VOLUMETRIC AND VARIFOCAL-OCCLUSION AUGMENTED REALITY DISPLAYS

Kishore Rathinavel

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2020

Approved by:

Henry Fuchs

David Luebke

Leonard McMillan

Montek Singh

Gordon Wetzstein

Turner Whitted

ABSTRACT

Kishore Rathinavel: Volumetric and Varifocal-occlusion augmented reality displays
(Under the direction of Henry Fuchs)

Augmented Reality displays are a next-generation computing platform that offer unprecedented user experience by seamlessly combining physical and digital content, and could revolutionize the way we communicate, visualize, and interact with digital information. However, providing a seamless and perceptually realistic experience requires displays capable of presenting photorealistic imagery, and especially, perceptually realistic depth cues, resulting in virtual imagery being presented at any depth and of any opacity. Today's commercial augmented reality displays are far from perceptually realistic because they do not support important depth cues such as mutual occlusion and accommodation, resulting in a transparent image overlaid onto the real-world at a fixed depth. Previous research prototypes fall short by presenting occlusion only for a fixed depth, and by presenting accommodation and defocus-blur only for a narrow depth-range, or with poor depth or spatial resolution.

To address these challenges, this thesis explores a computational display approach, where the display's optics, electronics, and algorithms are co-designed to improve performance or enable new capabilities. In one design, a Volumetric Near-eye Augmented Reality Display was developed to simultaneously present many virtual objects at different depths across a large depth range (15 - 400 cm) without sacrificing spatial resolution, frame rate, or bitdepth. This was accomplished by (1) synchronizing a high-speed Digital Micromirror Device (DMD) projector and a focus-tunable lens to periodically sweep out a volume composed of 280 single-color binary images in front of the user's eye, (2) a new voxel-oriented decomposition algorithm, and (3) per-depth-plane illumination control. In a separate design, for the first time, we demonstrate depth-correct occlusion in optical see-through augmented reality displays. This was accomplished by an optical system composed of two fixed-focus lenses and two focus-tunable lenses to dynamically move the occlusion and virtual image planes in depth, and designing the optics to ensure unit magnification of the see-through real world irrespective of the occlusion or virtual image plane distance.

Contributions of this thesis include new optical designs, new rendering algorithms, and prototype displays that demonstrate accommodation, defocus blur, and occlusion depth cues over an extended depth-range.

To Mom and Dad

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Henry Fuchs, who has guided me and supported me throughout my PhD career. I want to thank my committee members: David Luebke, Montek Singh, Leonard McMillan, Gordon Wetzstein, and Turner Whitted, for investing a lot of time and effort in guiding me along this PhD. Their brilliant ideas and useful suggestions often changed the course of my research for the better. Thanks Gordon for inspiring discussions, and for advising me on my occlusion paper, and later too.

I want to thank my internship supervisors: Gordon Wetzstein, David Luebke, Kaan Aksit, Fu-Chung Huang, Josef Spjut, and Jaron Lanier for their fruitful collaboration, helpful discussions, and teaching me so many things about doing research and writing papers.

Thanks to my research lab members Alex Blate, YoungWoon Cha, Praneeth Chakravarthula, David Dunn, Peter Lincoln, Andrew Maimone, Jim Mahaney, and Hanpeng Wang for a fun lab atmosphere and many helpful discussions.

I want to thank my parents for their numerous sacrifices in ensuring an excellent education for me and instilling in me a strong set of values and principles. I offer my utmost love and appreciation to you. Thanks to Keshav for giving me a lifetime's worth of beautiful memories. I want to thank my wife, Anusha Lalitha, whose constant support, encouragement, and several visits while pursuing her own PhD across the country made this PhD possible. Your constant belief in me made all the difference. Thanks to Prasad uncle, Padmaja aunty, and Sameer for your kindness, understanding, and cheerful encouragement.

My graduate life has been memorable and enjoyable thanks to friends: Sridutt Balachandra, Srihari Pratapa, Atul Rungta, and many others.

Also, thanks to the National Science Foundation and Intel for funding my PhD research. Thanks to Ronald Azuma and Greg Leeming from Intel for helpful discussions and feedback.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
AR	Augmented Reality
ASIC	Application Specific Integrated Circuit
CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DDS	Direct Digital Synthesis
DLP	Digital Light Processing
DMD	Digital Micromirror Device
DVI	Digital Visual Interface
fps	frames per second
FoV	Field of View
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
HDR	High Dynamic Range
HMD	Head Mounted Display
Hz	Hertz
LCD	Liquid Crystal Display
LCoS	Liquid Crystal on Silicon
LED	Light Emitting Diode
NED	Near Eye Display
PC	Personal Computer
RAM	Random Access Memory
RGB	Red, Green, and Blue
SLM	Spatial Light Modulator
SSIM	Structural Similarity Index
VR	Virtual Reality
PSNR	Peak Signal-to-Noise Ratio

CHAPTER 1: INTRODUCTION

1.1 Motivation

> *' A display connected to a digital computer gives us a chance to gain familiarity with concepts not realizable in the physical world. It is a looking glass into a mathematical wonderland. '* Sutherland (1965)

The goal of augmented reality is to seamlessly integrate the real world and the digital world. Although Augmented Reality hopes to address the combination of real and digital worlds for all our sense-modalities, i.e., sight, smell, touch, and hearing, in this dissertation, we shall only consider Augmented Reality for sight. There has been significant progress over the decades to create synthetic photorealistic imagery for movies, games, etc. But until recently, the most common way to interact with digital worlds was through 2D displays such as televisions, computer monitors, or mobile phones and tablets. These displays are 2D displays and subtend a narrow field-of-view. Since we live in a 3D world and our eyes see a wide field-of-view image, today's displays severely limit our communication, visualization, and interaction with the digital world. Providing a seamless, perceptually realistic experience requires more than just rendering photorealistic imagery. *Perceptual realism* requires displays that can present wide field-of-view high-quality imagery, and in particular, it requires that these displays support all depth cues of the human visual system Palmer (1999); Howard and Rogers (2002) accurately. Display technologies that enable a seamless combination of the real and digital worlds could revolutionize the way we communicate, visualize and interact with digital information. Fig. 1.1 shows some compelling envisioned applications for future Augmented Reality systems, and below is a brief description of each envisioned application:

Multiple screens anywhere: an augmented reality (AR) system can not only fully replace our current 2D displays (televisions and computer monitors) but can place virtual versions of our 2D displays at any location around us. The future workspace would be more customizable and productive than ever before.

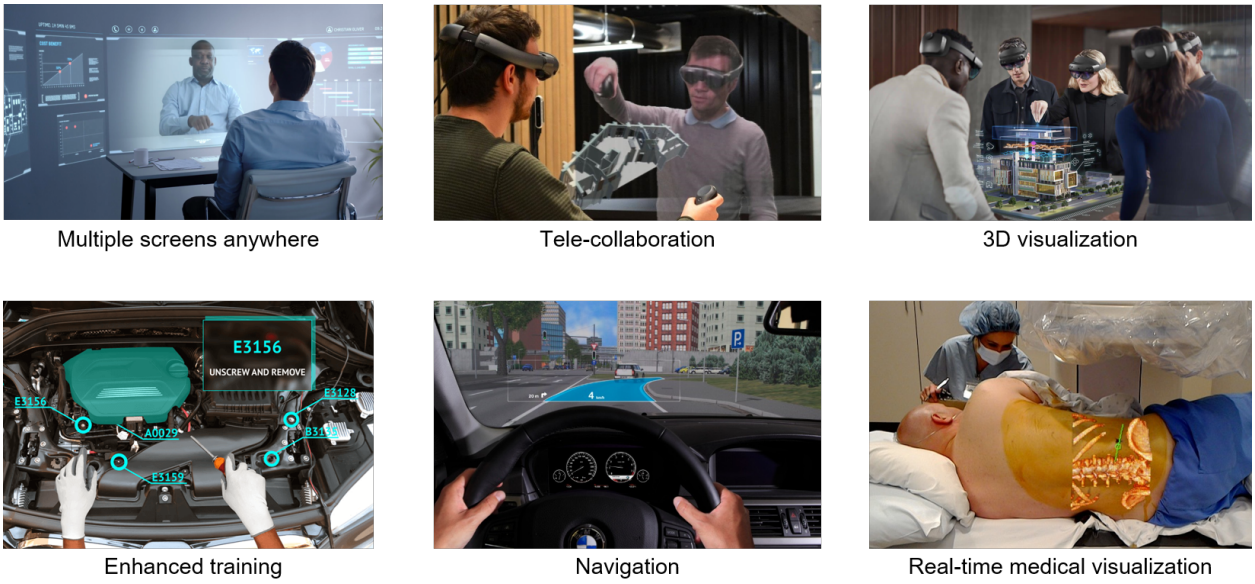

Figure 1.1: Augmented Reality provides new and better methods to communicate and visualize. Figure shows some envisioned applications. Image credits are given below:
(top,left): https://www.pinterest.com/pin/353462270745212902/
(top,middle): https://www.mimesysvr.com/
(top,right): https://medium.com/cgi-norway/augmented-reality-in-business-and-future-for-consumers-d4b8fcdee4a6
(bottom,left): https://www.growthbusiness.co.uk/uk-based-ar-firm-makes-largest-ar-fundraise-of-the-year-2555730/
(bottom,middle): https://www.driversalert.com/augmented-reality-heads-up-display-distracted-driving/
(bottom,right): https://hololens.reality.news/news/hololens-assists-live-surgery-0178887/

Tele-collaboration: an AR system comprising of an AR display and a 3D reconstruction system can enable an advanced version of today's video teleconferencing. In the envisioned system, the remote AR system would capture the remote person's geometry and colors and transmit it to the local AR system, which would render and display the remote person such that they appear to be in the local physical space.

3D visualization: future AR system would enable us to visualize 3D structures and volumetric effects (fog, fluid simulations, etc.) in very informative ways. Sometimes, future AR systems may even be better than recreating such 3D structures in the real-world because in a digital system, we could have the option to visualize not just the structure as a whole but informative versions of the structure, e.g., cross-sectional view, part-wise view, etc. It may even be possible to visualize information that is ordinarily invisble, e.g., visualization of energy propagation has been explored in Lanier et al. (2016, 2018).

Enhanced training: Augmented Reality can spatially register training instructions onto the real-world objects which the user needs to learn to use.

Navigation: Similar to the point above, navigation is another example of contextual information that can be presented better with an AR display because it is spatially registered to where the user is already looking.

Real-time medical visualization: one of the most important applications for future AR is probably in medical sciences, e.g., spatially registering visualization of the internal organs of the current patient undergoing surgery.

1.2 Scope of this dissertation

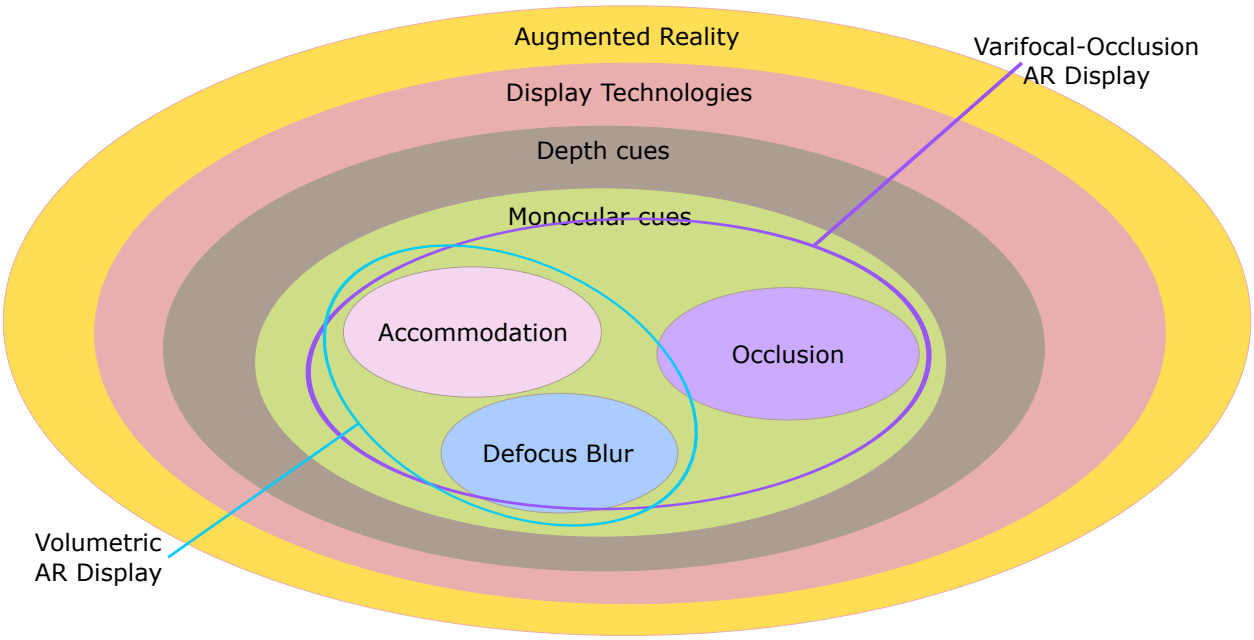

Figure 1.2: Figure shows the scope of this dissertation.

Fig. 1.2 depicts the scope of this dissertation. There are many enabling technologies and active research areas for Augmented Reality, e.g., display technologies, 3D reconstruction, tracking, virtual assistants, redirected walking, etc. This dissertation focusses on display technologies. Even for just the topic of display technologies, there are a number of desired requirements outlined in Sec. 2.4. This dissertation considers only three specific monocular depth cues, namely: occlusion, accommodation, and defocus-blur. These depth-cues are explained and discussed in detail in Sec. 2.1.2, but briefly, these three depth cues are important monocular depth cues whose absence in AR displays has demonstrated reduced task performance and increased discomfort Lambooij et al. (2009); Shibata et al. (2011); Wann et al. (1995). For

these depth cues, this dissertation develops two display technologies that improve the presentation of these depth cues over a large depth-range.

1.3 Thesis Statement

The use of computational displays, where the optics, electronics, and algorithms are co-designed, will improve accommodation, defocus-blur, and occlusion in AR displays.

1.4 Contributions of this dissertation

For accommodation and defocus blur, this dissertation's contributions are: (1) A volumetric near-eye display (NED) exhibiting 280 perceptually simultaneous binary depth planes, each an arbitrary RGB color, situated between 15 cm (6.7 diopters) and 400 cm (0.25 diopters) from the viewer. (2) A fixed-pipeline decomposition algorithm that converts a 3D color volume to a set of single-color binary depth planes, such that 24 bpp color voxels are displayed at 280 unique depth positions. (3) Adaptive color decomposition algorithms that convert a 3D color volume to a set of single-color binary depth planes which show improvements in depth-blur, perceptual loss metrics, and allow depicting transparent objects.

For depth-dependent occlusion, accommodation, and defocus blur, this dissertation's contributions are: (1) Varifocal occlusion as an AR display capability that adaptively changes the focal distance of an occlusion mask to enable depth-dependent hard-edge occlusion. (2) Complementary approaches of optimization and closed-form solutions for arriving at an optical design that enables a focus-tunable optical system to achieve varifocal occlusion in a perceptually realistic manner without optically distorting the observed scene. (3) A monocular varifocal occlusion-capable AR display prototype that demonstrates depth-dependent occlusion over a large depth range (30 cm to 300 cm).

The broad contributions of this dissertation are new optical designs, new real-time rendering algorithms, and prototype displays that demonstrate monocular depth-cues such as accommodation, defocus-blur, and hard-edge occlusion over a large depth range.

CHAPTER 2: BACKGROUND

This chapter breifly discusses the background knowledge required to understand the work presented in subsequent chapters. We first discuss the relevant properties of the human eye and the depth cues that are avaiable to us. We then discuss different technological approaches that have tried to combine the physical and virtual worlds. We explain why augmented reality head mounted displays are more suitable to our goal and briefly mention the state of the art augmented reality displays and their limitations in addressing the depth cues that we are particularly interested, i.e., accommodation, defocus blur, and occlusion. A in-depth discussion of previous augmented reality displays is presented in Sections 3.2 and 4.2.

2.1 The Human Visual System

2.1.1 The human eye

All the depth cues that are addressed in this dissertation can be explained by analyzing the image formation mechanism in the human eye. Hence, we start with a brief description of the human eye's components and mechanism involved in imaging incoming light, so that we can build display systems that can provide the appropriate light information.

Fig. 2.1 shows a schematic diagram of the human eye. Light from the external world passes through the pupil and is focused by the lens onto the retina. The pupil acts like an aperture stop in cameras. The pupil adapts its radius to adjust to the world's level of brightness, e.g., in a dark room, the pupil's radius is larger to allow more light into the eye. The lens' shape is deformable by the ciliary muscles. The lens' shape controls the focal length of the eye which in turn determines the distance at which the eye is focussed to.

Our eye's retina is composed of two types of sensory cells: cones and rods. Cones and rods serve different purposes, e.g., cones are capable of color vision, whereas rods have an achromatic response, but rods are extremely sensitive to light and hence are useful for low-light conditions. Interestingly, each cone is connected to an optic nerve, but multiple rods share an optic nerve. Hence, cones have a higher visual

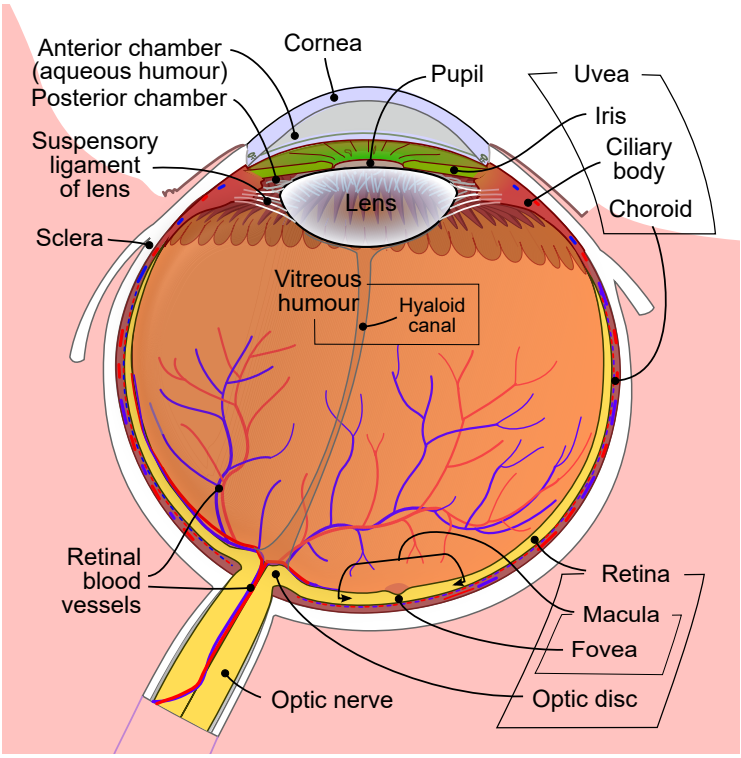

Figure 2.1: Schematic diagram of human eye. Image source: Rhcastilhos. and Jmarchn. (2007)

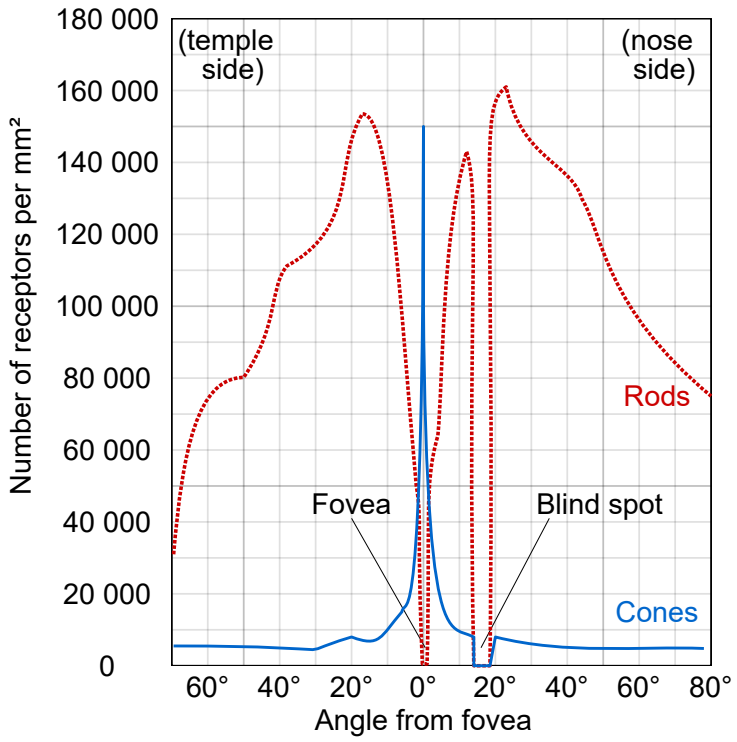

Figure 2.2: Distribution of rods and cones along a line passing through the fovea and the blind spot of a human eye. Image source: Cmglee (2019)

acuity than rods. Interestingly, the distribution of cones and rods on our retina is non-uniform. Fig. 2.2 shows the distribution of rods and cones. Observe how there is a high concentration of cones in a narrow region. This region is called the *fovea*.

