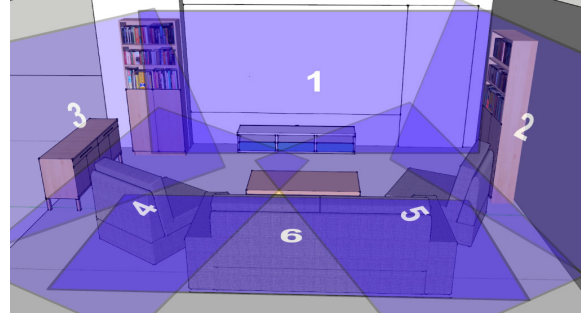


Figure 4.3: . We have installed RoomAlive in a large (18 ft x 12 ft) living room using six overlapping procam units, with three units pointed towards the three walls of the room, and another three are pointed downwards to cover the floor.

Figure 4.3.

RoomAlive is implemented as a plug-in to the Unity3D commercial game engine. Unity3D enables designers to easily author game content using an intuitive 3D editor and an interactive development environment. There is an active developer community with over two million developers and an assets store for easy sharing of created content. Within a few clicks, game designers can integrate in a wide range of content and rapidly prototype interactive experiences.

With RoomAlive procam units are connected through a distributed client-server model. Each client node is responsible for tracking the players and the room geometry within the view of its own local depth sensor. The clients are also responsible for any un-synchronized local game state (e.g. intensive physics simulations) and for rendering their local view of the global scene, including view dependent rendering. The master server node synchronizes game state and user tracking state across units. The master server node also acts as a client, handling its own game state and rendering.

4.2.1 Procram Hardware

To demonstrate the flexibility of the approach, the procam units are built using a variety of commodity wide field of view projectors (InFocus IN1126, BenQ W770ST, BenQ W1080ST). Each unit also contains a Microsoft Kinect for Windows sensor. As the projector and depth camera must overlap to create I/O pixels, any projectors used must be greater than or equal to the field of view of the Kinect (57° horizontal).

In our prototype, each unit is connected to its own personal computer with a high-end consumer grade graphics card (e.g. NVidia GTX 660 Ti). We use high-end personal computers to explore rich, computationally intensive interactions (e.g. GPU based optical flow) that will be possible in smaller form factors in the future. All procam units are currently mounted to the ceiling of the living room, however the approach is agnostic to the mounting solution. Procams could also be mounted on a tripod or placed on a bookshelf

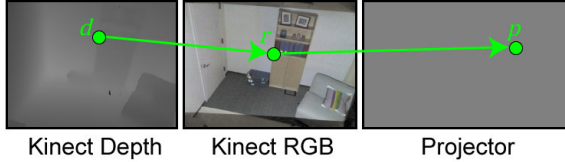


Figure 4.4: To establish correspondences between the projector and 3D world points, we map a depth pixel (d) to an RGB pixel (r) via the transfer map provided by the Kinect SDK. We then look up the corresponding projector pixel (p) via the decoded Gray codes. This is done for each depth pixel, giving dense correspondences between 3d world points and 2d projector image points.

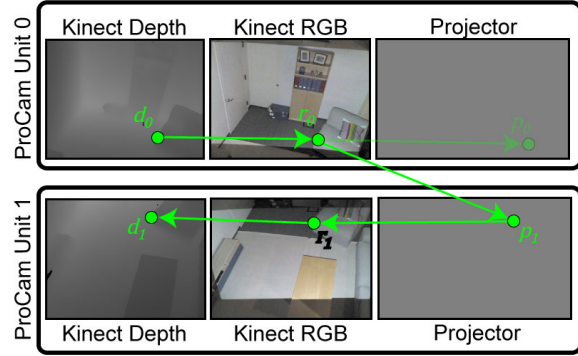


Figure 4.5: To compute correspondences between two units (Unit 0 and Unit 1), we map a depth pixel in Unit 0, (d_0), to an RGB pixel in Unit 0, (r_0). Next, we look up the corresponding projector pixel in Unit 1, (p_1), via the decoded Gray codes. We then invert the Gray code correspondences to look up the RGB pixel in Unit 1, (r_1). Finally, we invert the Kinect’s transfer map resulting in depth pixel (d_1).

or a coffee table, or generally any location that maximizes coverage and minimizes occlusions.

4.2.2 Automatic Calibration

In order to display virtual projected content in precise registration with the physical environment, we must calibrate each projector and depth sensor. RoomAlive’s calibration process is completely automatic, requiring no user intervention and is distributed across multiple procam units. Installers simply need to mount the procam units in the room, and the automatic calibration does the rest of the work. The calibration is bootstrapped off the “factory calibrated” Kinect Sensor that provides metric 3D measurements.

Internal Calibration

Each procam unit is first internally calibrated establishing intrinsic (e.g. field of view) and extrinsic (rotation & translation) parameters of the projector and depth camera. This is accomplished via an automatic projector, depth-camera calibration process adapted from [48]. First, the projector displays Gray code sequences [4], to establish dense correspondences between the projector’s image and the Kinect’s RGB camera. These Gray codes are decoded, resulting in a projector row and column for each Kinect RGB pixel (rejecting invalid code pixels). We then transform each correspondence point into the Kinect’s depth image, via a transfer map provided by the Kinect SDK, which is a black box function that is specific to the depth of the scene being imaged. We now look up the depth value associated with each correspondence, resulting in dense correspondences between 2D projector image points and 3D world coordinates (see Figure 4.4). In practice, the Kinect’s depth map is noisy at large distances, therefore we take a series of 60 depth frames.

We then reject any depth pixels that are inconsistent across the sequence of frames, taking the average of the reliable pixels.

We then solve for the intrinsic and extrinsic parameters of the projector using the dense correspondences. We use a Levenberg-Marquardt optimization algorithm which uses an initial guess at the projector intrinsics and minimizes the global re-projection error. This will only converge to a unique solution if the scene being imaged is non-planar. In our example living room, each procam unit has at least one non-planar element in view (e.g. a corner of a room, a chair, etc.) If the procam unit is viewing an entirely planar surface, the unit must be temporarily rotated to image a non-planar scene just for internal calibration.

The optimization can be sensitive to outliers in the correspondences, which can be caused by errors in the factory Kinect calibration, transfer maps or misread Gray codes. We therefore run this optimization as the inner loop of a RANSAC [30] procedure, to make the process robust to outliers. Furthermore, in the RANSAC procedure we only take initial guesses where the data points are non-coplanar. This results in a robust estimate of the intrinsics and extrinsics of the projector with respect to the depth map.

External Calibration

Next, we must calibrate between the procam units, solving for the rotation and translation of each unit with respect to a common global coordinate system. We establish dense correspondences between every pair of procam units with Gray code sequences. Each procam unit displays its Gray code sequences and they are simultaneously captured by all other procam units. By chaining correspondences together (see Figure 4.5), we achieve correspondences between 3D points in each pair of procam units. We actually have two sets of correspondences for each procam pair (forwards and backwards links), which we merge into a single set.

We estimate a rigid transform (rotation and translation only) between each pair of procam units using a singular value decomposition fit as the inner loop in a RANSAC procedure. We refine this fit via iterative closest point between the two sets of 3D points.

In order to establish a single global coordinate system, we need a minimal set of links between units that connect all units. To do this, we compute a minimal spanning tree, using the inverse of the number of inliers in the estimated rigid transform as the weight of each edge between units. We choose a node as the global coordinates system (Unit 0) and then chain together the pairwise transformations into a global transformation for each unit. To simplify content creation we rotate this coordinate system such that gravity points downwards, via Unit 0's Kinect's accelerometer. The end result is that we've roughly localized each procam unit, to within a few centimetres of its actual location in the room.

4.2.3 Automatic Scene Analysis

Following the calibration, we know the physical position and location of each unit in the room. We also have an average depth map acquired by each unit. Combining this information, we have a dense point cloud

of the room in a global coordinate system. To automatically map content to any arbitrary room, we would like to provide higher level information about the physical environment to game designers. To do so we extract the flat surfaces in the room out of the global point cloud, resulting in a multitude of polygons, see Figure 4.6.

We employ a greedy strategy, starting from an empty set and repeatedly adding planes. As a pre-processing step, we calculate the normal associated with each depth point via principal component analysis of a small window around each depth point. We then use the Hough transform [45] to select a finite set of planes. In our method, each 3D point and its surface normal votes for a plane equation parameterized by its azimuth, elevation, and distance from the origin. Each of these votes is accrued in an accumulator matrix.

Once we have a set of candidate planes we sort them in descending order by the number of votes they have received and iteratively associate points to each plane. A point can be associated to a plane if it has not been previously associated to any other plane and if its planar disparity and local surface normal difference are small enough (10 cm, 0.1 angle). As an additional heuristic, each new plane and its associated points are broken into a set of connected components ensuring that planes are locally connected.

Each polygon is then assigned a local (u,v) coordinate system, to be used as texture coordinates. For vertical planes, one axis is chosen to align with the gravity direction, while the orthogonal axis is parallel to the ground. Horizontal planes receive a (u,v) coordinate system that follows the principal axes of the polygon.

This process results in polygons that are unified across scans, for instance identifying in each procam unit the 3D points that belong to a single wall in the room. We also provide coarse semantic information by labeling these polygons as ‘vertical’, ‘horizontal’ or ‘other’ based on the orientation of the surface with respect to gravity. We also specifically identify the floor as the lowest horizontal plane. This enables designers to incorporate rough scene knowledge into their games. For instance, designers can tag a game object to automatically appear on the floor and only the floor.

4.2.4 Authoring Augmented Content

RoomAlive makes use of Unity3D’s modular plugin framework and scripting interface. Game designers need to simply add a single game object to their scene, which in turns loads the RoomAlive plugin. The game designer then has access to the API of RoomAlive directly within the scripting interface of Unity3D. When the designer presses the Play button, RoomAlive automatically loads in a high resolution 3D model of the room as seen from the current procam unit. The game window can be positioned on the projectors display to live preview the content in the room, augmented onto the physical environment. Alternatively, designers could load in a pre-scanned living room for testing purposes.

Game art assets can easily imported from external content authoring software (e.g. Autodesk Maya) and dragged directly into the augmented environment and previewed in situ. Behaviors can then be applied to

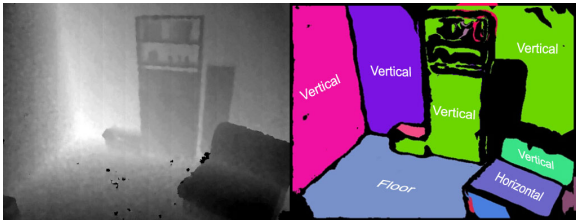


Figure 4.6: Planar surfaces are extracted from a unified point cloud of the living room environment. These planes are then classified as vertical, horizontal, floor and other, and then each 3D depth point is associated with a polygon (or no polygons at all).

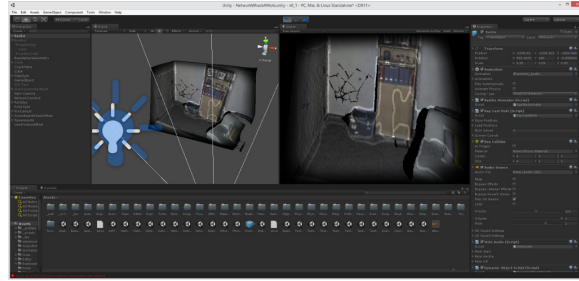


Figure 4.7: In the Unity3D editor’s Play mode (debug mode), a high resolution model of the current living room is automatically instantiated, along with virtual cameras (with correct intrinsics/extrinsics) for all the procam units in the scene. On the left is the scene view where artists can drag and drop game assets and view the game from any arbitrary viewpoint. On the right is a live-preview of the image that the projector will display.

a game object through a C# scripting interface, e.g., how a virtual object reacts to gun fire. These scripts have access to the RoomAlive API, so they can do things like query whether they are currently on the floor, or listen for touch collision events from players.

4.2.5 Mapping Content

One of the key technical challenges of making a projection based experience work in any living room is the question of where to place virtual game elements. Unlike traditional game design where elements like the game terrain are known a priori, RoomAlive experiences must operate in any living room. The mapping of game elements to the physical environment must be done at run-time. As stated previously, when in debug mode the current living room is loaded into the Unity3D editor. However, when not in debug mode, no physical environment appears in the editor. This is to enforce that the game must be created independent of the particular room that it is played in. Instead content must be mapped dynamically based on a set of rules. RoomAlive employs four techniques for mapping: uniform mapping, semantic mapping, user constrained mapping, and user driving mapping.

The easiest technique to employ in a living room is to map content in a uniformly random way (uniform mapping). For example, if an experience involves a user shooting targets, the targets can be mapped to randomly appear anywhere around the room. Alternatively, additional semantic information can be used to map content (semantic mapping). For instance, grass and rock game objects could be mapped to only appear in locations on the floor of the room. To accomplish this, the game script queries the RoomAlive API, on start-up, to supply a list of points that belong to the floor of the room, which can then be sampled to instantiate the game content. Content can also be constrained based on the user position or user tracking

state (user constrained mapping). For example, a canon programmed to shoot a user in the head, can be dynamically placed at the user’s eye-level in the room. Finally, users themselves can arrange content in the room (user driven mapping). Users can spawn content during a gaming experience by touching a physical surface or pointing with a gun at surfaces in the room. This enables users to level-edit their games, or simply re-decorate their room.

