FUNNEL VISION

Low-Cost Auto-Stereoscopic 360-degree Display with Conical Reflection and Radial Lenticular + Contextual Artificially Intelligent Character with Procedural Animation

Emily M. Salvador
B.A. in Computer Science and Music, Massachusetts Institute of Technology, 2016

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

June 2019

Signature of Author: Signature redacted

Emily M. Salvador
Program in Media Arts and Sciences
June 2019

Certified by: Signature redacted

V. Michael Bove Jr.
Principal Research Scientist
Thesis Supervisor

Accepted by: Signature redacted

Tod Machover
Academic Head, Program in Media Arts and Sciences

FUNNEL VISION

Low-Cost Auto-Stereoscopic 360-degree Display with Conical Reflection and Radial Lenticular + Contextual Artificially Intelligent Character with Procedural Animation

Emily M. Salvador

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning on May 2019,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

abstract:
Every year, we progress towards a new reality where digital content blends seamlessly with our physical world. Augmented reality (AR) headsets and volumetric displays afford us the ability to create worlds that are as rich as the movies we watch and interact with media in ways that contextualize our environments. However, AR headset devices can only be experienced by an individual, and most volumetric displays typically are fragile, expensive and/or have reciprocating components.

Funnel Vision aims to bring 3D lightfields to the physical realm using lenticular rendering, conical reflection, and a 4K monitor. The need for inexpensive, reliable 3D, 360-degree display technologies grows as AR applications continue to increase in popularity. In real-time, this system creates an AR experience that can be viewed from any angle with primarily inexpensive, readily-available components. The radial optics partitions views generated real-time in Unity, which are then reflected off a mirrored cone, to produce a volumetric image.

Additionally, this system provides affordances not available with existing devices given that it is portable, perspective-occluding, and collaborative. I created a 3D character that animates and responds in real-time based on human interaction and emotional evaluations to highlight the capabilities of this unique display.

Ultimately, I hope this thesis will inspire the entertainment and consumer electronics industries to pursue this novel display technology that brings characters to life and showcases effects at the same fidelity as they exist in the digital world.

Thesis Supervisor: Signature redacted

V. Michael Bove Jr.
Principal Research Scientist
Object Based Media Group, MIT Media Lab

FUNNEL VISION

Low-Cost Auto-Stereoscopic 360-degree Display with Conical Reflection and Radial Lenticular + Contextual Artificially Intelligent Character with Procedural Animation

Emily M. Salvador

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning on May 2019,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

abstract:
Every year, we progress towards a new reality where digital content blends seamlessly with our physical world. Augmented reality (AR) headsets and volumetric displays afford us the ability to create worlds that are as rich as the movies we watch and interact with media in ways that contextualize our environments. However, AR headset devices can only be experiences by an individual, and most volumetric displays typically are fragile, expensive and/or have reciprocating components.

Funnel Vision aims to bring 3D lightfields to the physical realm using lenticular rendering, conical reflection, and a 4K monitor. The need for inexpensive, reliable 3D, 360-degree display technologies grows as AR applications continue to increase in popularity. In real-time, this system creates an AR experience that can be viewed from any angle with primarily inexpensive, readily-available components. The radial optics partitions views generated real-time in Unity, which are then reflected off a mirrored cone, to produce a volumetric image.

Additionally, this system provides affordances not available with existing devices given that it is portable, perspective-occluding, and collaborative. I created a 3D character that animates and responds in real-time based on human interaction and emotional evaluations to highlight the capabilities of this unique display.

Ultimately, I hope this thesis will inspire the entertainment and consumer electronic industries to pursue this novel display technology that brings characters to life and showcases effects at the same fidelity as they exist in the digital world.

Thesis Reader: Signature redacted

Cynthia Breazeal
Associate Professor of Media Arts and Sciences
Personal Robots Group, MIT Media Lab

FUNNEL VISION

Low-Cost Auto-Stereoscopic 360-degree Display with Conical Reflection and Radial Lenticular + Contextual Artificially Intelligent Character with Procedural Animation

Emily M. Salvador

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning on May 2019,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

abstract:
Every year, we progress towards a new reality where digital content blends seamlessly with our physical world. Augmented reality (AR) headsets and volumetric displays afford us the ability to create worlds that are as rich as the movies we watch and interact with media in ways that contextualize our environments. However, AR headset devices can only be experiences by an individual, and most volumetric displays typically are fragile, expensive and/or have reciprocating components.

Funnel Vision aims to bring 3D lightfields to the physical realm using lenticular rendering, conical reflection, and a 4K monitor. The need for inexpensive, reliable 3D, 360-degree display technologies grows as AR applications continue to increase in popularity. In real-time, this system creates an AR experience that can be viewed from any angle with primarily inexpensive, readily-available components. The radial optics partitions views generated real-time in Unity, which are then reflected off a mirrored cone, to produce a volumetric image.

Additionally, this system provides affordances not available with existing devices given that it is portable, perspective-occluding, and collaborative. I created a 3D character that animates and responds in real-time based on human interaction and emotional evaluations to highlight the capabilities of this unique display.

Ultimately, I hope this thesis will inspire the entertainment and consumer electronic industries to pursue this novel display technology that brings characters to life and showcases effects at the same fidelity as they exist in the digital world.

Signature redacted

Thesis Reader:

Ken Perlin
Professor of Computer Science
Future Reality Lab, New York University

ACKNOWLEDGEMENTS

The past two years at the MIT Media Lab has transformed my perspective and motivation in large thanks to my classmates, mentors and the unconditional generosity of this special community. The experiences I've had at the lab, whether through my thesis or the opportunities I had, will continue to inspire me for a lifetime.

First and foremost, I would like to thank my advisor, Michael Bove, for accepting me into his group, and giving me the space to explore and create. His valuable mentorship and freedom gave me the tools to ask the right questions and gave new purpose to my research goals. His confidence in his students and his genuine support of us following our passions to do our best work gave me a new confidence in my ability to imagine and achieve my goals.

Thank you to Cynthia Breazeal and Ken Perlin for your support and feedback as I shaped this thesis. Cynthia, your constructive feedback and challenging questions allowed me to ask better questions regarding this project and its purpose – and I will continue to keep in mind the advice you gave me as I work with new teams exploring ambiguous contexts. Ken, your intellect, kindness, and keen insights helped me develop this work and my own aspirations. Thank you for your generosity with your time and ideas. I would like to thank Hiroshi Ishii for your informal mentorship. I appreciate your willingness to provide feedback and support and for genuinely caring about how I can become the best version of myself. I will forever be humbled and inspired by Cynthia, Ken, and Hiroshi's minds and generosity.

I would like to thank Dan Novy for his "dad" moments and support throughout these past two years. Dan, your ability to elevate my mood in the face of challenges and projects, and your creative problem-solving skills is unmatched. This work is better because of your insight, troubleshooting skills, and positive attitude. Thank you Emily Van Belleghem for your preliminary work on this system – your contributions lit the path for me to further realize this project. To Nina, thank you for all you did to help breathe life and aesthetic into this system. Without your efforts, especially in completely reimagining the hardware, my project wouldn't be nearly as beautiful or functional. I know you're going completely amaze MIT as a graduate student - and I can't wait to see how you conquer the lab over the next several years.

I would like to extend a big thank you to the Object Based Media Group for your support, friendship, and inspiration. You're all so amazingly passionate, and it was such an incredible privilege to learn from each of you through your work and our relationships. To my dearest lab-mates and friends, Aramael, Bianca, Sunny and Pedro - you have the biggest brains and the warmest hearts and the perfect amount of realness. I will continue to treasure our friendships, which ranks highly as the most rewarding parts of my experience at the lab.

My brother, Danny - thank you for being the best brother a sister can ask for. You have so much to offer the world – believe in yourself!

To my partner, my best friend, and my infinite source of intellect, humor, and support - Jonathan, you inspire me every day as you pursue new concepts that bring happiness to people's lives. Thank you for helping me grow, for your infectious laughter, and for finding Kaia, the world's best dog. May you always be older and more chill, but never smarter than me!

And finally, a tremendous thank you to my parents - for everything you have done for me, and for shaping me into the person I have become. Thank you for your immense selflessness, persistent motivation, and unconditional love that allowed me to grow into who I am today. I don't think I'll ever be able to fully express my gratitude – as it is infinitely deep.

TABLE OF CONTENTS

FUNNEL VISION

Low-Cost Auto-Stereoscopic 360-degree Display with Conical Reflection and Radial Lenticular + Contextual Artificially Intelligent Character with Procedural Animation

Emily M. Salvador

Imagination is everything.
It is the preview of life's coming attractions.

Albert Einstein

Part 1
Background

INTRODUCTION

Illusions have been used to enchant spaces since the early conception of storytelling. 20,000 years ago, our earliest ancestors used the flickering fire light to animate cave paintings to better instruct and entertain younger generations [68]. Since then, the tools humans develop further enhance our ability to suspend reality which allows us to imagine, create and resonate. From the invention of the camera obscura [37] to the advent of computer graphics [19], technological innovations of the current time catalyze a renaissance of expression. By inciting a sense of curiosity and wonder, technological innovations provide a distinctly human purpose. The most famous magicians know that their best performances incorporate the most fantastical technologies (while obscuring their mechanisms), producing perplexing and completely magical results for their uninitiated audience. Illusionists are acutely aware that a mismatch between what the eyes see and what the brain interprets can have profound effects in creating extraordinary.

The inception of the Pepper's Ghost illusion, on which this thesis heavily relies, dates back to 1862 [53]. While the original Pepper's Ghost illusions were achieved by illuminating performers and objects via oil lamps, by integrating modern day displays and projectors into the illusion, we can seamlessly render digital content within our physical world. With applications ranging from resurrecting performers like Tupac and Amy Winehouse [17] to allowing for moderately affordable AR headsets like the Lenovo Mirage [55], the Pepper's Ghost illusion has cemented itself as an essential tool for placing digital content in our physical environments.

To understand how perfectly paired the author is to this thesis, here is a brief overview of her background. Emily Salvador, before joining the MIT Media Lab, worked for Walt Disney Imagineering and Universal Creative where she investigated novel uses for the Pepper's Ghost illusion, which is frequently used in theme parks to place digital content into a physical context. Given theme parks' increased incentives to recreate prominent intellectual properties as attractions for guests to enjoy, the Pepper's Ghost illusion is a helpful way to integrate content that cannot be physically reproduced. Examples of the Pepper's Ghost illusion in theme parks include the ghosts in the Haunted Mansion attractions [64], the appearance of Harry Potter characters in various attractions at Universal Studios [34], and, most recently, transforming a skeleton into a physical animatronic of Captain Jack Sparrow on the Shanghai Pirates of the Caribbean attraction. Theme parks take these carefully crafted spaces for storytelling and elevate their impact through illusion and complete immersion.

Emily's projects for these theme park companies specifically focused on real-time interaction and storytelling through the intersection of the digital and the physical. Once a member of the Object Based Media Group, it became clear that creating a high-fidelity digital character that could exist in the physical world without the use of personal hardware could extend beyond themed space applications. These smaller, communal Pepper's Ghost displays could be used in homes and classrooms to enchant and engage users in familiar environments.

Funnel Vision shines as an alternative to augmented reality (AR) headsets and other volumetric displays because of its low cost and easy maintenance. Because each of the optical components of the display can be manufactured for pennies worth of plastic and because there are no moving pieces, this system is an ideal candidate for democratizing AR in an increasingly mixed reality world. Additionally, Funnel Vision allows for multiple users to share in their interaction with the device, allowing for more socially engaged collaboration and experiences.

While just developing the Funnel Vision hardware and shader would've been a thesis in itself, it was important to showcase how this platform would be used in a meaningful context. The author chose to develop a digital character that users could interact with in real-time with an emphasis on capturing facial reactions to the character, as it's currently difficult to capture facial reactions when the face is occluded by AR headsets.

CHAPTER 1
DISPLAYS

The Funnel Vision system is a novel display that combines radial optical partitions and conical reflections. Its nearest neighbor devices include holograms, volumetric displays, light field displays, and head-mounted display systems. Funnel Vision, without the use of a head-mounted display or special glasses, creates an affordable, durable and non-perspective occluding 360-degree viewing experience.

1.1 PEPPER'S GHOSTS

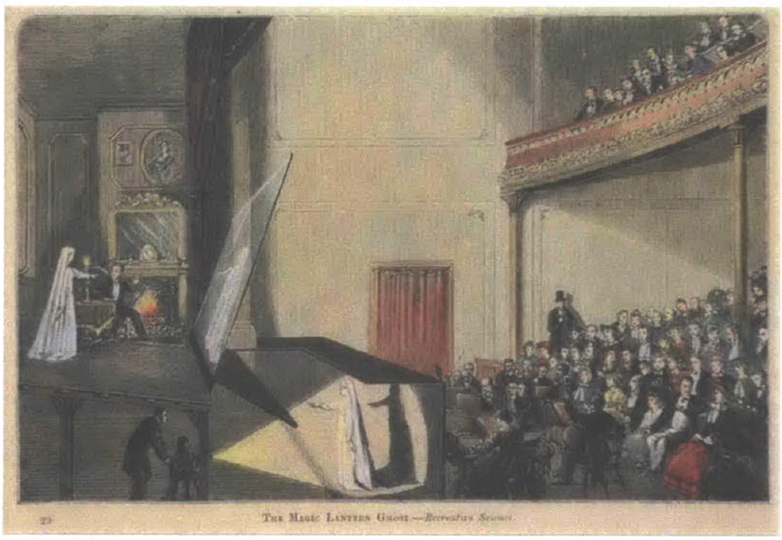


FIGURE 1.1: An Illustration of a Pepper's Ghost illusion in a theater [64].

The core mechanics of a Pepper's Ghost illusion can initially be dated back to 17th century where "ghostly" images could be produced by reflecting candlelight through a pane of glass [14]. However, its modern prevalence can be credited to Henry Dircks (1858) and later Professor John Pepper (1862) [12]. To produce the Pepper's Ghost illusion, a pane of glass (or half silvered mirror) must be positioned at a 45-degree angle relative to the viewer, so that a lit hidden object can reflect off the surface. An image of a glowing object appears as a reflection in the glass, rendered at the same distance as the illuminated object is away from the reflector. The 45-degree angle is typically desired because the reflected image will appear at 2θ, where θ is the angle of the reflective surface. As the reflective surface is moved away from the 45-degree angle, the reflected image appears to elongate or truncate as the new perspective is adjusted [18].

This phenomenon is further explained by Figure 1.2 below, in the modern context of using projectors or other displays to produce the illuminated source [13]. In the original Pepper's Ghost illusion, a three-dimensional object, like a performer or prop was often used to produce the effect. With the advent of display technologies like LCD screens and projectors, two-dimensional illuminated surfaces can produce two-dimensional dynamic reflections. While the original Pepper's Ghost illusion was intended for hidden live performers to haunt actors onstage, the combination of this illusion with modern emissive display technologies has proved transformative.

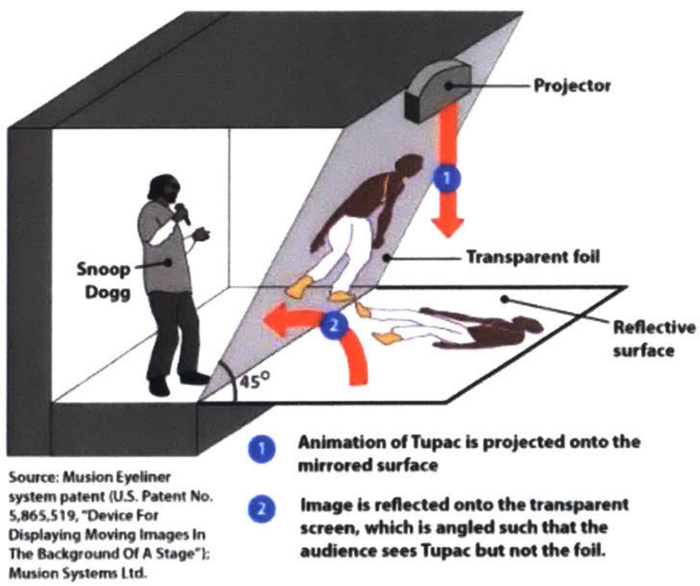


FIGURE 1.2: A diagram showcasing modern Pepper's Ghost techniques [13].

After the invention of digital displays, a Pepper's Ghost illusion could be produced from the light of a monitor instead of requiring illumination of an actor or physical object. The applications of this technology range from small scale illusions produced on smart phones and tablets to large stage illusions that can be viewed by thousands of people.

FIGURE 1.3: Examples of four-sided reflective pyramid Pepper's Ghost illusions across scales (left) [23] (right) [27].

This thesis was primarily inspired by projects like the commercial HoloMAX display [23] as well as "do-it-yourself hologram" tutorials from Instructables [27]. By employing a four-sided reflective pyramid format, the Pepper's Ghost illusion can be extended to allow 360-degrees of viewing. While this effect is absolutely impressive for such minimal hardware and software implementation, because of the use of planar mirrors, these devices present several obstacles for unobstructed viewing. First, because of the pyramid shape, the corner seams along the edge of each planar face disrupt the viewing experience. Additionally, these planar surfaces are sensitive to calibration errors and viewing position. Finally, the reflected illusion often deteriorates when viewed from oblique angles.

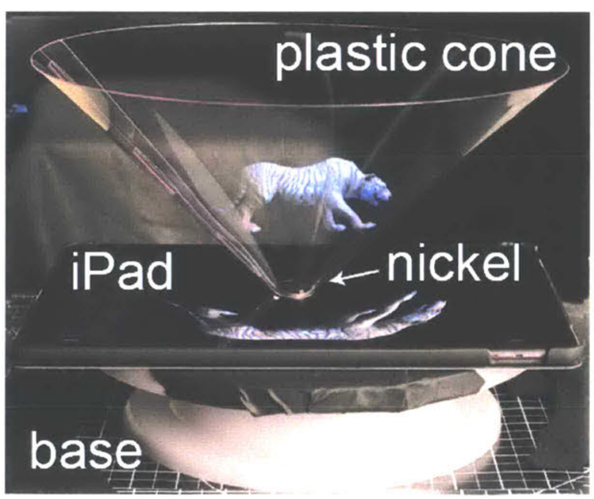


FIGURE 1.4: Conical Pepper's Ghost with Anamorphic Distortion [67].

Pepper's Cone [67], most notably, introduced the concept of reflecting anamorphically distorted, generated imagery onto a mirrored cone. While the Pepper's ghost illusion has existed in planar and 360-degree pyramidal form, the Pepper's Cone project [67], highlights the utility of curved reflective surfaces. This design feature allows for a seamless reflection surface [67].

1.2 HOLOGRAPHIC DISPLAYS

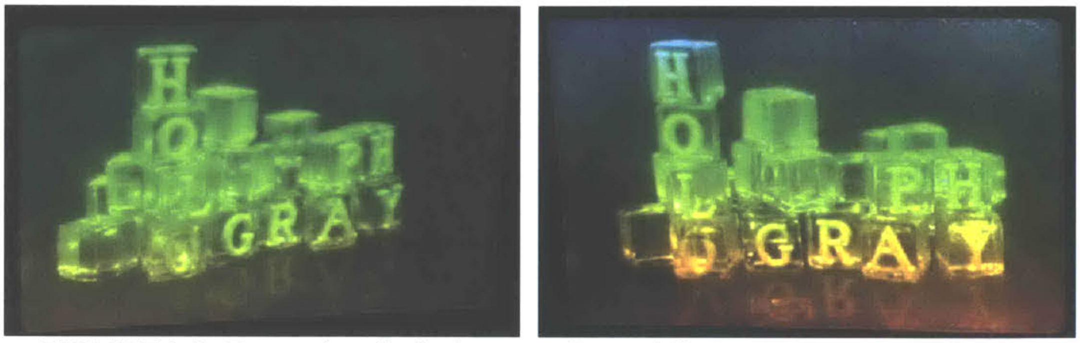


FIGURE 1.5: Example of a hologram from different viewing angles [4].

Holographic displays use laser interference to reproduce a continuous light field. Researchers have developed a variety of holographic displays using distinct materials and optical designs [4] [24] [31] [39]. Notably, the Object Based Media group itself develops techniques to produce holographic video [9]. Unfortunately, high resolution, fidelity, and frame rate holographic displays are still years away. Furthermore, devices like these will likely be expensive and require non-trivial maintenance in the short-term future.

1.3 VOLUMETRIC DISPLAYS

Through a variety of architectures, volumetric displays achieve a 3D matrix of individually addressable display elements. For example, by using a reciprocating mechanism, a 2D surface or mirror can be swept over a 3D volume at high speed and produce a volumetric image by synchronizing a projector or other display [50] [52]. Despite the components of the display moving to produce the volumetric image, due to persistence of vision, the user perceives a stationary 3D rendering. Other volumetric displays are achieved by coupling a projector with small optical reflectors, like fog, allows for a selectively illuminated volume [4]. Some volumetric displays provide correct stereo and parallax cues and can accommodate multiple simultaneous viewers. However, few can reproduce proper occlusions and view-dependent effects like specular highlights. Many volumetric displays suffer from low resolution due to the higher computation required to refresh the volume of display elements at an acceptable frame rate.