2.1.2 Depth cues

2.1.2.1 Binocular depth-cues

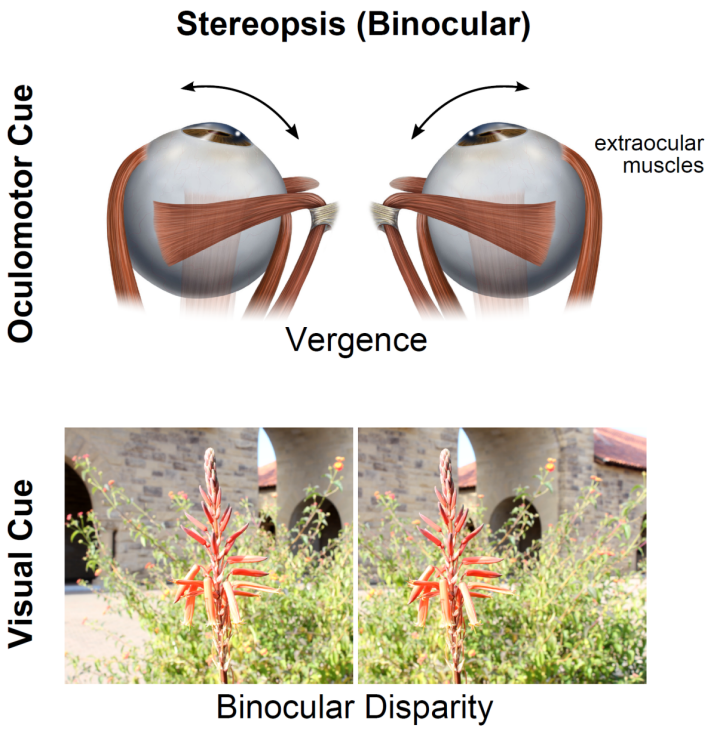

Figure 2.3: We perceive depth from these binocular cues (1) *convergence:* our eyes rotate to form the image of the object of attention at our retina, and (2) *disparity:* the images formed in our two eyes are slightly different. Image source: Adapted from Konrad et al. (2017)

Some depth cues that we are aware of depend on information from both eyes. Fig. 2.3 depicts two binocular depth cues and a brief explanation follows:

Disparity Since our eyes are at slightly different positions, the image seen by them is also slightly different. This slight shift in the content between the images formed in our eyes is called *disparity*.

The disparity between the images is also dependent upon the depth of the object from the eyes; the closer the object is, the higher the disparity in its image between the two eyes.

Convergence When we look at an object, our eyes automatically rotate in such a manner that the image of the object of attention is formed at the fovea of both our eyes. Depending on the distance of the object of attention, the angle of convergence changes. The convergence angle is greater for closer objects and less for farther objects. The depth of the object determines the angle by which our eyes have to rotate inwards. This depth cue is called *convergence*.

2.1.2.2 Monocular depth-cues

Depth cues that are available even when the world is viewed with one eye alone are called monocular depth cues. There are several monocular depth cues, namely: *occlusion, accommodation, defocus-blur, intrapupillary occlusion, chromablur,* etc. We discuss each of these monocular depth-cues below:

Accommodation Our eyes have a narrow opening called the pupil to let in light from the world, behind which, we have a lens which can be deformed to change the focal distance of the eyes. We automatically try to bring the object of attention into sharp focus on the retina by deforming this lens. This ability to change the focus is called accommodation and it provides us with an estimate of the distance.

Defocus blur A given lens state fixes the focal distance and brings objects at that focal distance into sharp focus at the retina, but makes objects at other distances blurred. This is a property that can be observed in any single-lens imaging system. Some display technologies have been developed that dynamically refocus the virtual plane distance to the eye's focal depth, thereby providing accommodation depth cues. In these displays, the objects that are at depths far away from the virtual plane's depth are computationally blurred to create a synthetic defocus blur effect.

Occlusion Occlusion is a relative depth order cue which arises when the nearer object partially obstructs the view of a farther object. Occlusion actually only informs us about the depth ordering and is not useful to estimate the magnitude of depth.

Such depth cues, which only provide information about the depth order (relative depth) instead of a quantitative estimate of the depth, are called nonmetrical depth cues. Other depth cues discussed so far (disparity, convergence, accommodation, defocus blur) provide a quantitative estimate of the depth and are called metrical depth cues.

Even though occlusion is a nonmetric depth cue, it is nonetheless the most important depth cue. In Cutting and Vishton (1995), an experiment was done where two objects A and B are shown at different

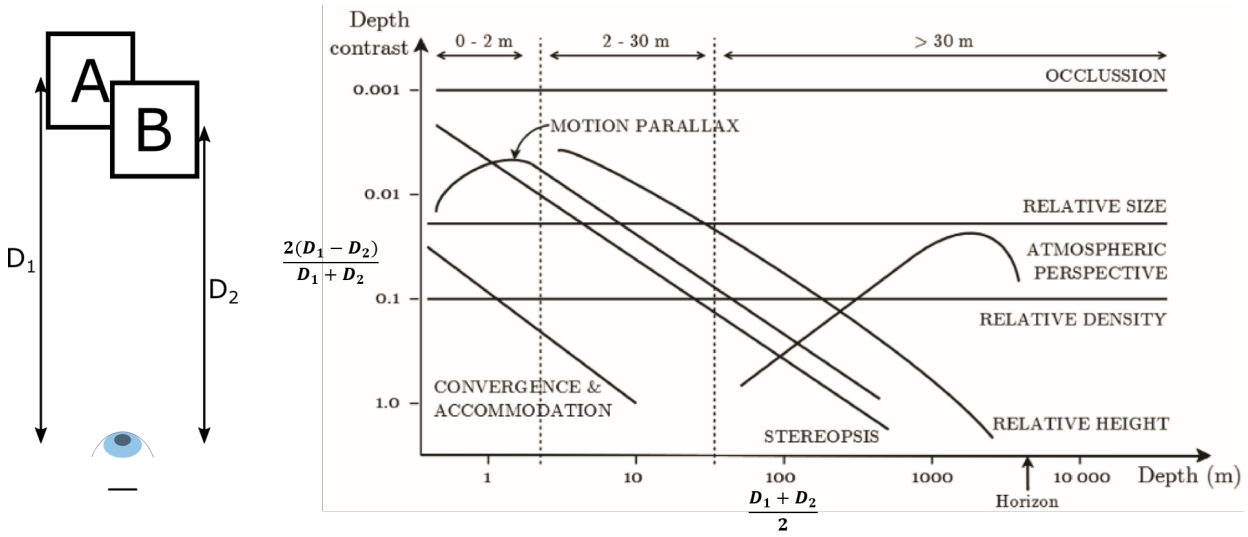

Figure 2.4: Figure shows the relative importance of different depth cues. Source: Adapted from Cutting and Vishton (1995)

depths, say D_1 and D_2, and the user is then asked to make a forced-choice as to which object is closer. To study the effect of different depth cues, various trials were conducted with only a few depth cues active in insolation. And the results of the experiments are summarized by the graph shown in Fig. 2.4. The horizontal axis of Fig. 2.4 shows the average distance of the two objects, i.e., $\frac{D_1+D_2}{2}$ and the vertical axis shows the depth contrast between the two objects, i.e., $\frac{2(D_1+D_2)}{D_1+D_2}$. And each curve shows minimum depth contrast required for correct ordering at a given depth and given depth cue. We can see that for occlusion, the depth contrast can be very low, and we'd still be able to order the objects correctly.

Other monocular depth cues There are some other monocular depth cues which are not addressed in this dissertation, and these are listed below:

- Chromatic aberration, which is present in almost all imaging and display systems, causes slight dependence of the size of the defocus blur on the wavelength of the light Cholewiak et al. (2017).
- Intra-pupillary occlusions Zannoli et al. (2016): Refers to the view-dependent occlusion and disocclusion effects seen across the area of the pupil. The effects of these view-dependent effects is an asymmetry in occlusion boundaries based on the accommodation state of the eye: When the eye is focused at a nearby object, its occlusion boundary is seen sharply against a blurred background. However, when the eye is focussed at the background, the occlusion boundary of the nearby object is blurred onto the background.

- Occular parallax Konrad et al. (2020): Refers to the view-dependent occlusion and disocclusion effects seen for the different pupil positions.
- Depth cues due to motion: There are multiple depth cues we infer from our own motion, the motion of objects, or the motion of light sources:
  - Motion parallax: our motion causes different relative motion for objects at different depths against a fixed background.
  - Kinetic depth effect: Sometimes, we can extrapolate the geometry of an object by observing its shadow or projection provided the three-dimensional object is moving.
  - Depth from motion: We can estimate the distance to an object if it moves relative to us.
- Pictorial depth cues: These are depth cues we develop and use from our understanding of our world, e.g., familiar size, relative size, aerial perspective, ligting and shading.

2.2 Beyond 2D Displays

In this section, we discuss the approaches to extend the virtual world beyond the traditional 2D displays. There are mainly three approaches: (1) 3D displays, (2) shader lamps, (3) Near-Eye Displays. Here's a brief description of each approach:

2.2.1 3D Displays

These displays often resemble the form-factor of traditional computer monitors and present virtual scene with 3D depth cues by presenting each eye with viewpoint-dependent imagery Geng (2013); Holliman et al. (2011). Typically, these displays present only binocular cues. These displays sometimes time-multiplex the imagery between the two eyes in synchronization with shutter-glasses. Instead of using shutter-glasses, other techniques have been developed that try to re-create the light field of the target 3D scene either by using stacks of transparent displays or other novel optical and mechanical configurations Wetzstein et al. (2012); Jones et al. (2007). With respect to our goal to seamlessly combine the physical and digital worlds, this approach's main limitation is that the imagery is confined to a local volume or field-of-view.

2.2.2 Shader Lamps

These systems employ projectors to display virtual worlds onto physical surfaces such as walls, tables, etc Bimber et al. (2008); Raskar et al. (1998); Jones et al. (2013). Multiple cameras are used to track the physical surfaces and user interactions. Disadvantages with these systems include (1) lack of monocular depth cues, (2) shadows cast by objects and users pose a difficulty for both projectors and cameras, (3) ambient lighting of these specialized rooms needs to be controlled carefully.

2.2.3 Head-Mounted Displays

These are head-worn devices that present imagery to each eye. These systems can also be made completely self-contained where the head-worn display even performs tracking and 3D reconstruction of the environment with only on-unit cameras and sensors, e.g., HoloLens[1], Oculus[2]. Due to their potential to be lightweight self-contained units that can present wide field-of-view imagery, I consider head mounted displays (HMDs) to be the most promising direction for Augmented Reality. However, there are a number of challenges and approaches for Head-Mounted Displays, which are covered in the remainder of this chapter.

2.3 Head-Mounted Displays

NEDs are broadly of two categories, Virtual Reality Displays and Augmented Reality Displays. We briefly discuss each category before focussing on just Augmented Reality displays.

2.3.1 Virtual Reality Displays

Virtual reality (VR), which immerses the user in a completely synthetic environment, is a useful modality in some scenarios, e.g., immersive movies, immersive training for unusual scenarios, or even computer games. While useful in specific scenarios, in VR, interaction with the real-world is generally unavailable. Hence, it is unlikely that users would be willing to be completely cut-off from the real-world for extended periods of use. Therefore, it is difficult to imagine VR as the next productivity tool or computing platform.

[1] URL: https://www.microsoft.com/en-us/hololens

[2] URL: https://www.oculus.com/quest/

2.3.2 Augmented Reality Displays

Augmented Reality NEDs insert virtual objects into the view of the real-world. The user maintains the context of the real-world, and the inserted virtual objects are often contextual and spatially registered with the real-world.

2.3.2.1 Video see-through AR dislays

One proposed technology for AR displays is to use a VR display, but relay the real-world's view using outward-facing camera(s) Rolland and Fuchs (2000); Kanbara et al. (2000); State et al. (2005). This approach solves the occlusion problem trivially, however, this approach has a major limitation being that the view of the real-world is limited by the display's and the camera's resolution (spatial and angular), latency, dynamic range, distortions, field-of-view, and color fidelity. In other words, for a video see-through AR display to recreate the same experience of viewing the real-world without the display, camera technologies and display technologies need to advance significantly.

2.3.2.2 Optical see-through AR displays

Optical see-through AR displays optically insert virtual imagery into the user's view of the real world. Of all the display technologies that seek to integrate the real-world and the virtual world, only optical see-through AR displays propose to do in a portable manner and with minimum encumbrance to the real-world view.

2.3.3 Similarities and Differences between Virtual Reality and Augmented Reality

Augmented Reality and Virtual Reality are similar technologies in these respects: the need for presenting imagery with optics placed close to the eye, the need for head and eye tracking, and a similar rendering and display graphics pipeline, etc. However, there are also differences in the enabling technologies:

1. AR displays require the addition of an occlusion mask to depict opaque virtual objects.

2. Since a VR display completely blocks out the real world, the user's own body cannot be seen naturally. For the user to see their body within a VR HMD, it would be necessary to 3D reconstruct the user's body and display it within the virtual scene in real-time.

3. For spatially registering virtual objects to the real world, it is often necessary to 3D reconstruct the real-world in real-time.

4. Design requirements for AR displays are more stringent. AR displays should not encumber the view of the real-world (cannot have on-axis components that distort or block the real-world), need lower display latency for virtual objects to appear registered to the real-world, and need to support a wider range of brightness levels.

For a in-depth discussion on video see-through vs. optical see-through AR displays, please refer to Rolland and Fuchs (2000) and Rolland et al. (1995).

2.4 Requirements for Optical See-Through Augmented Reality Displays

The below requirements are arranged approximately in the descending order of subjective importance of the author.

1. **Compact form-factor:** Future AR headsets may consist of the display unit, multiple sensors (head and eye tracking, cameras for 3D reconstructing the environment, inertial measurement units (IMU)), computing units (CPU and GPU), communication units, and a battery. Despite integrating all these components, it is important that these devices are lightweight so that the users can wear these devices for long hours. Previous work Yan et al. (2018) recommends that VR headsets should be designed with uniform weight distribution and aim to keep the weight within 300 g.

2. **Wide eyebox:** Eyebox refers to the range of pupil positions from where the virtual image presented by the AR display can be seen. The eyebox is the same as the exit-pupil of the display. Many display technologies and prototypes have been demonstrated which have beautiful imagery but with narrow eyeboxes Maimone et al. (2017); Westheimer (1966). For such displays, eyebox replication techniques may be useful. Jang et al. (2017).

3. **Wide field of view:** The human visual system has a monocular field-of-view of about 120 degrees and a binocular field of view of about 210 degrees. To effectively integrate the real and digital worlds, AR displays should aim to present wide field-of-view imagery.

4. **High resolution:** The human visual system is capable of viewing resolution as high as 60 cycles-per-degree. However, the human visual system has such a high resolution only for a narrow region on the retina called the fovea. Beyond this region, the resolution drops drastically and is very low in the peripheral field-of-view. This non-uniform resolution across the field-of-view provides an opportunity to provide high-resolution imagery without having to build very high-resolution display panels but poses a challenge to dynamically change the display as the eye looks in different directions.

5. **Depth cues:** AR displays should provide depth cues similar to that available in the real world. The various depth cues that the human visual system uses are discussed in Sec. 2.1.2.

6. **Low latency:** AR displays need to respond fast to the user's head and eye movements by updating the imagery being displayed. It can be shown that even 10 milliseconds delay between head motion and display update can result in 5 centimeters of error for a virtual object situated at 2 meters away. Below is a brief derivation for 5 centimeters error for 10 milliseconds latency:

   Suppose ω_{head} denotes the head rotation speed in degrees-per-seconds and t_{latency} denotes the latency of the display system (i.e., the time between the tracking information used for rendering and the display of the currently rendered image), then the error of the currently displayed frame is:

   $$\theta_{\text{error}} = \omega_{\text{head}} * t_{\text{latency}}. \tag{2.1}$$

   For a virtual object that is displayed at distance d_{object} away, the lateral error is given by:

   $$d_{\text{error}} = \tan(\theta_{\text{error}}) * d_{\text{object}}. \tag{2.2}$$

   Suppose $\omega_{\text{head}} = 150$ degrees-per-second and $t_{\text{latency}} = 10$ milliseconds, then $\theta_{\text{error}} = 1.5$ degrees, and if $d_{\text{object}} = 2$ meters, then $d_{\text{error}} = 5.24$ centimeters.

A more detailed description of Augmented Reality and its requirements are covered by these review papers: Azuma et al. (2001) and Carmigniani et al. (2011).

2.5 Overview of previous work for depth cues in AR displays

Current commercially-available AR displays offer impressive capabilities, but they typically do not support important monocular depth cues such as accommodation or mutual occlusion, resulting in a transparent image overlaid onto the real-world at a fixed depth. To realize the vision of Augmented Reality, providing a seamless and perceptually realistic experience requires displays capable of presenting photorealistic imagery, and especially, perceptually realistic depth cues, resulting in virtual imagery being presented at any depth and of any opacity. Previous research prototypes fall short by presenting occlusion only for a fixed depth, and by presenting accommodation and defocus-blur only for a narrow depth-range, or with poor depth or spatial resolution. We briefly discuss major themes in previous work for addressing the lack of accommodation and depth-dependent occlusion. Later, in each technical chapter (Chapter 3 and 4), a detailed review of previous work is presented in context for the technology presented in that chapter.

2.5.1 Accommodation and defocus-blur

Previous AR displays that propose technologies to provide accommodation are broadly classified, and their limitations are mentioned:

1. *Varifocal Displays* (e.g., Konrad et al. (2016); Padmanaban et al. (2017); Dunn et al. (2017); Akşit et al. (2017)): Provides synthetic defocus-blur cues, requires to track accommodation-state of the eye, and has latency in moving the in-focus plane.

2. *Multifocal Displays* (e.g., Akeley et al. (2004); Narain et al. (2015)): Few focal planes which leads to partly synthetic focal cues and reduced spatial resolution for content in-between the few focal planes.

3. *Light-field Displays* (e.g., Lanman and Luebke (2013); Maimone et al. (2014); Huang et al. (2015)): Poor spatial resolution and narrow depth-range.