4.2.6 Tracking User Interaction

RoomAlive supports interacting with augmented content by whole body movement, touching, stomping, pointing/shooting and traditional controller input. For computational efficiency, processing is done locally on each client procam unit, and only resulting changes in game state are synchronized across units.

Proxy Particle Representation

To enable interaction through body movement, touching and stomping, we use the captured depth map along with a real-time physics simulation. Using a similar approach as [6, 42], moving objects (e.g., players) in the room are represented by a cloud of spherical ‘proxy particles’, that are capable of exerting frictional forces. This enables physically realistic interactions with virtual game objects. For instance, a collision event can be triggered by the proxy particles when the user touches or stomps the physical environment.

The proxy particles are tracked using a depth-aware optical flow algorithm. The 3D flow field pipeline uses a GPU implementation of Brox’s algorithm [16] to compute optical flow on the captured 2D depth video, updating the proxy particles to follow moving objects. While flow is typically computed on the RGB image, we do not use the color video, as projected content can lead to incorrect flow results. The computed 2D flow field is re-projected onto the depth data to generate the 3D displacements of the proxy particles.

Gun/Pointing Input

A peripheral gun controller was also used to support at a distance pointing tasks. The gun controller uses an integrated infrared LED which is activated when the user pulls the gun’s trigger. Optics within the gun focus the light, similar to a flashlight. The light is observed by the Kinect’s infrared camera, as the LED is precisely matched to the wavelength of the Kinect’s infrared band-pass filter. The infrared camera captures a background image of the scene and checks for changes in infrared scene illumination to detect the location of the gun’s infrared light, this is then converted into a 3D location via the Kinect’s depth map.

Traditional Controller Input

In addition to natural user interactions, RoomAlive also supports traditional physical game controllers, such as a Microsoft Wireless Xbox controller. This allows users to interact with whole room augmented games

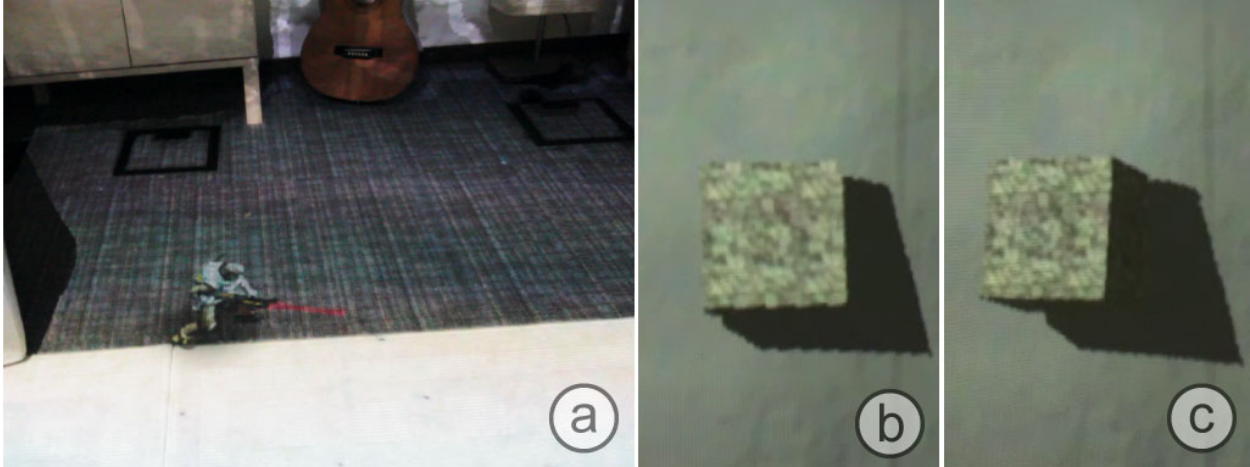


Figure 4.8: (a) Virtual objects that exist off-surface like this game character are rendered in a view dependent manner with radiometric compensation. (b) A virtual cube placed in front of a wall and viewed straight on. (c) The same cube viewed from a more oblique angle to the right.

using the same input affordances as traditional television gaming experiences. The controller is connected to the server, which distributes user interaction to the clients.

4.2.7 Rendering

Projection mapped content can only physically appear where there is a physical surface. However, virtual content can appear to be at any arbitrary 3D location, by displaying a perspective rendering of the virtual content on the surface, from the view direction of the player.

RoomAlive tracks the player’s head position and renders all virtual content with a two-pass view dependent rendering [11] (see Figure 4.8). Content that is aligned with the physical environment can be rendered realistically without a need for view dependent rendering. A unique challenge is the possible presence of multiple users. While multi-user viewpoint rendering remains an open problem [86], RoomAlive uses a simple approach by averaging users’ head positions. In practice, 3D scene geometry that strays farther than existing physical surfaces makes RoomAlive a single-viewer experience. For scenes where the virtual content is near the physical surfaces, rendering with the average head position offers a good working solution [36].

Another challenge arises from the use of existing room furniture as projection surfaces which may contain a non-white surface albedo. To overcome this problem, a radiometric compensation [76] is applied to compensate the projected image for the color of the surface. This process is limited by the brightness, dynamic range and color space of the projector and some desired surface colors maybe unachievable. For example, a solid red object in our physical environment cannot be made to appear completely green.

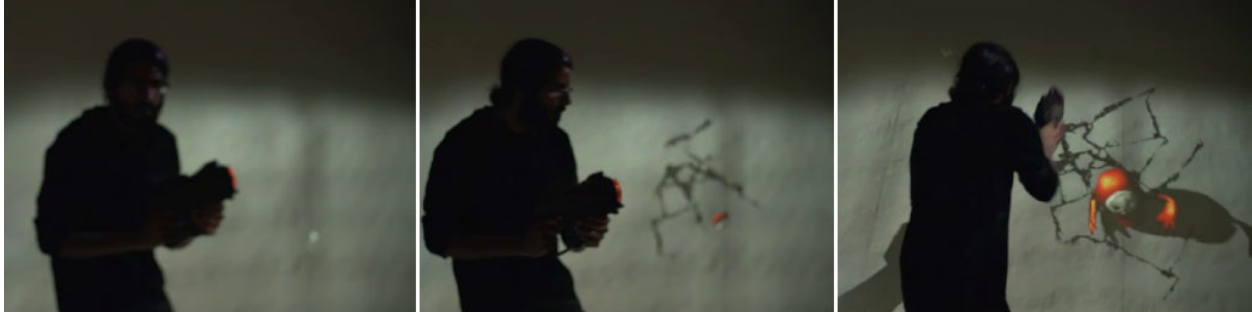


Figure 4.9: A user playing the Whack-A-Mole experience with an infrared gun.

4.3 Example Experiences

We envision the RoomAlive system supporting a diverse range of applications, limited only by the imagination of game designers. We have prototyped four example applications that demonstrate the potential of whole-room augmented experiences. These experiences were developed in partnership with a game designer, and were arrived at through numerous brainstorming sessions and iterative designs. These experiences represent an initial survey and are not an exhaustive list. We encourage the reader to see the accompanying video to best understand the mechanics of the experiences.

4.3.1 Whack-A-Mole

The Whack-A-Mole experience demonstrates a combination of whole body movement, touch and gun/pointing input. Similar to the popular arcade game, in Whack-A-Mole, users race to whack, stomp, and shoot moles that pop up randomly in the living room. The moles are generated uniformly across the entire room. First, a crack appears on a surface in the room (see Figure 4.9). Then an audio clip plays, taunting the user “I’m over here”, attracting the user’s attention to a specific part of the room. Finally, a 3D animated mole bursts out of the wall/floor, casting appropriate shadows onto the physical room and rendered with view dependent rendering. Figure 4.1c shows a user whacking a mole on a wall and Figure 4.1d shows a user shooting a mole on the floor. A virtual scoreboard on the wall keeps count of the number of points the user achieves.

4.3.2 Angry Bots

RoomAlive also supports non-physically realistic gaming mechanics that create entirely new gaming experiences in the living room. Angry Bots allows a user to control a virtual character that can run across the walls, floor, sofa, etc. (see Figure 4.10 and Figure 4.1e). The character is entirely constrained to the surfaces of the living room. The character can move forwards, backwards, left, right on the surface, but not off of the surface. As the character moves around the living room, surface constrained enemy ‘angry bots’ appear, tracking and firing weapons at the virtual character. The character must defend and shoot back against

the angry robots. The surface constrained nature of the experience enables the game characters to walk up walls, adapting their orientation to the normal vector of the surface. Weapons also follow a similar behavior, where ‘bullets’ remain surface constrained going up and around objects rather than bouncing away. This type of surface relative control scheme is a natural fit for using a traditional game controller.



Figure 4.10: (a) A virtual character runs up the side of a bookshelf. (b) A virtual character fights a bot, splashing battle damage.

4.3.3 Traps

While the previous experiences demonstrate virtual objects near a physical surface, virtual objects can also appear unattached to a physical surface. In Traps, a user is surrounded by virtual traps and must navigate the physical environment in order to escape being hit by virtual darts. This experience is inspired by adventure games, and simulates the perils that Indiana Jones must face. If a user navigates through a trigger volume, darts are emitted and collide with physics enabled proxy particles that represent the user and any dynamically moving object. Because the darts are free space virtual objects, they rendered view dependently. If the user is hit, blood particles are emitted at the collision location, see Figure 4.11.



Figure 4.11: Blow dart traps pop out of the wall, forcing the user to dodge. If the user is hit, battle damage shows on their body.

4.3.4 Setting the Stage

Imagine being able to instantly transform your entire room to match the environment of a video game or film. RoomAlive can use projected light to automatically augment the appearance of the entire room, procedurally changing the surface color of objects in the room environment. Figure 4.1 (a-b) shows two example procedurally generated environments. Any Unity3D material, including procedural materials, can be easily drag-and-dropped onto a polygon of the room geometry. The pre-generated (u,v) coordinates are used for each room polygon. Also, game designers can specify semantic groups to be given a texture (e.g. vertical surfaces). We also demonstrate special effects that are easy with 3D scene information. With a click of a button, a game designer can make it rain in a living room or send cock-roaches scuttling over every surface, see Figure 4.12.

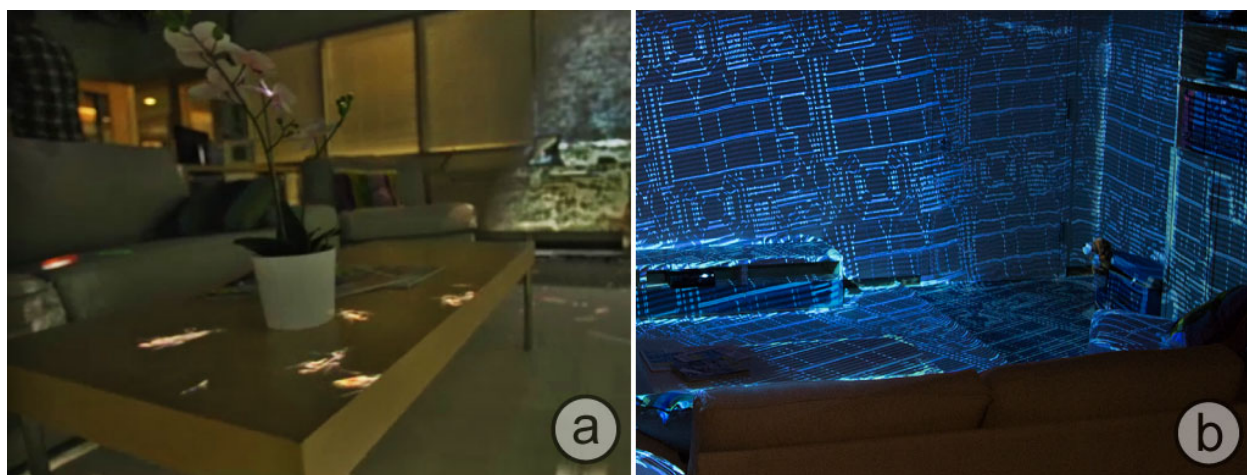


Figure 4.12: The room is transformed (via the extracted polygons and textured coordinates) into: (a) a cockroach infestation and (b) a Tron style game room.

4.4 Design Considerations

In order to create experiences for RoomAlive, game designers must consider the unique benefits and challenges associated with interactive, adaptive, projection mapped game content. We now highlight four critical aspects that should be considered when designing room sized augmented experiences: input choice tradeoffs, capturing user attention, game physics, and controlling player movement.

4.4.1 Input Choice Tradeoffs

RoomAlive supports a wide variety of techniques for input. In game experiences that require physical contact, users can provide input by touching or stomping. These interactions can be used to deliver experiences where users are constantly moving around the room promoting the physicality of the game experience.

However, certain parts of a living room environment may be less suitable for touch based experiences. For example, an expensive vase or painting may be inappropriate for direct contact. Particularly for children’s games, designers may want to limit touch interaction to the floor and walls (large vertical planes). Alternatively, the experiences could rely solely on whole body movement, pointing/shooting or traditional controller input. Pointing and traditional controller input are ideal for controlling objects at a distance, e.g. a virtual car driving across the wall. We generally found that experiences seemed the most fun when all forms of input were available to users, and users naturally do not whack fragile objects.