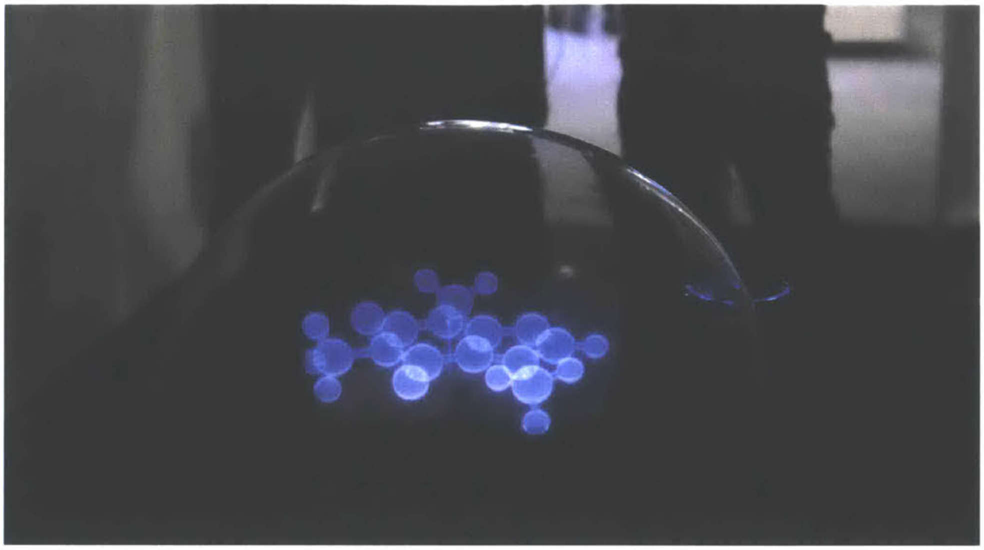


FIGURE 1.6: Example of volumetric display [61].

Standalone volumetric display products currently exist on the market. For example, Voxon offers a 360-degree field of view volumetric display by projecting imagery onto a high-speed reciprocating screen [61]. Projection frame slices reform an illuminated cross-sectional image at the right place in physical space. Through persistence of vision, the human eye blends hundreds of layers together to create an illusion of a complete volume. Content is ghosted, meaning users can see through to the back of the models, due to this generation technique.

1.4 LIGHT FIELD DISPLAYS

Autostereoscopic and light field displays are able to present a correct left/right stereoscopic image pair to a viewer over some viewing range without the need of special glasses [16]. Bonding a lenticular array [40] or aligning a parallax barrier [28] onto a conventional high resolution 2D display is a popular approach. Looking Glass leverages lenticular optics to create a stereoscopic image that can be viewed without wearing a headset or special glasses [59]. Another option is to combine multiple projectors using a reflective or transmissive screen that has a very narrow scattering profile.


FIGURE 1.7: Example of a lenticular-based light-field display [59].

Xia et al. [65] use light field generation to achieve a 360-degree surround viewable volumetric display with proper occlusion. Jones et al. [30] combine a fast spinning slanted anisotropic mirror with a synchronized projector to reproduce a light field that can be viewed from any angle. Many of these methods show the proper image to the viewer's left and right eye, which is critical for creating a believable volumetric illusion. It is non-trivial to accommodate multiple, simultaneous viewers, yet some devices are able to. However, there are some clear drawbacks to these types of displays. Many of the displays referenced are prohibitively expensive, complex to reproduce, and difficult to maintain. More recently, researchers have explored stacked arrangements of printed transparencies [62] and planar LCD panels [35] [63]. Although these systems can theoretically reconstruct a light field with proper focus cues, they do not allow examining an object from all sides, and current prototypes are either limited to static scenes or are inherently complex and prohibitively expensive.

1.5 HEAD-MOUNTED DISPLAYS

Head-mounted displays strive to augment our world and provide a ubiquitous, persistent, immersive platform. Due to the advent and coupling of high framerate displays with mobile high-bandwidth GPUs, large-scale consumer Virtual Reality (VR) efforts such as the Oculus Rift, HTC Vive, and Google Cardboard have become possible. Augmented Reality (AR) displays such as the Magic Leap and Microsoft Hololens overlay imagery onto the physical world through the use of micro projectors and mirrors or transparent waveguides. There are even AR displays that directly employ the Pepper's Ghost illusion to produce moderately affordable headset options [55]. Although it is impossible for Funnel Vision and other environmental Pepper's Ghost displays to achieve expansive viewing conditions, these alternatives don't require the use of personal hardware mounted to user's faces, allowing for a more natural and collaborative experience.

FIGURE 1.8: Example of an affordable Pepper's Ghost augmented reality headset [55].

1.6 FUNNEL VISION OPTICS

The Funnel Vision system uses a radial optical partition between the monitor and the conical reflection to separate the views for auto-stereoscopic viewing. Lenticular lenses and parallax barriers employ distinct mechanisms to separate an array of views underneath. Most prior art focuses on linear lenticular lenses and parallax barriers. Additionally, while the Pepper's Cone [67] cleverly employed anamorphic distortions for their conical reflection, anamorphic distortions have been used in other contexts.

1.6.1 PARALLAX BARRIERS

Parallax barriers were first conceptualized by Auguste Berthier, who published an article on stereoscopic images in 1896 [6]. Below, in Figure 1.9, there is a depiction of the core concept of a linear, two-view parallax barrier.

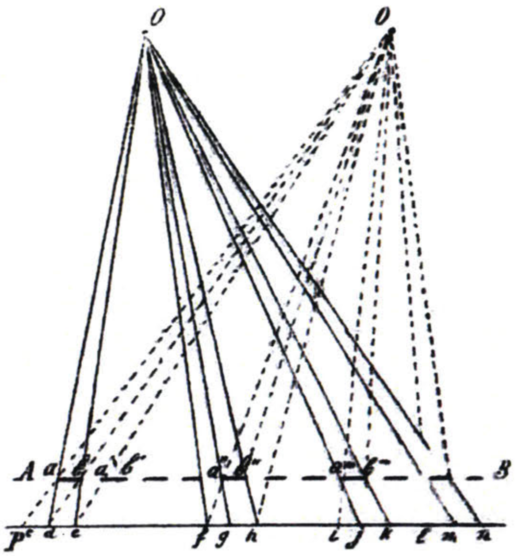


FIGURE 1.9: The first documented drawing of a parallax barrier [6].

The mathematics of a traditional parallax barrier depend on several key parameters. The setup for a linear, two-view parallax barrier is shown in Figure 1.10, taken from Towards a Common Framework for Parallax Barrier and Holographic 3D Display [49].

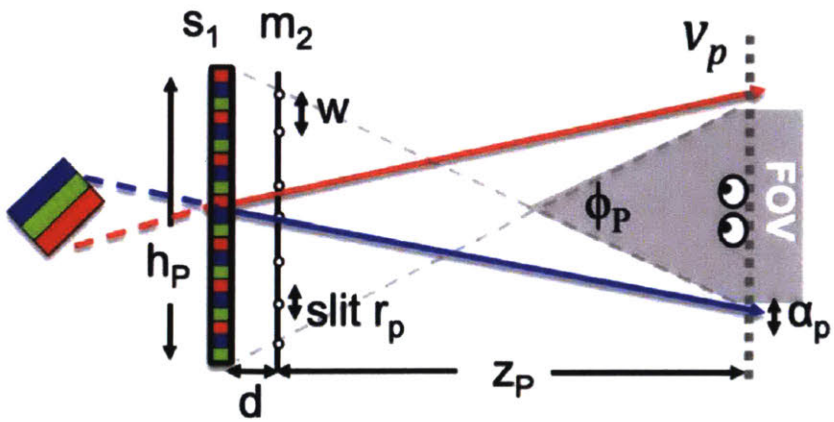


FIGURE 1.10: Setup of a linear, two-view parallax barrier [49].

The width of the opaque barrier and the adjacent, transparent aperture are identical and can be calculated through the pixel pitch of the monitor *p*. The distance between the parallax barrier and the pixel array, *d*, is given by:

$$d = \frac{z_p n p}{e}$$

where e is the approximate distance between the viewers eyes, z_p is viewing distance between the user and the optic, and n is the index of diffraction of the material between the parallax barrier and the screen [49]. Pixel pitch is defined as the width of a single pixel from the display, typically measured in micrometers.

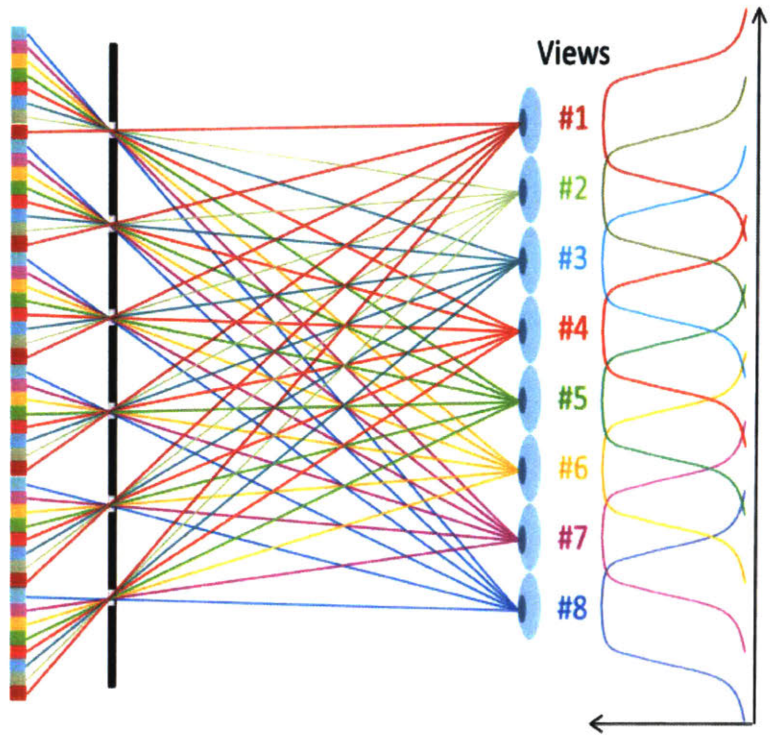


FIGURE 1.11: Setup of a linear, multi-view parallax barrier [21].

It is critical to note that this formula changes when more views are introduced to the system. The aperture of the parallax barrier is a ratio of the number of views total to the views intended to be seen in the aperture (typically one view) [21]. There are visible trade-offs for introducing more views to the display. There will be reduced brightness as only a small amount of light emitted from pixels passes through the parallel barriers. Additionally, for a display with N views, the resolution of any individual view is 1/N of the original display resolution for a linear parallax barrier. There is an image flipping artifact when crossing a viewing zone because the viewer's left eye may see the image intended for the right eye and vice versa when transitioning between views, which can distort the perceived depth. Additionally, when increasing the number of views, the width of the dark slit of the barrier increases, while the transparent aperture doesn't increase, causing display brightness decrease and a striped viewing effect.

Active research focuses on how to enhance the traditional parallax barrier effect using emerging technologies. Dynamic parallax barriers can adapt to where the user is positioned and where their gaze is pointed. With head tracking, the location of the user's eyes can be used to dynamically render the parallax barrier on a transparent LCD screen [48]. The Nintendo 3DS, a popular gaming console, does not use head tracking, however it creatively employs dynamic parallax barriers. The Nintendo 3DS dynamically toggles a transparent LCD screen with a parallax barrier mask over the underlying LCD to create a dynamic 3D effect [56]. Because the parallax barrier mask can be toggled on and off, the user can choose between 2D and 3D modes.

1.6.2 LENTICULARS

Integral imaging, first used in 1911, is technique for employed for creating three dimensional images from a series of photographs, commonly paired with lenticular lens arrays or parallax barriers [25]. The effect is accomplished without requiring the user to wear any optical device, such as 3D glasses. By relying on the user's perspective, the correct corresponding image can be displayed using optical techniques mapped to the image. Both optical partitions filter which views a user sees to give the effect of depth perception, however a lenticular approach is preferred in this context to a parallax barrier for several reasons.

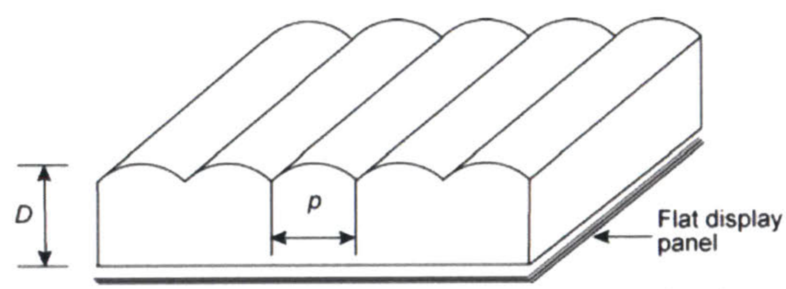


FIGURE 1.12: A up-close depiction of a piece of a lenticular sheet with cylindrical lens width p, and height D. [66].

First, rather than utilizing all of the light emitted from a screen for an illusion like a lenticular, a parallax barrier blocks a subsection of the emitted light with opaque masking bars. Because of this, the resulting image is typically dimmer due to the occlusion of a portion of the emitted light. The parallax barrier requires a distance between the barrier and generated image behind it in order for it to function as intended. This is sub-optimal for illusions with space restrictions and requires precise, persistent mapping. Comparatively, a lenticular can be bound to the integral image without any extra space, and the placement is depended on one axis instead of two. The lenticular is typically attached directly to the display, reducing the risk that the display will become misaligned over time.

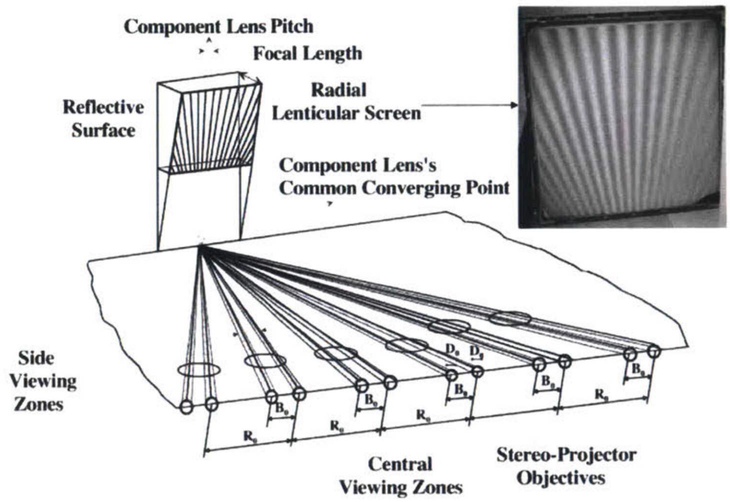


FIGURE 1.13: Radial lenticular design for 3D cinema, circa 1940 [7].

Lenticular sheets are usually made from one of the following materials: Acrylic (PMMA), APET, PETG, Polycarbonate, Polypropylene, PVC and Polystyrene. Those materials may be subtractive manufactured, 3D printed or cast in an injection mold.

1.6.3 ANAMORPHIC IMAGES

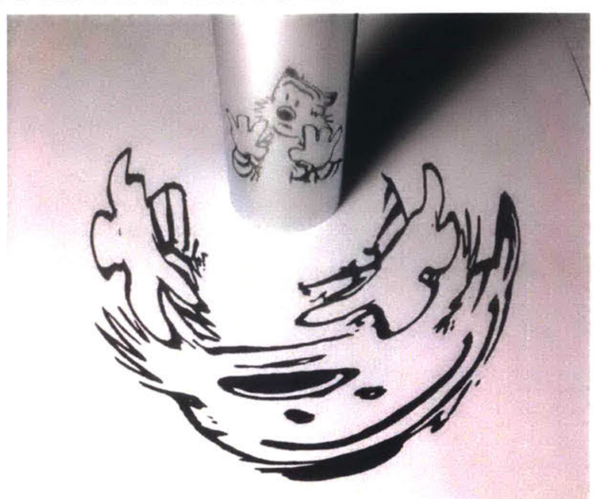


FIGURE 1.14: Anamorphically distorted image reflected in cylinder [27].

Anamorphic distortion can be defined as "a distorted projection or perspective; especially an image distorted in such a way that it becomes visible only when viewed in a special manner" [1]. Leonardo Da Vinci created the first recorded anamorphic art piece in 1485, however anamorphic renderings were not popularized until the 16th century [20]. Hans Holbein created a piece in 1533 entitled *The Ambassadors*, which is one of the most famous examples of anamorphic distortion to this day. The rendering in the foreground, if viewed from the correct angle, would appear as a skull [20]. There is even a popular meme in 2019 associated with anamorphic rendering – requiring the viewer to tilt their mobile phone to read secret messages.

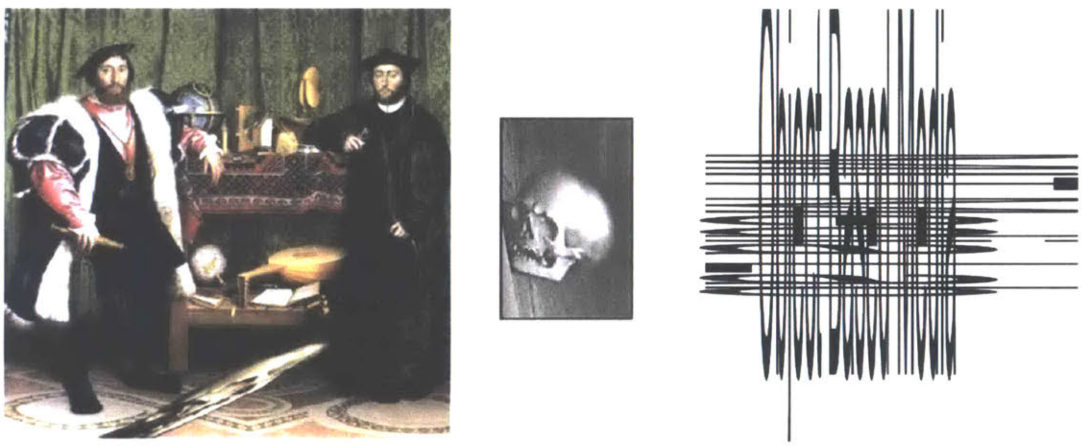


FIGURE 1.15: The Ambassadors by Hans Holbein as viewed straight on (left) and from the correct perspective to view the skull (middle) [20]. There is an example of text anamorphic distortion, on the right, which has trended as a popular meme in 2019.

Movie Maps [38] is an example of digital distortion to produce a correct anamorphic distortion in a conical mirror. Furthermore, in projection mapping, a technique where a projected image is distorted to map to the physical surface it's being projected on, corner pinning and projection matrixes are implemented to provide the correct perspective to the viewer. Dynamic moving eyepoint is employed on attractions at theme parks like Shanghai's Pirates of the Caribbean and the Na'vi River Journey to give the illusion of depth and motion on a projected surface [41].

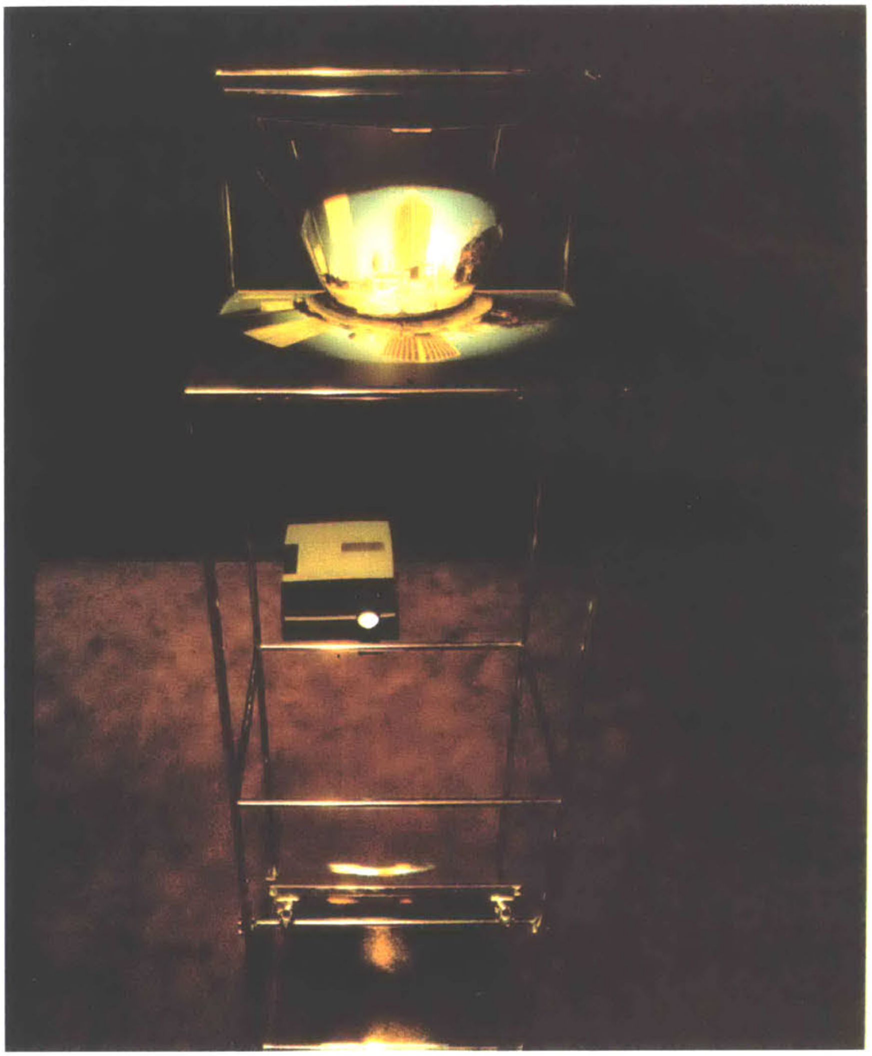


FIGURE 1.16: Optical System for viewing anamorphic images. Image is projected onto a horizontal screen with a conical mirror placed above. The image is viewed through a mirror [38].