4. *Holographic Displays* (e.g., Maimone et al. (2017); Shi et al. (2017)): Complex hardware, high computational costs, and hard trade-offs between eyebox, field-of-view, and depth-range.

2.5.2 Occlusion

Previous AR displays that propose technologies to provide occlusion support are broadly classified, and their limitations are mentioned:

1. *Fixed-focus occlusion displays* (e.g., Kiyokawa et al. (2000, 2001, 2003)): Preserve a high-quality of the see-through view, but present the occlusion mask at a fixed distance.

2. *Light-field occlusion displays* (e.g., Maimone and Fuchs (2013)): Attempt to provide depth-dependent occlusion by presenting a 4D light field occlusion mask using stacked *liquid crystal display* (LCD) layers placed out of focus in front of the eye, where the occluding patterns are calculated by light field factorization algorithms Lanman et al. (2010); Wetzstein et al. (2012). While theoretically capable of presenting depth-dependent occlusion cues, this approach's use of LCD panels causes severe diffraction and deterioration of the real-world's view.

CHAPTER 3: VOLUMETRIC AUGMENTED REALITY DISPLAY[1]

This chapter describes an augmented reality display that presents high-quality accommodation and defocus blur cues. Although the system has some limitations (such as bulky form-factor, static demonstration, lack of occlusion), this display is capable of combining the physical world and the digital world for a large depth-range and allows the user to refocus their eyes to any depth and immediately see the correct image.

3.1 Introduction

Near-eye displays that seamlessly integrate virtual content into the real world offer exciting possibilities. Real-virtual integration could induce a paradigm shift in multiple aspects of our lives, including education, communication, entertainment, and others. Near-eye displays, as compared to spatially augmented reality and 3-D displays, allow true immersion in the sense that the near-eye display user could truly experience a virtual world around them in all directions while preserving the user's natural experience and view of the real world. However, several challenges must be addressed to realize truly immersive see-through near-eye displays. One of these is the mismatch between the vergence and accommodation cues of depth perception. As described in Section 2.1.2.1, *Vergence* or *convergence* is the orienting of our eyes such that the image of a fixated object forms on the fovea. As described in Section 2.1.2.2, *Accommodation* is the eye lenses' ability to change their focal length to bring the object of fixation into proper focus on the fovea of both eyes. These are cross-coupled physiological effects. Their absence, mismatch, or incorrect representation (may also apply to other depth cues) can disrupt the sense of presence or immersion and may cause visual discomfort, eyestrain, and nausea Hoffman et al. (2008).

Some of the proposed solutions to the problem of providing such depth cues attempt to approximate focus cues. Varifocal displays, monovision displays, and even some implementations of multifocal displays

[1] Most of this chapter (Sections 3.1 - 3.6) previously appeared as an article in Transactions on Visualization and Computer Graphics. The new sections of this chapter are adaptive color-to-binary decomposition (Sec. 3.7), a real-time display (Sec. 3.8), and optical calibration for the display (Sec. 3.9). The original citation is as follows: Rathinavel, K., Wang, H., Blate, A., and Fuchs, H. (2018b). An extended depth-at-field volumetric near-eye augmented reality display. *IEEE transactions on visualization and computer graphics*, 24(11):2857–2866

are in this category. Some other proposed solutions, such as light field displays and holographic displays, provide accurate focus cues but have limitations. Current implementations of light field displays have poor resolution or are diffraction-limited. Current implementations of holographic displays are compute-intensive and typically have very small eyeboxes. Phase-only *spatial light modulator* (SLM) technologies also need improvement before holographic displays based on these technologies can become practical.

This chapter explores a new class of displays: volumetric near-eye displays. Our approach is to sweep the virtual image plane back and forth over a wide range of diopters and use a high-speed *digital micromirror device* (DMD) coupled with high-speed illumination to present a large number of multiple thin slices of a computer-generated volume. While this sounds similar to multifocal displays that show images at various fixed depths, there is a crucial difference: the number and granularity of the depth planes.

Traditionally, for multifocal displays, the computer-generated volume is decomposed into a series of image planes placed at different depths; for time-multiplexed multifocal displays, this necessitates that the focus-tunable lens or deformable mirror settle down in each focus state. Our approach is to oscillate the focus-tunable lens in a continuous state and display a stack of binary images at high-speed such that the displayed stack of images is perceived as slices of a continuous full-color volume. We decompose the computer-generated volume locally, on a per-voxel basis, and distribute the decomposition around the location of the voxel. A *voxel* is the 3D equivalent of a *pixel* which refers to each element of a 2D display panel. So, a *voxel* is the fundamental geometric unit of a 3D display. In our display, the voxels are actually tiny frusta rather than cubes and are better refered to as *froxel*, but in this dissertation we rather refer to them as *voxel* because it is more familiar to a larger audience. Thus, our rendering algorithm is aware of and leverages the fact that the focus-tunable lens is in continuous motion–rather than assuming a lens that moves and settles in discrete steps. Low-level hardware access to a high-speed DMD and a high-speed *high dynamic range* (HDR) *red-green-blue* (RGB) *light emitting diode* (LED) allows control of display pattern and illumination for each binary frame. We present a rendering pipeline for volumetric near-eye displays that utilizes such hardware.

One might be concerned about the computational complexity of our approach. In our implementation, we make some simplifying assumptions to reduce computational overhead. However, these simplifications might not be desirable in a human-wearable product; without these assumptions, our approach would be moderately computationally demanding. While this might be an encumbrance for today's embedded hardware, we assert that near-eye displays (NEDs) of the future must have substantially more compute

power to perform, e.g., low-latency corrections, head and eye tracking, real-world scene understanding, and so on. For example, onboard GPUs are already found in NEDs such as Microsoft HoloLens. While our current implementation is offline, we believe that future NEDs will have sufficient onboard computational resources to perform the required computations in real-time on the device.

3.1.1 Contributions

This chapter's main contributions are:

1. A volumetric NED exhibiting 280 perceptually simultaneous binary depth plane images, each an arbitrary RGB color, situated between 15cm (6.7 diopters) and 4M (0.25 diopters) from the viewer.

2. A rendering pipeline for the new NED that decomposes 3-D graphics primitives efficiently into the set of binary depth plane images illuminated by a single color, such that 24 bits-per-pixel color voxels can be displayed at 280 unique depth positions.

3.1.2 Benefits

In addition to supporting the current volumetric display implementation, our proposed system can emulate varifocal displays and previous multifocal displays. This could allow the system to become a test-bed for future perceptual studies on accommodation. Our display allows low-level access to many stages of the graphics pipeline between GPU and the actual emission of light rays that form a retinal image. This low-level access could be used to study alternative rendering pipelines for future near-eye displays and advanced projectors. Integration of our present work and previous work with similar hardware Lincoln et al. (2016, 2017) could lead to a near-eye display with several desirable properties (low-latency, high dynamic range, accommodation-capable).

3.2 Related Work

3.2.1 Volumetric Displays

Volumetric displays create multiple real or virtual light sources in a three-dimensional volume of space and can typically be seen from a wide range of angles around the display. These light sources are the 3-D

analog of pixels and are called *voxels*. Earlier designs of volumetric displays were table-top designs and the displayed volume was confined to the *physical volume of the display* Favalora et al. (2002); Sullivan (2004); Cossairt et al. (2007); Ochiai et al. (2016); Refai (2009); Smalley et al. (2018). One of the limitations of most of these displays is that the light sources are presented additively and view-dependent effects, such as occlusion, are absent. This limitation is overcome in Cossairt et al. (2007) and in Jones et al. (2007) by using anisotropic diffusers.

Our proposed display provides a methodology to create virtual light sources over an *extended volume external* to the display's physical volume. Applied to near-eye displays, this methodology has the potential to solve the vergence-accommodation conflict and reduces the need to track accommodation state in future eye-tracking technology. To clarify, our display needs eye-tracking in the sense that the *pupil position* must be tracked, but the *gaze direction* and *accommodation state* of the pupils need not be tracked.

3.2.2 Accommodation supporting NEDs

3.2.2.1 Multifocal near-eye displays

Multifocal near-eye displays, first proposed by Akeley et al. (2004), display a small number of images at different depths; the images are perceived additively Akeley et al. (2004); MacKenzie et al. (2010); Liu et al. (2010); Love et al. (2009); Hu and Hua (2015). In Akeley et al. (2004) and MacKenzie et al. (2010), subregions of an LCD panel were mapped to different focal planes using beamsplitters. Liu and Hua (2009), Love et al. (2009), and Liu et al. (2010) propose a switchable lens to multiplex between the multiple focal planes. Wang et al. (2018) propose a segmented lens and a fast optical shutter to create the focal planes. Hu and Hua (2014a,b, 2015) propose to use high-speed optical components, such as a DMD and a 1KHz deformable membrane mirror, to achieve a larger number of focal planes (six) than previously demonstrated.

Because a relatively small number of depth planes are used to represent objects occupying a large volume, multifocal plane displays need scene decomposition algorithms to optimally represent a 3-D scene using a few 2-D image planes. Content generated by these scene decomposition algorithms provide synthetic focus cues to represent objects that lie in between the focal planes. MacKenzie et al. (2010) propose a per-pixel linear blending approach. Narain et al. (2015) propose an optimized blending algorithm that can demonstrate occlusion, reflection, and non-Lambertian effects. Mercier et al. (2017) and Lee et al.

(2018a) propose a new scene decomposition techniques that are tolerant to eye movements. While scene decomposition algorithms help to depict imagery that lie between the focal planes, the spatial frequency of the fused image is inversely related to the focal plane separation Hu and Hua (2014a); Hua (2017).

Similar to multifocal displays, our display can also be thought of as a view-dependent and depth-fused multifocal display. Our display has about two orders of magnitude more focal planes than previous multifocal displays which approaches a *volumetric display's* performance. Like previous multifocal displays, our display also requires eye-tracking to provide correct occlusion and dis-occlusion effects. In this chapter, we assume that the pupil position is known. Like previous multifocal displays, we also share the problem of generating synthetic focus cues through scene decomposition to represent a large 3-D scene with 2-D image planes. However, while previous methods perform the scene decomposition in an image-oriented manner, we perform the scene decomposition in a voxel-oriented manner. This is discussed in detail in Section 3.5.

Matsuda et al. (2017) propose a multifocal display whose focal surfaces can acquire non-planar, scene-dependent surface geometry. Matsuda et al. (2017) propose a rendering pipeline that converts a 3-D scene to multiple piecewise smooth 2-D surface representations that are displayed in a time-multiplex manner. In comparison with their work, our rendering pipeline generates a single 2-D surface representation of the 3-D scene, and our display does not require piecewise smooth 2-D surfaces. Our display also exhibits more uniform image quality throughout the displayed volume.

Recently, Lee et al. (2018c,b) propose a multifocal plane display that uses synchronized DMD, LCD panel, and focus-tunable lens. With the exception of their LCD panel and our HDR LEDs, the hardware and operation seem similar to our display. But, because of their use of LCD panel and our use of HDR LEDs, the rendering pipelines of the two displays are different. In their display, during the focus-tunable lens' cycle, the DMD panel is used to illuminate portions of the LCD panel resulting in color sub-images at various depths. In our display, during the focus-tunable lens' cycle, the HDR LEDs and DMD create a series of single-color binary images that integrate together such that a color volume is perceived.

3.2.2.2 Light field near-eye displays

Light field displays synthesize the individual light rays that recreate the 3-D scene and can conceptually provide accurate focus cues and monocular occlusion. However, current implementations of light

field displays are limited by diffraction effects Maimone et al. (2014); Huang et al. (2015) or have poor resolution due to a spatial-angular resolution trade-off Lanman and Luebke (2013); Hua and Javidi (2014). While light field displays present a virtual pixel by displaying the light rays originating from the virtual pixel individually, our volumetric NED displays the entire set of light rays that originate from the virtual pixel simultaneously.

3.2.2.3 Holographic near-eye displays

Holographic displays precisely modulate the wave function of the image arriving at the pupil using a digital hologram displayed on a phase-only spatial light modulator (SLM) such as a phase-only *liquid crystal on silicon* (LCoS) panel. Conceptually, these displays can also provide accurate focus cues, monocular occlusion, vision correction, and non-Lambertian effects. Current implementations of holographic near-eye displays have a very small eyebox Maimone et al. (2017), and are computationally expensive Shi et al. (2017); Maimone et al. (2017); Matsuda et al. (2017). Our NED also has a small eyebox (4mm) and is moderately computationally intensive. Our NED's eyebox can be larger; the limiting factor for our eyebox is the focus-tunable lens's aperture (1cm). Maimone et al. (2017) demonstrate a NED that can provide per-pixel focus cues for a range of 10 - 32.5 cm. In comparison, our NED provides per-pixel near-accurate focus cues for a large depth range (15 - 400 cm).

3.2.2.4 Varifocal near-eye displays

Varifocal near-eye displays have a single image plane where the vergence and focus cues match, and this plane is moved by using focus-tunable lenses Padmanaban et al. (2017); Liu et al. (2008); Konrad et al. (2016); Xia et al. (2019), or deformable membrane mirrors Dunn et al. (2017), or by actuating fixed-focus optical components Akşit et al. (2017), or using interchangeable optical components Rathinavel et al. (2018a); Akşit et al. (2019). In a varifocal display, all pixels are at the same focal plane - so virtual pixels that do not lie on the plane of focus need to be synthetically blurred in proportion to their distance from the plane of focus. Varifocal displays need to track the accommodation state of the pupil Padmanaban et al. (2017) or assume that the pupils are accommodated to the eye convergence distance Dunn et al. (2017); Akşit et al. (2017).

3.2.3 Rendering pipeline for DMD-based NEDs

Previous NEDs have used DMDs and proposed different rendering pipelines Lincoln et al. (2016, 2017); Hu and Hua (2014a, 2015). We build upon their hardware but propose a new rendering pipeline. A detailed discussion is provided in Section 3.5.1.

3.3 System Overview

Figure 3.1 shows an overview of our NED's hardware and operation. Our proposed display consists of three main active optical components, namely: (1) an HDR Illuminator, (2) a DMD chip, and (3) a focus-tunable lens. These three optical components are driven at high-speed by an *field programmable gate array* (FPGA), a microcontroller, and custom electronics.

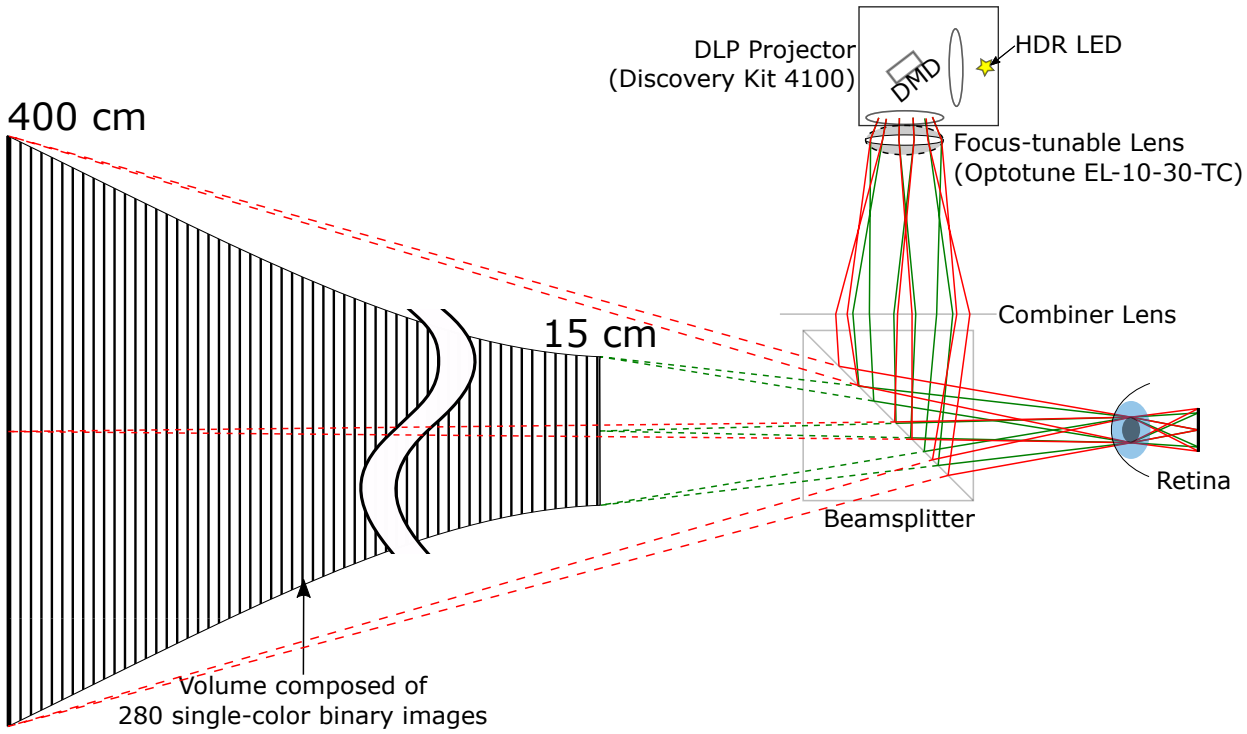

Figure 3.1: Figure shows an overview of the hardware and operation of our NED. The NED is composed of a high-speed HDR LED, high-speed projector, focus-tunable lens, and other common optical components. The NED's optics, rendering pipeline, and the synchronized operation of its active components (HDR LEDs, DMD, focus-tunable lens) work together to present a color volume spanning 15 cm (6.7 diopters) to 400 cm (0.25 diopters).

The focus-tunable lens is driven in a continuous mode such that its optical power follows a triangular or sinusoidal waveform. The DMD projector is synchronized with the focus-tunable lens to display a stack

of binary frames in each lens cycle, and the HDR illuminator illuminates the DMD chip with a distinct selected RGB color for each binary frame. Each cycle of the focus-tunable lens is one frame of the overall display. To avoid confusion, each frame of the DMD will be referred to as *single-color binary image*, whereas the 24-bit color rendering of the 3D scene will be referred to as the color image.

Our DMD's refresh rate is $f_{\text{DMD}} = 16,800$ Hz, and our target display refresh rate is $f_{\text{NED}} = 60$ Hz. Then the number of single-color binary images displayed by the DMD in each frametime of the NED is given by

$$N_b = \frac{f_{\text{DMD}}}{f_{\text{NED}}} = 280. \tag{3.1}$$

These 280 single-color binary images are distributed in optical depth along the user's line-of-sight from 15cm to 4M. Correct modeling of the depth distribution and field of view (FoV) of binary images is necessary for proper rendering and color decomposition.

The optical design of our NED is discussed in Section 3.4, and the rendering pipeline that converts 3D scene information into multiple single-color binary images is discussed in Section 3.5.

3.4 Optical Design

This section models the optical design and timing characteristics of our near-eye volumetric display to arrive at the geometry of the displayed volume, i.e., depth distribution and FoV of the binary images. The geometry of the volume is used in the rendering pipeline to decompose a 3D scene to a volume that is displayed by the NED.