4.4.2 Capturing User Attention

When the display surface consists of a large living room environment, directing a user’s attention becomes a critical aspect of designing the experience. If a user is on one side of the living room, a user may not realize that their attention is needed on the opposite end of the room. Surround sound can play a large role here, directing the user’s attention to a rough location in the room. Also the designer can incorporate a warm-up animation into the game design, for instance in Whack-A-Mole, a large crack appears seconds before the mole pops out, to attract the user in their peripheral vision.

4.4.3 Selecting Game Physics

Living room projected experiences enable game designers to create unique non-physically realistic interactions. The concept of gravity can be applied globally to all virtual objects in an experience as is done traditionally in a typical game. Alternatively, gravity can also be applied locally to a virtual object, constraining the object to a surface [49]. For example, a ball rolling on a surface can bounce off the wall or continue rolling up the wall depending on the desired outcome. A local surface adherence model requires a local coordinate system, where the virtual objects local “floor plane” lies tangent to a surface. For instance, a virtual character’s “front” would be its current ‘front facing’ direction and the character’s up direction would be the normal perpendicular to the tangent plane on the surface.

4.4.4 Controlling Player Movement

In traditional video games, the user is represented as an avatar in a virtual world. The game designer has explicit control over the actions that are possible in the virtual world, including where the user’s avatar can walk, what objects they can interact with, etc.

In a RoomAlive experience, a player’s movement and actions are completely controlled by the user, and therefore uncontrollable by the game designer. For instance, imagine a game that involves building physical pillow forts for a virtual snow fight. The game designer cannot stop a user from walking over to another user’s pillow fort and knocking it over. Similarly, a game designer cannot control where a user walks or what



Figure 4.13: Calibration errors between units result in ghosting artifacts only in projector overlapping regions. (left) The character in a single projector, and (right) in an overlapping region.

items in the room they interact with, everything is fair game. Therefore, the designer must take care to handle edge cases and ensure that the game mechanics themselves guide the user into the desired behavior.

4.5 System Limitations

While RoomAlive enables new and exciting interaction possibilities in the living room, there are several challenges with the current implementation. Foremost are calibration errors caused by non-metric distortions in the raw Kinect data and by deviations from factory calibration settings in the Kinect’s own internal calibration. This results in visual errors that causes virtual content to appear ghosted in overlapping projector regions (see Figure 4.13). In practice the overlap regions are small, and we smoothly blend between projector images, so visual artifacts are not as noticeable. In the future, more precise depth estimates of the Microsoft Kinect V2 should reduce the sensing noise and ensure better calibration between procam units. More advanced calibration techniques [1, 55, 59, 84] could also be used to correct for imperfections in calibration.

Finally, a key challenge in supporting a wide variety of interactions is handling tracking issues that arise from overlapping Kinect sensors. We found the tracking results from the skeleton tracker to be relatively unreliable in overlapping regions, and therefore used the optical flow proxy particle tracking approach. Existing structured light depth cameras can be mechanically augmented to improve noise [17]. New time-of-flight depth sensors (e.g., Microsoft Kinect v2) may have less noise in overlapping regions. Advanced approaches for pose and dynamic object tracking can also be used to improve the RoomAlive tracking system [24, 92].

4.6 Summary

In this Chapter, we have presented RoomAlive, a novel system for transforming a full living room environment into an interactive display. By calibrating and unifying multiple uniquely situated projector camera units, we demonstrate how a distributed rendering approach can be used to seamlessly merge our physical and

virtual worlds. Through the development of a range of example experiences, we have shown wide and varied interaction possibilities where RoomAlive can be put into the real world use creating a truly magical entertainment experience.

In the future we will explore approaches and experiences that involve multiple users and multi-user view-point rendering. We will also explore new ways to increase immersion with spatialized sound. RoomAlive provides a glimpse of how a scalable multi-projector system can transform any room into an immersive augmented gaming experience. Many new and exciting possibilities remain to be explored.

CHAPTER 5

PROJECTIBLES: OPTIMIZING SURFACE COLOR FOR PROJECTION

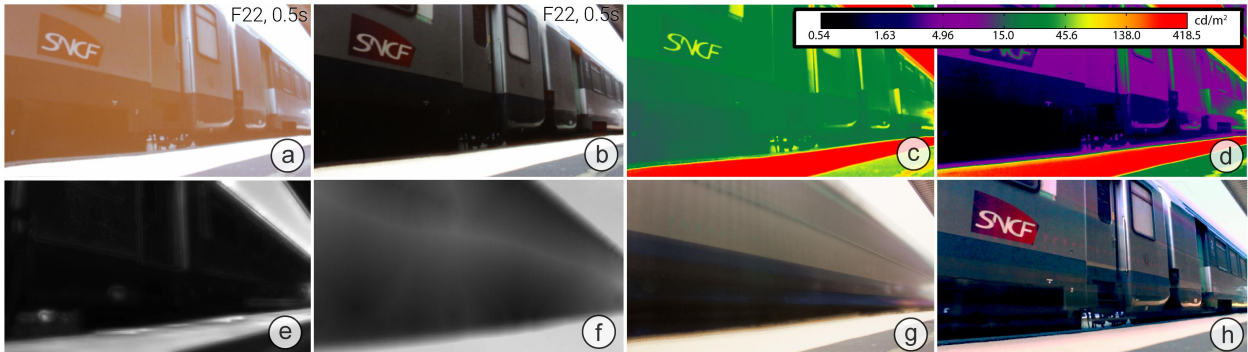


Figure 5.1: Results for a looped video sequence of a train passing by a station (see Figure 5.2a for source video). (a, b) Shows real world captured images ($F22, 0.5s, 800 ISO$) of (a) a normal projected image on a white screen, and (b) the Projectible combining projected light with a printed display surface. (c, d) Log plots of the real-world measured luminance in cd/m^2 for (c) the projector only, and (d) the Projectible display. We compute a perceptual error metric (e) for each frame of the sequence, and use it to smoothly vary (f) the contrast across the display. We then separate out the printed image (g) and the projected image (h), which combine to form the Projectible.

Typically video projectors are used with white screens (e.g. in conference rooms). Movie theaters frequently use gray screens to get darker black levels. Making the screen darker makes the content darker, effectively multiplying the appearance of the display by a constant. For darker video sequences like a city in twilight, a darker screen would look better because of darker black levels, but brighter sequences like a sunny sky would look better with a lighter screen. As displays are designed to show both bright and dark content, there is no optimal screen color.

In this chapter, we present *Projectibles* which use a spatially varying display surface color for video projection, greatly increasing the contrast and resolution of the combined display. For a particular video sequence, we algorithmically determine a static printed image, then overlay video projection onto the printed image, creating a high dynamic range (high contrast), high resolution display that can be larger than flat panel displays, have any contour (non-rectangular), and can even be 3D (non-flat).

Projectibles combine two low dynamic range (LDR) displays, a printer and projector, to create a high dynamic range (HDR) display. High dynamic range is important because the dynamic range of many real-world scenes exceeds the capabilities of current display technology by several orders of magnitude [87].

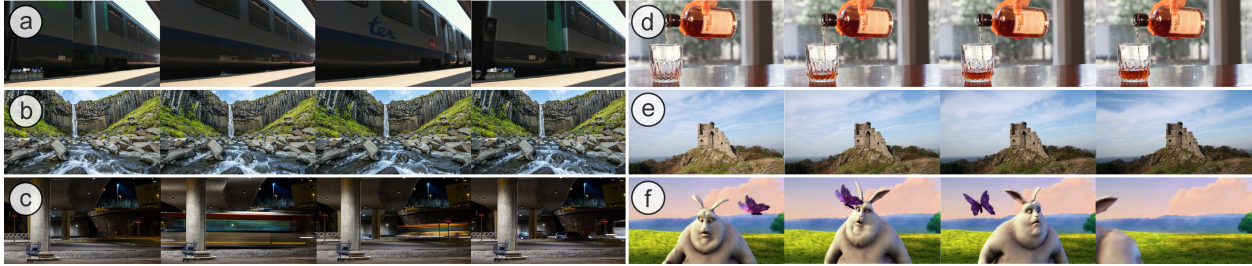


Figure 5.2: Example frames of the looped video sequences (see <http://brettrjones.com>) (a) *Train*: an exposure bracketed HDR video shot with a DSLR of a train passing a station. (b) *Svartifoss*: an ultra resolution [7360,4140] RAW time-lapse of the Svartifoss waterfall (credit: Stian Rekdal, <http://www.stianrekdal.com>). (c) *Underpass*: a high res time-lapse video of an underpass at night (credit: Eric Solheim, <http://eirikso.com>). (d) *Whiskey*: A video of pouring whiskey made by combining a high res still image with 1080p video. (e) *Castle*: A high res time-lapse of Mow Cop Castle (credit: NatureClip, <http://natureclip.co.nr>). (f) *Bunny*: An animated sequence from Big Buck Bunny (credit: Blender Foundation, <https://peach.blender.org>).

For instance, the sun is 8 orders of magnitude brighter than twilight. Traditional displays simply cannot reproduce the feeling of seeing a sunset in real life, or the twinkling lights of a city at night.

Projectibles have the most effect on videos with some spatial and temporal consistency. These types of videos are often called *Cinemagraphs* or *Cliplets* [53, 60], which are a recent trend in art and advertising that combine still images with video sequences, and are often distributed as ‘.gif’ sequences or looped videos (for examples, see Figure 5.2 and the supplementary video¹). These videos are frequently used for advertising, as the subtle motion is great at capturing attention.

With Projectibles, for each pixel in the video, if the pixel has similar colors through time, we can alter the screen color at this location to make the darks darker or the brights brighter. For instance, Figure 5.1 shows a looped video sequence of a train passing by a station. Parts of the image are consistently brighter (the sky), and others are consistently darker (the bottom of the train). We can determine a static printed image to serve as the display surface for video projection that increases the contrast of the overall video. The process of determining the printed image is non-trivial as it is limited by the dynamic range of the printer and projector, and the amount of motion in the input video. An incorrect choice of the printed image could lead to incorrect colors in the combined display.

Projectibles can be much higher resolution than traditional digital displays. Printers have a much higher effective resolution than projectors. Projectibles use the printed image to increase the resolution of the display in regions with temporally consistent high resolution details. For instance, in the waterfall sequence (Figure 5.2b) while the water and sky are moving, the cliffs and moss are stationary with only lighting changes from passing clouds. By printing high resolution details of the cliff and moss, the resolution of the combined display can be increased in these regions (see Figure 5.6). Projectibles have spatially varying contrast and resolution, with large improvements in contrast and resolution in areas with temporal consistency throughout the video.

¹see <http://brettrjones.com>

The core problem of Projectibles is to decompose an input video into a static (printed) component and a dynamic (projected) component. Naive algorithms for decomposition result in distracting visual artifacts, or poor contrast/resolution. We present two algorithms for splitting a video into a static and dynamic component, based on models of human perception, that maximize the contrast ratio and resolution, while minimizing visual artifacts. We present an offline constrained optimization method, and a faster heuristic solution which enables the content author to adjust the parameters for the solution in near real-time. Finally, we present empirical measurements of real-world results, along with subjective ratings of viewer feedback, and discuss the benefits and limitations of Projectibles. Overall, this is the first approach to combine a static display and a dynamic display for video, and the first to optimize surface color for the projection of video.

A note: This work was done at the University of Illinois at Urbana-Champaign in collaboration with Rajinder Sodhi, Pulkit Budhiraja, Kevin Karsch, Brian Bailey, and David Forsyth.

5.1 Background

Most closely related to our work, Bimber et al. [9] demonstrated a static HDR display by combining a static printed image with a static projected image. For static images, splitting into printed and projected is as simple taking a square root of the input after linearizing each device. As we will describe later, this approach only works for static images and will lead to visual artifacts for video sequences. We derive inspiration from this approach, and create algorithms that work on video sequences.

Previous work has also investigated combining two dynamic LDR displays to create an HDR display. For instance, by stacking two LCD panels on top of each other [33], combining LCD panels and LED back-lights [95, 87, 88], or LCD panels with projectors [87]. These displays are capable of creating HDR scenes for arbitrary input videos and therefore show great promise, but require expensive custom hardware solutions. Projectibles uses commodity hardware to create HDR displays for looped video sequences, and unlike previous work Projectibles can be made into any shape or contour.

Projectibles optimize the surface color for projection, but a wide range of previous work has established methods to compensate the projected image for an existing surface color. If we want to achieve a desired appearance, we can model the surface reflectance and the incoming light in the scene. Then given an existing surface color, we can solve for the required projected light. The lighting model can be as simple as a Lambertian model [32, 63, 2, 99, 58], include color mixing between projector channels [68], or be as complex as a global illumination model [101, 89].

Usually, radiometric compensation is accomplished by using a camera to establish a geometric transfer function between camera pixels and projector pixels, often using structured light [58, 4, 96]. In this chapter, we focus on the core problem of optimizing the surface color and we use a homography for geometric registration. Future work could extend our approach to support more complex geometric registration.

Unfortunately, radiometric compensation is not perfect and is limited by the existing surface color and the

dynamic range of the projector. This results in distracting visual artifacts. One solution is to use multiple projectors [1], increasing the brightness and decreasing artifacts, but also increasing setup complexity and system cost.