CHAPTER 2 CHARACTERS

Procedural character design and animation is certainly an established field. With work ranging from emotional blend shapes [43] to real-time contextual animation [47], there is an impressive catalog to draw inspiration from.

2.1 SOCIAL ROBOTS

FIGURE 2.1: Examples of social, physically embodied robots with personality. From L to R, Vyloo [45], Tega [33], and Jibo [10].

The Vyloo animatronic characters located in the Guardians of the Galaxy: Mission Breakout queue at Disney California Adventure respond to emotional cues based on the face pose of the users they interact with [45]. The Vyloo animatronics personalities are procedurally generated based on a limited number of emotional parameterized indexes which inform how the character will physically animate [45]. Tega is a storytelling robot developed in the Personal Robots group [33]. Jibo, developed by Cynthia Breazeal, adds physically embodied character to the voice assistant paradigm (like Google Home or Amazon Alexa) [10]. Leonardo, another social robot developed in the Personal Robots group, added a new level of artificially intelligent social engagement, as the character could make permanent emotional associations about objects and people as they were presented – remembering the associations later [11].

2.2 DIGITAL CHARACTERS

FIGURE 2.2: Examples of digital character companions. On the left, Tamagotchi, incredibly popular in the 90s, relied on 8-bit graphics and simple button based interactions. Graphics techniques and affordability of high-resolution displays have improved since, allowing for devices like the Nintendo Labo (right) [36].

Responsive digital characters manifest in many platforms and contexts. Tamagotchi in the 90's used simple button interactions to establish a relationship with an 8-bit sprite character. The Nintendo Labo device allows users to introduce physical props into the play experience with a responsive character [36]. In Project Porg, developed by ILMxLAB and Magic Leap, through a series of episodic interactions, users build a relationship with an augmented reality family of alien bird creatures from the Star Wars universe [22].

Furthermore, digital characters that emote and sympathize have a long-standing history both at the MIT Media Lab and in the broader research arena. The work from Bickmore and Picard [7] on using empathy as a mechanism to improve long-term relations with computer agents, as well as the integration of empathy as a key element in the design of synthetic characters inspire current innovation in this field. This work is formative as it demonstrated how humans can both perceive and project empathy onto technological devices. From a computational perspective, there is active research on how to create characters that understand emotion and how to procedurally generate empathetic, clear signals. Sabourin [51] created a framework for algorithmically assessing generated empathy.

CHAPTER 3
MIXED WORLDS

3.1 PHYSICAL SPACES

FIGURE 3.1: Examples of VR/AR mapped to the physical world in a liminal space. On the left, players use VR headsets to be transported to a completely digital, interactive world [60]. On the right, augmented reality content is rendered in the physical environment [3].

Mixed reality, by some definition, is an established field. From circuses and carnivals in the early 19th century to projection mapped stage shows on Broadway, in the entertainment context, our reality in the physical world can be influenced as we enter special, controlled spaces.

Theme parks like Disneyland and Universal Studios enchant their audience by providing a completely immersive and expansive physical environment that entices all the senses. From traditional dark rides like the Haunted Mansion and Pirates of the Caribbean to mixed environments that seamlessly blend screens with the physical world like The Amazing Adventures of Spiderman and Star Wars: Rise of the Resistance, the theme park industry creates spaces that completely subvert mimetic reality.

Reimagined and reenchanted physical spaces are trending as digital markets saturate and stagnate market share. From Amazon's acquisition of Whole Foods to Google's Pixel 3 themed takeover in Piccadilly Square in London, technology focused companies want to reimagine interaction and content distribution in the physical world. Apple's former head of retail is quoted as saying "I think as humans we still need gathering places, and when you are serving digital natives, the thing they long for more than anything is human connection. Eye contact... Stores need to become living, breathing spaces, not just two-dimensional boxes." Some brands have achieved this more successfully than others. In 2019, Spotify launched a buzzword filled 'interactive, immersive music experience' for Billie Eilish's new album, but the space was gimmicky, sparse, and overall lackluster.

The VOID and similar companies have produced impressive digitally overlaid physical worlds, where virtual objects, set pieces and characters feel solid due to mapping in the real world [60]. Taking the VOID's concept a step further, AT&T recently partnered with Magic Leap and HBO to bring mixed reality demos to the public. At select AT&T stores, consumers have the opportunity to demo Magic Leap headsets and fight white walkers in an environment themed to Game of Thrones [3].

3.2 CONSUMER PRODUCTS

FIGURE 3.2: Pokémon Go is a popular example of mobile augmented reality and location-based interaction [42].

In addition to physical spaces embedded with specialized hardware to project digital content into the world, there are now a number of consumer products and devices that enhance immersion. We've already covered

VR/AR headsets in Section 1.4, although they are the leading example of consumer products that overlay digital content on our physical world.

Mobile applications like Snapchat and Pokemon Go allow individual interaction and control of digital content mapped to the physical world. Snapchat allows its users to place digital content in their environments and can even map and distort content on the user's face. Pokemon Go has catalyzed the popularity of mobile AR contextualized in the world – garnering a record number of users in a short window of time [42].

Lightform Inc. is a consumer product that enables its users to easily compute projection maps and deploy content to a physical scene. With the technical literacy level of a digital graphic designer, artists and hobbyists can now easily map and project digital content on physical objects and scenes [58].

The Disney Play App allows guests at the Disney resorts to indirectly manipulate their surroundings through their smartphone applications. Localizing the user through GPS and Bluetooth beacons allows the guests to influence the environments they're geolocated to [15].

Part 2
Funnel Vision

CHAPTER 4
HARDWARE

4.1 THE SYSTEM

Funnel Vision is a 360-degree display that generates discrete imagery based on angular position around the display by using a radial mirror, radial lenticular or parallax barrier and a monitor. A 3D scene is rendered in a game engine, virtual cameras are generated to capture the scene, and a shader is applied to partition the views from the cameras appropriately on the monitor. This system is uniquely balanced to allow for the optical components to operate efficiently as a hardware system. The Funnel Vision system is housed in a custom created cabinet that holds the computer underneath the monitor and optical setup. The cabinet allows for a 360-degree view of the conical reflection, but can be modified to block ambient-light, which subsequently increases contrast in the conical reflection.

4.2 RADIAL OPTICS

4.2.1 LENTICULAR APPROACH

Integral imaging allows for the volumetric information associated with a 3D object to be packaged for display onto a 2D surface. The "3D information" is rendered specifically for use with an associated optical barrier or partition. Without the optical barrier or partition, the integrated, rendered image is blurry due to the fact that the views have been sliced and repositioned into an interleaving pattern. The optical partition, whether a parallax barrier or lenticular, then filters the views so the viewer will only see at most one view per eye (allowing for a stereoscopic effect). This is critical for the purposes of this project as a flat image, displayed on a monitor, can transform into a volumetric illusion.

This Funnel Vision display can create autostereoscopic imagery up to 360 degrees around a 4K monitor. The radial lenticular allows for horizontal motion parallax, which allows the hardware system to display different imagery as users move angularly around the display. As a user walks circumferentially around the display, the appropriate views are cast in that particular viewing zone. The rays from the monitor pass through the lenticular lens which partitions the pixels that are directly underneath the lens into angular paths. As the user moves off axis of the lenticular lens, new views come into focus and are magnified under the lens. Those rays that are cast radially out of the lenticular lens then bounce off the radial mirror, further deviating the path of the image. If the user is standing directly in front of a lenticular lens, the view directly under the lens will be in focus and bounce straight off the mirror directly towards the user. However, the lenticular lenses off-axis to the user will focus a particular view based on the angle, which will then bounce off the radial mirror, magnifying that offset angle.

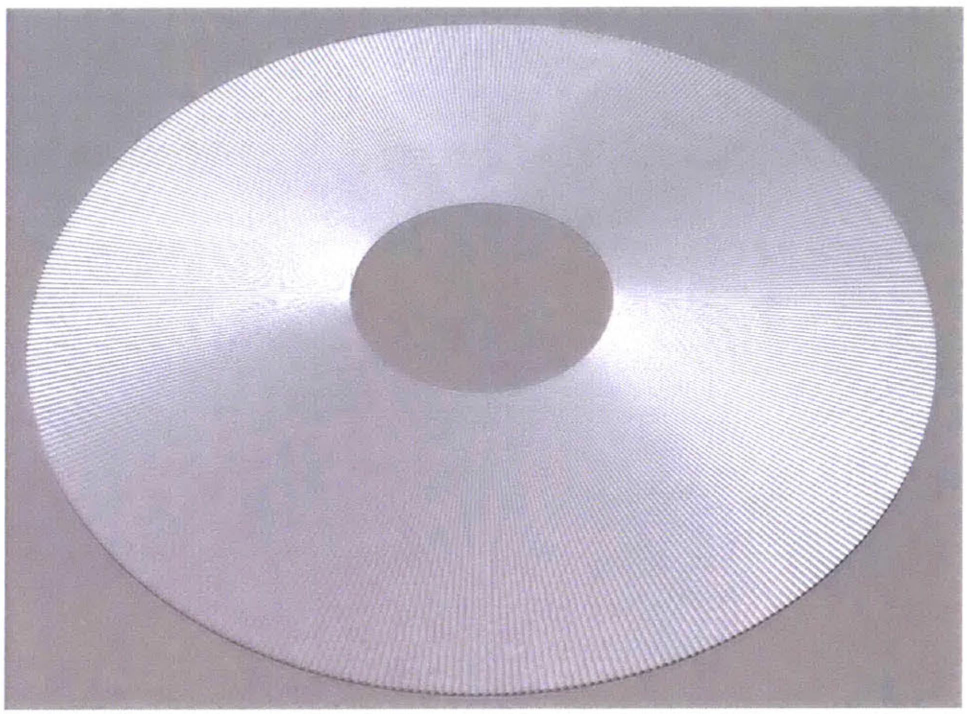


FIGURE 4.1: Rendering of initial radial lenticular design in Fusion 360.

The shape of the radial lenticular lenses diverges from linear lenticular lenses. Radial lenticular lenses are wedge shaped (the width of the lenses increase towards the peripheral of the display), because of the polar distortion and maximizing use of the pixel space on the monitor. Consequently, the height of the lenticular lenses must increase as the width increases to maintain focus. Alternatively, the lenses can change profile rather than height so as to maintain the same focal length as they become wider. The focal length, F, can be calculated by the radius of the curvature of the lenticular, r, divided by the lens's index of refraction, n, minus one.

$$F = \frac{r}{n-1}$$

Usually, the focal length is calculated to extend to the base of the lenticular. Depending on what the lenticular is designed to focus to, there may be an additional distance to reach the desired focus plan. For example, many monitors have a transparent barrier protecting the light emitting component and the width of that barrier should be considered when determining the focal length of the lenticular lenses. The shape of the lens influences the viewing angle for each lenticular lens which results in a shorter or longer phase distance between views under each lenticular lens.

Additionally, the angle within the lens is variable. It is difficult to manufacture perfect half-circles lenses because of limitations on existing tools, therefore many lenticular manufacturers will adjust the angle within the lens resulting in a substrate of variable height below the surface of the lenses. This can be calculated by the formula below:

$$e = \frac{nr}{n-1}$$

where *n* represents the index of refraction and *r* represents the radius of the curvature of the lenticular. Additionally, this absolutely impacts the angle outside the lens or the viewing angle of the lenticular lens. That angle, *O*, and the internal angle, *I*, can be calculated using the following formulas:

$$O = 2(\sin^{-1}\left(\frac{p}{2r}\right) - I\ )$$

$$I = \ \sin^{-1}(\frac{n\sin(R)}{n_a})$$

where *p* represents the width of each lenticular lens, *r* represents the radius of the curvature of each lenticular lens, *n* represents the lens's index of refreaction, n_a represents the refractive index of air, and *R* represents the angle between the extreme ray and the normal point where it exits the lens.

Keep in mind, that because this was a wedge shaped radial lenticular design, *p* scaled accordingly as the width increased towards the periphery of the display. In the case of applying radial lenticular lens array to the Funnel Vision display, these constraints proved problematic. Because the monitor emits light, and the thickness of the substrate on which the lenticular was printed on was substantial, views that were intended for one lens would appear under adjacent lenses producing a "flipping" effect.

4.2.2 PARALLAX BARRIER APPROACH

For a radial parallax barrier, the barrier can be printed using an inkjet printer onto transparency film. As long as there is a transparent aperture and an opaque barrier, the parallax barrier should function. There should be a gap between the monitor and the parallax barrier plane. When increasing the gap, the relative size of each parallax barrier wedge should increase to accommodate the offset. The gap size influences how quickly views shift into the aperture of the parallax barrier. Additionally, the relative size of the aperture to the size of the barrier per each wedge of the parallax barrier is a ratio of the number of views under each wedge – 1 to 1. For example, if there were 5 views under each parallax barrier wedge, then the barrier portion of the wedge should represent 80% of the wedge while the aperture would represent 20%.

4.3 CONICAL REFLECTION

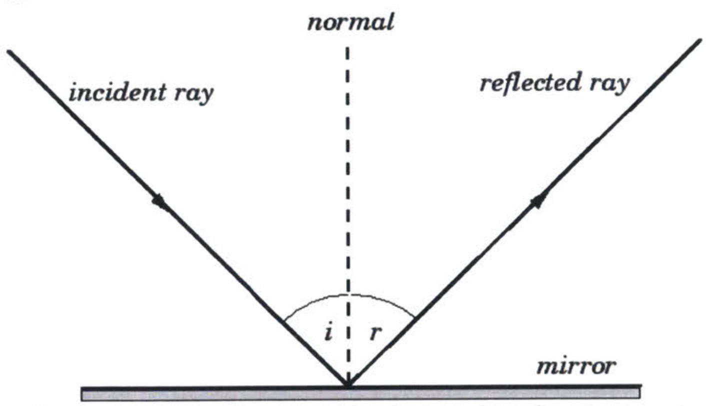


FIGURE 4.2: A diagram representing the Law of Reflection, where the incident ray produces a reflected ray perpendicular to the normal [18].

The Law of Reflection, demonstrated in Figure 4.3, is the primary mechanic that generates the Pepper's Ghost illusion. An incidence ray of light aimed at a mirror at an angle *i* forms a 90-degree angle combined with the reflected ray's angle *r* [18]. This law applies to all reflective materials, however the amount of light conserved in the reflective bounce step depends on how reflective the piece of material is; this ranges from a pane of glass that reflects very little light to a mirror that fully reflects the light.

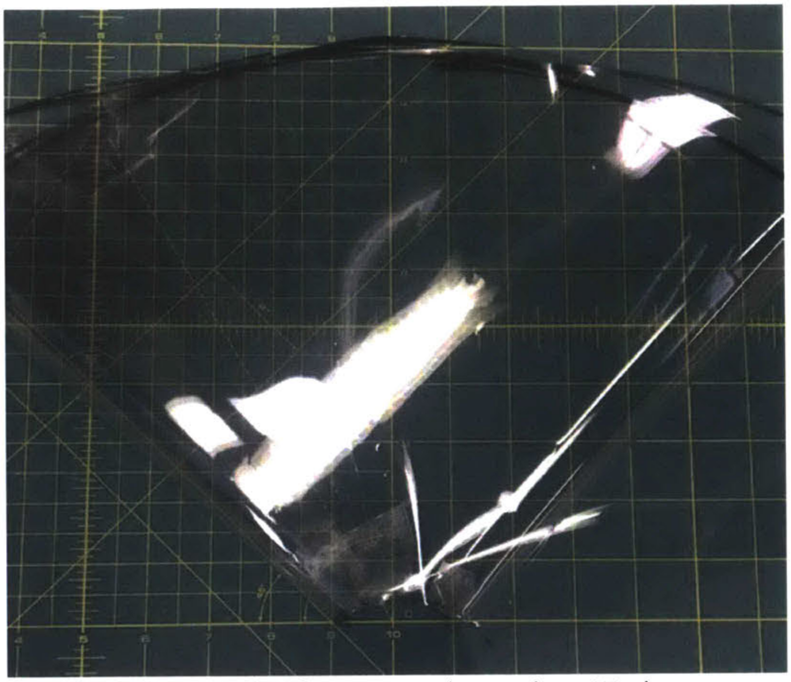


FIGURE 4.3: A cone made of reflective mylar with a 45-degree angle edge.

The shape of the radial mirror could be the frustum of a cone, elliptic paraboloid, sphere, cylinder, or of any other object with one axis of rotational symmetry, however the underlying shader (or integrated image) would need to be modified to account for the shape's anamorphic distortion. The material of the mirror should be reflective, and examples include half-silvered mylar, acrylic, non-transparent reflective acrylic or full mirror. Mylar was chosen as the ideal material for the cone, because of its high refractive index relative to its semi-transparency. Exactly how much light is lost and at what angle is best modeled by using the Fresnel equations [46] and Snell's Law [54] shown below.

Frensel's Equations:

$$t_s = \frac{n_1\cos(q_i) - n_2\cos(q_t)}{n_1\cos(q_i) + n_2\cos(q_t)} \qquad r_s = \frac{2n_1\cos(q_i)}{n_1\cos(q_i) + n_2\cos(q_t)}$$

$$r_p = \frac{n_2\cos(q_i) - n_1\cos(q_t)}{n_1\cos(q_i) + n_2\cos(q_t)} \qquad r_s = \frac{2n_1\cos(q_i)}{n_1\cos(q_t) + n_2\cos(q_i)}$$

Snell's Law:

$$n_1\sin(q_i) = n_2\sin(q_t)$$

4.4 SYSTEM EFFECTS

As the user moves circumferentially around the display, the rendered views that are opposite the user (i.e. on the other side of the radial mirror) cannot be seen. Because of this assertion, it became evident that the user doesn't need to see all of the potential views from a static position, therefore radial lenticulars operate differently than linear lenticulars. With a linear lenticular, every view that will be displayed needs to exist under every lenticular lens for the effect to work. For example, if there are 8 images that have been integrated, there would be a slice of each of those 8 views underneath each lenticular lens. This results in a per-view image resolution of 1/N where N is the number of views being rendered. As the number of views increase, the per-view resolution decreases.

However, with this setup, it is not necessary to produce every view under every lenticular lens because each lenticular lens only needs to display the views that the user could potentially see from that lens. Because of that, there is a rolling assignment system for which views to display under each lenticular lens. If the user is standing directly in front of where a viewing zone is located for a particular image, the lenticular lenses closest to the user should have that view centrally located under the lenticular. The lenticulars that are adjacent to that area should still have information that can reach the user, however the views have to shift to accommodate the off-axis position. In that way, there is a rolling priority system, where the generated view that is closer to the user will appear centrally under the lenticular lens and incrementally move off-center and eventually disappear as the user moves radially around the display. This is unique to this system due to the favorable, synergetic conditions of the radial optical partition and the conical mirror.