3.4.1 Overview of optical design

Our optical system is composed of multiple lenses (see Figure 3.2). The left diagram of Figure 3.2 shows the image formation process for any projector. Such a projector can be converted to a near-eye display by placing an eyepiece or combiner lens just after the projected image. Since the projected image for most off-the-shelf projectors would be too large, a converging lens could be placed between the projector and the combiner lens; this helps in reducing the magnification of the projected image and in reducing the form-factor of the NED. In our NED, instead of placing a static converging lens between the projector and the combiner lens, we place a focus-tunable lens (see right diagram of Figure 3.2) and configure the optical

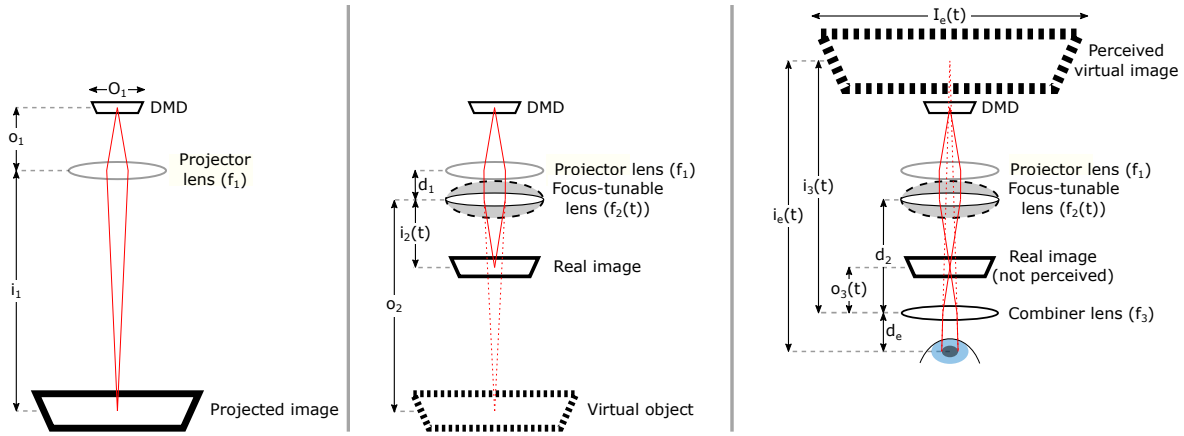

Figure 3.2: Our NED's optics can be analyzed in three stages. Figure shows the unfolded optics and ray diagram for each stage. *Left:* Image formation for the DMD projector using manufacture-provided projection optics. *Middle:* Adding a focus-tunable lens at the exit pupil of the DMD projector causes the real image of the DMD to be formed closer; Configuring the focus-tunable lens power to continuously oscillate causes the real image of the DMD to also oscillate. *Right:* A combiner lens finally creates a virtual image of the DMD that can be seen by the eye.

power of the focus-tunable lens to sweep the real image of the DMD close to the combiner lens. To see a virtual image, the combiner lens's focal length has to be less than the distance between the lens and the real image (i.e., $f_3 < o_3(t)$).

3.4.2 Modeling of optical design to derive volume geometry

We begin with stating the Gaussian thin-lens equation

$$\frac{1}{f} = \frac{1}{o} + \frac{1}{i}, \tag{3.2}$$

and associated equations

$$i = \frac{fo}{o - f}, \quad M = \frac{I}{O} = -\frac{i}{o}, \tag{3.3}$$

where f denotes focal lens of a thin lens, o denotes object distance, i denotes image distance, O denotes the object size, I denotes image size, and M denotes the magnification of the lens.

Due to the presence of multiple lenses in the optical stack (see Figure 3.2), we analyze the image formation of each lens separately and consider the image formed by each lens as the object for the next

lens. This gives the following geometric relations:

$$o_2 = i_1 - d_1, \quad o_3(t) = d_2 - i_2(t), \quad i_e(t) = i_3(t) + d_e. \tag{3.4}$$

The relationship between the distance from the DMD to the projection lens (o_1), focal length of the projection lens f_1, and the projected image distance (i_1) is given by

$$i_1 = \frac{f_1 o_1}{o_1 - f_1}. \tag{3.5}$$

The relationship between the object distance $o_2 = i_1 - d_1$, focal length ($f_2(t)$), and image distance ($i_2(t)$) for the focus-tunable lens is given by

$$i_2(t) = \frac{f_2(t) o_2}{o_2 - f_2(t)} = \frac{f_2(t)(i_1 - d_1)}{(i_1 - d_1) - f_2(t)}. \tag{3.6}$$

The optical power of the focus-tunable lens ($f_2(t)$) can be configured to maintain a constant value or follow a time-varying square, triangular, or sinusoidal waveform. Other waveforms may be possible with custom electronics, but for this chapter, we analyze only the triangular and sinusoidal waveforms of lens power.

To define optical power of the focus-tunable lens as a function of time, we define some standard parameters for time-varying signals: *DC bias*, *amplitude*, and *half-time period*. Let the DC bias, which is the average value of the signal over one full-time period be denoted by D. Let the amplitude, which is half of the peak-to-peak value, be denoted by A. Note that each cycle of the lens is the frametime of the NED. Hence, the frequency of the lens is equal to the refresh rate of the NED (f_{NED}). Let a denote half a time period ($a = \frac{1}{2f_{\text{NED}}}$). The optical power of the focus-tunable lens, when following a triangular waveform, can be modeled as

$$f_2(t) = D - A\left(\frac{1}{2} - \left|\frac{t-a}{a}\right|\right), \tag{3.7}$$

and when following a sinusoidal waveform, it can be modeled as

$$f_2(t) = D + A\text{sin}(2\pi f_{\text{NED}} t). \tag{3.8}$$

And finally, the relationship between the object distance ($o_3(t)$), focal length (f_3), and image distance ($i_3(t)$) for the combiner lens is given by

$$i_3(t) = \frac{f_3 o_3(t)}{o_3(t) - f_3} = \frac{f_3(i_2(t) + d_2)}{(i_2(t) + d_2) - f_3}. \tag{3.9}$$

The above equations (3.4) to (3.9) are sufficient to calculate the depth of each binary image plane. The FoV of the virtual binary images is found by repeated application of the magnification formula from Equation (3.2):

$$M = M_1 M_2 M_3, \tag{3.10}$$

$$I_e = M O_1, \tag{3.11}$$

$$\theta_{\text{FoV}} = 2\tan^{-1}\left(\frac{I_e}{2i_e}\right). \tag{3.12}$$

In our system, these are the values for the known quantities: $f_{\text{DMD}} = 16,800$ Hz, $f_{\text{NED}} = 60$ Hz, $f_1 = 2.96$cm, $o_1 = 3$cm, $O_1 = 1.778$cm (diagonal size of the DMD module), $d_1 = 3$cm, $f_3 = 6$cm, $d_2 = 12$cm, $d_e = 3$cm, $D = 14$, $A = 4$, and $a = 8.33$ ms. Equations (3.1) to (3.12) are evaluated with the above values to calculate the geometry of the displayed volume. The geometry of the volume is graphed in Figure 3.3.

The above formulation and graphs in Figure 3.3 shows only 140 unique depth planes over the time period because depth values in the first half of the time period are repeated in the second half of the time period. Our implementation is slightly different from this - we apply a small phase difference to equations (3.7) to (3.8) to get 280 unique depth values.

3.4.3 Sinusoidal vs. triangular waveforms

In optical imaging systems, including the human eye, the blur of an object that is defocused is directly proportional to the difference between the actual distance of focus and the distance of the object in units of diopter. In our NED, the depth distribution of images should ideally be dioptrically equidistant from each other. From Figure 3.3, it can be seen that the lens power following a triangular waveform results

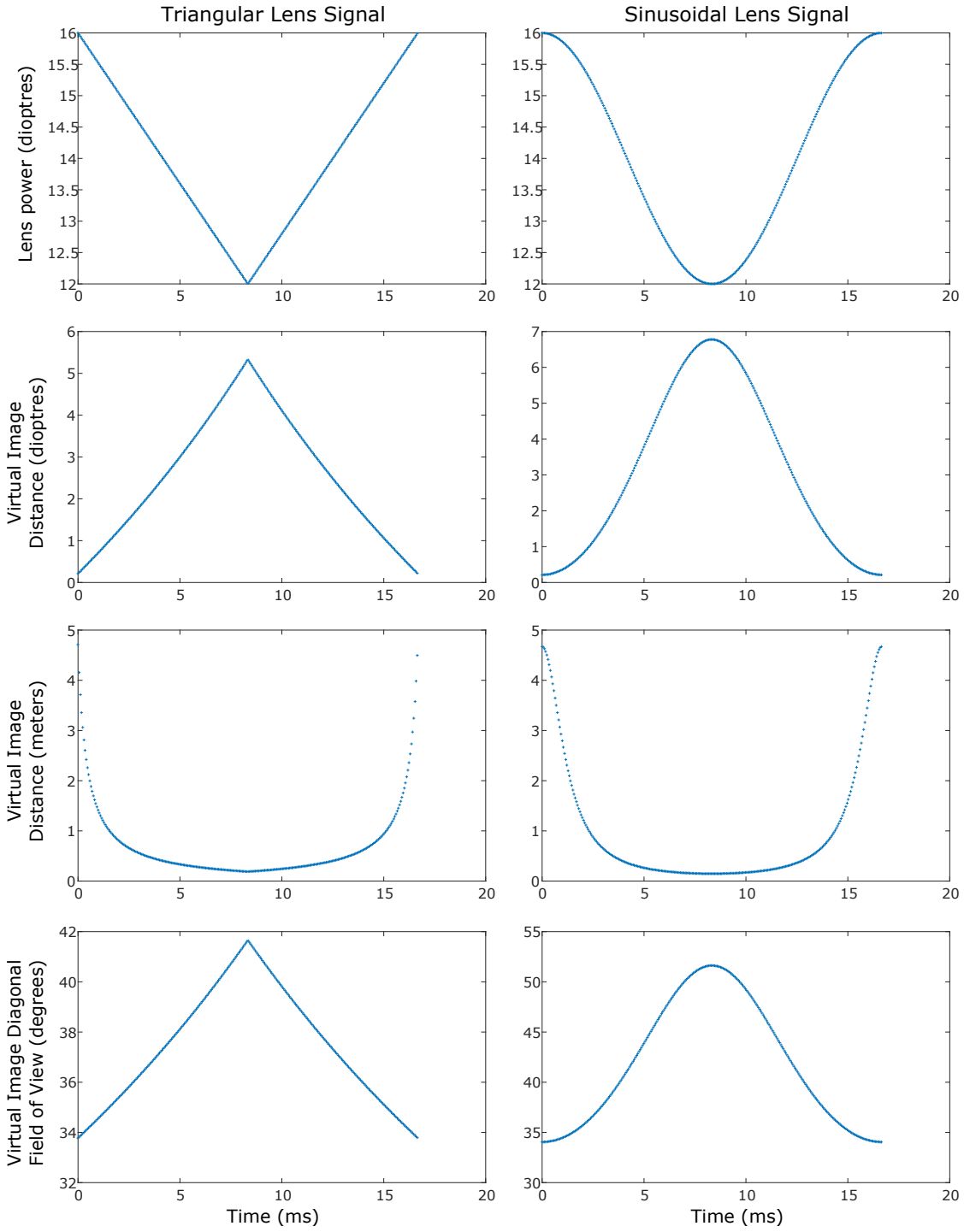

Figure 3.3: Graphs modeling the depth distribution and FoV of the displayed single-color binary images that compose the volume formed by synchronizing the DMD projector and a continuously oscillating focus-tunable lens. The oscillating lens's optical power can follow a triangular waveform *(Left column)* or a sinusoidal waveform *(Right column)*. Data presented in these graphs are used in the rendering pipeline to convert 3-D scene information to multiple single-color binary images that are displayed by the NED. Equations used to generate these graphs are described in Section 3.4.

in a near-linear and equidistant distribution of virtual image planes in dioptric space. Hence, we implemented the rendering pipeline and electronic synchronization assuming that the lens sweeps a triangular waveform. However, when we used the sinusoidal waveform in place of a triangular waveform, keeping everything else such as the color decomposition, and electronic synchronization the same, we didn't notice a significant difference in the displayed volume geometry and image quality. This may be either because the difference between the triangular and sinusoidal waveforms is negligible compared to the minimum dioptric difference required to make a perceptual difference or because the lens' triangular and sinusoidal waveforms are similar, which can often happen with physical systems due to inertia/friction, etc., especially at higher frequencies. It may be possible to interface a closed-loop feedback system to monitor the lens' focal length and adjust for deviations from the desired waveform, but we did not implement such a system.

3.5 Rendering Pipeline

In this section, we first discuss the rendering pipelines of previous DMD-based NEDs, then describe our full rendering pipeline from graphics primitives to single-color binary images, and finally discuss the benefits and limitations of our rendering pipeline.

3.5.1 Rendering pipeline for previous DMD-based NEDs

3.5.1.1 Low latency and HDR NEDs

Most display technologies that employ a DMD also use a constant intensity or bivalent illumination source and use pulse train modulation to create grayscale or color imagery Lincoln et al. (2016). Recently, Lincoln et al. (2017) demonstrated a DMD-based display system which used a controllable high-speed HDR illuminator. They demonstrated that the intensity and color of the illumination could be changed over a wide range on a per-binary frame basis. They also proposed a new color-to-binary decomposition method, which they call *Direct Digital Synthesis (DDS)*. Let d be the desired color intensity value, g be the generated color intensity value, and s be the step-index of the binary representation of the value of d. Then, DDS decomposition from color to binary values can be represented per color channel, as shown below:

$$g = \sum_{s=0}^{n-1} \left(2^s \times \text{bit}(d, s)\right). \tag{3.13}$$

3.5.1.2 Multifocal plane NEDs

Previous DMD-based multi-focal plane displays Hu and Hua (2014a, 2015) decomposed a 3D scene to a stack of *color images* fixed at the various depths. In these approaches, the focus-tunable lens or deformable membrane mirrors would step through a set of focal lengths, and at each focal length, *after the lens stabilizes*, a series of binary images was displayed by one of the classical pulse train modulation schemes to generate color imagery. For such color image plane based approaches, we provide equations below for the relationship between the DMD's frame rate (f_{DMD}), number of focal planes (N_{planes}), frame rate of the NED (f_{NED}), and the color depth per color channel (N_{gray}). Up to some extent, a multifocal NED with a larger number of N_{planes} can present better imagery because the scene decomposition algorithms of depth fused multifocal displays trade-off the spatial frequency of the fused image and the focal plane separation Hu and Hua (2014a); Hua (2017).

In case of classical pulse train modulation schemes, the number of focal planes is given by

$$N_{\text{planes}} = \frac{f_{\text{DMD}}}{3 f_{\text{NED}} (2^{N_{\text{gray}}} - 1)}. \tag{3.14}$$

Using DDS decomposition, N_{planes} can be increased significantly as shown by the following equation

$$N_{\text{planes}} = \frac{f_{\text{DMD}}}{3 f_{\text{NED}} N_{\text{gray}}}. \tag{3.15}$$

A DMD-based multifocal display which decomposes 3D scene information to color image planes which are in turn decomposed from color images to binary images based on DDS decomposition has not been demonstrated. If it were demonstrated with our hardware ($f_{\text{DMD}} = 16,800$ Hz, $N_{\text{gray}} = 8$, $f_{\text{NED}} = 60$ Hz), we would achieve $N_{\text{planes}} = 11$ color image planes. However, we propose a further improvement below based on voxel-oriented decomposition rather than image-oriented decomposition.

3.5.2 An overview of our rendering pipeline

The pipeline currently handles only opaque polygons; transparency and other primitives are left to future work. Our rendering pipeline is composed of two steps: (1) *voxelization*, i.e., the process of converting 3D polygonal data to a 2D surface composed of color voxels (3D equivalent of pixels) that best approximates 3D polygonal data; and (2) *decomposition* of the color voxels into a series of binary images and corresponding illumination values; these data are used by the display to present a series of single-color binary images to the viewer.

3.5.3 Voxelization: Graphics primitives to 2D surface

Using 3D models and scene data, an OpenGL renderer generates an RGB image and a linearized 16-bit depth map of the current scene at the resolution of the DMD display (1024×768). The 16-bit values of the depth map are remapped to the 280 depth values of the focal planes supported by our optical design. This results in a 2D surface, composed of color voxels, in a $1024 \times 768 \times 280$ volume. By *2D surface* we mean a surface defined in 3D space with a bijective mapping to an image plane (the RGB image or the depth map). The view of the surface from the front of the frustum is isomorphic to the rendered RGB image.

3.5.4 Binary Decomposition: Color voxels to binary images

Our key observation is that the binary representation of a color voxel need not start or end at one of the modulo $3 \times (2^{N_{\text{gray}}} - 1)$ planes as proposed by earlier binary multifocal displays. It need not start or end at one of the modulo $3 \times N_{\text{gray}}$ also, as would be the case for a multifocal NED which displays color image planes using DDS decomposition. Instead, the decomposition of a color voxel to binary voxel can begin and end at arbitrary depths.

When converting from color volume data to binary images, the intensity and color of each color voxel tell us the binary pattern that represents it, and the depth of the color voxels tells us the center around which the binary pattern should be distributed. The binary voxels that encode the color voxel are distributed along the perspective projection lines that pass through the color voxel's location, and the distribution is centered around the color voxel's location. Figure 3.4 provides a visualization of our rendering pipeline. For ease of representation, Figure 3.4 depicts the rendering pipeline for equidistant focal planes

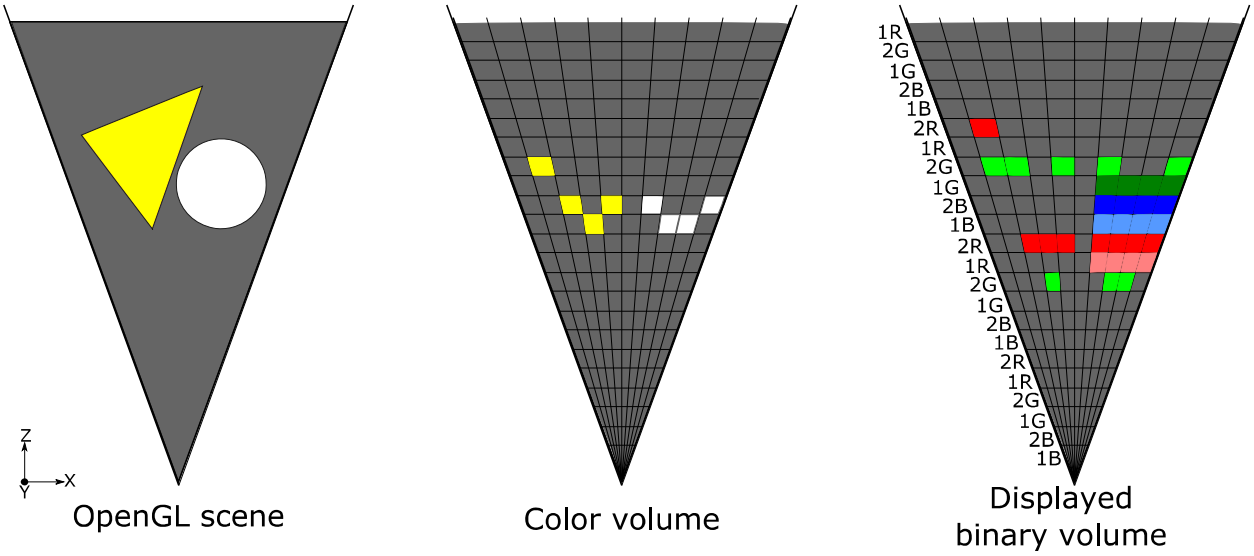

Figure 3.4: Diagram shows the stages of our rendering pipeline: voxelization (see Section 3.5.3) and binary decomposition (see Section 3.5.4). For ease of representation, the figure depicts the rendering pipeline for a simple 2-D graphics and 6 bits-per-pixel imagery. Actual implementation uses 3-D graphics and 24 bits-per-pixel imagery. The numbers along the displayed binary volume's frustum indicate the intensity level and color of the RGB LED that illuminates the current binary image.

and for 2D graphics generating six bits-per-pixel imagery. Our implementation handles 24-bits-per-pixel color imagery.

If this decomposition was implemented in an acyclic manner, the number of unique color voxel depths would be $N_b - (3 \times N_{\text{gray}}) + 1$, which is 257 planes in our case. However, we could implement this decomposition in a cyclic manner, and in this case, the number of unique color voxel depths would be equal to N_b, which is 280 in our case.