In this work, we combine a wide background of previous work in radiometric compensation, building on previous methods that optimize the black/white level based on the surface color, the projector dynamic range, and mathematical models of human perception [20, 63]. We also build on content-dependent methods that optimize over the particular video being shown, particularly previous work in offline optimizations [99, 2, 58] and real-time systems [32]. This previous work all assumes an existing fixed display surface color. We take parts and pieces from each method, and present two novel algorithms for optimizing the surface color for projection of a video sequence.

5.2 Lighting Model

In this chapter we use a Lambertian lighting model, as the display surface is diffuse professional quality matte paper. We use a homography to register the projected content onto the printed page. Initially, we will assume that the printer and projector are equal in resolution.

We adapt the lighting model for a projector-camera system from [54]. At each pixel on the display surface (x, y) , and for each frame t of the video sequence, the final appearance $L(x, y, t)$ can be modeled as follows,

$$L(x, y, t) = k(s(x, y)) \left(i(d(x, y, t)) f(x, y) + a(x, y) \right). \quad (5.1)$$

The reflectivity coefficient of the surface (i.e. surface color) is defined by k , the printer’s spatially invariant transfer function. It is a function of the normalized input *static* image $s(x, y)$, sent to the printer. The contribution from incoming light can be broken down into three components. First, $i(d)$ is a spatially invariant transfer function that describes the color response of the projector, where $d(x, y, t)$ is the normalized input image for each frame sent to the projector. Second, $f(x, y)$ is a spatially variant function incorporating the vignetting in the optics of the projector, the form factor between the projector and the surface normal, and light falloff. Finally, the contribution of ambient light, $a(x, y)$, is a spatially variant function over the display surface, incorporating all incoming ambient light and display surface form factors.

To measure the above terms, we use a ColorMunki spectrophotometer and HDR photographs from a DSLR camera. We measure k and i directly using the spectrophotometer. We measure f by taking an HDR exposure bracketed image sequence of a projected white image without ambient light. We measure a with an HDR image of the ambient light only.

After calibration, we simplify the lighting equation by creating two new terms representing the overall contributions from the printed component $K(x, y) = k(s(x, y))$ and the projected component $D(x, y, t) =$

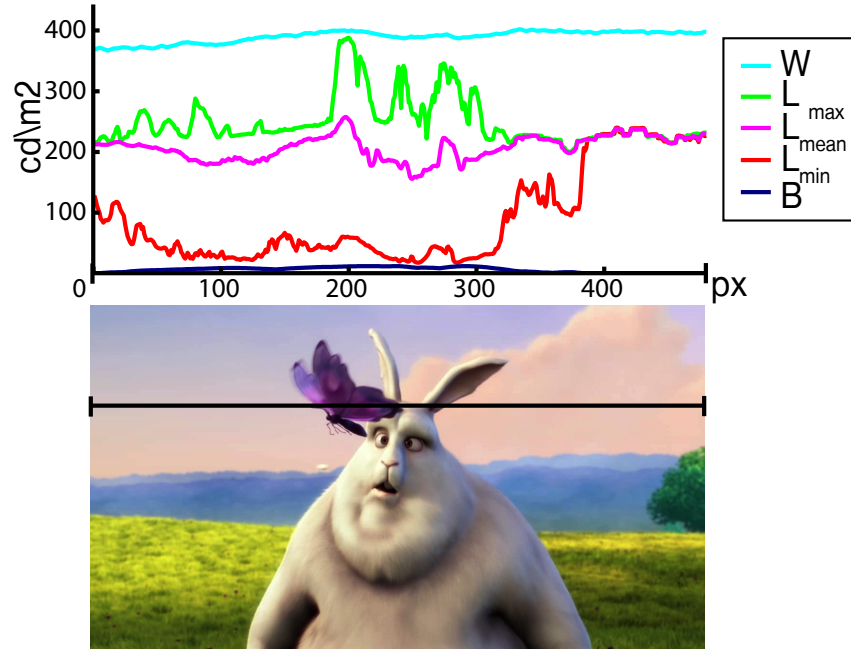


Figure 5.3: Statistics of the luminance of a single row (noted in black) from the Bunny sequence. The butterfly flies across the screen to the left, and the bunny exits to the left. The right quarter of the video is static. We plot the white level $W(x, y)$, black level $B(x, y)$, maximum luminance $L_{max} = \max_t(L_g(x, y, t))$, minimum luminance $L_{min} = \min_t(L_g(x, y, t))$ and mean luminance $L_{mean} = \text{mean}_t(L_g(x, y, t))$.

$i(d(x, y, t))f(x, y) + a(x, y)$. The lighting equation becomes

$$L(x, y, t) = K(x, y)D(x, y, t). \quad (5.2)$$

5.3 Problem Formulation

The core challenge of Projectibles to decompose a dynamic input video into a static printed component and a dynamic projected component, while maximizing the display's contrast ratio and minimizing any visual artifacts from the separation process. We can break this down into two main steps, solving for the black/white level of the display, and solving for the printed/projected component.

5.3.1 Black/White Level

The maximal value a display can show at each pixel is called the white level of the display, and the minimal value is the black level. We start with an input video sequence $I(x, y, t)$ in CIE XYZ coordinates with the luminance values (Y coordinate) normalized. The desired goal appearance of the Projectible $L_g(x, y, t)$

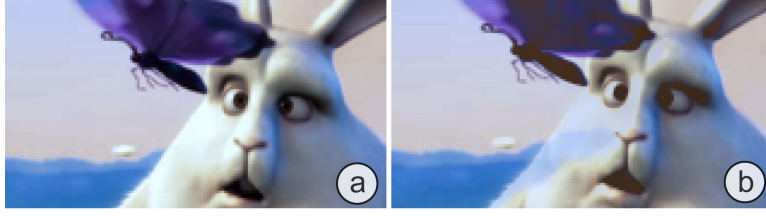


Figure 5.4: Artifacts result when $L(x, y, t) \neq L_g(x, y, t)$. (a) The desired appearance $L_g(x, y, t)$ is determined by the black and white level of the display. (b) The actual physical appearance $L(x, y, t)$ is determined by the printed and projected components. Here we have a sub-optimal solution (printing the square root of the temporal mean with $B = B_{min}$ and $W = W_{max}$), resulting in visual artifacts (b) on the left side of the bunny’s face.

should be a linear interpolation between the black level $B(x, y)$ and the white level $W(x, y)$.

$$L_g(x, y, t) = I(x, y, t) \left(W(x, y) - B(x, y) \right) + B(x, y) \quad (5.3)$$

We are combining two displays and have a black/white level for each display. For the printed component $K(x, y)$, the white level (paper white) is a measured constant K_w , and the black level (paper white with black ink) is a measured constant K_b . For the projected component $D(x, y, t)$, the white level (projected white and ambient light) is a spatially varying measured value $D_w(x, y)$, and the black level is a spatially varying measured value $D_b(x, y)$. The projected black/white levels are spatially varying because projected displays are brighter at the center of projection [63]. The ideal combined display would utilize the maximal possible contrast ratio, and have a maximal white level $W_{max}(x, y) = K_w D_w(x, y)$ and a minimal black level $B_{min}(x, y) = K_b D_b(x, y)$.

5.3.2 Printed/Projected Components

The actual physical appearance of each pixel of the display $L(x, y, t)$ is the combination of the printed and projected components. We want the actual appearance of each pixel to be the same as the desired appearance, $L(x, y, t) = L_g(x, y, t)$. Thus, given a printed component $K(x, y)$, we can solve for the required projected component for each frame, $D(x, y, t) = L_g(x, y, t)/K(x, y)$.

We will have visual artifacts if $L(x, y, t) \neq L_g(x, y, t)$ (see Figure 5.4). For a given printed component $K(x, y)$, we will have visual artifacts if the solution for the projected component exceeds the dynamic range of the projector, which is defined by the maximum/minimum luminance values.

$$\frac{L_g(x, y, t)}{K(x, y)} > D_w(x, y), \quad \frac{L_g(x, y, t)}{K(x, y)} < D_b(x, y) \quad (5.4)$$

Overall, we will have clipping artifacts if the temporal dynamic range of our target appearance exceeds the

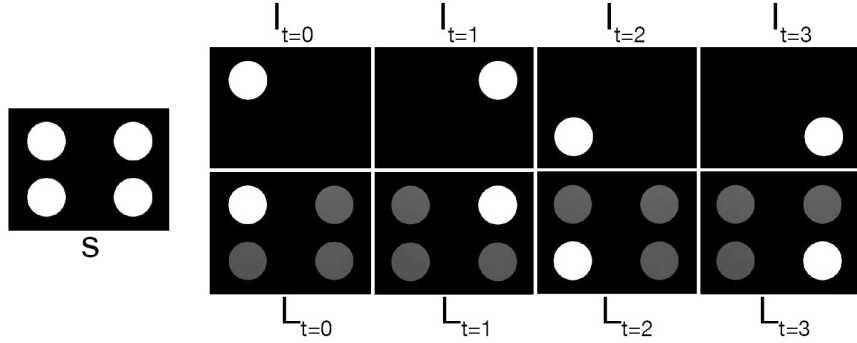


Figure 5.5: An example 4 frame input video I of a moving white circle on a black background. A naive solution that uses the maximal contrast ratio and gives preference to brighter content would concentrate the error into darker content in a non-spatially uniform way, leading to permanent gray circles in the background of the entire video sequence.

projector’s temporal dynamic range.

$$\frac{\max_t (L_g(x, y, t))}{\min_t (L_g(x, y, t))} > \frac{D_w(x, y)}{D_b(x, y)}. \quad (5.5)$$

We need to determine what our printed image should be given our input looped video sequence. As with previous static/static [9] and dynamic/dynamic [33, 95, 87] HDR displays, the logical place to start is to separate the input video into each display equally, which places equal bits of information into each display, and maximizes their combined contrast ratio. In a static/static display [9], we could choose $s(x, y) = d(x, y) = \sqrt{I(x, y)}$, where I is the input image. But this doesn’t work for video sequences as we have to choose a single printed image for our entire video sequence. Instead, we could use the square root of the average value of each pixel, or $s(x, y) = \sqrt{\text{mean}_t(I(x, y, t))}$. However, this will result in visual artifacts whenever Equation 5.5 is true, thus clipping pixels with high temporal dynamic range (see Figure 5.4).

Instead, we could choose to accurately reproduce the brightest colors, by setting

$$K(x, y) = \max_t (L_g(x, y, t)) / D_w(x, y), \quad (5.6)$$

which concentrates the reproduction errors into darker content. As Figure 5.5 shows, always giving preference to brighter content could result in prominent visual artifacts in darker content that last the entire video sequence.

5.3.3 Adjusting the Black/White Level

To reduce visual artifacts we can reduce the contrast ratio of the display, by artificially raising the black level or lowering the white level. For each pixel in the display, we solve for $B(x, y)$ and $W(x, y)$ that minimizes our reproduction error $(L(x, y, t) - L_g(x, y, t))$, subject to the constraints that $B(x, y) > B_{min}(x, y)$ and

$W(x, y) < W_{max}(x, y)$. Additionally, we want the black/white levels to be smooth. Consider two pixels with equal values in the source image $I(x_0, y_0, t_0) = I(x_1, y_1, t_1)$, ideally the final display values will also be equal, $L(x_0, y_0, t_0) = L(x_1, y_1, t_1)$, but due to spatially varying black/white levels the values may not be equal. However, if the change in black/white levels is smooth enough the human visual system will not notice it [63].

In Projectibles, we allow the content author to choose between altering the white level or black level of the display. For content that is predominately brighter (through space and time), it would be very noticeable to lower the white level, and less noticeable to raise the black level, and vice versa for predominately darker content. Previous work in radiometric compensation focused on projection onto existing surfaces, and could not control the surface color and only solved for the white level of a display. However, as our display surface is completely controllable, we can either make the entire display brighter or darker based on the content author’s goals.

5.3.4 Perceptual Error

So far we have only considered the physical error ($(L_g(x, y, t) - L(x, y, t))$ in (cd/m^2) , and have not taken into account how perceptually noticeable these errors are to the viewer. Similar to [32, 99] we use the perceptual physical error metric developed in [75] which leverages three known properties of human visual perception to approximate perceptual error instead of physical error. First, our visual system is more sensitive at lower luminance levels than at higher luminance levels. Second, our visual sensitivity is dependent on spatial frequencies, being most sensitive at frequencies around 2 to 4 cycles per degree. Third, complex images contain visual patterns at different frequencies which non-linearly mix to influence our perception [27]. These factors combine into a perceptual physical error metric, $M(x, y, t)$, that predicts for a given input frame the amount each pixel can change in physical units (cd/m^2) without being noticed. We pre-compute $M(x, y, t)$ over the entire video sequence, using $B = B_{min}$, $W = W_{max}$ as a proxy for the final luminance, and we use a viewer position of 3 feet away from the display. Figure 5.1e demonstrates the perceptual physical error metric for the train sequence. Overall, the error metric will tolerate more error on edges and in bright areas.

5.4 Splitting Static and Dynamic

We present two algorithms for splitting into a printed and projected component: an optimal solution based on a constrained optimization, and a heuristic method that leverages approximations to generate results in near real-time. Overall, we would like to develop methods that globally minimize the visual errors through time of the display, while maximizing the contrast ratio and maintaining a smooth black/white level.