CHAPTER 5 SHADERS

5.1 LENTICULAR RENDERING

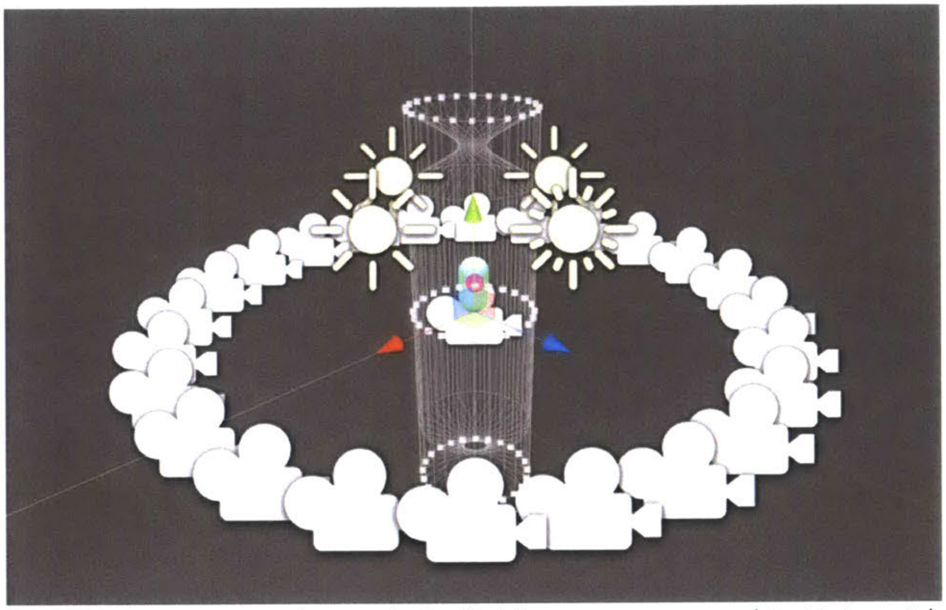


FIGURE 5.1: Cameras rendering light field auto-generated in Unity with a C# script. Note, the volumetric capture area is located within the frustum of all the cameras.

Harkening back to the previous section, this system required a different, more efficient approach to lenticular rendering. As the user moves circumferentially around the display, the rendered views that are opposite the user (i.e. on the other side of the radial mirror) cannot be seen. Because of this assertion that the user doesn't need to see all of the potential views from a static position, radial lenticulars operate differently than linear lenticulars. Therefore, it became clear that if the user is standing directly in front of where a viewing zone is located for a particular image, the lenticular lenses closest to the user should have that view centrally located under the lenticular. The lenticulars that are adjacent to that area should still have information that can reach the user, however the views have to shift to accommodate the off-axis position. In that way, there is a rolling priority system, where the generated view that is closer to the user will appear centrally under the lenticular lens and incrementally move off-center and eventually disappear as the user moves circumferentially around the display.

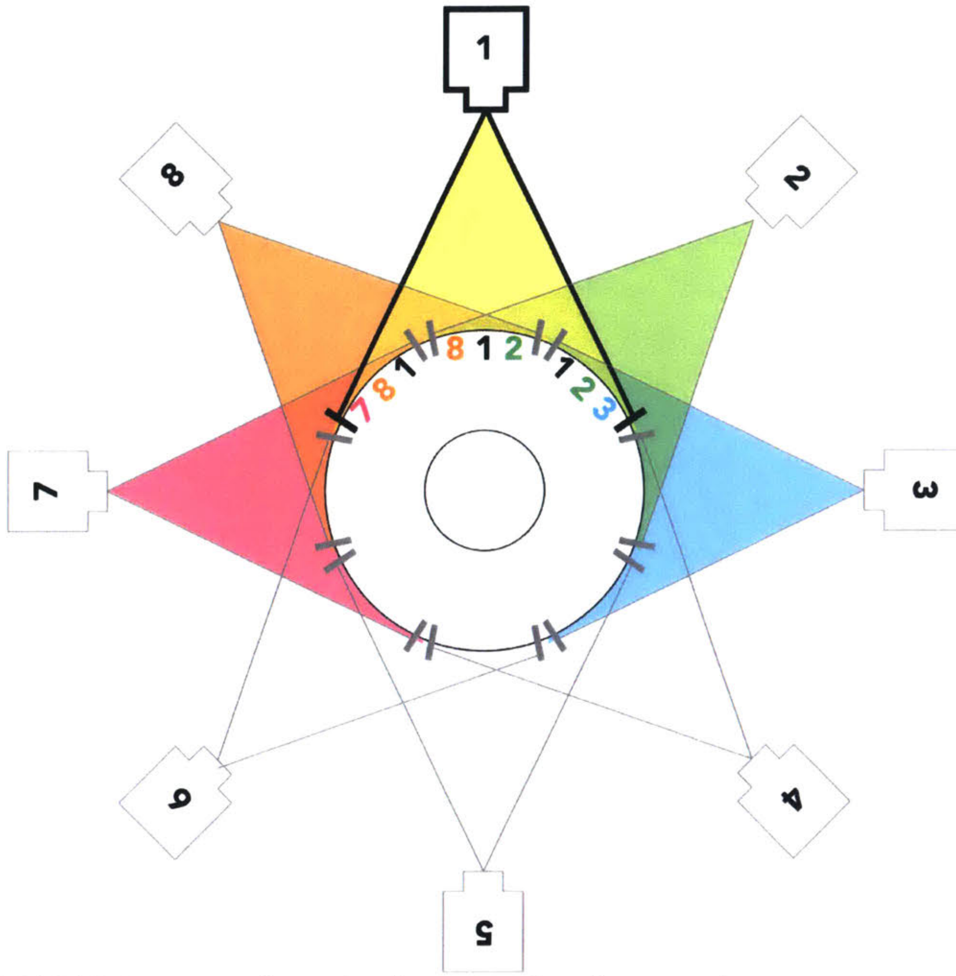


FIGURE 5.2: Diagram depicting how the virtual camera images are processed and displayed in each portion of the display screen.

In order to generate the views that are rendered on the display based on the 3D scene, there are three independent variables that must be considered. First, the number of the virtual cameras/views/viewing zones must be decided. In order to decrease transitional areas in the display, the goal is to optimize for number of viewing zones, which means maximizing the number of virtual cameras in the scene. Second, the number of views per lenticular lens is variable depending on how quickly views should change based on the user's radial position. Third, the number of times a specific view zone lenticular pattern should repeat should be considered. For example, an integrated pattern of virtual camera views 1-5 might repeat many times to increase horizontal resolution and size of the virtual image. These three independent variables impact the system substantially for the following dependent variables which are outlined below.

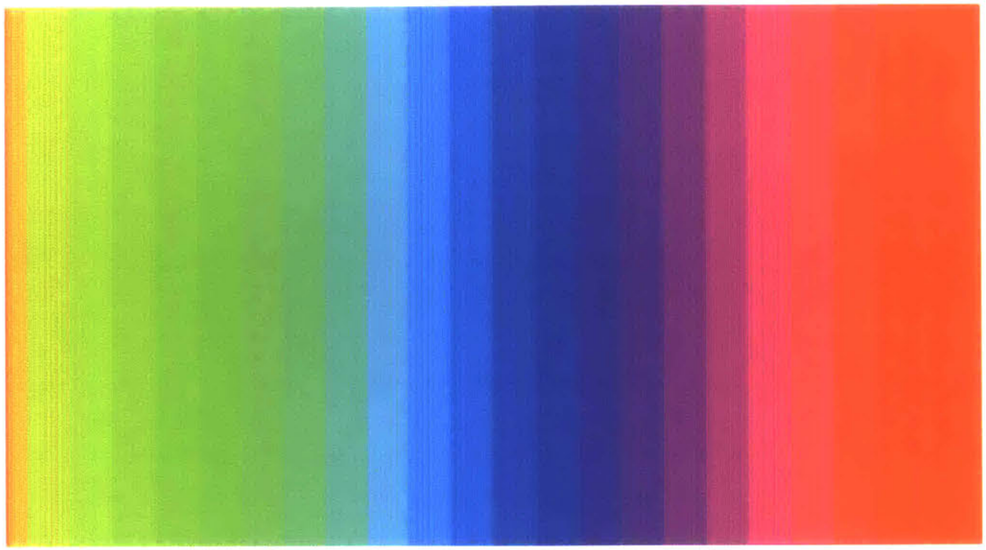


FIGURE 5.3: Lenticular rendering of 24 camera views, where each camera view is a different hue (see appendix for higher resolution).

The first dependent variable that is impacted is the angular influence that a particular camera view will have on the monitor. This is important, because if the angle is too small, the virtual image generated will not take advantage of the full horizontal size, and if it's too large, the view will disappear off the radial mirror and not be relevant for the user, thus unnecessarily contributing to a lower resolution per view. The angular influence for a 360-degree display can be described as 360 degrees multiplied by the number of views per lenticular lens divided by the number of cameras/views. For example, if there are 20 cameras and 5 views per lenticular, the angular influence on the monitor for a particular view would be $360° \times 5 \div 20$ which would equal 90-degree angular influence.

The next variable that can be calculated with the three independent variables is the angular size of each lenticular lens. This is another important value to consider because of the physical, manufacturing and optical constraints of the radial lenticular. To calculate the angular size of each lenticular lens for a 360-degree display, divide 360° by the number of cameras/views and the number of times each that specific view configuration repeats. For example, if there are 20 cameras/views and for a specific view there are 20 repeats of that integrated pattern, the angular size of each lenticular lens would be $360° \div (20 \times 20)$ which equals .90 degrees.

Because we are constrained to an existing monitor with a specific pixel pitch, we must calculate the minimum viewing radius on the display. That radius is the point where there is one pixel along the circumference of the circle for every slice of every view on the display. To calculate the minimum viewing radius, multiply the number of cameras/views by the number of views per lenticular and the number of times each view configuration repeats. Then divide that number by 2π. For example, in a system with 20 cameras/views, 5 views per lenticular lens, and 20 repeats for each configuration, the minimum viewing radius is $20 \times 5 \times 20 \div (2\pi)$ which equals 318.3px. Using the pixel pitch of the monitor, that value can be converted to other units.

The remaining dependent variables describe the size and resolution of the virtual image generated in the radial mirror. The resolution of the virtual image increases vertically because of the wedge shape of each lenticular. There are more available pixels at the edge of the monitor as the wedge widens which results in a higher pixel resolution. The physical height of the virtual image remains constant because of the anamorphic distortion applied to the camera/view. The resolution of the width of the virtual image also is variable along the vertical axis. To calculate the minimum resolution for the width of the virtual image, multiply the number of views per lenticular by the number of times that configuration of views repeats. That will return a pixel width which can then be converted using the pixel pitch of the monitor into the horizontal size of the virtual image.

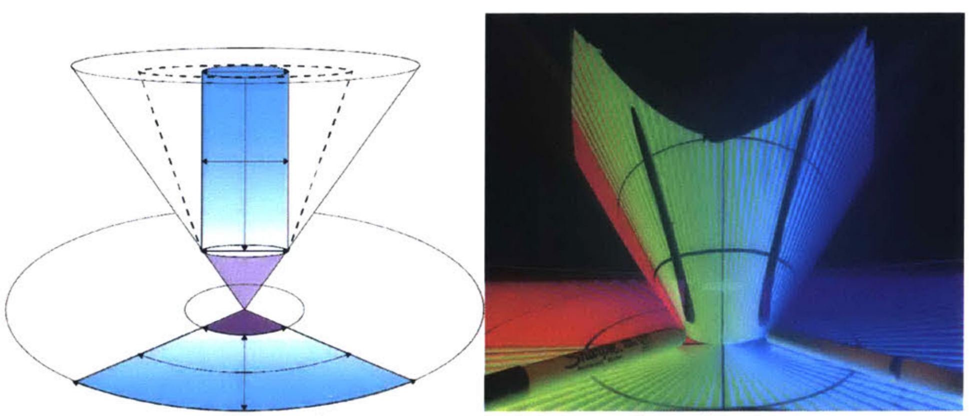


FIGURE 5.4: On the left, a diagram showing that the imagery generated on the monitor corresponds to a conical viewing volume, where the image is rendered on a place that circumvents the cylinder. On the right, imagery generated on the monitor is reflected in the cone, producing the conical viewing volume.

When increasing the number of cameras/views, all other variables equal, the horizontal and vertical resolution and size of the virtual image will decrease. Additionally, the minimum viewing radius will increase and the angular extent of each individual view will decrease. The angular size of each lenticular lens will shrink.

When increasing the number of views per lenticular lens, all other variables equal, the horizontal and vertical resolution and size of the virtual image will decrease. Additionally, the minimum viewing radius will increase and the angular extent of each individual view will increase. The angular size of each lenticular lens will remain constant.

When increasing the number of repeated configurations of integrated slices, all other variables equal, the horizontal and vertical size of the virtual image generated will increase, however the resolution of the image generated will decrease. Additionally, the minimum viewing radius will increase and the angular extent of each individual view will increase. The angular size of each lenticular lens will increase.

FIGURE 5.5: This image illustrates that as the selected camera rendering angularly changes, the image captured must shift its origin accordingly.

In the 3D scene, cameras are generated radially around the 3D stage. In software, the cameras field of view can be adjusted to accommodate for the angular extent of the view on the monitor and the number of views being generated. Additionally, the radius of the camera ring is also variable. After capturing the scene from each of the cameras' perspectives, the frames from each camera are sliced and rearranged in a shader onto a render texture. The height of the render texture is equal to the maximum radius of the display (in pixels) that can generate a circle (not many monitors have a square aspect radio). The width of the render texture is equal to the number of cameras/views multiplied by the number of views per lenticular and the number of times that specific view configuration repeats.

5.2 ANAMORPHIC DISTORTION

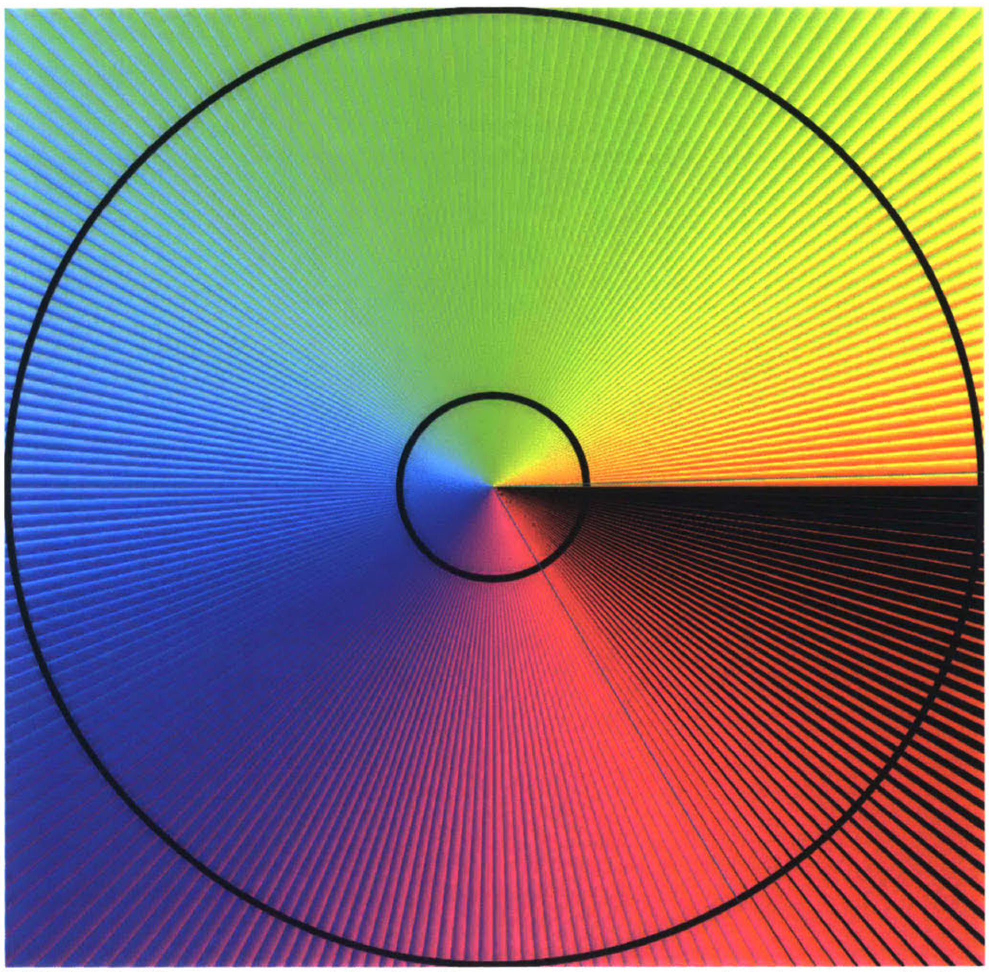


FIGURE 5.6: The anamorphically distorted, integrated debug rendering. The inner and outer circle represent the minimum and maximum viewing range accordingly.

Next, that light-field render texture is sent through a second shader process that radially distorts the image around the center of the monitor. This process turns the single pixel wide lines from the render texture into wedges that extend from the center of the monitor. As each wedge extends from the monitor, the number of pixels per wedge increases. To avoid stretching and repeating pixels and to maximize the pixels that are sampled, this process could be streamlined to sample pixels based on the radially distorted location on the monitor.

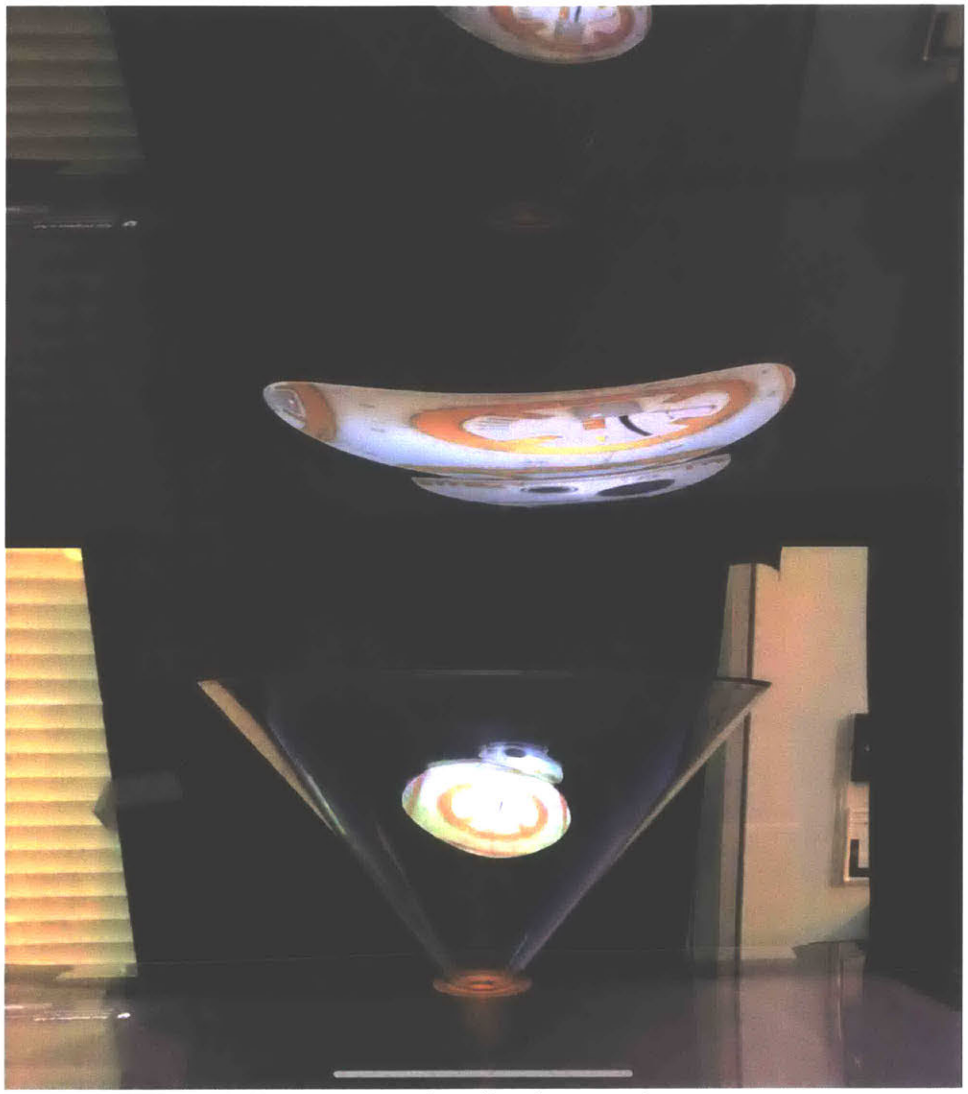


FIGURE 5.7: Above, the anamorphically distorted image as it appears on the monitor. Below, the same image appears un-distorted after it has reflected off the conical mirror.

Part 3
Procedural Character

CHAPTER 6
PROCEDURAL CHARACTERS

After creating a real-time system in Unity and hardware, the next step was to provide a meaningful interaction to validate the utility of this design. Because of the ubiquity of real-time voice assistants like Amazon Alexa and Google Home, we decided to pursue a digital character companion to showcase the potential of this display. Additionally, because this display differentiates itself from augmented reality glasses which obscures the user's face, we chose to develop interactions that depended on the affective state of user, particularly face pose analysis.

6.1 DESIGN

FIGURE 6.1: Character design for procedural character animation [2].