Even though we depict in Figure 3.4 that the decomposition happens in a perspectively shaped volume, it can be implemented as a decomposition on a rectangularly shaped volume. This is indeed the case in our implementation. This is not an issue because when the NED displays the single-color binary images, it does a near-inverse perspective transformation.

3.5.5 Display: Binary images to Retinal image

The binary images generated are displayed on a DMD in sync with a focus-tunable lens sweeping a sinusoidal or triangular waveform for the optical power of the lens. The single-color binary images displayed by the NED are integrated by the eye to see a color volume. Displaying the binary images in our prototype display is a near inverse-perspective transformation. It is not a perfect inverse-perspective

transformation due to the slight change in FoV of images seen over the cycle of the lens (see Figure 3.3). This near inverse-perspective transformation allows us to perform the transformation of RGB and depth images to color voxels, and the transformation of color voxels to binary voxels in an orthographic space.

3.5.6 Limitations

3.5.6.1 Depth and Spatial resolution

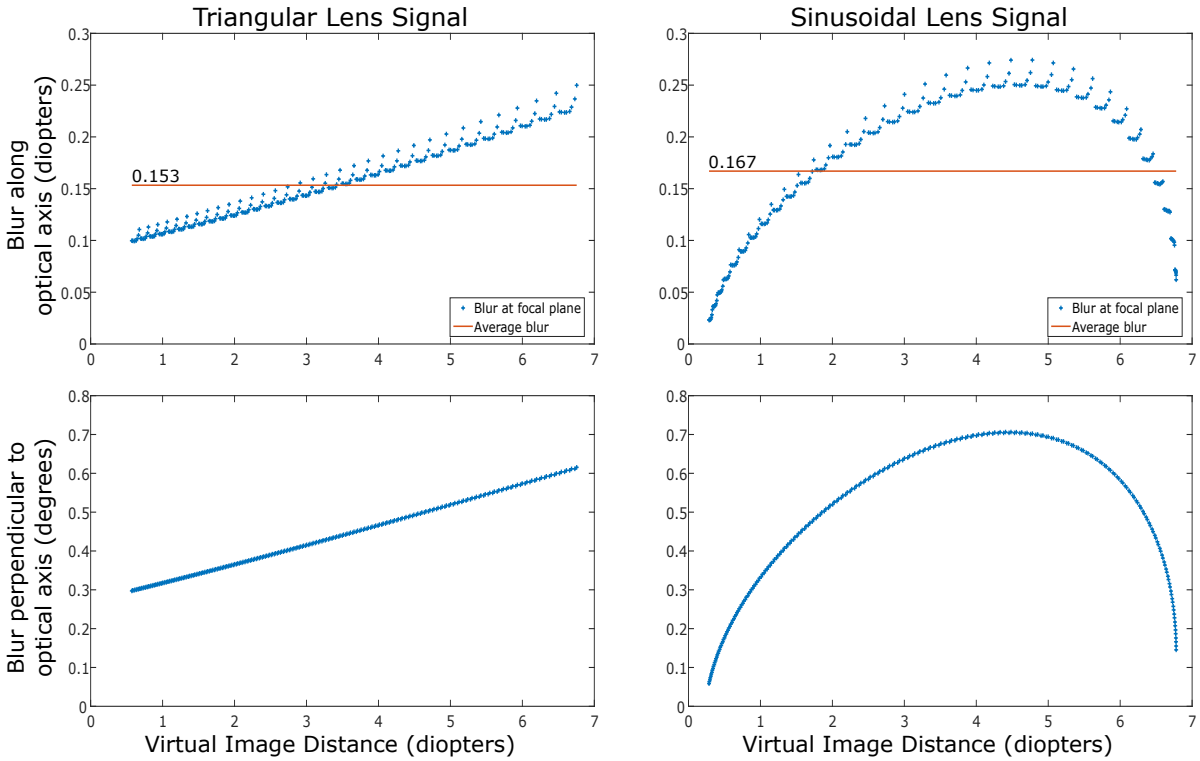

Figure 3.5: *Top row:* Graphs indicate the depth blur for a color voxel at each depth plane and the average depth blur for color voxels of all depth planes. The depth blur arises because the rendering pipeline decomposes each color voxel to multiple single-color binary voxels, which are spread along the perspective projection lines. *Bottom row:* The optics of our NED cause the FoV of the virtual image to slightly change over the lens cycle; this changing FoV is graphed. This creates a blur perpendicular to the optical axis leading to a loss in spatial resolution.

Conceptually, the minimum non-zero separation in depth between two voxels in our display is 1 depth plane which averages to $\frac{6.7 \text{ diopters}}{280 \text{ focal planes}} = 0.024$ diopters. However, because the binary voxels are spread across multiple binary image planes, we should expect to see a blur for the color voxel along the optical axis which could lead to a loss in the depth resolution of the NED. Since each color voxel is represented by multiple binary voxels and the brighter binary voxels are going to be perceived more strongly, we calculate

the depth blur as the *weighted* standard deviation of sorted depth values for a moving window of length $3 \times N_{\text{gray}} = 24$; this is graphed in the top row of Figure 3.5.

Similarly, due to the slightly changing FoV of binary images across the lens cycle (shown in Figure 3.3), we should expect to see a blur perpendicular to the optical axis which could lead to a loss in the spatial resolution of the NED. This blur is minimum for pixels close to the optical axis and maximum for pixels at the periphery. The maximum angular blur perpendicular to the optical axis is calculated as the standard deviation of FoV values for a moving window of length $3 \times N_{\text{gray}} = 24$; this is graphed in the bottom row of Figure 3.5.

The blur perpendicular to the optical axis can be reduced by performing a calibration to determine the actual FoV of each binary image plane and modifying the color to binary decomposition algorithm to take into account the deformed volume geometry. We perform this calibration in Section 3.9. The blur along the optical axis, however, is more fundamental to the display technology. It could be reduced by advanced color volume to binary volume decomposition schemes as presented in Section 3.7.

Previous works have suggested slightly different values for the focal plane separation required for a good multifocal display. Rolland et al. (1999) suggest 0.143 diopters, Akeley et al. (2004) design their prototype with image spacings of 0.67 diopters, Liu and Hua (2010) and Simon J. Watt (2012) suggest 0.6 diopters, and MacKenzie et al. (2010) suggest 1 diopter. As shown in Figure 3.5, our display has a maximum depth-blur of 0.3 diopters and an average depth-blur of 0.167 diopters.

3.5.6.2 Voxel-fighting in a dynamic display implementation

Here we discuss a minor limitation in extending our proposed offline rendering pipeline to a dynamic display. Observe that to decompose a single color voxel for a 24 bits-per-pixel image, we require 24 binary voxels. In the case of a static display and a cyclic implementation of our decomposition algorithm, this means that a color voxel at, say, the 280th focal plane would be decomposed into binary voxels that range from binary image indices 268 to 12. However, in a dynamic display case, we run into the issue that a new frame is received for each display cycle.

If the incoming frame information completely replaces the previous frame information, there could be a loss of brightness and bit-depth for the color voxels in the last few focal planes. Alternatively, if we design the NED to start displaying the new frame information only after it finishes displaying the previous

frame information, the DMD display's cycle would quickly fall out-of-sync with the lens cycle. With a modified rendering pipeline, for which the frame rate of the NED is slightly lower than the frequency of the focus-tunable lens, and very good synchronization of the lens and the DMD, this would not be an issue. Alternatively, we could carry over the information of the last few focal planes of the previous frame to the new frame while giving priority/preference to the new frame's information.

3.6 Static System

This chapter presents two rendering pipelines for our display system: (1) One pipeline enables the display to present static volumes. This pipeline is described in this section in a detailed manner. (2) The other pipeline would enable the display to present interactive and dynamically changing volumes. This is still largely under development and this dissertation presents preliminary results in Section 3.8.

3.6.1 Overview and Software

To test our ideas, we developed a hardware prototype of a monocular near-eye display and implemented an offline version of our proposed rendering pipeline. The offline rendering pipeline begins with the rendering of a virtual scene using OpenGL/GLSL to generate an RGB image and a linearized 16-bit depth map of the virtual scene. The RGB image and depth map are processed in MATLAB to generate a series of binary images and RGB LED brightness values. The binary images are uploaded to and displayed by the DMD controller, and the RGB LED brightness values are used by a custom RGB LED driver for precise high-speed control over each LED's brightness. An ARM-based microcontroller provides synchronization between the lens, the DMD controller, and the custom RGB LED driver. Below we discuss each hardware component in detail.

3.6.2 Hardware

Focus-tunable Lens The focus-tunable lens used is the Optotune EL-10-30-TC-VIS. The optical power of the lens is controlled via a manufacturer-provided software and a USB-connected lens driver. The optical power can be set to a static value, or it can be set to follow a rectangular, sinusoidal, or triangular signal for a wide range of frequencies (0.25 Hz to 2000 Hz). For this chapter, all experiments were conducted with

the optical power of the lens configured to follow a triangular signal of 60 Hz frequency, and the maximum and minimum lens powers of the triangular signal were approximately $50m^{-1}$ and $120m^{-1}$.

Optics Other than the focus-tunable lens, we use the manufacturer-provided optical engine of the TI Discovery 4100 Kit (STAR-07 optical module), a Fresnel lens (60mm focal length), and a beamsplitter that allows the display to optically integrate the real world view and the imagery of the virtual scene.

DMD controller The DMD controller we use is the Texas Instruments (TI) Digital Light Processing (DLP) Discovery 4100 Kit which drives an XGA (1024 x 768) DMD module. The display system is capable of displaying binary images at up to 17241 Hz which would allow 287 binary images to be displayed in each lens cycle. This would need precise synchronization between the lens signal and the DMD controller, which is not afforded by the current implementation. For a more robust system, we display 280 images in each lens cycle and design the system such that the 280 images are guaranteed to be displayed slightly before the beginning of the next lens cycle.

Custom RGB LED Illuminator A PCB mounted RGB LED is controlled using electronics consisting of Digital Analog Converters (DACs), Op-Amps, etc. The board listens for three 16-bit binary codes over Serial Peripheral Interface (SPI) protocol over three parallel buses and sets each color LED to the brightness level corresponding to the received code. The board is capable of illuminating the DMD with a wide range of brightnesses and color combinations. 2^{16} levels of intensity are possible for each color LED, and all color LEDs can be driven in parallel. The full-scale rise and fall times of each channel are approximately 500ns; every binary frame can be illuminated at a distinct intensity and color mix. This RGB LED illuminator is the same as the one used in Lincoln et al. (2017); please refer to that chapter for more details.

PC A PC using Intel Xeon E5-2630 2.4 GHz processor with an NVIDIA GeForce GTX 980 running Windows 7 is used to implement an offline version of the proposed rendering pipeline.

3.6.3 Operational detail

Figure 3.6 gives an overview of the operation of the NED. The binary images are uploaded to the DMD controller using the ALP 4.1 Controller Suite. The DMD controller is configured to advance frames each time it receives a trigger signal from the microcontroller. At the end of the sequence of images, the

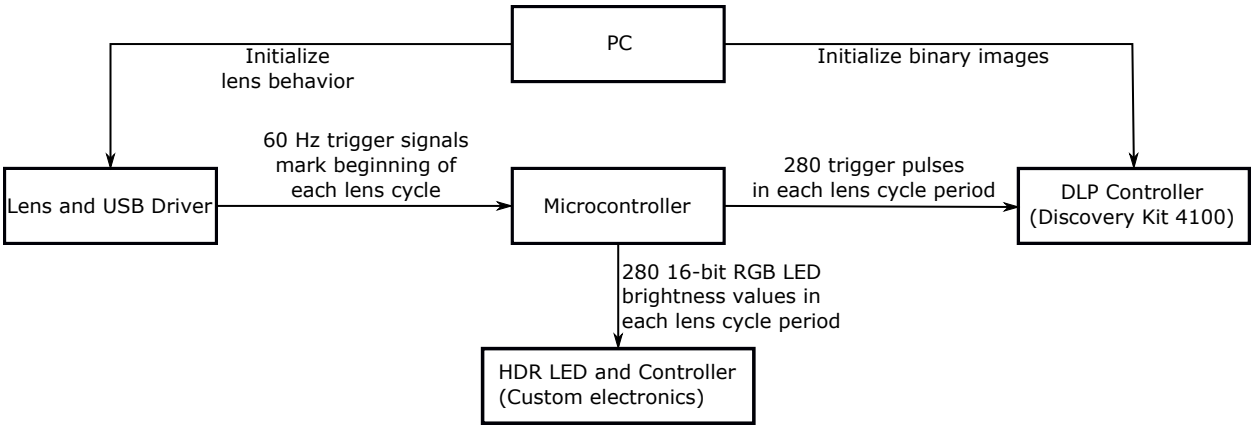

Figure 3.6: Diagram shows the various hardware components and their timing relations to each other in the display's operational state.

DMD cycles back to the first image. The frametime of the DMD was set to the minimum possible frametime of 58 μs.

The lens controller outputs a trigger signal whose rising edges correspond to the beginning of each lens's cycle. The lens operates at a frequency of 60 Hz. The lens' trigger signal is detected by the microcontroller, which then performs 280 instances of these operations before the next lens' trigger signal: (1) microcontroller outputs a trigger signal to the DMD controller, and (2) microcontroller sends three 16-bit words to the LED controller. Each 16-bit word specifies the brightness of a color LED. The microcontroller ensures that the DMD updates and illumination values are phase-locked to the lens cycle. Figure 3.7 shows an image of our display's hardware and the experimental setup.

3.6.3.1 Calibrating phase delay

In our experience, we've found that there is a phase delay between the lens signal and the displayed image plane depth estimated in Figure 3.3. This phase delay was calibrated visually by generating a synthetic stack of images in which each image has a single feature (like a cross-hair), but the feature is placed at a different location in each image. By setting the camera lens to nearest focus, it was visually determined that the 180th image out of the 280 image stack is in focus, which meant that the lens trigger signal and our system's display of the virtual images are out of phase by $\frac{180-140}{280} \times 16.67ms = 2.38ms$. To correct for this, the binary images uploaded to the DMD controller were cyclically rearranged such that the 180th binary image is moved to the 140th index.

3.6.4 Future implementation improvements

The results can be visually improved by performing white-balance correction, gamma curve calibration, and calibrating for the non-uniform frustum as shown in the last row of Figure 3.3. In our experience, we didn't find the change in FoV to reduce the image quality significantly, but it does make long straight objects slightly curved, especially when the straight objects are places towards the periphery of the display.

3.6.5 Results

Cameras To record images and videos of the see-through view of the display, cameras that approximate the human eye were placed behind the display (see Figure 3.7) at a distance that approximated the eye relief of a human viewer (2cm away from the beamsplitter). See-through image results presented in this chapter were recorded using a Canon T6i Rebel camera with a Canon 24-70mm f/2.8 lens. See-through video results (in accompanying video submission) were recorded using a Point Grey Chameleon3 camera with a Fujinon 2.8-8mm f/1.2 lens. A 4mm aperture was used in both cameras to emulate the human pupil diameter while collecting results. When using PointGrey cameras, the nearest distance was chosen to be 15cm (6.7 diopters), and when using the DSLR camera, the minimum distance was 20cm (5 diopters) because the lens could not focus closer.

Our display is capable of presenting virtual imagery closer than 15cm and farther than 4M, but we were constrained by the recording camera (for the minimum distance) and by the lab space (for the farthest distance). Although virtual images closer or farther than what is demonstrated here may not be required for near-eye displays, this may be of use in some other application.

Setup Figure 3.7 shows the monocular display prototype, the positioning of the camera in place of a viewer's eye and the staged real-world scene consisting of a large poster, and smaller objects such as a Rubik's cube, a wristwatch and a tiny rubber ducky (2cm height). The real-world objects are arranged from small to large progressively away from the display approximately along the line of sight of the see-through view. Virtual objects are scaled progressively from small to large away from the display for the virtual objects to subtend approximately the same angle at the camera.

See-through images Using the color-to-binary decomposition algorithm described in Section 3.5.4, we can display full-color (24 bit-depth) volume, such as shown in Figure 3.9. This image shows the see-

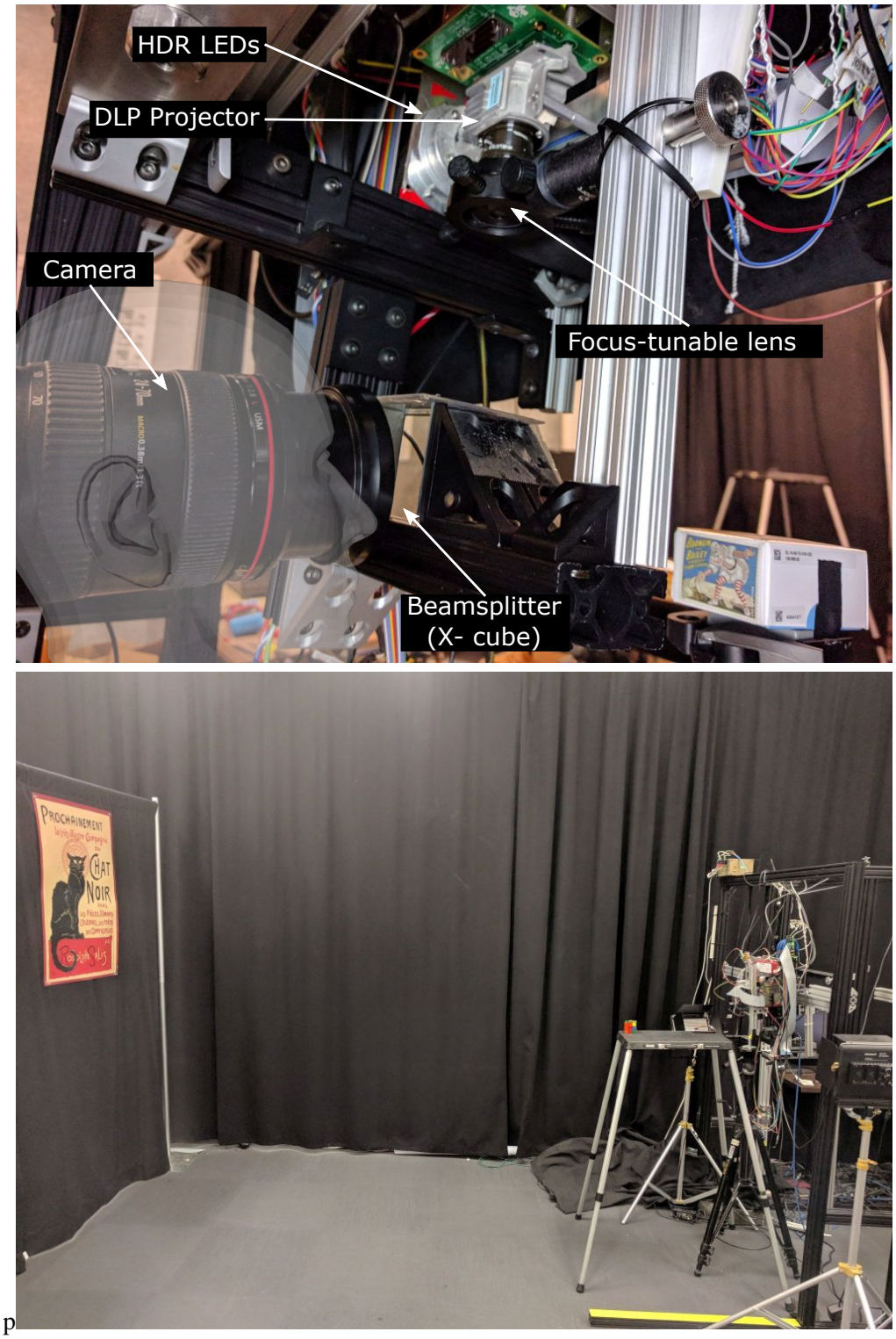

Figure 3.7: *Top:* Our prototype display. *Bottom:* The staged real-world scene used to collect all see-through images and videos. Multiple objects (a tiny rubber ducky (2cm height), a wristwatch, a Rubik's cube, and a wall poster) are arranged progressively from near to far. Virtual objects are rendered in this staged real-world scene such that each virtual object is located at the same depth as one of the real-world objects (See Figure 3.9).