5.4.1 Constrained Optimization

We can formulate our problem as a constrained optimization, similar to the approach taken in [2]. We minimize an error function E over the entire video sequence, solving for $B(x, y)$, $W(x, y)$ and $K(x, y)$, given the constraints $K_b \leq K(x, y) \leq K_w$, $B(x, y) \geq B_{min}(x, y)$, and $W(x, y) \leq W_{max}(x, y)$. Our error function E consists of seven terms, each with a corresponding weighting constant α_i .

$$E = \sum_t \sum_{xy} (\alpha_0 f_0 + \alpha_1 f_1) + \sum_{xy} \sum_{i=2}^6 \alpha_i f_i \quad (5.7)$$

The first term f_0 , accounts for the total error through time in reproducing our target image, $(L_g(x, y, t) - L(x, y, t))^2$. This can be re-written as a piecewise function, with errors when the projector clips at its maximal/minimal values.

$$f_0 = \begin{cases} (L_g(x, y, t) - K(x, y)D_w(x, y))^2, & (i) \\ (L_g(x, y, t) - K(x, y)D_b(x, y))^2, & (ii) \\ 0, & otherwise \end{cases}$$

where (i) $L_g(x, y, t)/K(x, y) > D_w(x, y)$,

$$(ii) L_g(x, y, t)/K(x, y) < D_b(x, y) \quad (5.8)$$

The above term measures error in physical units, but we would like to minimize perceptual error. We use the perceptual error metric $M(x, y, t)$ in two ways. First, we can increase the tolerance for an acceptable solution (similar to [32]). Secondly, we divide the absolute error by this perceptual error threshold to get an approximate perceptual error.

$$f_0 = \begin{cases} \frac{(L_g(x, y, t) - M(x, y, t) - K(x, y)D_w(x, y))^2}{M(x, y, t)^2}, & (i) \\ \frac{(L_g(x, y, t) + M(x, y, t) - K(x, y)D_b(x, y))^2}{M(x, y, t)^2}, & (ii) \\ 0, & otherwise \end{cases}$$

where (i) $(L_g(x, y, t) - M(x, y, t))/K(x, y) > D_w(x, y)$,

$$(ii) (L_g(x, y, t) - M(x, y, t))/K(x, y) < D_b(x, y) \quad (5.9)$$

This leaves the printed image underconstrained. In these situations, we would like to place equal amounts of information into the printer and projector [9, 33, 87]. We add a term that encourages the normalized

printed s image and normalized projected d images to be equal.

$$f_1 = \left(s(x, y) - d(x, y, t) \right)^2 \quad (5.10)$$

We want the white/black levels to be smooth so the changes are not noticeable. Similar to [2], we form two terms that minimize the sum of the gradients of B and W , $f_2 = (\nabla B(x, y))^2$, and $f_3 = (\nabla W(x, y))^2$. Because we are optimizing our display for a particular video sequence, we can smooth the gradients further. Using the perceptual error metric $M(x, y, t)$, we can tolerate a faster change in our black/white level in regions where the error tolerance is higher, and a slower change where the error tolerance is lower. For instance, in Figure 5.1f the white level of the train sequence varies greatly, and changes most rapidly in bright regions and on edges where the changes will be least noticeable. We divide our previous terms by the minimum error tolerance through time.

$$f_2 = \frac{(\nabla B(x, y))^2}{\min_t (M(x, y, t))^2}, \quad f_3 = \frac{(\nabla W(x, y))^2}{\min_t (M(x, y, t))^2} \quad (5.11)$$

To ensure we use the maximum contrast ratio possible, we want the white level and black level to be as close as possible to their ideal values. The content author can prefer brighter content or prefer darker content by adjusting the α_4 and α_5 parameters, which weight f_4 and f_5 .

$$f_4 = (B - B_{min})^2, \quad f_5 = (W - W_{max})^2 \quad (5.12)$$

Until this point we have assumed exact alignment between projected pixels and printed ‘pixels’. In fact projector pixels are not rectangles but small ellipsis of light [1], and printers do not have pixels at all, but use overlapping dots of ink of different colors. It is possible to achieve sub-pixel alignment between projected content and a display surface [1], however in practice it is extremely difficult and any slight movement (someone walking by) could cause misalignment. In this chapter, we do not assume precise geometric alignment, so we add an additional term that slightly smooths the printed image to ensure mis-registrations will not yield visual artifacts, $f_6 = (\nabla s(x, y))^2$. In practice, we do not need to smooth the entire printed image. Tight spatial registration is most crucial when compensation is necessary, i.e. when the projector and printer have drastically different values. We create a masking term ω which is large for pixels with high temporal dynamic range and with a small error tolerance.

$$f_6 = \omega (\nabla s(x, y))^2, \\ \text{where } \omega = \frac{(I_{max}(x, y, t) - I_{min}(x, y, t))^2}{\min_t (M(x, y, t))^2} \quad (5.13)$$

We solve this constrained optimization problem in Matlab using a projected quasi-Newton method, min-

Conf_PQN². We tried other constrained solvers, and found the best performance with minConf_PQN. On the Bunny sequence at [270, 480] with 150 frames, when initialized with random input it took 3 hours, 44 minutes to converge (on a laptop with an i7 2.3GHz processor). Performance could be improved with multi-grid solvers [34] and using native code, however, simply evaluating the function once can take seconds to loop through every pixel and frame.

5.4.2 Heuristic Method

The above approach yields an optimal solution given the seven input parameters, α_i . Due to the computational complexity, this approach does not allow the content author to explore the parameter space. We would like a method which can be evaluated fast enough for the content author to drive the parameters in real-time. In our heuristic method, we split the problem into two main parts, solving for the black/white level and solving for the printed image.

Solving for the White/Black Level

To ensure there are no visual artifacts, we can reduce the contrast of the desired image until there are no errors, by lowering the white level and/or raising the black level of the display. To simplify the problem, let us start by fixing the black level at its maximal value $W(x, y) = W_{max}(x, y)$, and solve for the black level of the display. At pixels that would result in visual artifacts, Equation 5.5 become an equality, and we can solve for $B(x, y)$, and at pixels with no visual artifacts we can set $B(x, y) = B_{min}(x, y)$.

This results in a non-smooth black level. Smoothing with a simple Gaussian filter would yield pixels that violate the constraints [34], so we borrow a technique from Mujumder et al. [63]. We solve a constrained gradient blurring problem, finding a new black level $B'(x, y)$ such that it is minimized across the image subject to the constraints that $B'(x, y) \geq B(x, y)$ and $\|\nabla B'(x, y)\| \leq B'(x, y)/\delta_0$, where δ_0 is a constant. This can be solved using dynamic programming in real-time. However, the approach from [63] results in a very smoothly changing black/white level, because the approach is not content dependent. We are optimizing our display for a single video sequence, so we can have a faster changing white/black level by hiding the changes in places where it will be least noticeable (where M is the highest). Similar to Equation 5.9, we modify the heuristic approach to use the perceptual physical error metric, where δ_0, δ_1 are constants.

$$\|\nabla B'(x, y)\| \leq \frac{B'(x, y)}{\delta_0} + \delta_1 \min_t(M(x, y, t)), \quad (5.14)$$

To control whether we adjust the black level or white level, we expose a single parameter to the content author which we call *prefer black/white*, $\rho_{bw} = [0, 1]$. We first solve for the adjusted B' via the process described above. If we prefer white ($\rho_{bw} = 1$) then we want to use the adjusted black level B' , however if

²<http://www.cs.ubc.ca/~schmidtm/Software/minConf.html>

we prefer black ($\rho_{bw} = 0$) then we want to use the original minimal black level B_{min} and lower the white level instead. Thus, our adjusted black level B'' is

$$B''(x, y) = B_{min}(x, y)(1 - \rho_{bw}) + B'(x, y)(\rho_{bw}). \quad (5.15)$$

We then solve for the white level given B'' , and smooth it using the constrained gradient blur. The end result is a smooth black level and smooth white level that are guaranteed to not introduce any visual artifacts.

Allow Artifacts

Not allowing artifacts can actually be a bad thing. The constrained optimization solution balances visual artifacts and contrast ratio through the α_i parameters. However, the heuristic method described above will ensure zero artifacts, maximally clipping the display contrast. Thus, we provide the content author with an additional parameter which we call *allow artifacts*, $\rho_{aa} = [0, 1]$. Similar to ρ_{bw} , this parameter is a simple linear interpolation between the naive maximal contrast black/white levels and the conservative black/white levels that introduce no artifacts. We take the black level above and solve for an adjusted black level B''' , and repeat a similar process for the white level.

$$B'''(x, y) = B''(x, y)(1 - \rho_{aa}) + B_{min}(x, y)(\rho_{aa}) \quad (5.16)$$

Solving for the Printed Image

Now we need to solve for the printed image given the black/white level. For static pixels with no motion, we would like the normalized printed and projected images to be identical. We can solve the equation below for the printed image in static regions $K_s(x, y)$ by substituting $D(x, y, t) = L_g(x, y)/K_s(x, y)$.

$$\frac{K_s(x, y) - K_b}{K_w - K_b} = \frac{D(x, y, t) - D_b(x, y)}{D_w(x, y) - D_b(x, y)} \quad (5.17)$$

For dynamic pixels, to allow zero artifacts the printed image will be constrained by the brightest desired image, and the darkest desired image. The darkest the printed image can be is determined by the brightest desired image, divided by the brightest the projector can display, $K_{min}(x, y) = \max_t(L_g(x, y, t))/D_w(x, y)$, and the brightest the printed image can be is determined by the darkest desired image divided by the darkest the projector can display, $K_{max}(x, y) = \min_t(L_g(x, y, t))/D_b(x, y)$. Thus we have the constraints $K_{min}(x, y) \leq K(x, y) \leq K_{max}(x, y)$. If $\rho_{aa} = 0$, then we should have already adjusted our black/white levels such that there can be no artifacts, and we can just ensure the above constraints are met. However, if $\rho_{aa} > 0$, then we may have artifacts and we need to choose whether to introduce artifacts in darker or brighter content.

The constrained optimization solution chooses a printed image that minimizes errors over time, and the above constraints do not incorporate time. For pixels that are mostly bright over time, we want to introduce artifacts into the darker frames for these pixels. And for pixels that are mostly dark over time, we want to introduce artifacts into the brighter frames for these pixels. To accomplish this we use the mean $I_{mean} = mean_t(I(x, y, t))$ to interpolate between K_{min} and K_{max} ,

$$K = K_{max}(1 - I_{mean}^\gamma) + K_{min}(I_{mean}^\gamma). \quad (5.18)$$

Due to the multiplicative nature of light and surface color, we want to error on the side of a brighter surface color. The one exception to this is if we have a very dark scene (like Figure 5.5), as we don't want to introduce visually significant artifacts into the background throughout the whole sequence. Therefore, we apply a $\gamma = 0.1$, to give preference to a brighter printed image except at primarily dark pixels.

Authoring Interface

To enable the content author to guide the solution in real-time, we created a GUI with sliders for each parameter of the heuristic algorithm. We also created a series of visualizations in two sets of tabs that the content author can view side by side while editing the Projectible. The content author can scrub through the video sequence via a time-line and play/pause controls. In the tabs, the content author can view the source content, the target image, the simulated Projectible result, the printed image, the projected image, the black level, the white level, and the perceptual physical error tolerance. These results can be viewed through a variety of tone mapping operators³.

5.4.3 Temporal Adaption

So far we have not incorporated a perceptual model of how our visual system responds to input over time. This can lead to excessive clipping. For instance, in the Castle time-lapse sequence there are visitors walking around the castle, which are only visible in a single frame each. We don't want to reduce the contrast of the scene to properly display them for such a short period of time. We use an exponential decay function [32] as our temporal adaption model, which decreases the perceptual error for ephemeral content and is based on the temporal response for photopic vision. We pre-process the input video sequence before running the constrained optimization or heuristic methods.

5.4.4 Up-Res Printed

Until now we have assumed that the projector and printer are equal resolution. However, printers are much higher resolution than projectors, especially for larger format displays. The ideal approach would incorporate

³<http://advancedhdrbook.com>

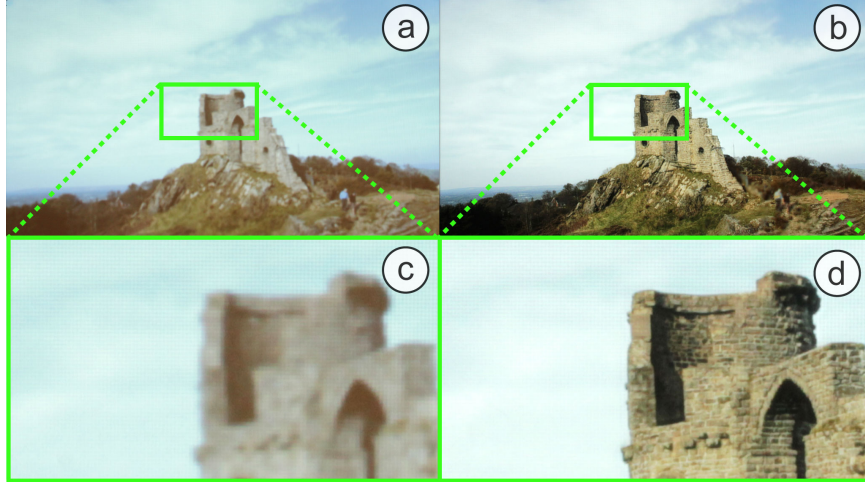


Figure 5.6: Projectibles have much higher resolution in areas where fine details are consistent throughout the video sequence. Despite the lighting changes, the high res details in the rocks are consistent in the sequence. (a) Projector only, (b) Projectible. (c-d) Close ups.

the respective resolutions (and pixel modeling) into the splitting algorithms, solving for a solution at the printer’s resolution. As the printer can be easily $>10X$ higher resolution than the projector, this would impractically increase the running time. We compute the splitting at the projector’s resolution, then up-res the answer where possible to the printer’s resolution.