For displaying an emotive character to users in the system and given the limited resolution of the monitor's reflection, it was important to carefully consider character design. For this thesis, the above dog character with a large head relative to the body and high contrasting eyes and mouth was chosen. Additionally, a dog character seemed like a logical choice because there are established associations for physically embodied emotion for dogs (i.e. wagging tail indicates the dog is excited, downturned ears and head indicates sympathy or sadness). We purchased an already rigged model from Anko3D of a cartoonish dog character that met these criteria.

FIGURE 6.2: Rendering of terminal blend shape positions for various emotion-based parameters [2].

The dog character, named CAIA (after a real-life dog and as an acronym meaning Contextual Artificially Intelligent Agent), was created in 3DSMax with skeletal rigging and blend shapes for the face that define hero poses for each of the possible emotional embodiments the character might have. These facial blend shapes allow the character to animate its mouth, eyes and other features smoothly by interpolating points on the mesh between the various poses. The character was then imported into Unity, where the character could be procedurally animated in real-time based on the parameter analysis from various input devices (i.e. Intel Realsense camera or microphone).

FIGURE 6.3: Skeletal rigging and blend shape parameters for the digital character exposed [2].

6.2 HARDWARE

Intel Realsense and similar depth sensing devices like Kinect and Leap Motion provide SDKs that allow developers to readily stream depth and color data into Unity. For this thesis, the Intel Realsense camera was chosen for a variety of reasons, including peripheral integration, ease of use, and resolution. The Intel Realsense data can be further processed and interpreted using software like Nuitrack which uses machine learning to compute and perform skeleton tracking which exposes body pose and joint positions. This information can be used to allow the system to know when a user is close enough to the display or standing a particular place. The data collected can be further interpreted to allow for gesture recognition as well. The color data from the Intel Realsense camera can be processed in OpenCV or similar libraries which have Unity SDKs or libraries. With OpenCV or similar, the color data can be analyzed for object detection, face detection and recognition and face pose analysis because of the resolution of the Intel Realsense camera. This allows the system to recognize objects, people and face pose, which could be used to interpret affective state. Face and gaze detection could be used in lieu of trigger words like "OK Google" and "Alexa" as presumably the users have intent to interact with the system if they're looking at the character/sensors.

For a system like this, microphones can be easily integrated to further enhance the character interaction. Although we didn't include this hardware in this initial design, sound recordings could be used to detect volume, pitch and speech. The speech can be analyzed using a cloud-based service, which then streams the input and a response back to Unity to influence how the character animates and responds. Speech analysis can be used to interpret special assigned words to trigger activities, games or special animations or content in the display. The response generated by the cloud-service can be used to animate the characters' mouth if the character audibly responds to users.

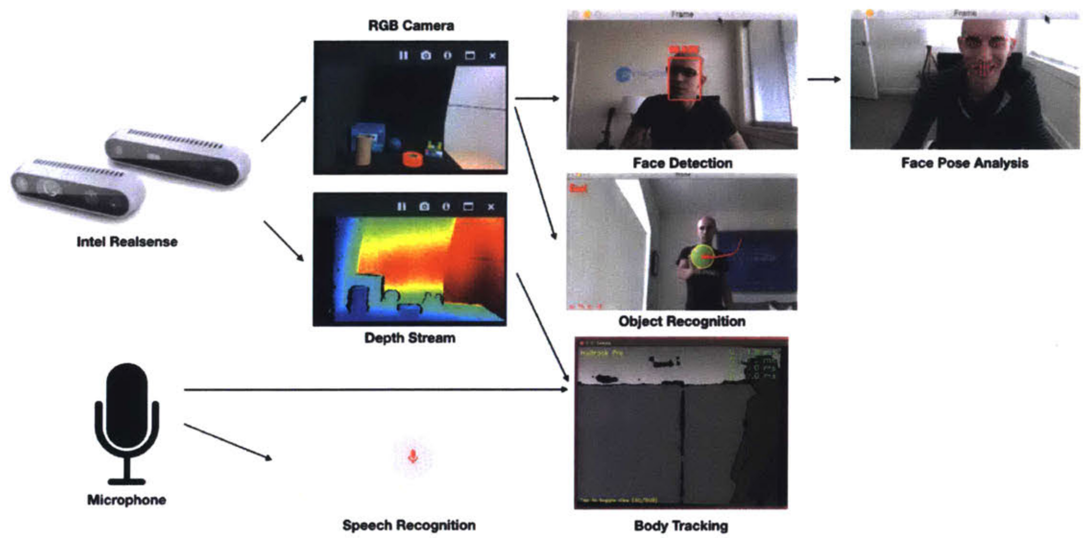


FIGURE 6.4: Diagram showing the input device pipeline.

Because the dog character couldn't look at multiple users at once, a LED strip was added along the circumference of the display to convey auxiliary information. For example, if the system detects a user, a subsection of the LED strip lights up that corresponds to the user's affective state and engagement. For the later developed user study, the LEDs indicated which activity mode the user had entered by changing to a specific color on the card. If the microphone had been integrated, when the system is listening to the user, using FFT, the width of the area might modulate to indicate that the system is engaged and interpreting. This color information is streamed from Unity using serial messaging to an Arduino connected to the LEDs and the computer via USB.

6.3 SOFTWARE

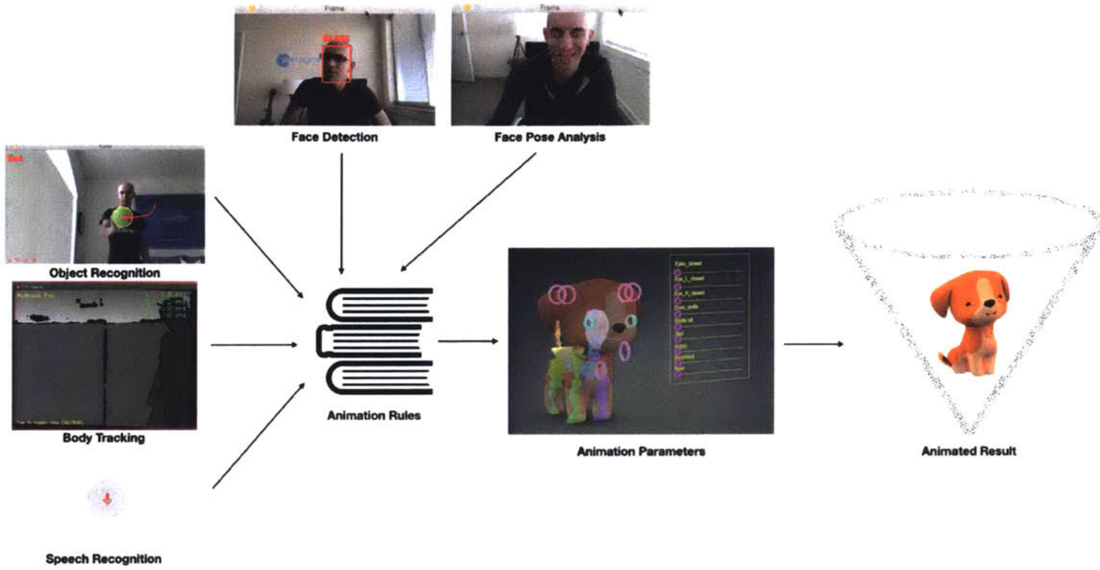


FIGURE 6.5: Diagram explaining how the processed input channels influence the character animation.

In game engines like Unity and Unreal, it is straight-forward to connect various sensor and stream data real-time in the system. To fully explore the real-time interaction potential of this device, the procedural character was rendered in the display volume, animating based on data from external sensors. To interact with the character, Intel Realsense depth/RGB cameras were connected via USB to the computer which then streams depth and color data to Unity. The depth information streamed allowed the system to understand where users are in the space as well as body pose. The color information streamed allowed for body pose detection, object/image detection, face detection and recognition, and face pose detection.

An omni-directional microphone could've been connected to the computer and the audio streamed into Unity where it would've been processed for volume and speech to text. Many third-party companies like IBM, Google and Amazon allow for cloud voice analysis which is how this system will interpret speech and assign responses. FFT (fast Fourier transform) could have been applied to audio streaming directly in Unity which can be used to detect pitch and volume.

The first software that was written in Unity to animate the character was a script which controlled animation based on the position of a virtual red ball relative to the dog. In this first experiment, if the ball moved closer to the dog, the dog's tail would wag faster, indicating excitement. Additionally, the dog's head pose would rotate to always face the ball which allowed for an enhanced sense of engagement from the character. After this initial experiment, the software was expanded to allow the dog to sympathetically respond to the user's emotion. Using Affectiva's Unity plug-in, software that uses machine learning to understand facial pose affect, the user's face pose could be parameterized to allow for nuanced emotional interaction. If the user was smiling, the dog would smile and wag its tail, but if the user frowned or looked angry, the dog would stop wagging its tail and look nervous or sad. These first interactions provided a framework for creating the user study.

6.4 INTERACTION

The coupling of depth/RGB data allowed for more nuanced understanding of a user's intent and affective state (which would have been further enhanced by the use of a microphone and natural language processing). In combination, these inputs were used to drive sympathetic animations from the character.

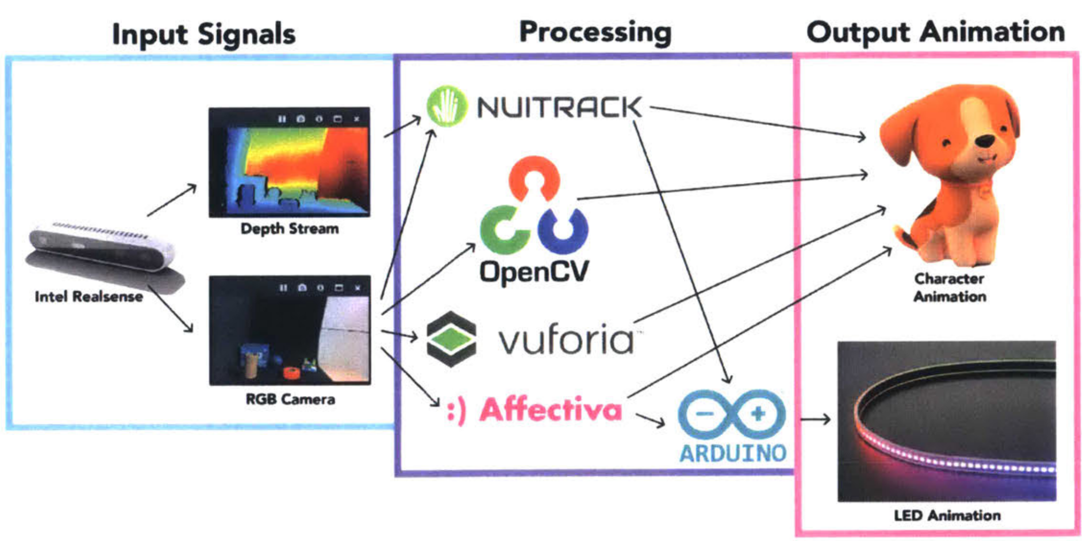


FIGURE 6.6: Diagram showing how the data streamed from the Intel Realsense camera is used by the processing libraries to generate appropriate animations for the character and LEDs.

To illustrate the capabilities of this real-time display, an experiment was designed that encouraged the user to interact physically and emotionally with the dog character. Custom designed cards were created in Adobe Illustrator to be used to trigger activity modes during the user study. The cards, seen below, were designed specifically to be playful and intuitive while also providing enough detail for the image to be recognized by the camera. Vuforia, which is an SDK primarily used for tracking augmented reality content, was repurposed in Unity to recognize these image glyphs. These registered images allowed for the Unity system to understand where special markers were placed in three-dimensional space. This allowed the dog character to look at the markers convincingly.

FIGURE 6.7: These "image marker" cards were designed in Adobe Illustrator and processed in Vuforia for image detection and tracking in Unity. These designs were printed and mounted on poster board for the user study.

The first activity created was called "play". This seemed like an obvious first interaction with the dog character as the user would have an intuitive understanding of how the dog might behave when presented with a ball. When the user holds up the "play" card, the digital dog picks up a digital version of the green ball on the card. If the user starts smiling while shaking the card, the dog will move faster as it loops through the play animation with the virtual ball. The LEDs on the periphery of the Funnel Vision housing turned green to match the bar at the top of the "play" card.

FIGURE 6.8: This is an example of an intervention that happens when the camera recognizes the "play" activity card. The dog character now has a 3D model of the 2D ball represented on the card.

The next activity was called "treat". Again, this seemed like a straight-forward and familiar association for the user to make. When presented with the "treat" card, the dog character would start panting and smiling while looking directly at the physical treats. If the user moved the treat card physically passed a virtual threshold, a virtual treat would be dispensed allowing the dog to celebrate. The LEDs on the periphery of the Funnel Vision housing turned blue to match the bar at the top of the "treat" card.

For the "sleep" card, the user would again present the card to the Funnel Vision system. Because this would signal to the character to go to sleep, the virtual lighting would dim and shade to purple, as a lullaby would play allowing the character drifted off to sleep. The LEDs on the periphery of the Funnel Vision housing turned purple to match the bar at the bottom of the "sleep" card and the changed reflected in the virtual lighting.

For the final "style" card, the user might not have had a clear mental model for how the character would behave. The goal of this interaction was to allow the user to change the color of the dog's collar based on the user's mood. As the user held the card up to the Funnel Vision system, the Intel Realsense camera would stream data through Affectiva which would then influence what color the collar should turn. The LEDs on the periphery of the Funnel Vision housing turned orange to match the bar at the bottom of the "style" card.

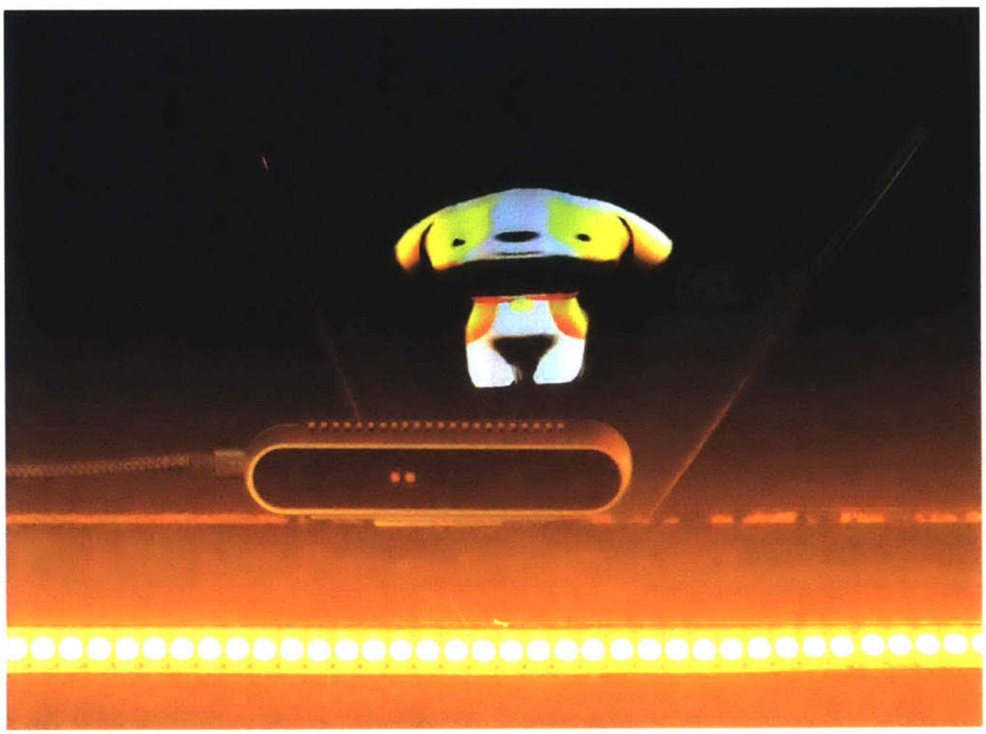


FIGURE 6.9: This image showcases the complete Funnel Vision system, with the procedural dog character, the Intel Realsense camera and the auxiliary LED strip.

For all four of these interaction modes, the LEDs directly in front of the user would change color based on the user's perceived affective state using the Affectiva integration. While the interactions were limited in their scope for the purpose of the user study, one could imagine how a character like this could be playful and informative at the same time. For the purpose of the user study, every interaction was being generated in real-time based on exclusively the current state of the user. Because the

RGB data allows for face recognition, the character could potentially store information about users to be retrieved whenever that user interacts with the system.

6.5 REACTION

For the user study, I wanted to understand how users interacted with the character and whether they felt the character had more presence when situated in the physical environment. This was the metric I decided to base the success of the system on, because ultimately, I feel the most ubiquitous use case of a device like this would be formatted as a digital character companion.

6.5.1 REASONING

I decided to compare the Funnel Vision system to 2D monitor because of the familiarity and affordability of traditional displays. Furthermore, because this study was designed to measure engagement with the character using the affective pose of the user, doing a comparative study using an augmented reality headset would've occluded the user's face. For the purposes of this study, the number of rendered views were minimized to two, because the interaction was focused primarily on the character, not on the 360-viewpoint effect. This allowed for a brighter, clearer image of the character to be reflected in the cone.

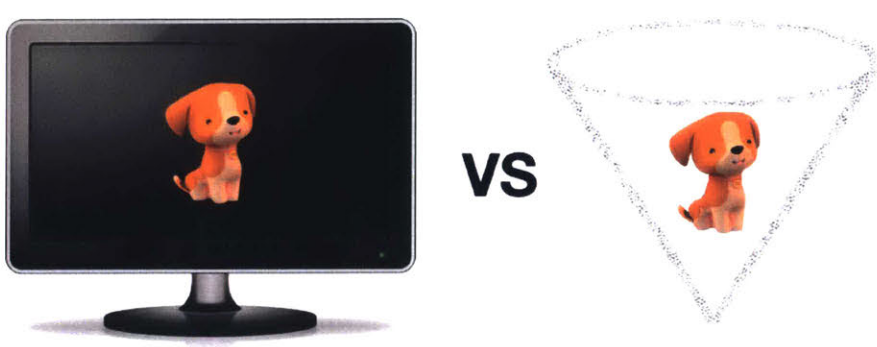


FIGURE 6.10: For the user study, it was important to compare the character interaction to a familiar, affordable display.

The user study was designed to measure three features of the display: engagement, fidelity, and enjoyment. These features were chosen because ultimately, this display is intended to be engaging and enjoyable to use, while providing a high fidelity experience. Because of the optical layering with the system, I already knew it wouldn't be as high fidelity compared to a traditional 2D monitor, so I also asked questions about perception of the character's attention.

6.5.2 HYPOTHESIS

Before running the user study, I wasn't entirely sure what the participants would think of the display. About half of the participants hadn't used augmented reality glasses before, so I predicted they might have a more enjoyable time using this display because they wouldn't have seen higher resolution augmented reality content. For the users who had used augmented reality glasses before, I predicted that they might enjoy this display because it doesn't require wearing a headset, which is an inconvenience or cumbersome for some users. As I designed the user study, I was hoping that the activities would be enjoyable for all users who participated in the study, so I predicted that enjoyment would be high for both the 2D monitor and the Funnel Vision system. As for engagement, I believed that for some of the activities, the Funnel Vision system would be more engaging overall, but I had concerns for activities that required higher resolution. For example, the dog's facial pose would change based on the user's face pose, but those changes might be more difficult to notice on the Funnel Vision display. Additionally, I didn't tell the participants that the camera was tracking their facial pose (although many users could infer that because the LED lights would change to indicate engagement and perceived emotion). Because of that, I predicted that engagement might be higher on both systems if the participant started with the 2D monitor, because they could experience the character in full-resolution in a familiar format before transitioning to the Funnel Vision system.

6.5.3 PROCEDURE

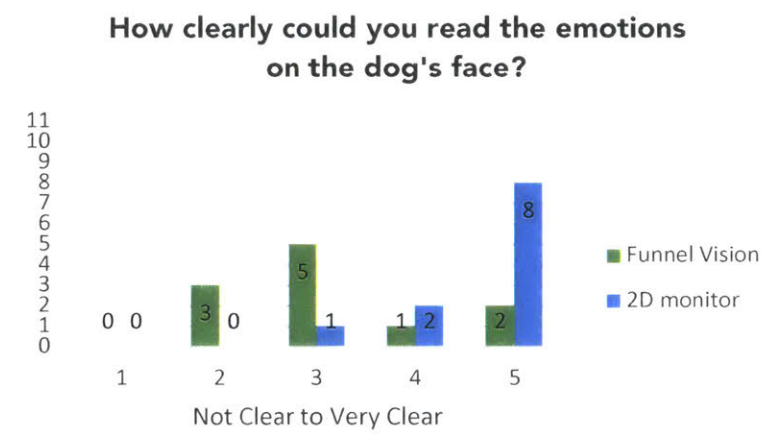
The full list of questions for the user study can be viewed in the appendix. For the procedure of the user study, an email was sent out asking for voluntary, uncompensated people to participate in the study. The form asked two questions, whether they could actually commit to the study and if they had used augmented reality of any kind before. In total, there were 11 participants for this study, with most participants in their 20s and several older adult participants. The gender balance was split with six female participants and five male participants. When the participant arrived for the study, the completed a pre-survey that asked the user how they were feeling today and when they thought augmented reality would be a mature technology. Next the user was asked to either free-style engage with the Funnel Vision system or the 2D monitor, after a brief explanation of how to use the system. Since I was in the room, watching the participants as they interacted with each system, I could answer their questions if they felt lost while interacting with the character. During the demo, I pointed out where the edges of the camera's viewable area was. Due to the limitations of the Vuforia integration, before the participants interacted with the system, I demonstrated that in order to achieve the best tracking result, the participant should hold the image glyph parallel to the camera.