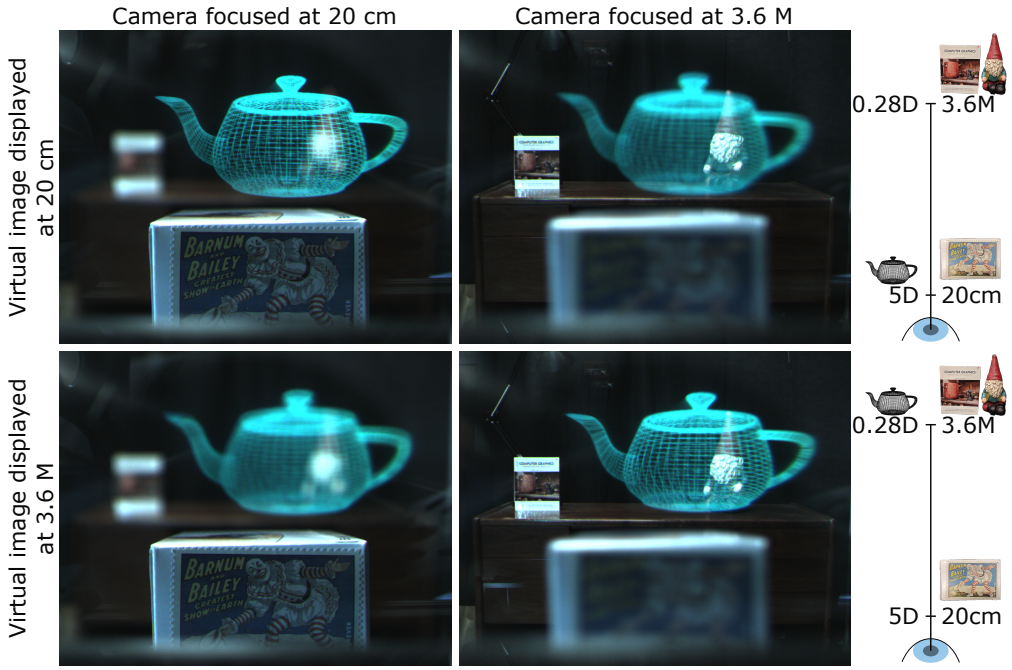

Figure 3.8: View through our near-eye display when only one out of the 280 binary image planes is encoded with a binary image. This figure gives an idea of how each binary image is perceived by an eye or a camera. When all binary images are encoded with appropriate content, a time-integrated color volume occupying a large depth range can be seen (see Figure 3.9).

through views of our NED when displaying a virtual scene registered to the staged real-world scene. Even though our RGB LED illuminator can produce high-dynamic range and consequentially very bright virtual imagery Lincoln et al. (2017), we're currently displaying moderately bright imagery at 24 bits-per-pixel. A black background screen is used to improve the visibility and contrast of the virtual objects.

Figure 3.8 shows the see-through view when only one of the 280 binary depth planes is encoded with the image of a wire model of a teapot. As discussed earlier, the field of view of the virtual image changes during the lens cycle. While an ideal system would calibrate for this effect, we found the resulting degradation minor in practice and did not perform this calibration for collecting this set of results. However, we do explore how to do this calibration in Section 3.9. Other first-order and second-order optical aberrations present in all optical systems may also need to be calibrated. Each of these optical aberrations likely varies with depth across the volume. Optical aberrations are observable in our system, e.g., in the second row of Figure 3.9, even when focused at the correct depth, the bottom portion of the Jack card is blurred relative to the top.

Figure 3.10 shows a comparison between the see-through views when the lens signal follows a triangular and a sinusoidal waveform. This is discussed in Section 3.4.3.

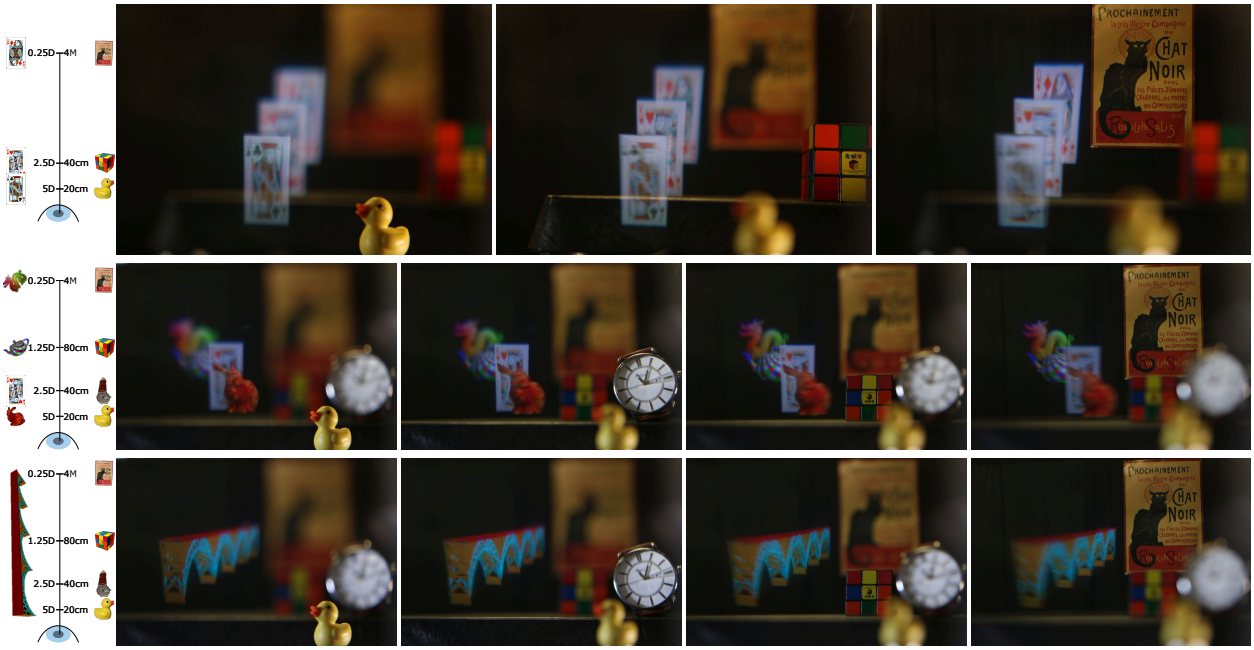

Figure 3.9: View through our volumetric near-eye display where virtual objects are placed among real objects at a range of distances. *Extreme left:* Overhead depiction of scene geometry. Icons to the left of the optical axis correspond to virtual objects, while icons to the right of the optical axis correspond to real objects. *Other images, left to right, in each row:* Photos taken through the display where the focus of the camera is adjusted progressively from near to far. In each row, the only difference between the see-through views is the camera's focus settings - this demonstrates the ability of the display to provide proper focus cues for all virtual pixels simultaneously, allowing the viewer to freely accommodate in the scene without any feedback to the display.

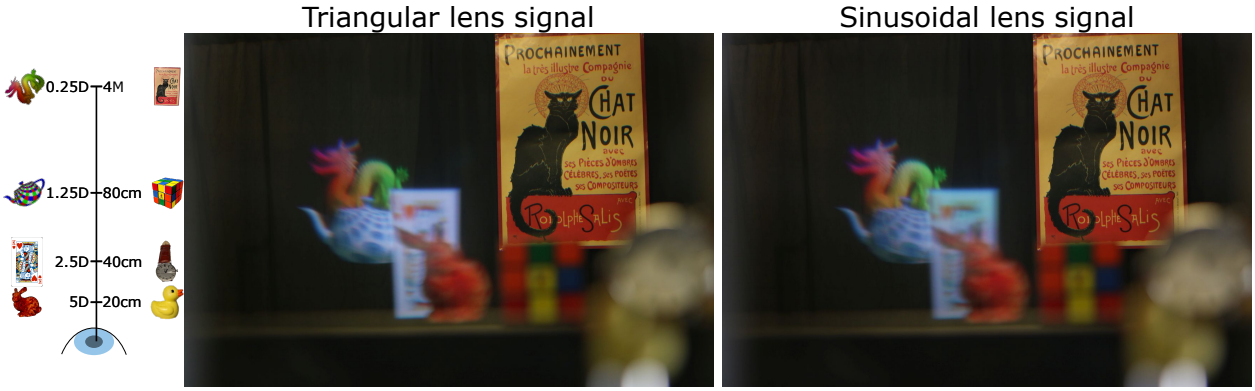

Figure 3.10: Images show the see-through view through the display when the optical power of the focus-tunable lens follows a triangular waveform and a sinusoidal waveform. For both these images, the voxelization and color volume to binary volume decomposition was performed assuming a triangular waveform. We don't observe a significant difference in the see-through images.

Video results A video was recorded as a demonstration of this display. This video was submitted as part of this dissertation and is also available publically at this URL: `https://www.youtube.com/watch?v=oDcOQ_NotRU`. The video results show a larger range of depth (15cm - 4M) compared to the

image results (20cm - 4M) because the camera used to record the video was able to focus closer. In these videos, a flicker is seen propagating back and forth through the displayed volume. This flicker is an artifact of the video capture and is not human-visible. The flicker arises because of the slight discrepancy between the display frame time (16.67 ms) and the minimum shutter speed possible on the camera (16.74 ms). The flicker moves back and forth in the volume because the camera samples the whole volume once and a small portion of the volume twice - and because of this, it starts to sample the volume in the subsequent frame from a slightly different starting position of the volume.

3.7 Adaptive Color-to-Binary Decomposition Algorithms

In this section, we present methods for more efficient decomposition from color-volume to binary-volume, efficiency here referring to the number of binary depth planes that are used to represent a color depth plane.

3.7.1 Motivation

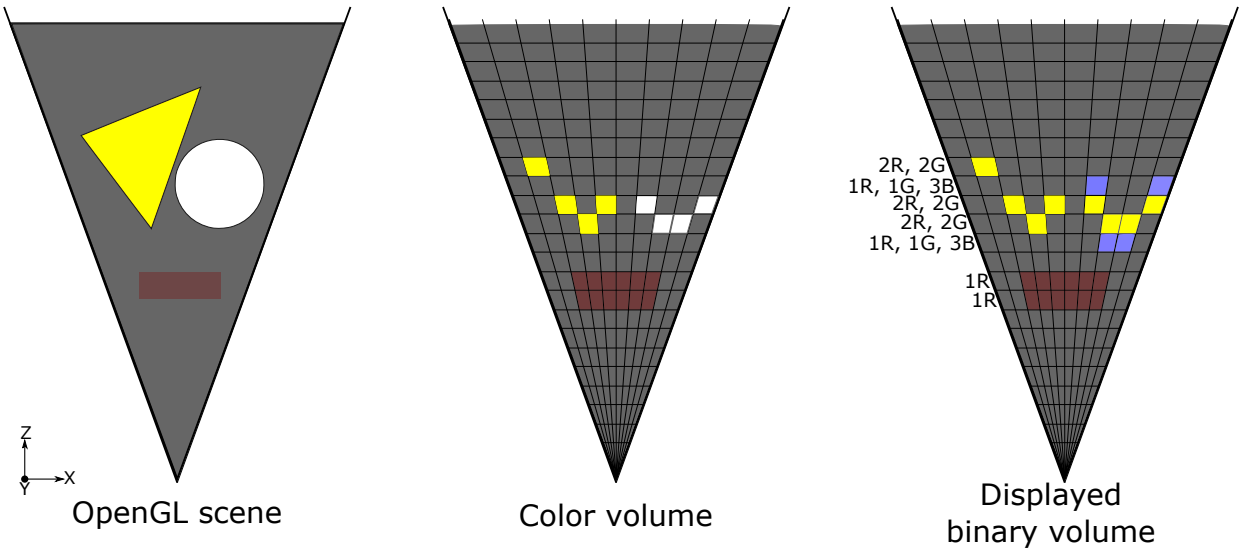

Figure 3.11: Figure shows a concept diagram for adaptive color-to-binary decomposition. Note how the LED values written on the left side of the binary volume in this figure do not follow a repeating pattern as shown in Fig. 3.4

In the method presented in Sec. 3.5.4, each color voxel was decomposed into the nearest 24 binary depth planes. Since our display's LEDs can change color and intensity over a very large range on a per-binary frame basis, we don't need to limit our display to a fixed pattern of LED colors and intensities or

to 8 bits-per-color. A more optimal decomposition method might use a fewer number of binary voxels to represent the same color voxel. Fig. 3.11 shows a conceptual diagram of how the resulting binary volume of such adaptive color-to-binary decomposition algorithms may look. It is useful to reduce the number of binary voxels that represent each color voxel for the following reasons:

1. This will reduce the depth-blur associated with each color voxel.
2. This may be useful for more compact prototypes because the more compact DMD projectors have a slower refresh rate than the DMD projector used in our display.
3. We may be able to achieve High Dynamic Range imagery.
4. With fewer number of binary voxels representing each color voxel, we can represent objects that are transparent and closer to each other than with the fixed pipeline decomposition. With the fixed pipeline decomposition, we can not represent objects along the same depth that are closer than $3 \times$ color bit-depth.

For the fixed-pipeline algorithm, let us parameterize the minimum separation required to cleanly display two objects that project to the same pixel (i.e. the minimum separation for presenting transparencies). If we assume that the depth planes are distributed linearly in dioptric depth, and if the display's optics are designed for N_{planes} depth planes with a depth range of $D_{\min}$ - $D_{\max}$ diopters, and the display has a per-color-channel bit-depth of C_{depth}, then the minimum separation corresponds to: $S_{\min} = 3C_{\text{depth}} \left(\frac{D_{\max} - D_{\min}}{N_{\text{planes}}} \right)$. For the results shown in Fig. 3.9, the relevant parameters are $\{C_{\text{depth}} = 8, D_{\min} = 0.25, D_{\max} = 5, N_{\text{planes}} = 280\}$, and $S_{\min} = 0.407$ diopters. Suppose we changed the number of depth planes to a much smaller value, say $N_{\text{planes}} = 50$, then $S_{\min} = 2.28$ diopters which is a very large separation! Below, we demonstrate that color adaptive decomposition methods can do a much better job at presenting virtual scene with significantly reduced depth blur and virtually no separation between objects projected onto the same pixel.

3.7.2 Approach

The basic idea behind the color adaptive decomposition methods explored here is that of error propagation or diffusion. Starting at the nearest depth plane, we consider the slice of the color volume and best

approximate it with a binary image and an arbitrary LED color. Unavoidably, there will be errors, which are propagated onto the next depth plane. At the next depth plane, the current depth's slice of the color volume and the propagated errors are added and considered as the target image for the decomposition. The pseudo-code for this approach is given in Algorithm. 1.

Algorithm 1 Outline of error propagation appoach

Input: Color Volume
Output: Binary Images, LEDs

Residual $\leftarrow$ zeros
for $d \leftarrow 1$ **to** N_{planes} **do**
    Color Volume Slice $\leftarrow$ Color Volume $[:, :, d]$
    Target $\leftarrow$ Color Volume Slice + Residual
    $(\text{Binary Image}_d, \text{LED}_d) \leftarrow \text{Decompose}(\text{Target})$
    Residual $\leftarrow$ Target $-$ Reconstruct$(\text{Binary Image}_d, \text{LED}_d)$
end for

The methods explored here differ from each other in the way the target image at each depth is decomposed, i.e., the *Decompose* function in Algorithm. 1. In the sections below, we explore some possible ways to decompose the target image at each depth. For each of the methods below, the decomposition step tries to minimize the L2-norm between the decomposition and the target images:

$$\underset{\text{Binary Image}_d, \text{LED}_d}{\operatorname{argmin}} \; ||\text{Target} - \text{Reconstruct}(\text{Binary Image}_d, \text{LED}_d)||^2 \tag{3.16}$$

However, the problem with trying to decompose a 24-bit target RGB image into a 1-bit DMD image and an RGB LED value is that because the DMD pattern is common for the three color channels, the decomposition is not separable among the color-channels, e.g., a particular DMD pattern may have a very low loss for the red channel but very high for the green channel such that the combined error might be lesser than completely leaving out the green channel.

3.7.2.1 Combinatorial Optimization

In this method, we calculate the decomposition for every combination of color channel. These are the difference combinations of color channels: $\text{Combination}_1 = \{ 1, 0, 0 \}$, $\text{Combination}_2 = \{ 0, 1, 0 \}$, $\text{Combination}_3 = \{ 0, 0, 1 \}$, $\text{Combination}_4 = \{ 1, 1, 0 \}$, $\text{Combination}_5 = \{ 1, 0, 1 \}$, $\text{Combination}_6 =$

$\{ 0, 1, 1 \}$, Combination$_7 = \{ 1, 1, 1 \}$, where 0 indicates that the color channel is not considered for the optimization and 1 indicatest that the color channel is considered for the optimization.

Algorithm 2 Combinatorial color decomposition

Input: Target, Residual, Color Volume Slice
Output: Binary Image, LED, Residual

1: Target $\leftarrow$ Color Volume Slice + Residual
2: $\{\text{Target}_R, \text{Target}_G, \text{Target}_B\} \leftarrow$ Split(Target)
3: **for** $i \leftarrow 1$ **to** 7 **do**
4: **for** $j \leftarrow \{R, G, B\}$ **do**
5: **if** Combination$_i[j] == 0$ **then**
6: LED$_j \leftarrow 0$
7: Binary Image$_j \leftarrow I_{\text{ones}}$
8: **else**
9: LED$_j \leftarrow$ Mean $\left(\{\text{Target}_j : \text{Target}_j \neq 0\}\right)$
10: Binary Image$_j \leftarrow$ Signum $\left(\frac{\text{Target}_j}{\text{LED}_j} - \text{Threshold}\right)$
11: **end if**
12: **end for**
13: Binary Image$_i \leftarrow$ Binary Image$_1 \odot$ Binary Image$_2 \odot$ Binary Image$_3$
14: **for** $j \leftarrow \{R, G, B\}$ **do**
15: **if** Combination$_i[j] == 0$ **then**
16: LED$_j \leftarrow 0$
17: **else**
18: Target$'_j \leftarrow$ Target$_j \odot$ Binary Image$_i$
19: LED$_j \leftarrow \min\left(\{\text{Target}'_j : \text{Target}'_j \neq 0\}\right)$
20: **end if**
21: **end for**
22: LED$_i \leftarrow$ Combine $(\{\text{LED}_R, \text{LED}_G, \text{LED}_B\})$
23: Reconstruction$_i \leftarrow$ Reconstruct(Binary Image$_i$, LED$_i$)
24: Residual$_i \leftarrow$ Target $-$ Reconstruction$_i$
25: Energy$_i \leftarrow$ Loss(Residual$_i$)
26: **end for**
27: $k \leftarrow \text{argmin}(\text{Energy}_1, \text{Energy}_2, \ldots, \text{Energy}_7)$
28: Binary Image $\leftarrow$ Binary Image$_k$
29: LED $\leftarrow$ LED$_k$
30: Residual $\leftarrow$ Residual$_k$

See Algorithm 2 for pseudo-code for this approach. A step-wise explanation for the algorithm follows:

- Lines 1-2: At each depth plane, we assign the target image by adding the previous residual to the current depth plane's image. We split this target image into its color channels because we process the color channel's independently in the initial part of the algorithm to estimate the common optimal binary image.