As the resolution improvement is only from the printed image, we can only increase the resolution of the separation if the high frequency information at the printer’s resolution remains constant throughout the video sequence. For instance, in the Castle sequence (Figure 5.6), the colors of the stone change significantly due to passing clouds, but the high resolution details on the stone remain the same, so we can increase the resolution. To separate out the high contrast details, we run a Gaussian high pass filter over each frame of the video sequence and compute the temporal mean and standard deviation of this high pass signal. We can up-res the printed image at the lower projector resolution K^d , to the higher printed resolution K^s by transferring high resolution details from the temporal mean image $L_{mean}^s = mean_t(L_g^s(x, y, t))$, in areas β_h where the temporal standard deviation of the Gaussian high pass filter is low.

$$K^s = (1 - \beta_h) K^d + (\beta_h) K^d \frac{L_{mean}^s}{G(L_{mean}^s)} \quad (5.19)$$

5.5 Authoring Content

In order to make use of the resolution and contrast improvements that Projectibles can provide, the source content should ideally be high resolution (4K or greater) and high dynamic range (12 bits per channel or greater). To acquire such content, the simplest option is to use a 4K HDR camera, however, these cameras are expensive and not ubiquitous yet.

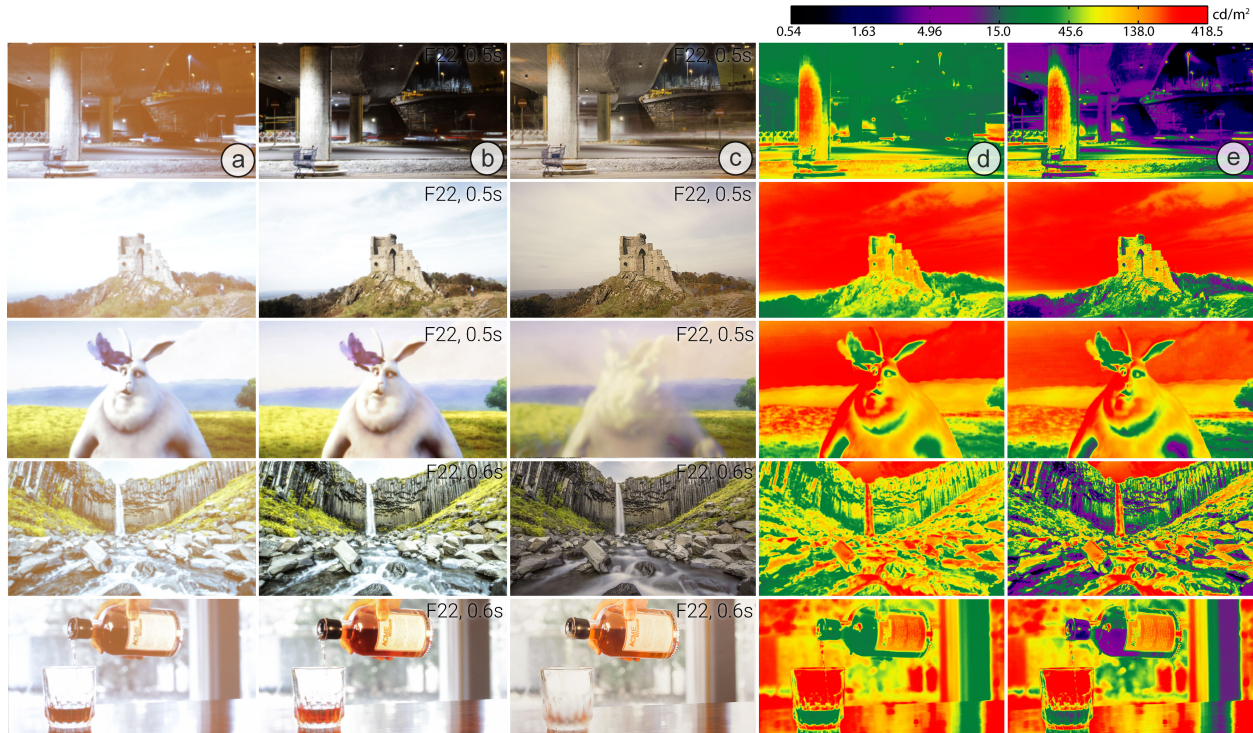


Figure 5.7: (a, b) Captured images ($F22, 0.5s, 800 ISO$) of the (a) projector only and (b) combined Projectible. (Zoom in to see the resolution differences.) (c) Printed image. (d, e) Log plots of the luminance in cd/m^2 captured with an HDR image: (d) projector only, (e) combined Projectible.

Time-lapse photography is frequently used to create ultra-high resolution video from standard DSLR cameras. Time-lapse videos composite full resolution RAW images taken at regular intervals into a video sequence. For instance, the Svartifoss waterfall sequence is a [7360, 4140], 16 bit TIFF image sequence. Another content source for Projectibles are virtual animated sequences which can be easily rendered out to high resolution, HDR images (e.g. the Bunny sequence). Standard DSLR video cameras can also shoot HDR video using exposure bracketing with custom firmware⁴ (e.g. the Train sequence), however only at standard resolutions $\leq [1920, 1080]$. To create high resolution Projectibles with standard DSLR cameras, we can composite a high resolution still image with lower resolution video. For instance, in the Whiskey sequence, we shot 1080p video of the sequence, and captured a 15 megapixel still image, then composited them together in Adobe After Effects.

5.6 Results

We demonstrate real world results for a variety of video sequences, with empirical measurements of the display's contrast ratios and subjective viewer feedback. All the results in this chapter were captured in a light controlled environment, with high ambient light to better demonstrate the splitting algorithm and the

⁴Magic Lantern (<http://www.magiclantern.fm>)

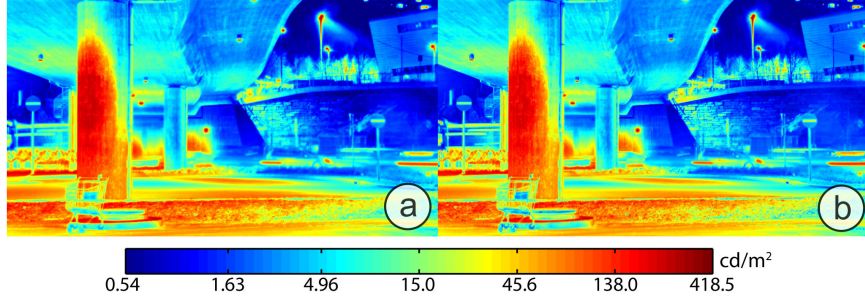


Figure 5.8: Verification of our lighting model and calibration process. Log plots of luminance values in cd/m^2 of (a) real world result captured with an HDR photograph of the Projectibles display and (b) the simulated result from our lighting model.

robustness of the methods used. The ambient light was a 60W compact fluorescent bulb which resulted in a reddish white level of [20.72, 18.80, 6.15] in CIE XYZ.

We used low cost commodity hardware. The projector (InFocus IN126ST) cost \$719 and had [1280, 800] resolution, and we tiled 2 Projectibles vertically resulting in [480, 270] pixels each. The projector was 3000 lumens with a bluish white level [405.0, 447.8, 563.1], without ambient light the dynamic range was [1.9, 447.7] with a contrast ratio of 232:1, with ambient light the dynamic range was [20.7, 466.6] with a contrast ratio of 29:1. The printer (Epson R2000) cost \$499, has a DPI of [5760, 1440], and when used with matte paper (Epson Ultra Premium Presentation Paper Matte) the dynamic range was [0.026, 0.897] with a 34:1 contrast ratio. Under ambient light, the ideal combined display would have a dynamic range of [0.58, 418.5], with a contrast ratio of 721:1. Using more expensive projectors and printers, with no ambient light, can drastically increase the combined contrast [9].

In order to measure our example results we captured calibrated exposure bracketed HDR image sequences. In Figures 5.1 and 5.7, we show log plots of the real-world measured luminance values for the projector only image on white paper, compared to the combined Projectible display. These results demonstrate the significant contrast and resolution enhancement that Projectibles can provide. We also evaluate our calibration and lighting model, Figure 5.8 compares the predicted combined display $L(x, y, t)$ with the empirical real-world results captured via an HDR image.

Similar to other image editing applications (e.g. in Photoshop), the processing time is highly dependent on the resolution of the projector/printer. As an example, for the Bunny sequence at [480, 270] projector resolution, calculating the black/white level via the heuristic method takes 0.98 seconds, and calculating the printed image (at projector resolution) takes 0.44 seconds. While not real-time, the content author can easily tune the parameters to achieve an ideal result. For the constrained optimization method, solving for the same resolution from random input took 3 hours, 44 minutes.

It is not possible to empirically compare the results between the constrained optimization and heuristic solutions as the mathematical formulations are different, so the two methods *should not* converge to the same answer. However, Figure 5.9 shows that after manually tuning the parameters on each method, the



Figure 5.9: The constrained optimization (left) and heuristic solution (right) yield similar solutions. (a) printed image, (b) white level, (c) black level, and (d) real world image of combined Projectible.

results can be quite similar.

5.6.1 Viewer Feedback

We also gathered feedback from 10 participants comparing a heuristic Projectible display to a traditional projected display on white paper (the control). The displays were stacked vertically and viewed simultaneously by participants for four example scenes (Train, Svartifoss, Underpass, Bunny). We divided the participants into two groups, and randomized the position of the displays along with the ordering of the scenes for each group. Users were instructed to view the displays from multiple positions and angles, and then rate on a Likert scale (1-5) their overall satisfaction, and judge the contrast, resolution and visual artifacts for each scene.

Figure 5.10 shows the results from the user study, demonstrating that the ratings for overall satisfaction, contrast and resolution were much higher for Projectibles. Via t-tests, the results for overall satisfaction, contrast and resolution were statistically significant for the Train, Svartifoss and Underpass sequences, but not for the Bunny sequence. Due to the motion of the dark butterfly and the white bunny across the scene in the Bunny sequence, the Projectible process has a lower contrast boost than the other examples. Viewers were also asked to rate the visual artifacts in the display (if something didn't look right). The results were only statistically significant for the Underpass sequence, and viewers rated the Projectible as having less artifacts. Thus, overall visual artifacts were not clearly noticeable and didn't effect overall satisfaction rates.

5.7 Discussion

In this work, we focus on the core algorithms for splitting printed and projected. Extensions to the approach could enable all kinds of unique visual experiences (see Figure 5.11). We could optimize the surface color of

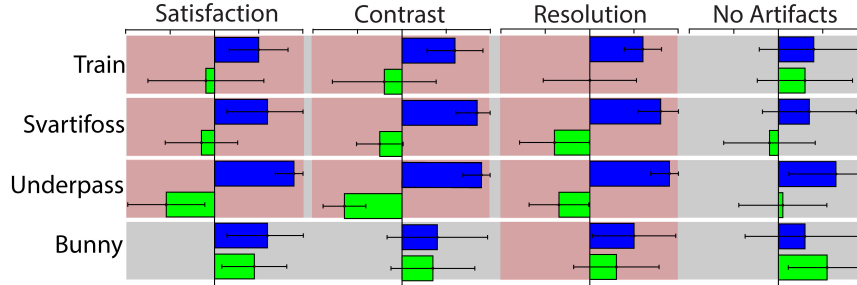


Figure 5.10: Results of Likert data from a user study (N=10) for four sequences (Train, Svartifoss, Underpass, Bunny), (blue) Projectible, (green) Projector only. Viewers clearly preferred the Projectible display for overall satisfaction, resolution and brightness. Viewers were asked to rate visual artifacts (if something didn’t look right), and the results show that viewers couldn’t clearly differentiate artifacts. (Results with a pink background were statistically significant $p < 0.05$ via t-tests).

3D objects, either constructed from paper cutouts, multi-material 3D printers, or industrial manufacturing processes. We could extend our lighting model to support transparency printing for rear projection surfaces (e.g. for store-front windows), or for other non-matte printed surfaces. We can easily create much larger Projectibles for truly immersive experiences, e.g. Figure 5.11d shows a large 5x3ft 3D version of the Svartifoss waterfall.

Overall, Projectibles work well for looped video sequences with a combination of static and dynamic elements. For sequences that are mostly static (e.g. Whiskey sequence), Projectibles can achieve ideal contrast ratios and significant resolution improvements. For sequences that have significant motion (e.g. Train sequence), resolution improvements are impossible but contrast improvements can still be achieved. The worst case scenario is a video sequence that goes from uniform white to uniform black, in which case the printed component cannot improve resolution or contrast. Thus, Projectibles work best with looped video content, and we showcase a range of contrast and resolution improvements on six different sequences.

In environments with fixed high ambient light, normal projected displays on a white display surface are too washed out for practical use. Projectibles could be used in these locations to create projected displays that are non-rectangular or non-flat with greatly improved contrast ratios and resolution. In environments with low ambient light, Projectibles are capable of achieving HDR display contrast ratios beyond those of normal displays [9]. However, environments with variable lighting like skylights and windows don’t work with the existing implementation. Projectibles could be extended with a real-time color calibrated camera that could adjust the projected image in real-time, but the printed image would still have to be optimized for a certain range of ambient light.