After the participant felt like they understood each activity (which they could try and re-attempt in any order), the participant would complete a survey that asked questions about their interaction with the display they just used. The questions pertained to engagement, fidelity and enjoyment. After completing the survey, the participant would switch to the next display and again, interact with the dog using their facial pose and the provided activity cards. I again demoed the procedure and the approximate view range of the camera to avoid confusion with the sensor to the participant could focus on their interaction with the character. After finishing the activities, the participant completed an identical survey about the second display they interacted with. Upon completion of that survey, the participant completed a final survey that comparatively addressed each display, with a second page of follow-up questions on what direction they believe this technology could take. The follow up question page specifically asked about multiple-user interaction. Because of the affordances of the Funnel Vision system, a potential next user study could include multiple users at one time interacting with the system.

6.5.4 ANALYSIS

The user study provided interesting results for interpretation on the utility of this system and for the direction of future work. Overall, the participants seemed to enjoy interacting with the Funnel Vision system, despite the 2D monitor imagery rendering more clearly. Fidelity was higher for the 2D monitor, but engagement and enjoyment was reported higher for Funnel Vision.

When asked how clearly the user could read the emotions and the character animations better on Funnel Vision and the 2D monitor, all participants either ranked the experience as the same or as the 2D monitor being more readable. This absolutely makes sense, for two key reasons. First, the 2D monitor is opaque, meaning the features of the dog character will contrast much more on the 2D monitor compared to the ghosting effect that happens on the conical reflection. Second, the 2D monitor (which was also 4K resolution), was not rendering a stereoscopic image, which means that there were more pixels dedicated to the dog character displayed on the 2D monitor compared to the Funnel Vision display.



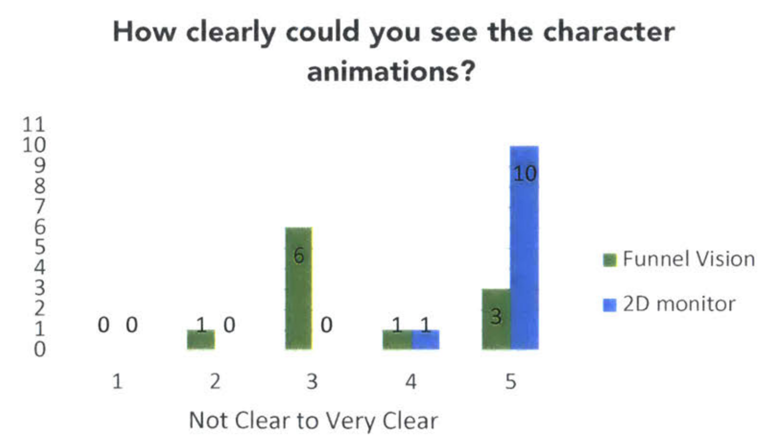
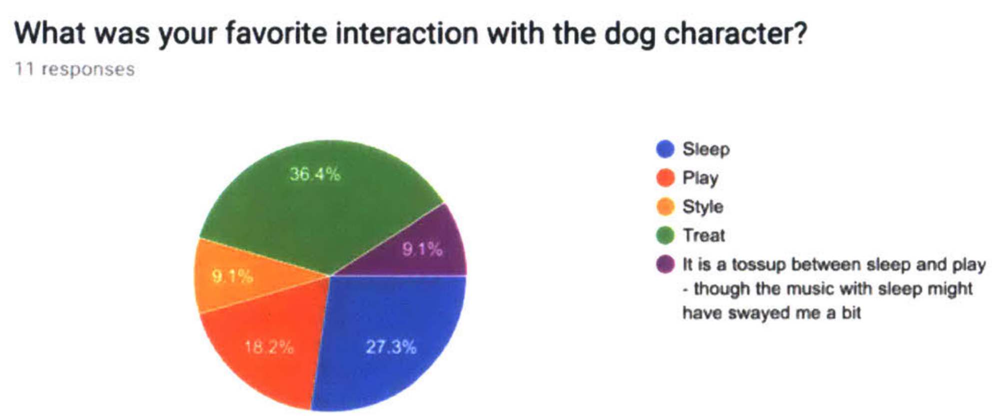


FIGURE 6.11: It was easier to read the emotional pose on the dog character's face on the 2D monitor compared to Funnel Vision. However, overall, the emotional poses were still visible on both displays. It was easier to see the character animation on the 2D monitor compared to Funnel Vision.

Next, I wanted to analyze indirectly how the participants might've enjoyed their experience with the two displays. The "style" and "treat" activities involved more facial animation than the "sleep" and "play" activities. When looking at the activity preferences between Funnel Vision and the 2D monitor, the results were surprising, as I expected the users to prefer "sleep" and "play" on Funnel Vision (as they required less face animations on the character). Instead, "treat" followed by "sleep" were the preferred activities on Funnel Vision. On the 2D monitor, the participants ranked "play" as the most enjoyable activity. The results are illustrated below in figure 6.13. "Treat" did not rank as a favorite activity at all on the 2D monitor. "Play" and "sleep" ranked highly overall as an aggregated total between the two results. "Style" was ranked higher on the 2D monitor compared to the Funnel Vision display. Because the "style" activity involved face pose analysis and the collar color modification was dimmer overall on the Funnel Vision display, perhaps it was more difficult to appreciate on the Funnel Vision display.

FUNNEL VISION



2D MONITOR

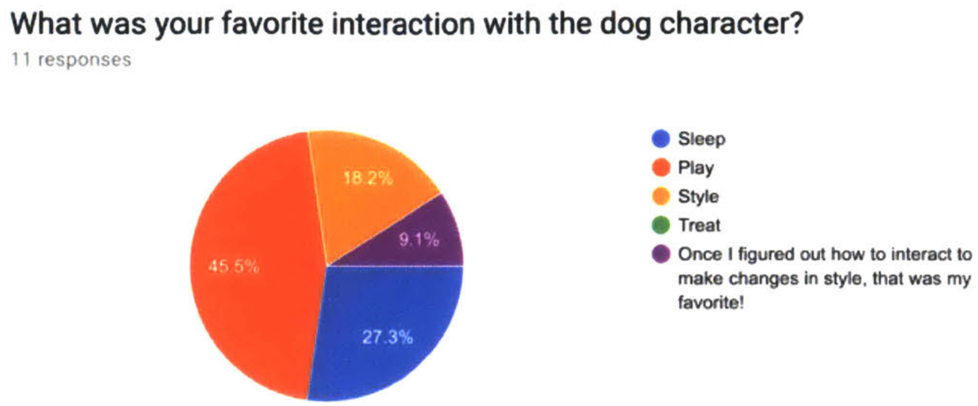
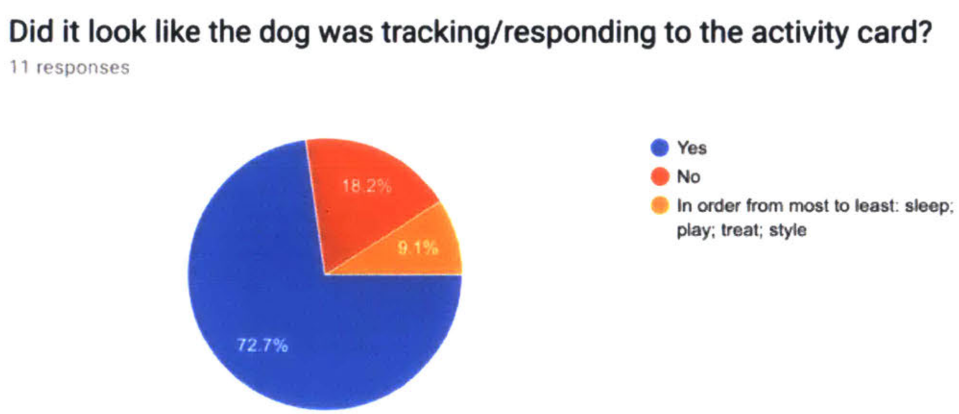


FIGURE 6.13: Based on the above charts, there was more variety in which interactions the user's enjoyed on Funnel Vision display. Almost half of the participants enjoyed "play" the most on the 2D monitor.

When asked if the participants noticed that the dog was able to track the cards in 3D space and respond to the activities, most of the participant said yes. Every participant that said they noticed the dog character tracking/responding to the activity card on the 2D monitor also noticed that behavior on Funnel Vision, regardless of the order introduced. Because this was an intuitive signal, embedded in the character animation, it was encouraging to learn that most participants noticed the behavior. If I were to do another user study, I might've toggled certain behaviors for different participants to understand if this was perception based or a clear understanding of the behavior.

FUNNEL VISION



2D MONITOR

Did it look like the dog was tracking/responding to the activity card?

11 responses

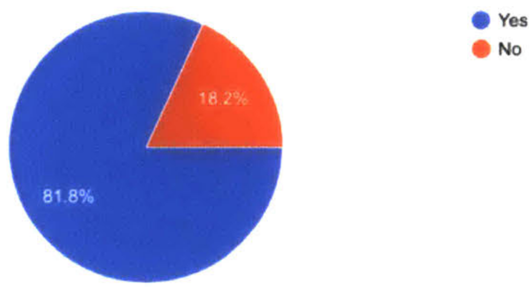


FIGURE 6.14: All the participants that said they noticed the dog tracking/responding to the activity card on the 2D monitor, noticed on Funnel Vision as well, regardless of the order introduced.

When asked on which display it was easier to read the character's facial features, most participants selected the 2D monitor. Because of the increased pixel canvas on the 2D monitor, its contrast and its familiar visual format, it was not surprising that the 2D monitor out-performed Funnel Vision on this metric. I had concerns that because the users found it easier to read the character's facial features on the 2D display, that Funnel Vision would be unsuccessful overall as an interactive experience.

On which display was it easier to read the character's facial features?

11 responses

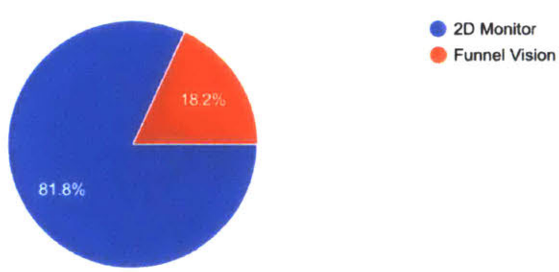


FIGURE 6.15: The character's facial features and animations were more clearly visible on the 2D monitor compared to Funnel Vision.

However, the participants ranked Funnel Vision as having more character presence, despite the facial feature's fidelity being higher on the 2D monitor. The participants might have ranked character presence higher on the Funnel Vision display for a couple key factors. First, because the Funnel Vision system situates the character in the physical environment, the participants of the study may have felt that the character had more presence. Second, because of the conical reflection, there may have been some angular distortion that contributed to this sentiment. Some participants spent time moving their head back and forth laterally as they looked at the character and this may have contributed to a sense of depth or presence for the character that doesn't exist with a 2D monitor.

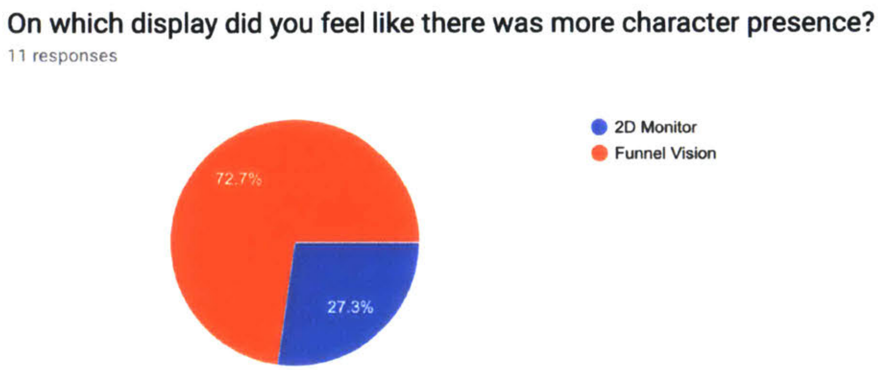


FIGURE 6.16: Users reported more character presence on the Funnel Vision display compared to the 2D monitor.

When asked which display the participants enjoyed interacting with more, the same participants that reported more character presence on the Funnel Vision display compared to the 2D monitor enjoyed interacting with the character more on the Funnel Vision display. If the participant was able to connect or establish presence with the character on the Funnel Vision display, the experience was more enjoyable. Overall, the participants seemed to greatly enjoy interacting with both displays, with all participants reporting ending the study happier or the same level of happy than when they arrived.

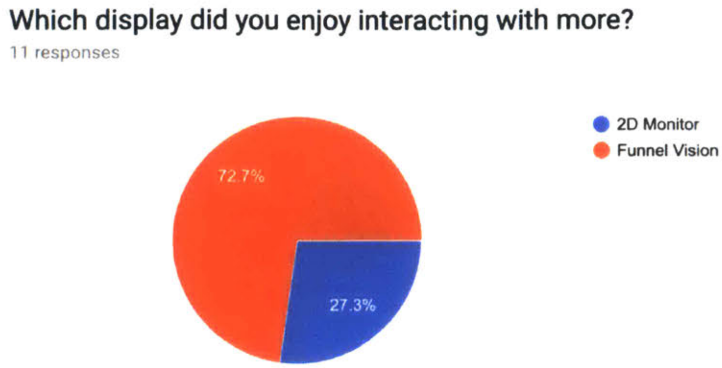


FIGURE 6.17: The same users that reported more character presence on the Funnel Vision display compared to the 2D monitor enjoyed interacting with the character more on the Funnel Vision display.

I thought it was important to ask, after completing both activities, if the participants noticed that their emotional face pose was being tracked and interpreted by the system. I never explicitly told the subjects that their faces were involved with the character interaction. Most participants recognized that the character had access to the user's face pose, despite not being told this information prior to interacting with the system. The participants that recognized that their face pose influenced the animation reported more enjoyment with both the "play" and "style" activities, where the face pose of the participant heavily influenced the interaction. For the "play" activity, if the user smiled while moving the "play" card around, the dog character would more vigorously play with the virtual ball that appeared in the scene. If the user didn't understand that their face was being tracked, the dog character wouldn't exhibit excited sentiments when presented with the "play" card. For the "style" activity, understanding that the facial pose was being interpreted was critical for the interaction. One participant rated "style" as not-enjoyable on either display. This might have been because this participant didn't notice that the system was using their face pose as input.

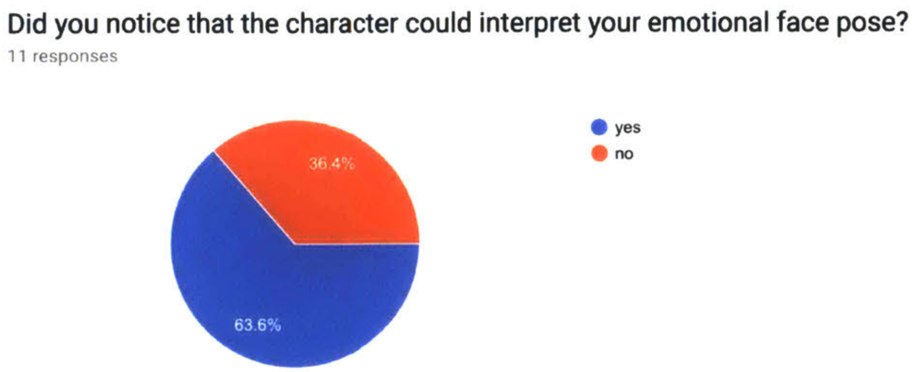


FIGURE 6.18: Most participants recognized that the character had access to the user's face pose, despite not being told this information prior to interacting with the system.

It is unclear if the users who didn't notice the face pose, didn't notice because the reciprocating animations the character did were too subtle or because they weren't conditioned to look for that response during the study. For example, on the iPhone, users can create Animoji sequences where their face pose is used to directly animate a character mesh. The messaging around that interaction was made abundantly clear through marketing, text instructions, and visual signals while using the interface. However, for the dog character animations, the animations weren't directly mapped to the user's face pose, but instead were reciprocating the perceived emotion. Additionally, because the dog's emotional animations were interpolated with the more dominant activity animations, it might have been difficult to recognize this design feature under the circumstances of a limited interaction.

What I do think is assuring, based on this response, is that most participants physically distorted their faces during the course of the evaluation to trigger an understanding of this design feature. Because most participant changed their facial pose while interacting with the character, they were able to notice that it has influence on the system. This has interesting implications because most existing character engagements don't require the user's active emotional participation. However, because the participants left the study feeling more positive, I imagine emotional engagement had a positive influence on the overall experience.

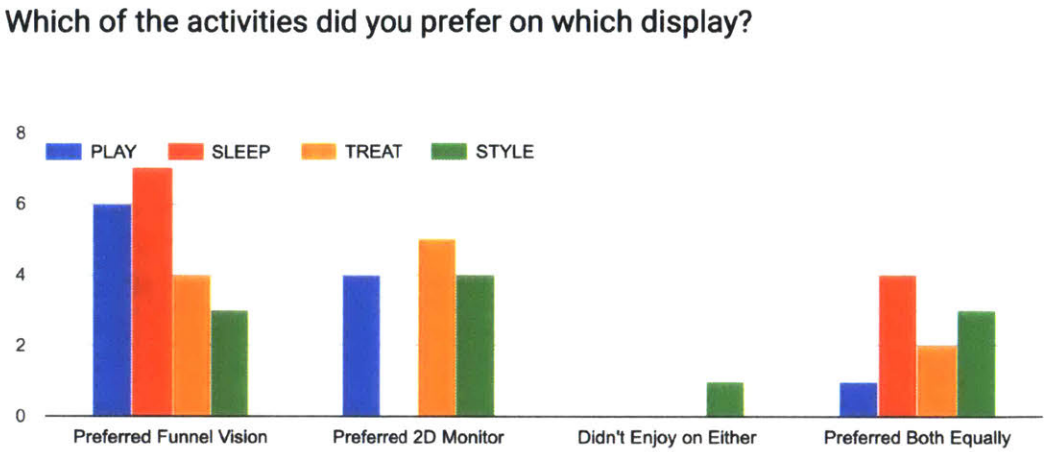


FIGURE 6.19: Overall the participants enjoyed the 'sleep' and 'play' activities more on Funnel Vision, and the 'style' and 'treat' activities more on the 2D monitor.

After completing the multiple-choice forms as part of the study, the participants were offered an opportunity to give informal feedback through a serious of open-ended questions. This was completely voluntary and the questions were framed to provide feedback for future directions this project could take rather than evaluations of the current iteration. When asked how Funnel Vision compared to augmented reality glasses, the participants who had used some form of augmented reality before, overall reported Funnel Vision as a more intuitive, collaborative system. One participant said "I thought that not having to physically augment my own head/body was a great improvement on AR glasses and similar systems from an accessibility and comfort standpoint. I felt like the face tracking performance was similar in quality to what I've experienced with existing systems, but that the added physicality of Funnel Vision made the dog character and entire experience more emotionally compelling." Another said that the Funnel Vision system has different goals and applications than traditional augmented reality, but that the Funnel Vision system was more comfortable. The main feedback to the question focused on comfort and character permanence which lead to a heightened sense of physicality and engagement.

The participants were then asked what new features they would like to see developed for the platform. Many of the participants recommended speech recognition, which seems like a natural direction for this system to take, as its directly inspired by voice assistants like Google Home and Amazon Alexa. Some participants were curious how this system would work with more than one participant at a time. Additionally, another participant inquired about a mobile version of this display, that they would be able to pick up and carry around with them.

As a pointed follow-up, the questionnaire then asked the users how they imagined this system being used in a collaborative context. I'm glad I included that question, as many of the participants were curious about the multi-user opportunity this setup affords. Most of the participants suggested telepresence as a multi-user scenario, while others suggested specifically designed game play as a new direction to take. Two of the participants suggested that the digital character could act as an impartial mediator or therapist in tense group dynamics.