- Line 3: In the outer for-loop, we consider each of the seven possible options for whether an LED is on or off, i.e., Combination_1, Combination_2, ..., Combination_7, and save these calculated values: LED values, binary image, the reconstructed image, the residual, and the energy or loss.

- Lines 5-13: calculate the best LED value and binary pattern when each color channel is considered independently and in accordance with whether that color channel's LED should be on or off. If that color channel's LED should be off, then the LED value for that color channel is set to 0, and the binary pattern is set to all ones. If, however, the LED should be on, the LED value is calculated as the mean value of the non-zero pixel values of that color channel, and the binary pattern is calculated by thresholding the pixel values divided by the calculated LED value. Typically, this threshold is set to 1.

- Line 14: Since the DMD pattern is common to all three color channels, we need to calculate the common binary pattern by simply multiplying together all the color channel's binary patterns.

- Lines 15-24: Because the common binary pattern may be different from each of the color channel's binary patterns, we need to calculate the new optimum LED values. This for-loop accomplishes that by assigning each LED value to the minimum pixel value in its color channel.

- Lines 27-30: We find which option resulted in the least energy and output that option's binary pattern, LED values, and residual image.

3.7.2.2 Highest Energy Channel Minimization

Algorithm 3 Highest Energy Channel Minimization

Input: Target, Residual, Color Volume Slice
Output: Binary Image, LED, Residual

1: $\text{Target} \leftarrow \text{Color Volume Slice} + \text{Residual}$
2: $\{\text{Target}_R, \text{Target}_G, \text{Target}_B\} \leftarrow \text{Split}(\text{Target})$
3: $\{\text{Energy}_R, \text{Energy}_G, \text{Energy}_B\} \leftarrow \text{Loss}(\{\text{Target}_R, \text{Target}_G, \text{Target}_B\})$
4: $k \leftarrow \text{argmin}(\text{Energy}_R, \text{Energy}_G, \text{Energy}_B)$
5: $\text{LED} \leftarrow \{0, 0, 0\}$
6: $\text{LED}_k \leftarrow \text{Mean}(\text{Target}_k : \text{Target}_k \neq 0)$
7: $\text{Binary Image} \leftarrow \text{Signum}(\frac{\text{Target}_k}{\text{LED}_k} - \text{Threshold})$
8: $\text{Reconstruction} \leftarrow \text{Reconstruct}(\text{Binary Image}, \text{LED})$
9: $\text{Residual} \leftarrow \text{Target} - \text{Reconstruction}$

See Algorithm 3 for a pseudo-code for this approach. A step-wise explanation for the algorithm follows:

- Line 1: At each depth plane, we assign the target image by adding the previous residual to the current depth plane's image.
- Line 2: Split the target image into its color channels because we process the color channel's independently.
- Line 3: Calculate the energy of each color channel.
- Line 4: Find the color channel that has the maximum energy. For the rest of the algorithm, it is only this color channel that is of interest to us. The other two color channels are ignored.
- Lines 5-6: Each LED color is first assigned to 0. Then, only for the color channel of maximum energy, the LED color is calculated as the mean of the non-zero pixel values of that color channel.
- Line 7: The binary image is calculated by thresholding the color channel of the maximum energy channel against the calculated LED color.
- Line 8-9: Calculating the reconstructed image and the residual image.

3.7.2.3 Projected Gradient

This approach is based on non-negative matrix factorization methods as used in recent compressive displays Wetzstein et al. (2012); Huang et al. (2015, 2017). However, the previous papers were concerned with decomposing a continuous-valued image into two or more continuous-valued images. Here, by continuous, we mean that the values are considered to be in continuous domain even through in a computer they may be represented by an 8-bit per color channel image. However, in our display, we need to decompose continuous-valued images (slices of the color volume) into a binary image and a continuous-valued triplet for the RGB LED. This is a significantly harder problem because it is a combination of the traditional optimization as explored by non-negative matrix factorization algorithms and combinatorial optimization which has not been investigated previously in the context of computational displays. In this section, we derive the update rules for non-negative matrix factorization algorithms, and we write down the algorithm and explain it. We shall see later in the Sec. 3.7.3 that this algorithm does poorly in terms of visual quality.

Newton's method adapted for optimization Non-negative matrix factorization's update rules can be understood from studying Newton's method adapted for optimization:

Taylor expansion:

$$f(x) = f(x_0) + \Delta f'(x_0) + \frac{\Delta x^2 f''(x_0)}{2} + ... \tag{3.17}$$

where $\Delta x = x - x_0$.

Let us consider only the first three terms of the Taylor expansion. We differentiate the above equation wrt Δx to determine the optimum step size to guarantee fast convergence.

$$\frac{df(x)}{d(\Delta x)} = \frac{d\left(f(x_0) + \Delta f'(x_0) + \frac{\Delta x^2 f''(x_0)}{2}\right)}{d(\Delta x)} \tag{3.18}$$

$$0 = f'(x_0) + \Delta x f''(x_0) \tag{3.19}$$

Rearranging:

$$\Delta x = -\frac{f'(x_0)}{f''(x_0)} \tag{3.20}$$

Then,

$$x_{n+1} = x_n - \frac{f'(x_0)}{f''(x_o)} \tag{3.21}$$

Applying Newton's method for our problem This is the mathematics for the matlab file named *adaptive_color_decomposition_all_channels.m*.

Now, let us apply Newton's method to our problem. Let us first denote the color volume by V_c, the binary volume by V_b, the color sub-volume image to be C, the binary subvolume image to be B, the LED color and brightness associated with B to be α. We now define the energy function that is to be optimized:

$$E = ||C - \alpha B||^2 = ||C^T - \alpha B^T||^2 \tag{3.22}$$

Note that the energy function is the L2-norm of the residual, R, defined as:

$$R = C - \alpha B = C^T - \alpha B^T \tag{3.23}$$

Deriving update rule for B Expanding $E = ||C^T - \alpha B^T||^2$:

$$E = C^T C - 2C^T \alpha B + \alpha^2 B^T B \tag{3.24}$$

$$\frac{dE}{dB} = -2\alpha C^T + 2\alpha^2 B^T \tag{3.25}$$

and

$$\frac{d^2 E}{dB^2} = 2\alpha \tag{3.26}$$

Then,

$$\Delta B = \frac{2\alpha C^T - 2\alpha^2 B^T}{2\alpha^2} = \frac{C^T - \alpha B^T}{\alpha} = \frac{R^T}{\alpha} \tag{3.27}$$

and

$$B_{n+1} = B_n + \frac{R^T}{\alpha} \tag{3.28}$$

Deriving update rule for α: Expanding $E = ||C^T - \alpha B^T||^2$:

$$E = C^T C - 2C^T \alpha B + \alpha^2 B^T B \tag{3.29}$$

$$\frac{dE}{d\alpha} = -2C^T B + 2\alpha B^T B \tag{3.30}$$

$$\frac{d^2 E}{d\alpha^2} = 2B^T B \tag{3.31}$$

Then,

$$\Delta\alpha = \frac{2C^T B - 2\alpha B^T B}{2B^T B} = \frac{C^T B - \alpha B^T B}{B^T B} = \frac{R^T B}{B^T B} \tag{3.32}$$

and

$$\alpha_{n+1} = \alpha_n + \frac{R^T B}{B^T B} \tag{3.33}$$

See Algorithm 4 for a pseudo-code for this approach. A step-wise explanation for the algorithm follows:

Algorithm 4 Projected Gradients

Input: Target, Residual, Color Volume Slice
Output: Binary Image, LED, Residual

1: Target $\leftarrow$ Color Volume Slice + Residual
2: $\{\text{Target}_R, \text{Target}_G, \text{Target}_B\} \leftarrow$ Split(Target)
3: **for** $j \leftarrow$ R, G, B **do**
4: $\quad \text{LED}_j \leftarrow \text{Mean}(\text{Target}_j : \text{Target}_j \neq 0)$
5: $\quad \text{Modulation}_j \leftarrow \frac{\text{Target}_j}{\text{LED}_j}$
6: **end for**
7: Binary Image $\leftarrow \text{sgn}(\text{Modulation}_R + \text{Modulation}_G + \text{Modulation}_B - \text{Threshold})$
8: Reconstruction $\leftarrow$ Reconstruct(Binary Image, LED)
9: Residue $\leftarrow$ Target $-$ Reconstruction
10: **for** $j \leftarrow 1$ **to** 2 **do**
11: $\quad$ **for** $j \leftarrow$ R, G, B **do**
12: $\quad\quad \text{LED}_j \leftarrow \text{LED}_j + \frac{\text{Reduce Sum}(\text{Residual}_j \odot \text{Binary Image})}{\text{Reduce Sum}(\text{Binary Image} \odot \text{Binary Image})}$
13: $\quad$ **end for**
14: $\quad$ Reconstruction $\leftarrow$ Reconstruct(Binary Image, LED)
15: $\quad$ Residual $\leftarrow$ Target $-$ Reconstruction
16: $\quad$ **for** $j \leftarrow$ R, G, B **do**
17: $\quad\quad \text{Modulation}_j \leftarrow \frac{\text{Residual}_j}{\text{LED}_j}$
18: $\quad$ **end for**
19: $\quad$ Binary Image $\leftarrow \text{Signum}(\text{Binary Image} + \text{Modulation}_R + \text{Modulation}_G + \text{Modulation}_B)$
20: $\quad$ Reconstruction $\leftarrow$ Reconstruct(Binary Image, LED)
21: $\quad$ Residual $\leftarrow$ Target $-$ Reconstruction
22: **end for**

- Line 1: At each depth plane, we assign the target image by adding the previous residual to the current depth plane's image.
- Line 2: Split the target image into its color channels because we process the color channel's independently.
- Lines 3-6: For each color channel, initialize the LED value to be the mean of that color channel's non-zero pixel values. Assuming that the DMD image is a per-channel continuous-valued image, calculate a modulation image by normalizing the color image by the mean of the color image.
- Line 7: Combine the per-channel continuous-valued images into a binary image by adding and thresholding.
- Line 8-9: Calculate the reconstructed image and the residual image.
- Line 10,22: for-loop for two iterations of the non-negative matrix factorization algorithm.

- Lines 11-13: Update the LED values according to Eq. 3.33.
- Lines 14-15: Calculate the reconstructed image and the residual image.
- Lines 16-18: Assuming that the DMD image is a per-channel continuous-valued image, calculate the gradient according to Eq. 3.28.
- Line 19: Combine the per-channel continuous-valued image-gradients into a binary image by adding and thresholding.
- Lines 20-21: Calculate the reconstructed image and the residual image.

3.7.2.4 Heuristic

This approach is a judicious combination of the brute-force approach (which yields the best image quality, least number of binary voxels, but longest runtime) and the projected gradients approach (which yields very poor image quality, a comparable number of binary voxels, but fast runtimes). In this approach, we treat the problem as a combination of combinatorial optimization (for selecting which combination of color channels the DMD image and LEDs should try to address) and continuous-valued optimization (for choosing the LED values). In this approach, the combinatorial optimization portion (selecting the combination of color channels to address) automatically determines the DMD pattern too.

See Algorithm 5 for a pseudo-code for this approach. A step-wise explanation for the algorithm follows:

- Line 1: At each depth plane, we assign the target image by adding the previous residual to the current depth plane's image.
- Line 2: Split the target image into its color channels because we process the color channel's independently.
- Lines 3-6: For each color channel, initialize the LED value to be the mean of that color channel's non-zero pixel values. Assuming that the DMD is capable of an independent per-color-channel binary image, calculate binary images.
- Lines 7-15: Calculate the best binary image for each of the seven possible options for whether an LED is on or off, i.e., Combination_1, Combination_2, ..., Combination_7, and calculate the number of

Algorithm 5 Heuristic approach

Input: Target, Residual, Color Volume Slice
Output: Binary Image, LED, Residual

1: Target $\leftarrow$ Color Volume Slice + Residual
2: $\{\text{Target}_R, \text{Target}_G, \text{Target}_B\} \leftarrow$ Split(Target)
3: **for** $j \leftarrow$ R, G, B **do**
4: $\quad \text{LED}_j \leftarrow \text{Mean}(\text{Target}_j : \text{Target}_j \neq 0)$
5: $\quad \text{Binary Mask}_j \leftarrow \text{Signum}(\frac{\text{Target}_j}{\text{LED}_j} - \text{Threshold})$
6: **end for**
7: **for** $i \leftarrow 1$ **to** 7 **do**
8: $\quad \text{Binary Image}_i \leftarrow I_{\text{ones}}$
9: $\quad$ **for** $j \leftarrow$ R, G, B **do**
10: $\quad\quad$ **if** $\text{Combination}_i[j] == 1$ **then**
11: $\quad\quad\quad \text{Binary Image}_i \leftarrow \text{Binary Image}_i \odot \text{Binary Mask}_j$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad \text{Pixels}_i \leftarrow \text{Reduce sum}(\text{Binary Image}_i) \cdot \text{Sum}(\text{Combination}_i)$
15: **end for**
16: $k \leftarrow \text{argmax}(\text{Pixels}_1, \text{Pixels}_2, \ldots, \text{Pixels}_7)$
17: Binary Image $\leftarrow \text{Binary Image}_k$
18: LED $\leftarrow$ LED $\cdot \text{Combination}_k$
19: Reconstruction $\leftarrow$ Reconstruct(Binary Image, LED)
20: Residue $\leftarrow$ Target $\cdot \text{Combination}_k -$ Reconstruction
21: **for** $j \leftarrow$ R, G, B **do**
22: $\quad \text{LED}_j \leftarrow \text{LED}_j + \frac{\text{Reduce Sum}(\text{Residual}_j \odot \text{Binary Image})}{\text{Reduce Sum}(\text{Binary Image} \odot \text{Binary Image})}$
23: **end for**
24: Reconstruction $\leftarrow$ Reconstruct(Binary Image, LED)
25: Residual $\leftarrow$ Target $-$ Reconstruction

pixels that are addressed by the binary image. Here, we count the number of pixels slightly differently: The number of pixels is the number of non-zero binary pixels of the binary image multiplied by the number of LEDs that are ON for that combination, e.g., if the number of non-zero binary pixels are 25 for $\text{Combination}_4 = \{1, 1, 0\}$, we count it as 50 pixels, and if the number of non-zero binary pixels are 99 for $\text{Combination}_1 = \{1, 0, 0\}$, it is counted as 99.

- Line 16: Find the combination which yields the maximum number of pixels as per the above method of counting pixels.
- Line 17-18: Copy the best combination's binary image to the output binary image. Copy the best combination's LED values to the optimization routine's LED value initialization.
- Lines 19-20: Calculate the reconstruction image and the residual image.

- Lines 21-23: Update the LED values according to the Eq. 3.33.
- Lines 24-25: Calculate the reconstruction image and the residual image.

3.7.3 Results

To study the above decomposition algorithms, we simulate the perceived images of all the decomposition algorithms proposed so far with each other, i.e., fixed-pipeline (Sec. 3.5.4), combinatorial (Sec. 3.7.2.1), highest energy channel minimization (Sec. 3.7.2.2), projected gradients (Sec. 3.7.2.3), and a heuristic approach (Sec. 3.7.2.4). For all the results shown in this section, we assume that the field-of-view of the display is constant.

We model the perceived images by two methods:

1. By accumulating the perspective projection of every binary image. In other words, by assuming that the human pupil is a pinhole camera. This method allows us to get a rough idea of the visual quality of the reconstruction in a more compact representation. It also allows us to count the number of binary voxels that each color voxel is decomposed into. See Fig. 3.12 for an example — this figure will be explained below in more detail. The disadvantage with this method is that it does not model focus cues such as accommodation or defocus blur.
2. To model focus cues such as accommodation and defocus blur as seen by a human pupil, we model the pupil as an area aperture of 4 mm. We render a set of images for a set of corresponding eye's focal lengths. This is called a *focal stack*. See Fig. 3.14 for an example — this figure will be explained below in more detail.

For both methods of generating perceived images, we present visual results, *peak signal-to-noise ratio* (PSNR) values, and *structural similarity index* (SSIM) values.

3.7.3.1 Pinhole camera simulation results

The top row of Figures 3.12 and 3.13 show the simulated perceived image assuming a pinhole pupil for a display of 280 depth planes. Observe the images for *projected gradients* in these figures — in Fig. 3.12, the image looks like a gray scale image and in Fig. 3.13, the image has color artifacts. As mentioned before, this algorithm is suitable when the decomposed units can take continuous values, but since our

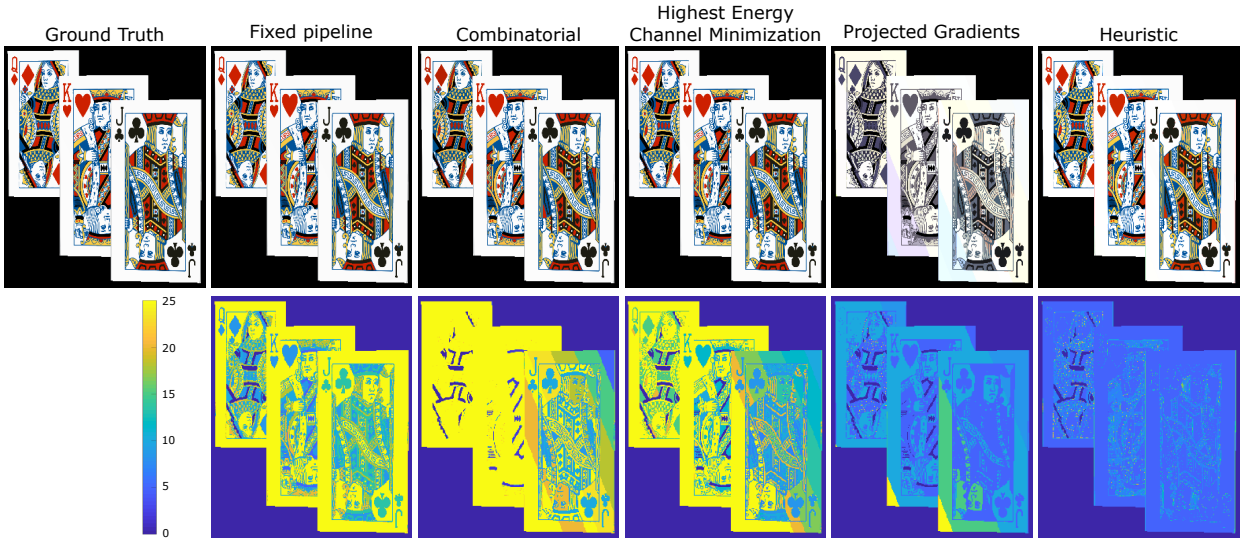

Figure 3.12: Simulation results. *Top row:* Visual quality when assuming that the imaging camera is a pinhole camera. *Bottom row:* Number of binary voxels for each color voxel.