Another limitation is that the perceptual physical error tolerance is based on a fixed viewer location. In the user study, the viewers were asked to view the display from a variety of distance and angles, and did not report noticing any differences in perceived appearance. Furthermore, through the authoring interface, we provide the content author with an accurate simulation of the combined result, which can be viewed from multiple angles and distances.

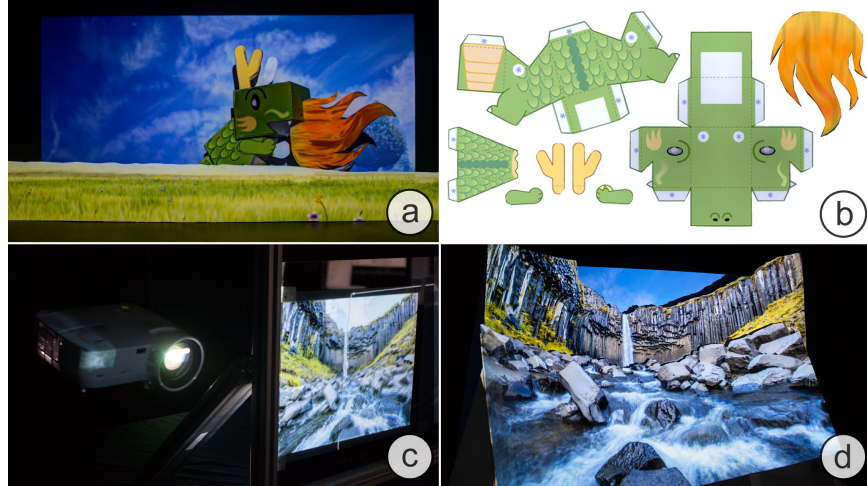


Figure 5.11: Projectibles could be extended in a variety of ways including: (a,b) optimizing the surface of animated 3D objects using paper cut-outs, (c) using rear projection surfaces, or (d) large scale 3D projection mapping installations (shown here at 5x3 ft).

To commercially deploy Projectibles, the system would need an automated geometric registration process (e.g. [9]). If precise sub millimeter physical alignment of the printed image with the projected image were possible, sub-pixel super-resolution could also be achieved [1], and the printed image could compensate for the size and shape of each projector pixel. Also, for use in medical imaging, the JNDs of the display could be optimized [9].

5.8 Summary

Compared to a traditional display, Projectibles can be higher resolution, higher contrast (in low ambient light), and can be made into large format, non-rectangular and even non-flat displays. This work is the first to combine a static display with a dynamic display for showing high dynamic range, high resolution video. It is also the first to optimize surface color for projection of video. We presented an offline constrained optimization method, and a near real-time heuristic method for splitting a dynamic video into a printed and projected component. We showed real world results for six example sequences, demonstrating Projectibles' significant contrast and resolution improvements, and presented viewer feedback confirming these results. For advertising and entertainment applications, Projectibles combine the benefits of printed signage with traditional digital displays, to create high resolution, high dynamic range viewing experiences.

Note: We would like to thank Stian Rekdal (<http://stianrekdal.com>) for the Svartifoss waterfall sequence, Dan at NatureClip (<http://natureclip.co.nr>) for the Castle sequence, and Eric Solheim (<http://eirikso.com>) for the Underpass sequence.

CHAPTER 6

SUMMARY AND CONCLUSION

This thesis has presented three projects that explore novel systems and methods for content creation for seamless augmented experiences with projection mapping. We have described two projects, *IllumiRoom* and *RoomAlive*, which use real-time depth data to adapt projected content to the color, geometry and semantics of the particular room. Whereas most projection mapping content is manually created for a specific display object, this adaptive projection mapping enables content to be created once by a game designer, then reused across any room configuration. In each project we demonstrated unique interactive experiences, detailed the implementation of the proof-of-concept systems and discussed the design space of potential adaptive projection mapped effects.

We also explored a novel method for designing projection mapped content, *Projectibles*, where artist co-design physical and virtual simultaneously. By carefully combining static printed colors with dynamic virtual projection mapping content, we can create high resolution, high dynamic range displays that can have any contour and be arranged into 3D shapes. We detailed two algorithms for splitting an input video into a printed and a projected component. We presented empirical measured results for six example sequences, subjective viewer feedback, and we discussed the benefits and limitations of Projectibles.

This work is unified by its quest to create projection mapping content that adapts to the physical environment, and to adapt the physical environment to the projected content. While the three prototype systems demonstrate the magical experiences that might be possible in the future, much work remains to be done.

6.1 Limitations and Future Work

Projection mapping has many unique advantages over other forms of virtual and augmented reality (e.g. mobile augmented reality, or head mounted displays). Namely, it is an apparently unmediated experience that does not encumber the user and does not degrade the users' perception of the existing physical environment. However, projection mapping itself has unique limitations.

Primarily, projection mapping is limited to only displaying content on the surfaces of physical objects. As described in Chapter 3 & 4, in order to display a virtual object that is not located on a physical object's surface, view dependent projection must be used. This view dependent projection can accurately create the illusion of a 3D object for a single user, but only if the display surface has no discontinuities and is not a

dark color or heavily textured. If the display surface is not an ideal projection surface, then the illusion of the 3D object can be broken (see [6] for an interesting user study on this topic). Furthermore, certain display surfaces are unable to receive projected light at all (e.g. black surfaces, or glass). The projected effects described in this thesis are simply impossible on these surfaces. Future work could explore adaptive projection mapping that ensure only high quality display surfaces are used. For instance, in a projected living room, the system could only project enemy robots on the white walls and avoid the glass windows.

Along these lines, not every surface is suitable for interaction. It is logical to use touch based interaction on a table-top or coffee table, but you don't want your children whacking a mole that appears on a priceless vase. One solution, would be for users to annotate their rooms and tag objects that are too fragile for play. Alternatively, additional semantic scene understanding could aid in eliminating fragile playing surfaces.

Projection mapping also has the inherent limitation that the projector must be mounted somewhere. In *IllumiRoom* and *RoomAlive* we argue that the procam unit could be placed on a coffee table, bookshelf or on a tripod. However, the optimal location (due to shadowing) for these procam units is on the ceiling. It is unlikely that commercial solutions could rely on end users to mount a procam unit on the ceiling, as running the power cable usually requires significant work and technical knowledge. Perhaps a business model that incorporates free setup and installation, like DirectTV, would work. Also, commercially we are seeing an increased trend toward short throw projectors. If this trend continues, and projectors become even more short throw, it may be possible to cover an entire living room with a projector on your coffee table.

Interaction with real-time projector camera systems is always impacted by system latency. Off-the-shelf projectors have significant latency, and when combined with latency due to the depth-camera, graphics card and processing, can lead to image ghosting effects where fast moving objects trail behind. Recent research has demonstrated hardware solutions to virtually eliminate latency [69].

Calibration accuracy comes up again and again in this thesis. For *IllumiRoom* and *RoomAlive* a great deal of effort was put into making the calibration procedures entirely automatic, requiring zero user input. This makes these systems plausible for real-world deployments in environments where users will not tolerate waving checkerboards around for calibration. However, more advanced optimization techniques could be used to fine-tune the calibration results [1, 55, 59, 84].

These projects just scratched the surface of what is possible with basic scene knowledge (floors, walls). More advanced techniques for scene understanding could yield (e.g. [43, 25, 38]) more detailed information about the physical environment. This in turn could be used to create richer augmented experiences. For instance, a game could spawn enemy robots at the door of the room, and have enemy helicopters entering through the windows.

6.2 Conclusion

Overall we have shown systems and methods for projection mapping content creation that can enable new interactive experiences where the physical environment and the virtual experiences seamlessly merge. This can increase users' sense of immersion and presence, leading to an overall “magical” experience.

It is yet to be seen if these ideas can transition from proof-of-concepts into fully fleshed out commercial solutions. While there are many technical and commercial challenges yet to be solved, I hope this thesis has demonstrated that they are worth answering.

REFERENCES

- [1] Aliaga, D. G., Yeung, Y. H., Law, A., Sajadi, B., and Majumder, A. Fast high-resolution appearance editing using superimposed projections. *ACM Transactions on Graphics* 31, 2 (Apr. 2012), 1–13.
- [2] Ashdown, M., Okabe, T., Sato, I., and Sato, Y. Robust content-dependent photometric projector compensation. In *IEEE CVPR* (2006).
- [3] Bandyopadhyay, D., Raskar, R., and Fuchs, H. Dynamic shader lamps : painting on movable objects. In *IEEE ISMAR* (2001).
- [4] Batlle, J., Mouaddib, E., and Salvi, J. Recent progress in coded structured light as a technique to solve the correspondence problem. *Pattern Recognition* 31, 7 (1998), 963–982.
- [5] Baudisch, P., Good, N., and Stewart, P. Focus plus context screens: combining display technology with visualization techniques. In *ACM UIST* (2001).
- [6] Benko, H., Jota, R., and Wilson, A. MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop. In *ACM CHI* (2012).
- [7] Bimber, O., Coriand, F., Kleppe, A., Bruns, E., Zollmann, S., and Langlotz, T. Superimposing pictorial artwork with projected imagery. *IEEE Multimedia* 12, 1 (2005), 16–26.
- [8] Bimber, O., Emmerling, A., and Klemmer, T. Embedded Entertainment with Smart Projectors. *IEEE Computer* 38, 1 (Jan. 2005), 16–26.
- [9] Bimber, O., and Iwai, D. Superimposing dynamic range. *ACM Transactions on Graphics* 27, 5 (Dec. 2008), 1.
- [10] Bimber, O., Iwai, D., Wetzstein, G., and Grundhöfer, A. The Visual Computing of Projector-Camera Systems. *Computer Graphics Forum* 27, 8 (2007), 2219–2245.
- [11] Bimber, O., and Raskar, R. *Spatial augmented reality: Merging real and virtual worlds*. AK Peters Ltd, 2005.
- [12] Bobick, A. F., Intille, S. S., Davis, J. W. J., Baird, F., Pinhanez, C. S., Campbell, L. W., Ivanov, Y. a., Schutte, A., Wilson, A. D., and Schütte, A. The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. *Presence: Teleoperators & Virtual Environments* 8, 4 (Aug. 1999), 369–393.
- [13] BotNDolly. <http://www.botndolly.com/box>, 2013.
- [14] Brown, M., Majumder, A., and Yang, R. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics* 11, 2 (2005), 193–206.
- [15] Brown, M., and Seales, W. A practical and flexible tiled display system. In *IEEE Computer Graphics and Applications* (2002).
- [16] Brox, T., Bruhn, a., Papenberg, N., and Weickert, J. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *European Conference on Computer Vision, ECCV'04*, vol. 4 (2004), 25–36.

- [17] Butler, D. A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S., and Kim, D. Shake'n'sense: reducing interference for overlapping structured light depth cameras. In *ACM CHI* (2012).
- [18] Cao, X., and Balakrishnan, R. Interacting with dynamically defined information spaces using a hand-held projector and pen. In *ACM UIST* (2006).
- [19] Chen, J., Turk, G., and MacIntyre, B. A non-photorealistic rendering framework with temporal coherence for augmented reality. In *IEEE ISMAR* (2012).
- [20] Chen, X., Yang, X., Xiao, S., and Li, M. A practical radiometric compensation method for projector-based augmentation. In *IEEE ISMAR* (2008).
- [21] Cockburn, A., Karlson, A., and Bederson, B. B. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys* 41, 1 (Dec. 2008), 1–31.
- [22] Cruz-Neira, C., Sandin, D., and DeFanti, T. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *ACM SIGGRAPH* (1993), 135–142.
- [23] Deadmau5. <https://www.youtube.com/watch?v=SX2Gd-kqV5s>, 2011.
- [24] Dou, M., Fuchs, H., and Frahm, J. M. Scanning and tracking dynamic objects with commodity depth cameras. In *IEEE ISMAR* (Oct. 2013).
- [25] Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *IEEE CVPR Workshops (CVPRW)* (2009).
- [26] Feiner, S., Macintyre, B., and Seligmann, D. Knowledge-based augmented reality. *Communications of the ACM* (1993).
- [27] Ferwerda, K. a., Shirley, P., Pattanaik, S. N., and Greenberg, D. P. A model of visual masking for computer graphics. In *ACM SIGGRAPH* (1997).
- [28] Fischer, J., Bartz, D., and Straber, W. Artistic Reality: Fast Brush Stroke Stylization for Augmented Reality. In *ACM VRST* (2005).
- [29] Fischer, J., Bartz, D., and Straber, W. Stylized augmented reality for improved immersion. In *IEEE VR* (2005).
- [30] Fischler, M. a., and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 6 (June 1981), 381–395.
- [31] Graham, M. E., Wang, W., and Stephens, P. S. Projection of images of computer models in three dimensional space, US Patent 5,687,305, 1997.
- [32] Grundhöfer, A., and Bimber, O. Real-time adaptive radiometric compensation. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 97–108.
- [33] Guarnieri, G., Albani, L., and Ramponi, G. Image-splitting techniques for a dual-layer high dynamic range LCD display. *Journal of Electronic Imaging* 17, 4 (Oct. 2008).
- [34] Guarnieri, G., Albani, L., and Ramponi, G. Minimum-error splitting algorithm for a dual layer LCD display, Part I: Implementation and results. *IEEE/OSA Journal of Display Technology* 4, 4 (2008), 391–397.
- [35] Haller, M., and Landerl, F. A mediated reality environment using a loose and sketchy rendering technique. In *IEEE ISMAR* (2005).
- [36] Hancock, M., Nacenta, M., Gutwin, C., and Carpendale, S. The Effects of Changing Projection Geometry on the Interpretation of 3D Orientation on Tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)* (2009).