When asked what features the participants would want in a consumer product version of this system, all of the feedback indicated that consumers would want a character assistant that understands their emotional state. What stood out to me is how expansive and helpful a digital character assistant that understands affect could be. For example, one participant remarked "I would love for a character assistant to, for example, hold notifications or notify me in a less disruptive way if I'm clearly stressed." In that example, the emotional state of the user drives an intuitive, subverted response from the character. While character companions have the potential to infringe on privacy and to exploit consumers emotional states, there is also a compelling advantage to a system that can better address user's needs.

Most participants want transparency in how they allow their data to be used, even in innocuous contexts. All participants recommended that emotional understanding be an opt-in feature for a display like Funnel Vision, even if it means the character loses some capabilities. The participants overall preferred visual signals like the LED strip indicators to perpetually acknowledge them that data was being collected and used to enhance their experience. A subset of the participants would prefer a subtle experience with the character, where they could opt-in to sharing their data, but the system would use that data in non-intuitive ways to improve the character interaction.

Part 4
Conclusions

CHAPTER 7
CONCLUSIONS

7.1 CONCLUSION

Funnel Vision and similarly made Pepper's Ghost displays offer a unique opportunity to provide a robust augmented reality experience at a fraction of the current price. Because of the collaborative nature of this display and its affordable price point, I hope that this project could be further developed to be used in classrooms. In order for that to become a reality, there is substantial amount of work required to finesse the radial lenticular and investigate best practices for interactions with this type of display. I would love to share the software for this project as a Unity package for anyone to develop on with a parallax barrier and cone while the logistics of the radial lenticular are further developed. I really enjoyed working on this project as it gave me an unforgettable and life-changing opportunity to work at the intersection of hardware, software and interaction design. Furthermore, I had the freedom to investigate procedural character animation in Unity through the context of human interaction and artificial intelligence. I am so excited to see what direction the Object Based Media group takes with this platform for low-cost augmented reality. Overall, I was surprised by how engaged participants were during the user study. There was genuine laughter and smiles in the room as the participants interacted with the character in the Funnel Vision display. Because of the structure of the MIT Media Lab, with plenty of visitors from various industries passing through the lab, I am comforted to know that there is a ton of imaginative directions this project could go, ranging from companionship to connected experience design to training simulations.

7.2 FUTURE WORK

Funnel Vision has two main branches of expansion for future researchers in the Object Based Media Group. First, the radial optics and conical reflections could be more robustly analyzed and developed. Unfortunately, the radial lenticular that was printed with an optical resin for this display was bonded to a thick substrate, so was unusable for this context. Because of the thickness of the substrate, views meant for adjacent lenticular lenses would bleed over as the viewer moves off-axis. This leads to flipping views (which is the same limitation the parallax barrier presents). Ideally, for a system like this to function correctly, a custom monitor with a radial pixel array would be developed to properly align the lenticular to the pixels. The radial lenticular itself would be created through a high resolution, subtractive manufacturing process. For an additive manufacturing process, perhaps the lenticular could be printed as a symmetrical subsection and then molded and casted to produce the final optical piece. Optical resin printers continually increase in resolution. The lenticular would then be bonded to the monitor for precise imaging of the views generated. In order to accomplish this task, researchers with a specific optics focus could model the system and produce the desired lenticular design. Additionally, material science expertise to understand how to reduce back reflections on the conical mirror would further enhance the viewing experience. Second, Funnel Vision could be a platform for collaborative digital character development. Researching character engagement involves natural language processing, interaction design, and signal processing to create a compelling system. Instead of using physical characters like social robots, perhaps Funnel Vision provides an affordable platform for developing digital telepresence and engaging characters. There is an initiative within the Object Based Media group to reimagine natural language processing pipelines to better serve personality driven agents, and that work could be showcased through a digitally embodied character on Funnel Vision. Additionally, as voice assistants (like Amazon Alexa, Siri, and Google Home) and embodied assistants (like Jibo) become more prevalent, researching interaction paradigms and sensor inputs for digital incarnations seems like a worthwhile research pursuit.

7.3 VISION STATEMENT

In our complex and connected world, it seems imperative to consider the how the technology developed forwards a vision for a more empathetic society. It is my hope that underutilized physical spaces can serve better purposes as community hubs, entertainment outposts, or extended education centers. Communities are stronger when they're educated and supported. Through the development of special, collaborative infrastructure and devices to be used in shared contexts, I hope that I can contribute to a future where we spend a little less time looking down and a little more time looking ahead. Safety goggles off.

REFERENCES

[1] "Anamorphism - Dictionary Definition." Vocabulary.com, www.vocabulary.com/dictionary/anamorphism.

[2] Anko3D. Cartoon Dog 2 3D Model. TurboSquid, 23 Apr. 2018, www.turbosquid.com/3d-models/cute-cartoon-dog-2-3d-model-1279763.

[3] "AT&T Brings Magic Leap One to Flagship Stores Next Week." AT&T News, AT&T, 27 Mar. 2019, about.att.com/story/2019/magic_leap.html.

[4] Barnum, Peter C., Srinivasa G. Narasimhan, and Takeo Kanade. 2010. A Multi-layered Display with Water Drops. ACM Transactions on Graphics 29, 4 (2010), 76:1–76:7.

[5] Benton, Stephen A. Optics in Entertainment, 1983, doi:10.1117/12.935062.

[6] Berthier, Auguste. (May 16 and 23, 1896). "Images stroscopiques de grand format" (in French). Cosmos 34 (590, 591): 205-210, 227-233.

[7] Bickmore, T. W., & Picard, R. W. (2005). Establishing and maintaining long-term humancomputer relationships. ACM Transactions on Computer-Human Interaction: A Publication of the Association for Computing Machinery, 12(2), 293-327.

[8] Boltanskaya, A. G. and N. A. Ovshahikova, "Stereoscopic cinema in soviet pavilion in EXPO-70," Techniques of Cinema and TV (Russian Journal), no. 10, 22–27, 1970.

[9] Bove, V. Michael, et al. "How to Build a Holographic Television System." 2013 3DTV Vision Beyond Depth (3DTV-CON), 2013, doi:10.1109/3dtv.2013.6676637.

[10] Breazeal, Cynthia. "Jibo Robot." Jibo, 2010, www.jibo.com/.

[11] Breazeal, Cynthia. "Leonardo." Personal Robots Group, MIT Media Lab, 2004, robotic.media.mit.edu/portfolio/leonardo/.

[12] Burdekin, Russell. "Pepper's Ghost At The Opera." Theatre Notebook 69.3 (2015): 152-164. International Bibliography of Theatre & Dance with Full Text.

[13] Chan, Casey. "This Is How the Tupac Hologram That Wasn't Really a Hologram Worked." Gizmodo, Gizmodo, 17 June 2013, gizmodo.com/this-is-how-the-tupac-hologram-that-wasnt-really-a-holo-5903092.

[14] Della Porta, Giambattista. "Magica Naturalis." Aubrii & Schleichius (1619) Web.

[15] "Disney Play App." Disneyworld.disney.go.com, Disney Parks and Resorts, 2018, disneyworld.disney.go.com/guest-services/play-app/.

[16] Dodgson, Neil A. 2005. Autostereoscopic 3D Displays. Computer 38, 8 (2005), 31–36.

[17] Farivar, Cyrus. "Tupac 'Hologram' Merely Pretty Cool Optical Illusion." Ars Technica, 17 Apr. 2012, arstechnica.com/science/2012/04/tupac-hologram-merely-pretty-cool-optical-illusion/.

[18] Fitzpatrick, Richard. "Law of Reflection." Law of Reflection, 14 July 2007, farside.ph.utexas.edu/teaching/302l/lectures/node127.html. Web.

[19] Foley, James D., and Andries Van Dam. Fundamentals of Interactive Computer Graphics. Addison-Wesley, 1981. 18-24

[20] Frantz, Michael. "A Perspective on Infinity: Anamorphism and Stereographic Projection." University of La Verne, 2005, 1-8.

[21] Geng, Jason. "Three-dimensional display technologies," Adv. Opt. Photon. 5, 456-535 (2013).

[22] "Hatching ILMxLAB's Star Wars: Project Porg." StarWars.com, 20 Feb. 2019, www.starwars.com/news/ilmxlab-star-wars-project-porg.

[23] "HoloMAX Holographic Experience." H Technology, hplustech.com/.

[24] Huebschman, Michael, Bala Munjuluri, and Harold Garner. 2003. Dynamic holographic 3-D image projection. Optics Express 11, 5 (2003), 437-439.

[25] "Integral Photography." Scientific American 105.8 (1911): 164. Print.

[26] Instructables. "Cylindrical Mirror Art." Instructables, Instructables, 22 Mar. 2018, www.instructables.com/id/Cylindrical-Mirror-Art/.

[27] Instructables, and Open Green Energy. "DIY 3D Hologram Pyramid." Instructables, Instructables, 1 Oct. 2017, www.instructables.com/id/DIY-Pyramid-Hologram/.

[28] Isono, Haruo, Minoru Yasuda, and Hideaki Sasazawa. 1993. Autostereoscopic 3-D display using LCD-generated parallax barrier. Electronics and Communications in Japan (Part II: Electronics) 76, 7 (1993), 77-84.

[29] Jones, Andrew, Ian McDowall, Hideshi Yamada, Mark Bolas, and Paul Debevec. 2007. An Interactive 360&Deg; Light Field Display. In ACM SIGGRAPH 2007 Emerging Technologies.

[30] Jones, Andrew, Jonas Unger, Koki Nagano, Jay Busch, Xueming Yu, Hsuan-Yueh Peng, Oleg Alexander, and Paul Debevec. 2014. Creating Life-sized Automultiscopic Morgan Spurlock for CNNs "Inside Man".

[31] Kajiki, Yoshihiro, Hiroshi Yoshikawa, and Toshio Honda. 1996. 3D display with focused light array. Proceedings of SPIE 2652 (1996), 106-117.

[32] Kathaperumal, M, and others. 2010. Holographic three-dimensional telepresence using large-area photorefractive polymer. Nature 468, 7320 (2010).

[33] Kory, Jacqueline, and Cynthia Breazeal. "Storytelling with Robots: Learning Companions for Preschool Childrens Language Development." The 23rd IEEE International Symposium on Robot and Human Interactive Communication, Aug. 2014, doi:10.1109/roman.2014.6926325.

[34] Kramer, Scott. "The Magic of Universal Orlando: Revealing the Technology Behind the Rides." Forbes, Forbes Magazine, 8 Mar. 2016, www.forbes.com/sites/scottkramer/2016/03/08/the-magic-of-universal-orlando-revealing-the-technology-behind-the-rides/#6a5fef5abd0b.

[35] Lanman, Douglas, Gordon Wetzstein, Matthew Hirsch, Wolfgang Heidrich, and Ramesh Raskar. 2011. Polarization Fields: Dynamic Light Field Display Using Multi-layer LCDs. ACM Transactions on Graphics 30, 6 (2011), 186:1-186:10.

[36] Lee, Dami. "Nintendo Labo VR Hands-on: Virtual Reality Goes DIY." The Verge, The Verge, 22 Mar. 2019, www.theverge.com/2019/3/21/18274405/nintendo-labo-vr-kit-switch-hands-on.

[37] Lefevre, Wolfgang. Inside the Camera Obscura: Optics and Art under the Spell of the Projected Image. Max-Planck Institute for the History of Science, 2007. 8-15.

[38] Lippman, Andrew.1980. Movie-maps: An application of the optical videodisc to computer graphics. In Acm Siggraph Computer Graphics, Vol. 14. ACM, 32-42.

[39] Maeno, Keiichi, Naoki Fukaya, Osamu Nishikawa, Koki Sato, and Toshio Honda. 1996. Electro-holographic display using 15mega pixels LCD. Proceedings of SPIE 2652 (1996), 15-23.

[40] Matusik, Wojciech and Hanspeter PÔ̈Åster. 2004. 3D TV: A Scalable System for Real-time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes. ACM Transactions on Graphics 23, 3 (2004), 814-824.

[41] Mine, Mark, et al. "Projection-Based Augmented Reality in Disney Theme Parks." 2012, web.cs.wpi.edu/~gogo/courses/cs525A /papers/Mine_2012_ProjectionAR.pdf.

[42] Morrison., Maureen. "Pokmon Go Goes Viral With No Big Marketing Blitz." Ad Age, 11 July 2016, adage.com/article/digital/p go-viral-marketing/304905/.

[43] Mothersill, Philippa, and V. Michael Bove. "The Form of Emotive

Design." MIT Media Lab, 28 Sept. 2015, doi:10.1145/2830315.

[44] Nayar, S. K. and V. N. Anand. 2007. 3D Display Using Passive Optical Scatterers. Computer 40, 7 (2007), 54–63.

[45] Panzarino, Matthew. "Disney Has Begun Populating Its Parks with Autonomous, Personality-Driven Robots – TechCrunch." TechCrunch, 8 Feb. 2018, techcrunch.com/2018/02/08/disney-has-begun-populating-its-parks-with-autonomous-personality-driven-robots/.

[46] Paschotta, Dr. Rdiger. "Fresnel Equations." Encyclopedia of Laser Physics and Technology - Fresnel Equa- tions, 23 Oct. 2017, https://www.rp-photonics.com/fresnel equations.html.

[47] Perlin, Ken. "Playable 3D Procedural Characters." Holojam, NYU Future Reality Lab, holojam.com/.

[48] Peterka, Tom, et al. "Advances in the Dynallax Solid-State Dynamic Parallax Barrier Autostereoscopic Visualization Display System." IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 3, 2008, pp. 487-499., doi:10.1109/TVCG.2007.70627. IEEE Xplore Digital Library.

[49] Raskar, Ramesh. "Towards a Common Framework for Parallax Barrier and Holographic 3D Display." MIT Media Lab, pp. 1-9.

[50] Rawson, Eric G. 1969. Vibrating varifocal mirrors for 3-D imaging. IEEE Spectrum 6, 9 (1969), 37–43.

[51] Sabourin, J., Mott, B., & Lester, J. (2011). Computational models of affect and empathy for pedagogical virtual agents. In Standards in emotion modeling, Lorentz Center International Center for workshops in the Sciences.

[52] Sawalha, L., M. P. Tull, M. B. Gately, J. J. Sluss, M. Yeary, and R. D. Barnes. 2012. A Large 3D Swept-Volume Video Display. Journal of Display Technology 8, 5 (2012), 256–268.

[53] Shaw, Marie. "Pepper's Ghost." Ideate, 20 Oct. 2015, ideate.xsead.cmu.edu/gallery/projects/static.

[54] "Snell's Law." Encyclopdia Britannica, Encyclopdia Britannica, Inc., 18 Apr. 2016, www.britannica.com/science/Snells-law.

[55] "Star Wars™: Jedi Challenges." Lenovo US, 2018, www.lenovo.com/us/en/jedichallenges.

[56] Strickland, Jonathan. "How the Nintendo 3DS Works." HowStuffWorks, 26 Apr. 2011, electronics.howstuffworks.com/nintendo-3ds5.htm.

[57] Suga, Chihiro and Itiro Siio. 2011. Anamorphicons: An extended display with a cylindrical mirror. In Proceedings of the ACM International Conference on

Interactive Tabletops and Surfaces. ACM.

[58] "The Design Tool for Projected AR." Lightform, 6 Mar. 2019, lightform.com/.

[59] "A Universal Holographic Display for 3D Creators." Looking Glass Factory, 2017, lookingglassfactory.com/.

[60] "The VOID." The VOID, 2016, www.thevoid.com/.

[61] "Voxon." Voxon Photonics, voxon.co/

[62] Wetzstein, Gordon, D. Lanman, W. Heidrich, and R. Raskar. 2011. Layered 3D: Tomographic Image Synthesis for Attenuation-based Light Field and High Dynamic Range Displays. ACM Trans. Graph. 30, 4 (2011).

[63] Wetzstein, Gordon, Douglas Lanman, Matthew Hirsch, and Ramesh Raskar. 2012. Tensor Displays: Compressive Light Field Synthesis Using Multilayer Displays with Directional Backlighting. ACM Transactions on Graphics 31, 4 (2012), 80, 1-11.

[64] Wong, Kevin. "The Science Behind the World's Most Convincing Ghost Effect." Vice, VICE, 28 Oct. 2016, www.vice.com/en_us/article/ aekye5/the-science-behind-the-worlds-most-convincing-ghost-effect.

[65] Xia, Xinxing, Xu Liu, Haifeng Li, Zhenrong Zheng, Han Wang, Yifan Peng, and Weidong Shen. 2013. A 360-degree floating 3D display based on light field regeneration. Optics Express 21, 9 (2013).

[66] Xie, HongBin, Xing Zhao, Yong Yang, Jing Bu, ZhiLiang Fang, and XiaoCong Yuan. "Cross-lenticular Lens Array for Full Parallax 3-D Display with Crosstalk Reduction." Science China Technological Sciences 55.3 (2012): 735-42. SpringerLink. Science China Technological Sciences, 28 Dec. 2011. Web. 11 Dec. 2016. http://link.springer.com/article/10.1007/s11431-011-4713-5.

[67] Xuan, Luo, et al. "Pepper's Cone: An Inexpensive Do-It-Yourself 3D Display." UIST, 2017, roxanneluo.github.io/papers/PeppersCone/ Pepper_s_Cone_UIST_2017.pdf.

[68] Zorich, Zach. "Early Humans Made Animated Art - Issue 11: Light." Nautilus, NautilusThink Inc, 27 Mar. 2014, nautil.us/issue/11/light/ early-humans-made-animated-art.

APPENDIX

8.1 RELEVANT CODE

ViewGenerator.cs

```
using UnityEngine;

[RequireComponent(typeof(Camera))]
public class ViewGenerator : MonoBehaviour {

    [Tooltip("Number of columns to tile in RenderTexture")]
    public int ViewDimX;

    [Tooltip("Number of rows to tile in RenderTexture")]
    public int ViewDimY;

  [Tooltip("Number of views per each lenticular lens")]
    public int viewsPerLenticular;

  [Tooltip("Number of repeated groupings of views")]
    public int numRepeats;

  [Tooltip("360 or the max angle divided by the number of cameras")]
  public float AngleOffset;

    [Tooltip("The width, in pixels, of the display and render texutre")]
    public int DisplayWidth;

    [Tooltip("The height, in pixels, of the display and render texture")]
    public int DisplayHeight;

    [Tooltip("The render texture combining the entire viewset")]
    public RenderTexture TiledViewsRenderTexture;

    [Tooltip("The render textures to use for each camera target")]
    public RenderTexture[] PerViewRenderTextures;

    // The view camera game object
    public GameObject[] views;

  // The distance of the cameras from the center
  public float radius;

    // Use this for initialization
    void Start () {

        GetComponent<Camera>().clearFlags = CameraClearFlags.SolidColor;
        GetComponent<Camera>().backgroundColor = Color.black;

        if(!TiledViewsRenderTexture)
            CreateRenderTextures();
        GenerateLightfieldCameras();

        // set the target texture to null after generating the views
  }

```

```
// Update is called once per frame
void Update (){

}

    void CreateRenderTextures()
    {
        TiledViewsRenderTexture = new RenderTexture(DisplayWidth, DisplayHeight, 24, RenderTextureFormat.ARGB32);
         TiledViewsRenderTexture.Create();

        PerViewRenderTextures = new RenderTexture[ViewDimX * ViewDimY];
        for (int i = 0; i < ViewDimX * ViewDimY; i++)
        {
            PerViewRenderTextures[i] = new RenderTexture(DisplayWidth / ViewDimX, DisplayHeight / ViewDimY, 24, RenderTextureFormat.ARGB32);
            PerViewRenderTextures[i].Create();
        }
    }

    void GenerateLightfieldCameras()
    {
        // Get the total number of views and create camera array
        int numViews = ViewDimX * ViewDimY;
        views = new GameObject[numViews];
        Vector3 center = new Vector3 (0f, 0f, 0f);

        for (int y = 0, idx = 0; y < ViewDimY; y++)
        {
            for (int x = 0; x < ViewDimX; x++, idx++)
            {
                // idx is the i-th view
                views[idx] = new GameObject("LightField View " + idx);

                // set the lightfield cameras to be children of this gameobject in the scene graph
                views[idx].transform.parent = transform;

                // attach a camera to the gameobject
                Camera theCam = views[idx].AddComponent<Camera>();

                float angle = idx * AngleOffset;
                            Debug.Log(angle);
                Vector3 pos = GenerateRadialPosition(center, radius, angle);
                       Quaternion rot = Quaternion.FromToRotation(Vector3.forward, center-pos);
                if (idx == 0) {
                               rot = new Quaternion (0.0f, 180.0f, 0.0f, 0.0f);//changed x for tilting down
                       }

                // change the position of the camera, FOV and whatever parameters your lightfield calculation calls for
                theCam.transform.position = pos;
                  theCam.transform.rotation = rot;
                  theCam.fieldOfView = 60;
                  theCam.nearClipPlane = radius-.5f;
                  theCam.farClipPlane = radius+.5f;

                  // Probably don't want a skybox
                  theCam.clearFlags = CameraClearFlags.SolidColor;
                  theCam.backgroundColor = Color.black;

```

```
                // set the viewport for each camera to its corresponding tile
                // this is normalized coordiantes where 0,0 is left and bottom, 1,1 is right and top
                //theCam.rect = new Rect((float)x / ViewDimX, (float)y / ViewDimY, 1.0f / ViewDimX, 1.0f / ViewDimY);

                // set each camera display to secondary display render texture
                theCam.targetTexture = PerViewRenderTextures[idx];
            }
        }
    }

    // generate the radial position for the cameras
    Vector3 GenerateRadialPosition (Vector3 center , float radius, float angle){
                float ang = angle;
                Vector3 pos;
                pos.x = center.x+ radius* Mathf.Sin(ang * Mathf.Deg2Rad);
                pos.y = center.y ;
                pos.z = center.z + radius * Mathf.Cos(ang * Mathf.Deg2Rad);
                return pos;
        }
}
```

LightfieldGenerator.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LightfieldGenerator : MonoBehaviour {

    [Tooltip("Shader file which will compute the light field from the render texture containing a grid of views")]
    public Shader ComputeLightFieldShader = null;
    public Shader ToPolarCoordsShader = null;
    public bool polarWarp = true;

    [Tooltip("The size of the pixels in your display (assuming square pixels), in meters")]
    public float PixelPitch;

    Material computeLightFieldMaterial;
    Material toPolarCoordsMaterial;
    ViewGenerator viewGenerator;
    RenderTexture transitionTexture;

  private string filepath = "/Users/emily/Documents/mit/funnelvision/debug/";

    // Use this for initialization
    void Start () {

        // Check if the shader is present
        if (!ComputeLightFieldShader)
            ComputeLightFieldShader = Shader.Find("LightField/ComputeLightField");

        // Check if the shader is present
        if (!ToPolarCoordsShader)
            ToPolarCoordsShader = Shader.Find("LightField/ToPolarCoords");

```

```
        // create the material which is used to set shader parameters, properties, etc.
        computeLightFieldMaterial = new Material(ComputeLightFieldShader);
        toPolarCoordsMaterial = new Material(ToPolarCoordsShader);

        // Check if the view generator is attached to the lightfield generator
        if(!viewGenerator)
            viewGenerator = GetComponent<ViewGenerator>();

        CreateTransitionTexture();
    }

// Update is called once per frame
void Update () {

        // create the tiled render texture for shader to sample from
          Texture2D tiledView = new Texture2D(viewGenerator.TiledViewsRenderTexture.width, viewGenerator.TiledViewsRenderTexture.height);
        tiledView.ReadPixels(new Rect(0, 0, viewGenerator.TiledViewsRenderTexture.width, viewGenerator.TiledViewsRenderTexture.height), 0, 0);
        byte[] bytes = tiledView.EncodeToPNG();
        System.IO.File.WriteAllBytes(filepath + "tiledView" + ".png", bytes);

        // toggles the polar warp shader
        if (Input.GetKeyDown("p")){
                 polarWarp = !polarWarp;
          }

        // Set the shader parameters via the material here
        computeLightFieldMaterial.SetInt("_ViewDimX", viewGenerator.ViewDimX);
        computeLightFieldMaterial.SetInt("_ViewDimY", viewGenerator.ViewDimY);
          computeLightFieldMaterial.SetInt("_ViewsPerLenticular", viewGenerator.viewsPerLenticular);
          computeLightFieldMaterial.SetInt("_NumRepeats", viewGenerator.numRepeats);
          computeLightFieldMaterial.SetInt("_DisplayWidth", viewGenerator.ViewDimX*viewGenerator.viewsPerLenticular*viewGenerator.numRepeats);
        computeLightFieldMaterial.SetInt("_DisplayHeight", viewGenerator.DisplayHeight);
        computeLightFieldMaterial.SetFloat("_PixelPitch", PixelPitch);
    }

    private void OnRenderImage(RenderTexture source, RenderTexture destination)
    {

          var viewWidthPx = viewGenerator.DisplayWidth / viewGenerator.ViewDimX;
        var viewHeightPx = viewGenerator.DisplayHeight / viewGenerator.ViewDimY;

        // This function will execute our shader using the views render texture
        // as the source and destination as our main camera render texture
        for (int y = 0, idx = 0; y < viewGenerator.ViewDimY; y++)
            for(int x = 0; x < viewGenerator.ViewDimX; x++, idx++)
            {
                Graphics.CopyTexture(
                    viewGenerator.PerViewRenderTextures[idx],
                    0,
                    0,
                    0, 0,
```

```
                    viewWidthPx,
                    viewHeightPx,
                    viewGenerator.TiledViewsRenderTexture,
                    0,
                    0,
                    viewWidthPx * x, viewHeightPx * y);
            }

        // If polar warping is enabled, do the anamorphic distorion to make the rendered image appear correctly in the conical mirror
        if (polarWarp){
            Graphics.Blit(viewGenerator.TiledViewsRenderTexture, transitionTexture, computeLightFieldMaterial);
            Graphics.Blit(transitionTexture, destination, toPolarCoordsMaterial);
        }

        else{
            Graphics.Blit(viewGenerator.TiledViewsRenderTexture, destination, computeLightFieldMaterial);
        }
    }

    void CreateTransitionTexture()
    {
        transitionTexture = new RenderTexture(viewGenerator.DisplayWidth, viewGenerator.DisplayHeight, 24, RenderTextureFormat.ARGB32);
        transitionTexture.Create();
    }
}
```

ComputeLightField.shader

```
Shader "LightField/ComputeLightField"
{
    Properties
    {
        _MainTex ("Texture", 2D) = "white" {}
        _ViewDimX("Number of view columns", Int) = 30
        _ViewDimY("Number of view rows", Int) = 1
        _ViewsPerLenticular("Number of view per lenticular lens", Int) = 6
        _NumRepeats("Number of repeated groupings of views", Int) = 20
        _DisplayWidth("Number of pixels in horizontal dimension of display", Int) = 3840
        _DisplayHeight("Number of pixels in the vertical dimension of display", Int) = 2160
        _PixelPitch("The width of the pixel (and height), in meters", Float) = 0.00016
    }
    SubShader
    {
        // No culling or depth
        Cull Off ZWrite Off ZTest Always

        Pass
        {
            CGPROGRAM
            #pragma vertex vert
            #pragma fragment frag

            #include "UnityCG.cginc"
```

```

            struct appdata
            {
                float4 vertex : POSITION;
                float2 uv : TEXCOORD0;
            };

            struct v2f
            {
                float2 uv : TEXCOORD0;
                float4 vertex : SV_POSITION;
            };

            v2f vert (appdata v)
            {
                v2f o;
                o.vertex = UnityObjectToClipPos(v.vertex);
                o.uv = v.uv;
                return o;
            }

            sampler2D _MainTex;
            int _ViewDimX;
            int _ViewDimY;
            int _ViewsPerLenticular;
            int _NumRepeats;
            int _DisplayWidth;
            int _DisplayHeight;
            float _PixelPitch;

            fixed4 frag (v2f i) : SV_Target
            {

// i.uv is a vector2 that tells us our current position in normalized device coordinates

// e.g. i.uv.x == 0 is left, i.uv.y == 0 is bottom, i.uv.x == 1 is right, i.uv.y == 1 is top determine our position, in pixel coordinates

                float2 screen_pos_px = i.uv * float2(_DisplayWidth, _DisplayHeight);

 // given our position in pixel coordinates, we can figure out what our position is on the display in a literal sense. this is determined by i.uv, resolution and pixel pitch(essentially we just multiply the size of the pixel by number of pixels in each dimension to determine the real size of the display)
                float2 screen_position_m = screen_pos_px * _PixelPitch;

// offset the position by half display width in horizontal dimension, and half height in vertical dimension, this shifts our real world coordinate system so that 0,0 is the center of the screen.
                screen_position_m = screen_position_m - float2((_DisplayWidth * _PixelPitch * 0.5f), (_DisplayHeight * _PixelPitch * 0.5f));

                uint2 view_dim_px = uint2(_DisplayWidth/ _ViewDimX, _DisplayHeight / _ViewDimY);

// get the index of the view for our current position
                uint2 view_idx = (screen_pos_px.x) % uint2( _ViewsPerLenticular, _ViewDimY) + screen_pos_px.x /_NumRepeats/_ViewsPerLenticular;

```

```
// tells us where that view starts in pixel coordinates
                uint2 view_offset_px = screen_pos_px / uint2(_ViewDimX, _ViewDimY);

// get the actual pixel position from the view we want to sample from
                float2 view_position_px = view_idx * view_dim_px;

// finally we can get the view position in normalized device coordinates (0-1)
                float2 view_position = view_position_px / float2(_DisplayWidth, _DisplayHeight);

                // sample the view texture at view_position
                fixed4 col = tex2D(_MainTex, view_position);

                return col;
            }

            ENDCG
        }
    }
}
```

PolarWarp.shader

```
Shader "LightField/ToPolarCoords"
{
    Properties
    {
        _MainTex ("Texture", 2D) = "white" {}
        M_PI("PI", Float) = 3.14159f
    }
    SubShader
    {
        // No culling or depth
        Cull Off ZWrite Off ZTest Always

        Pass
        {
            CGPROGRAM
            #pragma vertex vert
            #pragma fragment frag

            #include "UnityCG.cginc"

            static const float M_PI = 3.141592653589f;
            static const float aspect_ratio = 1.777777777f; // 16 W / 9 H = 1.777

            struct appdata
            {
                float4 vertex : POSITION;
                float2 uv : TEXCOORD0;
            };

            struct v2f
            {
                float2 uv : TEXCOORD0;
                float4 vertex : SV_POSITION;
            };

            v2f vert (appdata v)
```

```
            {
                v2f o;
                o.vertex = UnityObjectToClipPos(v.vertex);
                o.uv = v.uv;
                return o;
            }

            sampler2D _MainTex;

            fixed4 frag (v2f i) : SV_Target
            {
                float2 scaled = float2((i.uv.x-.5f)*aspect_ratio, i.uv.y -.5);

                float2 polar = (float2(
                    atan2(scaled.y, scaled.x) / (2.0*M_PI) + (scaled.y >= 0 ? 0 : 1), // angle
                    length(scaled*1.5) // radius
                ));

                fixed4 col = tex2D(_MainTex, polar);
                return col;

            }
            ENDCG
        }
    }
}
```

8.2 Additional Images

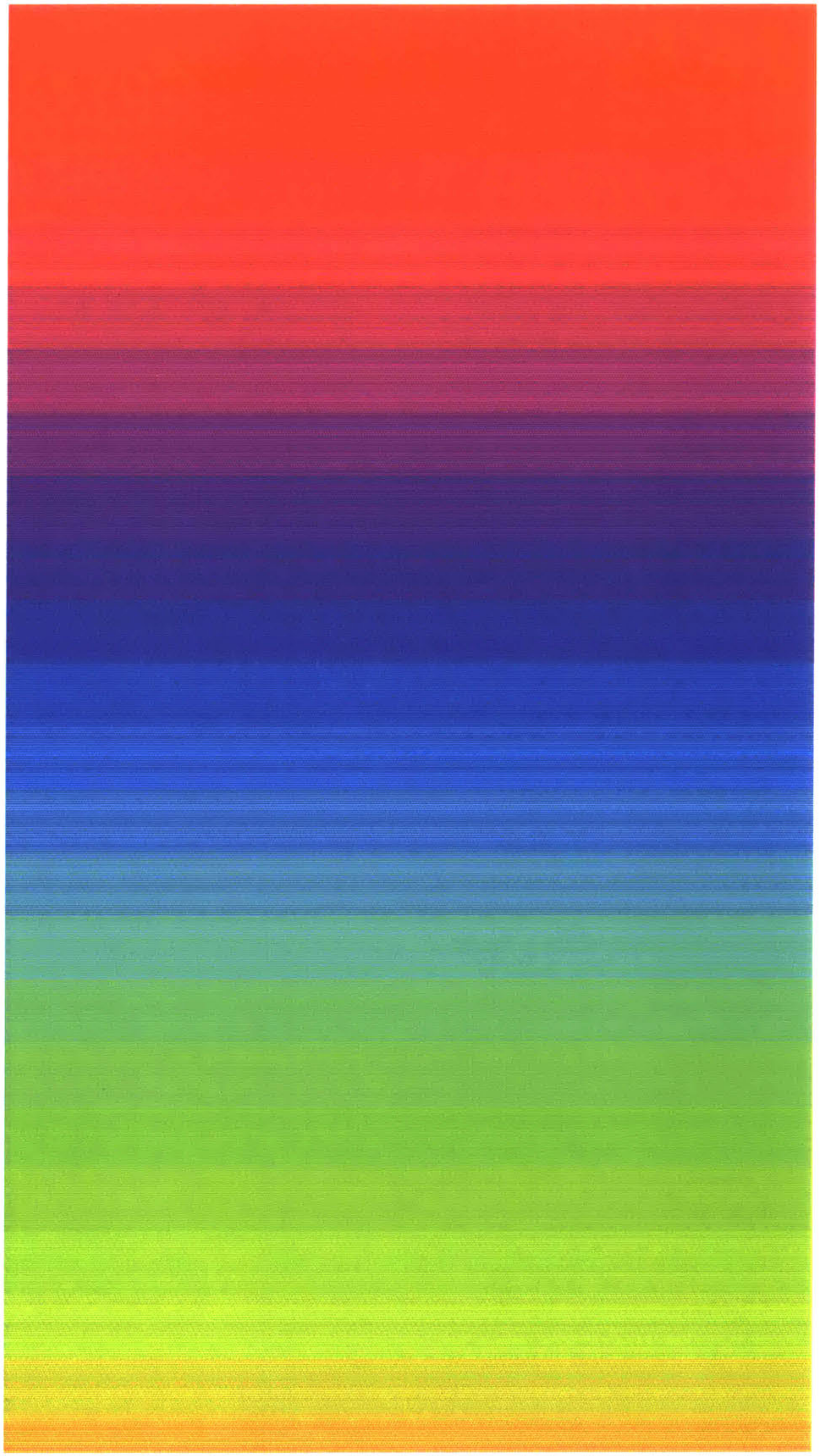


FIGURE 8.1: Larger view of lenticular rendering for radial optical partitioning.

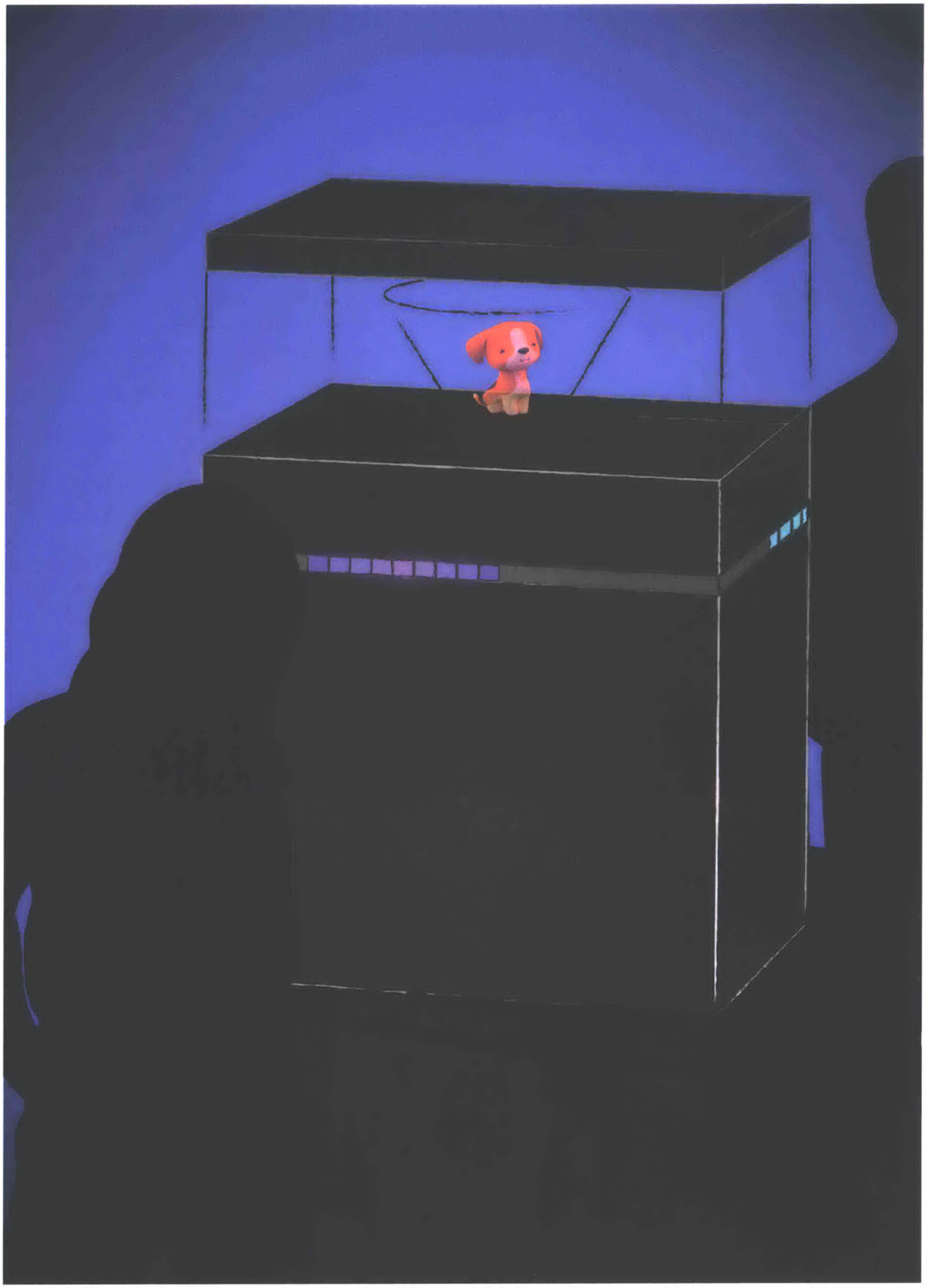


FIGURE 8.2: Concept art for the Funnel Vision display with integrated LEDs.

8.3 USER STUDY

PRE-STUDY QUESTIONS

→ **How positive do you feel today?**
[Ranking 1 – 5 from 'not positive at all' to 'very positive']

→ **Have you used an Augmented Reality device (like a Magic Leap headset) before?**
[Yes, No, Other]

→ **When do you think Augmented Reality (AR headsets, displays or other interventions) will become a viable product?**
[< 1 year, 1 – 2 years, 3 – 5 years, 5 – 10 years, 10+ years, I don't think AR will ever be a viable product]

IN PROGRESS QUESTIONS

→ **What was your favorite interaction with the dog character?**
[Play, Treat, Sleep, Style, Other]

→ **How clearly could you read the emotions on the dog's face?** [Ranking 1 – 5 from 'not clear' to 'clearly visible']

→ **How clearly could you see the character animations?**
[Ranking 1 – 5 from 'not clear' to 'clearly visible']

→ **Did it look like the dog was tracking/responding to the activity card?**
[Yes, No, Other]

→ **What do you think the LED indicators represent?**
[short text answer]

→ **Did you find the LED indicators helpful?**
[Yes, No, Other]

→ **Any feedback or comments you'd like to share at this point?**
[optional short text answer]

POST ACTIVITY QUESTIONS

→ **On which display was it easier to read the character's facial features?**
[2D Monitor, Funnel Vision]

→ **On which display did you feel like there was more character presence?**
[2D Monitor, Funnel Vision]

→ **Which display did you enjoy interacting with more?**
[2D Monitor, Funnel Vision]

→ **Are you more enthusiastic about Augmented Reality (AR) after trying the Funnel Vision system?**
[Yes, No, Other]

→ **Elaborate on the previous question if you'd like!**
[Optional short text answer]

→ **How positive do you personally feel after completing this user study?**
[Ranking 1 – 5 from 'not positive at all' to 'very positive']

→ **Did you notice that the character could interpret your emotional face pose?**
[Yes, No, Other]

→ **Would you be interested in a digital character assistant that understands your affective state?**
[Yes, No, Other]

→ **Which of the activities did you prefer on which display?**
[Play, Treat, Sleep, Style with options for Funnel Vision, 2D monitor, neither or both]

→ **If you've used augmented reality glasses before, how do you think Funnel Vision compared?**
[optional long text answer]

→ **What new activities/features would you like to see on Funnel Vision?**
[optional long text answer]

→ **How could you see Funnel Vision being used in a collaborative, multi-user context?**
[optional long text answer]

→ **If you could have a consumer product version of Funnel Vision in your home, what would you want the character assistant to understand about you? (example, your emotional state, your schedule, your location)**
[optional long text answer]

→ **Would you want the features from the question above to be opt in, transparent, or embedded?**
[optional long text answer]