- [37] Harrison, Benko, H., and Wilson, A. OmniTouch: wearable multitouch interaction everywhere. In *ACM UIST* (2011).
- [38] Hedau, V., Hoiem, D., and Forsyth, D. Recovering the spatial layout of cluttered rooms. In *IEEE CVPR* (2009), 1849–1856.
- [39] Henderson, J. M. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences* 7, 11 (Nov. 2003), 498–504.
- [40] Hereld, M., Judson, I. R., and Stevens, R. L. Introduction to building projection-based tiled display systems. *IEEE Computer Graphics and Applications* 20 (2000), 22–28.
- [41] Herring, S. R., Jones, B. R., and Bailey, B. P. Idea generation techniques among creative professionals. In *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS* (2009).
- [42] Hilliges, O., Kim, D., Izadi, S., Weiss, M., and Wilson, A. HoloDesk: direct 3d interactions with a situated see-through display. In *ACM CHI* (2012).
- [43] Hoiem, D., Rother, C., and Winn, J. 3D LayoutCRF for multi-view object class recognition and segmentation. In *IEEE CVPR* (2007).
- [44] Hou, J., Nam, Y., Peng, W., and Lee, K. M. Effects of screen size, viewing angle, and players’ immersion tendencies on game experience. *Computers in Human Behavior* 28, 2 (Mar. 2012), 617–623.
- [45] Illingworth, J., and Kittler, J. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing* 44, 44 (1988), 87–116.
- [46] Israel, T., Tech, I., and Avraham, T. Multiscale Ultrawide Foveated Video Extrapolation. In *IEEE ICCP* (2011).
- [47] Johnson, T., and Fuchs, H. A Unified Multi-Surface, Multi-Resolution Workspace with Camera-Based Scanning and Projector-Based Illumination. In *IPT-EGVE Symposium* (2007).
- [48] Jones, B. R., Benko, H., Ofek, E., and Wilson, A. D. IllumiRoom: peripheral projected illusions for interactive experiences. In *ACM CHI* (2013).
- [49] Jones, B. R., Sodhi, R., Campbell, R. H., Garnett, G., and Bailey, B. P. Build your world and play in it: Interacting with surface particles on complex objects. In *IEEE ISMAR* (Oct. 2010).
- [50] Jones, B. R., Sodhi, R., Forsyth, D., Bailey, B., and Maciocci, G. Around Device Interaction for Multiscale Navigation. In *ACM MobileHCI* (2012).
- [51] Jones, B. R., Sodhi, R., Karsch, K., and Casperson, M. Projection Mapping Central. <http://projection-mapping.org>, 2014.
- [52] Jones, B. R., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A. D., Ofek, E., Macintyre, B., Raghuvanshi, N., and Shapira, L. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units. In *ACM UIST* (2014).
- [53] Joshi, N., Mehta, S., Drucker, S., Stollnitz, E., Hoppe, H., Uyttendaele, M., and Cohen, M. Cliplets: juxtaposing still and dynamic imagery. In *ACM UIST* (2012).
- [54] Juang, R., and Majumder, A. Photometric Self-Calibration of a Projector-Camera System. In *IEEE CVPR* (2007).
- [55] Kainz, B., Hauswiesner, S., Reitmayr, G., Steinberger, M., Grasset, R., Gruber, L., Veas, E., Kalkofen, D., Seichter, H., and Schmalstieg, D. OmniKinect: Real-Time Dense Volumetric Data Acquisition and Applications. In *IEEE VR* (2012).

- [56] Kennedy, R. S., Lane, N. E., Berbaum, K. S., and Lilienthal, M. G. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology* 3, 3 (1993), 203–220.
- [57] Kjeldsen, R., Levas, A., and Pinhanez, C. Dynamically Reconfigurable Vision-Based User Interfaces. *Machine Vision and Applications* 16, 1 (2004), 6–12.
- [58] Law, A. J., Aliaga, D. G., Sajadi, B., Majumder, A., and Pizlo, Z. Perceptually based appearance modification for compliant appearance editing. *Computer Graphics Forum* 30, 8 (Dec. 2011), 2288–2300.
- [59] Lee, J., Dietz, P., Maynes-Aminzade, D., Raskar, R., and Hudson, S. Automatic projector calibration with embedded light sensors. In *ACM UIST* (2004).
- [60] Liao, Z., Joshi, N., and Hoppe, H. Automated video looping with progressive dynamism. *ACM Transactions on Graphics* 32, 4 (July 2013).
- [61] Lin, J.-W., Duh, H., Parker, D., Abi-Rached, H., and Furness, T. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *IEEE VR* (2002).
- [62] Majumder, A., and Stevens, R. Color Nonuniformity in Projection-Based Displays: Analysis and Solutions. *IEEE Transactions on Visualization and Computer Graphics* 10, 2 (2004), 177–188.
- [63] Majumder, A., and Stevens, R. Perceptual photometric seamlessness in projection-based tiled displays. *ACM Transactions on Graphics* 24, 1 (Jan. 2005), 118–139.
- [64] Marshmallow Laser Feast. Sony PS3 Video Store Realtime Projection Mapping, <https://vimeo.com/34025760>, 2012.
- [65] Milgram, P., Takemura, H., Utsumi, A., and Kishino, F. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for industrial applications*, International Society for Optics and Photonics (1995), 282–292.
- [66] Monroe, M. M., and Redmann, W. G. Apparatus and method for projection upon a three-dimensional object, U.S. Patent No. 5,325,473, 1994.
- [67] Naimark, M. Two Unusual Projection Spaces. *Presence: Teleoperators and Virtual Environments* 14, 5 (Oct. 2005), 597–605.
- [68] Nayar, S. K., Peri, H., Grossberg, M. D., and Belhumeur, P. N. A projection system with radiometric compensation for screen imperfections. In *ICCV Workshop on Projector-Camera Systems (PRO-CAMS)* (2003).
- [69] Ng, A., Lepinski, J., and Wigdor, D. Designing for low-latency direct-touch input. In *ACM UIST* (2012).
- [70] Novy, D., and Jr., V. M. B. Infinity-by-nine. <http://obm.media.mit.edu>.
- [71] Pinhanez, C. S. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In *Proc. UbiComp '01* (2001).
- [72] ProCams. <http://www.procams.org/>, 2007.
- [73] Quirk, P., Johnson, T., Skarbez, R., Towles, H., Gyrfas, F., and Fuchs, H. RANSAC-assisted display model reconstruction for projective display. In *IEEE VR* (2006).
- [74] R. Sekular, R. B., Sekular, R., and Blake, R. *Perception, 2nd Edition*. 1990.
- [75] Ramasubramanian, M., Pattanaik, S., and Greenberg, D. A perceptually based physical error metric for realistic image synthesis. In *ACM SIGGRAPH* (1999).

- [76] Raskar, R., and Beardsley, P. A self-correcting projector. In *IEEE CVPR* (2001).
- [77] Raskar, R., Beardsley, P., van Baar, J., Wand, Y., Dietz, P., Lee, J., Leigh, D., and Willwacher, T. RFIG Lamps: Interacting with a Self-Describing World via Photosensing Wireless Tags and Projectors. *Transactions on Graphics (TOG)* 23, 3 (2004), 406–415.
- [78] Raskar, R., Raskar, R., Van Baar, J., Van Baar, J., Beardsley, P., Beardsley, P., Willwacher, T., Willwacher, T., Rao, S., Rao, S., Forlines, C., and Forlines, C. iLamps: Geometrically Aware and Self-Configuring Projectors. *ACM SIGGRAPH 2006 Courses 223* (2003), 809–818.
- [79] Raskar, R., Raskar, R., Ziegler, R., Ziegler, R., Willwacher, T., and Willwacher, T. Cartoon Dioramas in Motion. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (2002), 7–13.
- [80] Raskar, R., and Welch, G. Shader lamps: Animating real objects with image-based illumination. In *Eurographics Rendering Techniques* (2001).
- [81] Raskar, R., Welch, G., Cutts, M., Lake, A., Stessin, L., and Fuchs, H. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *ACM SIGGRAPH* (1998).
- [82] Rekimoto, J., and Saitoh, M. Augmented Surfaces: A spatially continuous workspace for hybrid computing environments. In *ACM CHI* (1999).
- [83] Rosenberg, L. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *IEEE VR* (1993).
- [84] Sajadi, B., and Majumder, A. Auto-Calibration of Multi-Projector CAVE-like Immersive Environments. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (2012), 381–393.
- [85] Schild, J., LaViola, J., and Masuch, M. Understanding user experience in stereoscopic 3D games. In *ACM CHI* (2012).
- [86] Schulze, J. P., Acevedo, D., Mangan, J., Prudhomme, A., Nguyen, P., and Weber, P. Democratizing rendering for multiple viewers in surround VR systems. In *IEEE 3DUI* (2012).
- [87] Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trentacoste, M., Ghosh, A., and Vorozcovs, A. High dynamic range display systems. *ACM Transactions on Graphics* 23, 3 (Aug. 2004), 760.
- [88] Seetzen, H., Whitehead, L., and Ward, G. A High Dynamic Range Display Using Low and High Resolution Modulators. *SID Symposium Digest of Technical Papers* 34, 1 (2003), 1450–1453.
- [89] Sheng, Y., Yapo, T. C., and Cutler, B. Global illumination compensation for spatially augmented reality. *Computer Graphics Forum* 29, 2 (May 2010), 387–396.
- [90] Sodhi, R., Benko, H., and Wilson, A. LightGuide: projected visualizations for hand movement guidance. In *ACM CHI* (2012).
- [91] Sodhi, R. S., Jones, B. R., Forsyth, D., Bailey, B. P., and Maciocci, G. BeThere: 3D mobile collaboration with spatial input. In *ACM CHI* (2013).
- [92] Straka, M., Hauswiesner, S., Ruther, M., and Bischof, H. Rapid skin: Estimating the 3D human pose and shape in real-time. In *IEEE 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)* (2012).
- [93] Sutherland, I. E. A head-mounted three dimensional display. *Proceedings of the December 9-11, 1968, fall joint computer conference, part I* (1968), 757–764.
- [94] Tan, D. S., Gergle, D., Scupelli, P., and Pausch, R. With similar visual angles, larger displays improve spatial performance. In *ACM CHI* (2003).

- [95] Trentacoste, M., Heidrich, W., Whitehead, L., Seetzen, H., and Ward, G. Photometric image processing for high dynamic range displays. *Journal of Visual Communication and Image Representation* 18, 5 (Oct. 2007), 439–451.
- [96] Triebel, M., Oberländer, A., Thiele, S., and Bimber, O. PlayReal: Spatial Augmented Reality Games, <http://140.78.90.140/medien/ar/research.php>, 2008.
- [97] Underkoffler, J., Ullmer, B., and Ishii, H. Emancipated pixels: real-world graphics in the luminous room. In *ACM SIGGRAPH* (1999).
- [98] Urbanscreen. 555 Kubik. <https://vimeo.com/5595869>.
- [99] Wang, D. W. D., Sato, I., Okabe, T., and Sato, Y. Radiometric Compensation in a Projector-Camera System Based Properties of Human Vision System. In *IEEE CVPR Workshops (CVPRW)* (2005).
- [100] Weffers-Albu, A., S. de Waele, W. H., and Kwisthout., C. Immersive TV viewing with advanced ambientlight. In *IEEE Conference on Consumer Electronics* (Jan. 2011), 753–754.
- [101] Wetzstein, G., and Bimber, O. Radiometric Compensation through Inverse Light Transport. In *Pacific conference on computer graphics and applications* (2007), 391–399.
- [102] Willis, K., Poupyrev, I., and Shiratori, T. Motionbeam: a metaphor for character interaction with handheld projectors. In *ACM CHI* (2011).
- [103] Willis, K. D., Poupyrev, I., Hudson, S. E., and Mahler, M. SideBySide: ad-hoc multi-user interaction with handheld projectors. In *ACM UIST* (2011).
- [104] Willis, K. D. D., Shiratori, T., and Mahler, M. HideOut: mobile projector interaction with tangible objects and surfaces. In *ACM TEI* (2013).
- [105] Wilson, A. D. Depth-sensing video cameras for 3D tangible tabletop interaction. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems, TABLETOP'07* (2007).
- [106] Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *ACM UIST* (2010).
- [107] Wilson, A. D., Benko, H., Izadi, S., and Hilliges, O. Steerable augmented reality with the beamatron. In *ACM UIST* (2012).
- [108] Yang, R. Y. R., Gotz, D., Hensley, J., Towles, H., and Brown, M. PixelFlex: a reconfigurable multi-projector display system. In *Proceedings of the conference on Visualization'01*, IEEE (2001), 167–554.
- [109] Zhou, F., Dun, H. B. L., and Billinghurst, M. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *IEEE ISMAR* (2008).