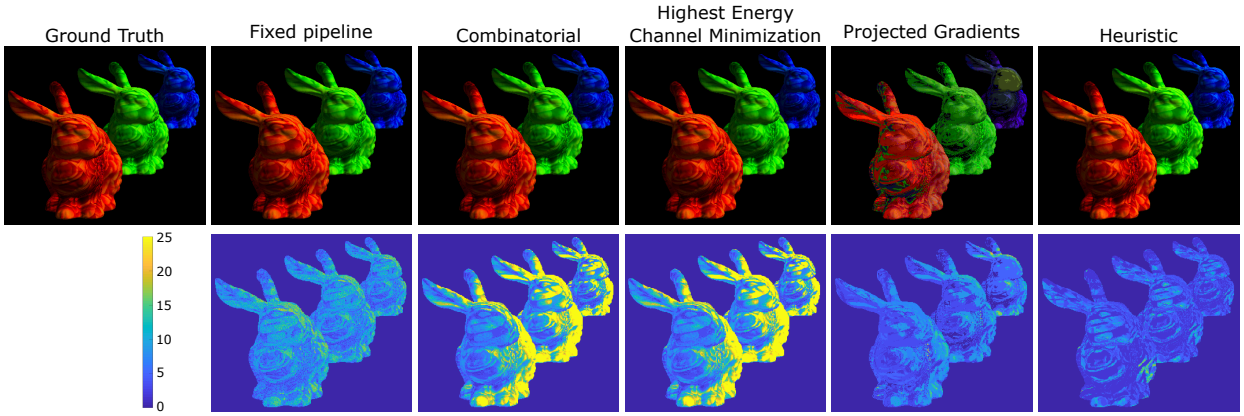

Figure 3.13: Simulation results. *Top row:* Visual quality when assuming that the imaging camera is a pinhole camera. *Bottom row:* Number of binary voxels for each color voxel.

display uses binary images, we need to convert continuous values predicted by the algorithm into binary values by thresholding. This thresholding leads to loss of information and results in artifacts.

The bottom row of Figures 3.12 and 3.13 show the number of binary voxels which represent each color voxel for a display of 280 depth planes. Observe how *projected gradients* and *heuristic* algorithm use a significantly lesser number of binary voxels. Also, observe how *combinatorial* and *highest energy channel minimization* algorithms use even more binary voxels than *fixed pipeline* — this is probably because calculating the LED values as the average of one or more color channel(s) is not the optimal choice. In *projected gradients* and *heuristic*, the LED values are initialized to the average of the color channels but are subsequently modified according to the optimization update rules.

Number of binary planes	Images	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
280	Cards	60.26	49.57	56.59	25.34	36.81
280	Bunnies	68.42	49.75	49.66	26.13	36.90
25	Cards	21.63	30.54	28.90	25.26	36.43
25	Bunnies	31.98	36.62	37.94	24.65	36.84
280	Transparency	35.47	44.38	50.43	20.65	33.85
25	Transparency	16.34	28.38	27.81	21.04	30.18

Table 3.1: Table shows PSNR values for the perceived image under the assumption that the pupil is a pinhole aperture for various decomposition algorithms and experiment settings.

Number of binary planes	Images	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
280	Cards	1.0000	0.9999	1.0000	0.9722	0.9976
280	Bunnies	0.9999	0.9999	0.9999	0.9720	0.9993
25	Cards	0.9736	0.9943	0.9895	0.9720	0.9987
25	Bunnies	0.9929	0.9975	0.9987	0.9602	0.9993
280	Transparency	0.9988	0.9998	0.9999	0.9389	0.9982
25	Transparency	0.9509	0.9908	0.9928	0.9435	0.9965

Table 3.2: Table shows SSIM values for the perceived image under the assumption that the pupil is a pinhole aperture for various decomposition algorithms and experiment settings.

Rows 1 and 2 of Tables 3.1 and 3.2 show the PSNR and SSIM values respectively for the images in Figures 3.12 and 3.13. These values indicate the *fixed pipeline* is the best algorithm and *projected gradients* is the worst when meausred with popular visual quality metrics. The reason *fixed pipeline* turns out as the best algorithm is because it is the only non-lossy decomposition. All the other methods invariably operate by thresholding the target color image against a calculated LED value.

But the above metrics do not indicate whether or not the decomposition happens at the correct depth. It may be possible that the reconstructed image has the same pixel values as the target image, but the binary representation of the color voxels are at a wrong location — such errors can lead to incorrect monocular cues. So, we introduce another way to compare the different decomposition algorithms below — by calculating the focal stack of the algorithm and comparing it against the focal stack of the color volume.

3.7.3.2 Reduced number of depth planes

Visual results Figures 3.14 and 3.15 show the focal stacks when the number of binary planes is 280 and 25 respectively. From these figures, we can see the first failure case for the *fixed pipeline* algorithm. When

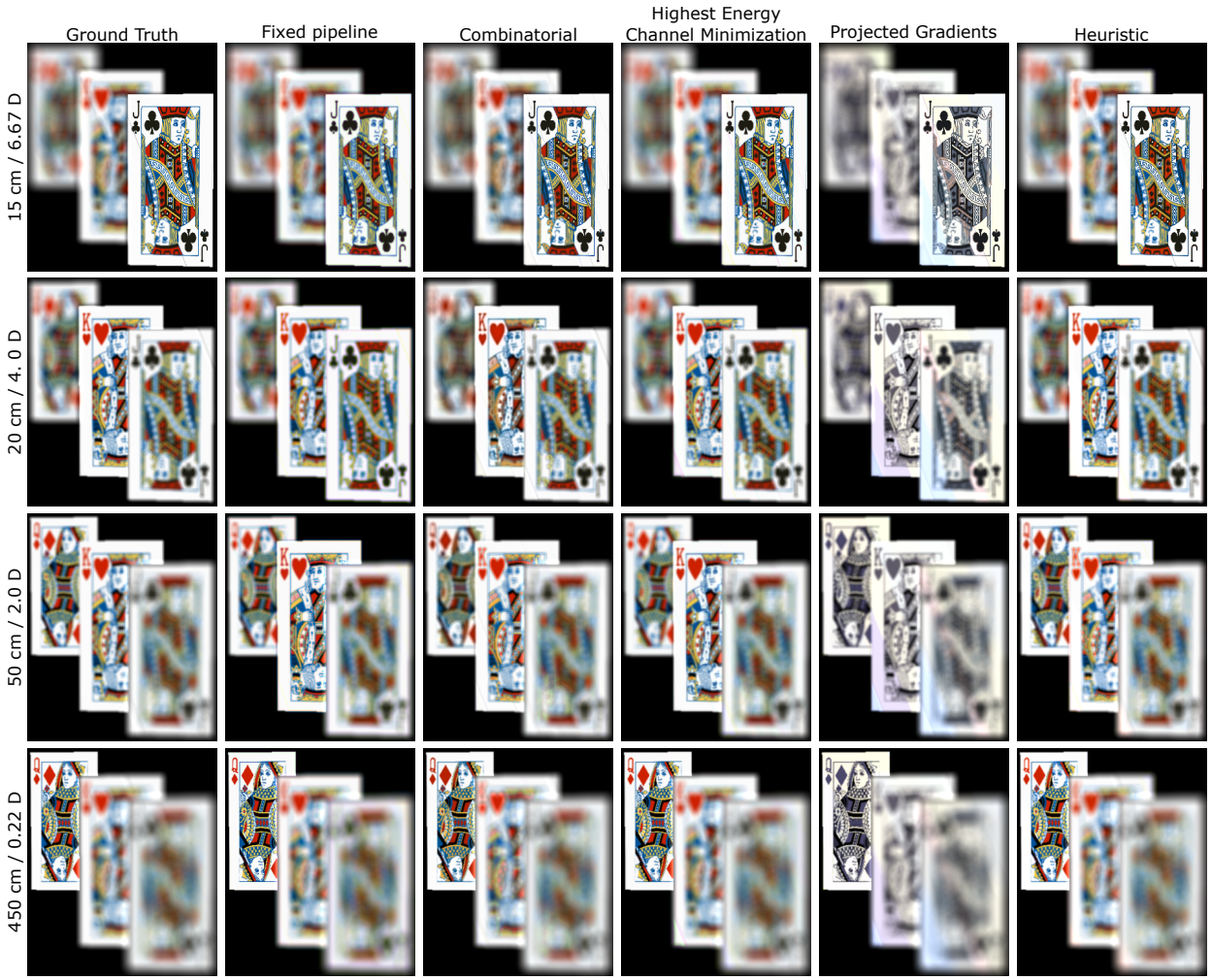

Figure 3.14: Figure shows focal stacks of the binary volume for the different decomposition algorithms. For this set of images, $N_{\text{planes}} = 280$.

$N_{\text{planes}} = 25$, the nearest and farthest cards exhibit color artifacts because their binary decompositions lie outside the displayable volume and hence get clipped. We also notice that *highest energy channel minimization* shows some darkening for the farther virtual objects, and that *combinatorial* incorrectly decomposes the farthest card to a grayscale card.

PSNR and SSIM results Tables 3.3 and 3.4 show the SSIM values for the focal stacks shown in Figures 3.14 and 3.15 respectively. From all these tables, Tables 3.5 and 3.6 show the SSIM values for the focal stacks shown in Figures 3.14 and 3.15 respectively.

From tables 3.3 and 3.5, we see that even though the PSNR and SSIM values for the pinhole aperture simulation predicts that *fixed pipeline* is the best algorithm, when focus cues such as accommodation and defocus blur are taken in account, *heuristic* algorithm is actually better. From all these tables, we see that

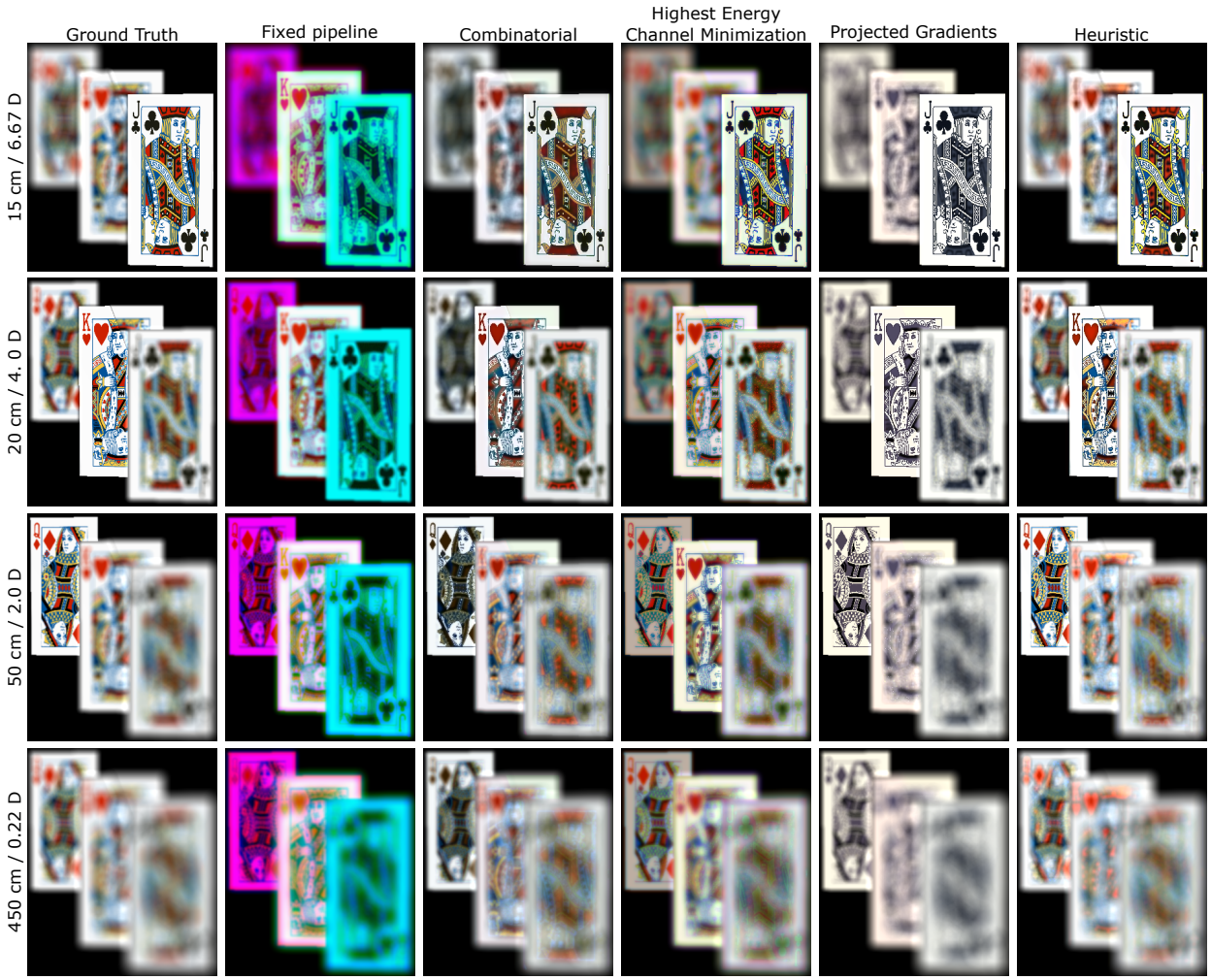

Figure 3.15: Figure shows focal stacks of the binary volume for the different decomposition algorithms. For this set of images, $N_{\text{planes}} = 25$.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	60.27	49.57	66.59	25.34	36.82
15 cm / 6.67 D	29.34	36.57	34.15	27.58	34.37
20 cm / 4.0 D	36.47	40.89	42.29	29.72	43.19
50 cm / 2.0 D	34.45	38.72	37.58	30.04	40.79
450 cm / 0.22 D	37.42	42.94	42.45	30.05	44.33

Table 3.3: Table shows PSNR values for focal stacks of the different algorithms. This table is for the cards target image and 280 binary depth planes.

heuristic algorithm shows the best performance in terms of PSNR and SSIM. The superiority of *heuristic* is obvious especially when $N_{\text{planes}} = 25$.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	18.49	30.55	28.90	25.26	36.43
15 cm / 6.67 D	18.29	28.56	27.14	26.11	31.01
20 cm / 4.0 D	18.75	28.88	26.41	26.83	31.59
50 cm / 2.0 D	18.97	29.97	27.12	28.54	33.06
450 cm / 0.22 D	19.58	31.86	29.31	32.85	38.01

Table 3.4: Table shows PSNR values for focal stacks of the different algorithms. This table is for the cards target image and 25 binary depth planes.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	1.000	1.000	1.000	0.972	0.998
15 cm / 6.67 D	0.985	0.996	0.994	0.966	0.996
20 cm / 4.0 D	0.991	0.997	0.997	0.965	0.999
50 cm / 2.0 D	0.987	0.994	0.993	0.964	0.997
450 cm / 0.22 D	0.994	0.998	0.998	0.964	0.999

Table 3.5: Table shows SSIM values for focal stacks of the different algorithms. This table is for the cards target image and 280 binary depth planes.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	0.953	0.994	0.990	0.972	0.999
15 cm / 6.67 D	0.910	0.977	0.970	0.964	0.992
20 cm / 4.0 D	0.914	0.983	0.968	0.963	0.991
50 cm / 2.0 D	0.918	0.979	0.964	0.960	0.986
450 cm / 0.22 D	0.904	0.976	0.969	0.958	0.990

Table 3.6: Table shows SSIM values for focal stacks of the different algorithms. This table is for the cards target image and 25 binary depth planes.

3.7.3.3 Transparencies

Visual results Figures 3.16 and 3.17 show the focal stacks when the number of binary planes is 280 and 25 respectively. When $N_{\text{plane}=280}$ (Fig. 3.16), we see that *fixed pipeline* has artifacts in some cases, e.g., the near card looks cyan and parts of the red or green bunny appear opaque. These artifacts arise when two color voxels are separated by less than 24 depth planes, which is the length of binary voxels required to represent any given color voxel. So, when two color voxels lie within 24 depth planes of each other, only one will be represented while the other is discarded When $N_{\text{plane}} = 25$ (Fig. 3.17), we see additional

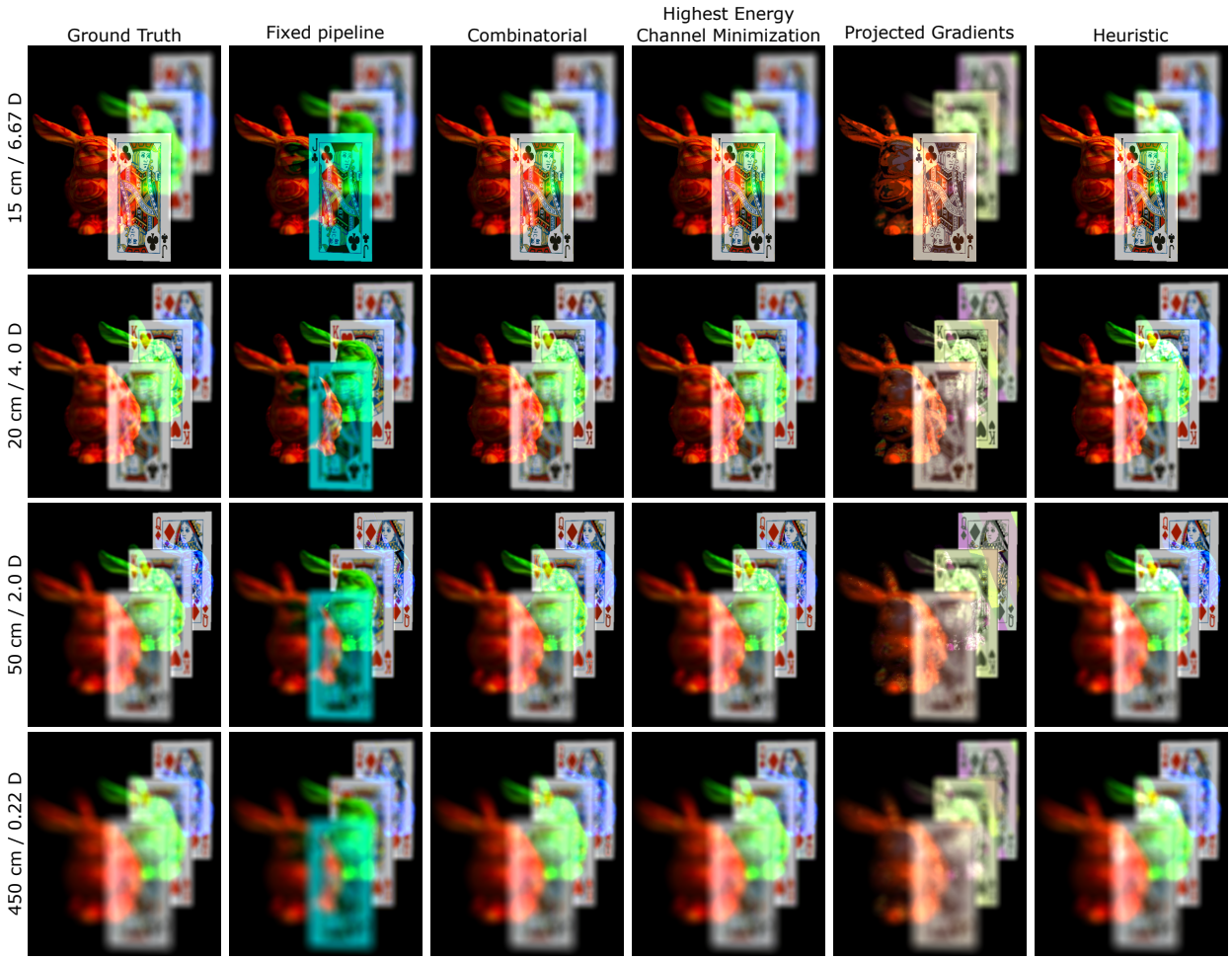

Figure 3.16: Figure shows focal stacks of the binary volume for the different decomposition algorithms. For this set of images, $N_{\text{planes}} = 280$.

artifacts in the focal stack of *fixed pipeline* due to the binary representation for near and far objects being clipped by the fewer binary depth planes.

PSNR and SSIM results Tables 3.7 and 3.8 show the PSNR values for the focal stacks in Figures 3.16 and 3.17 respectively. As expected, *fixed pipeline*'s values are much lower than the other algorithms. But, unlike previously discussed tables, here we see that *combinatorial* and *highest energy minimization* algorithms are slightly better than *heuristic*. However, when we look at the SSIM values in 3.9 and 3.10, it looks like *heuristic* is slightly better than *combinatorial* and *highest energy minimization*.

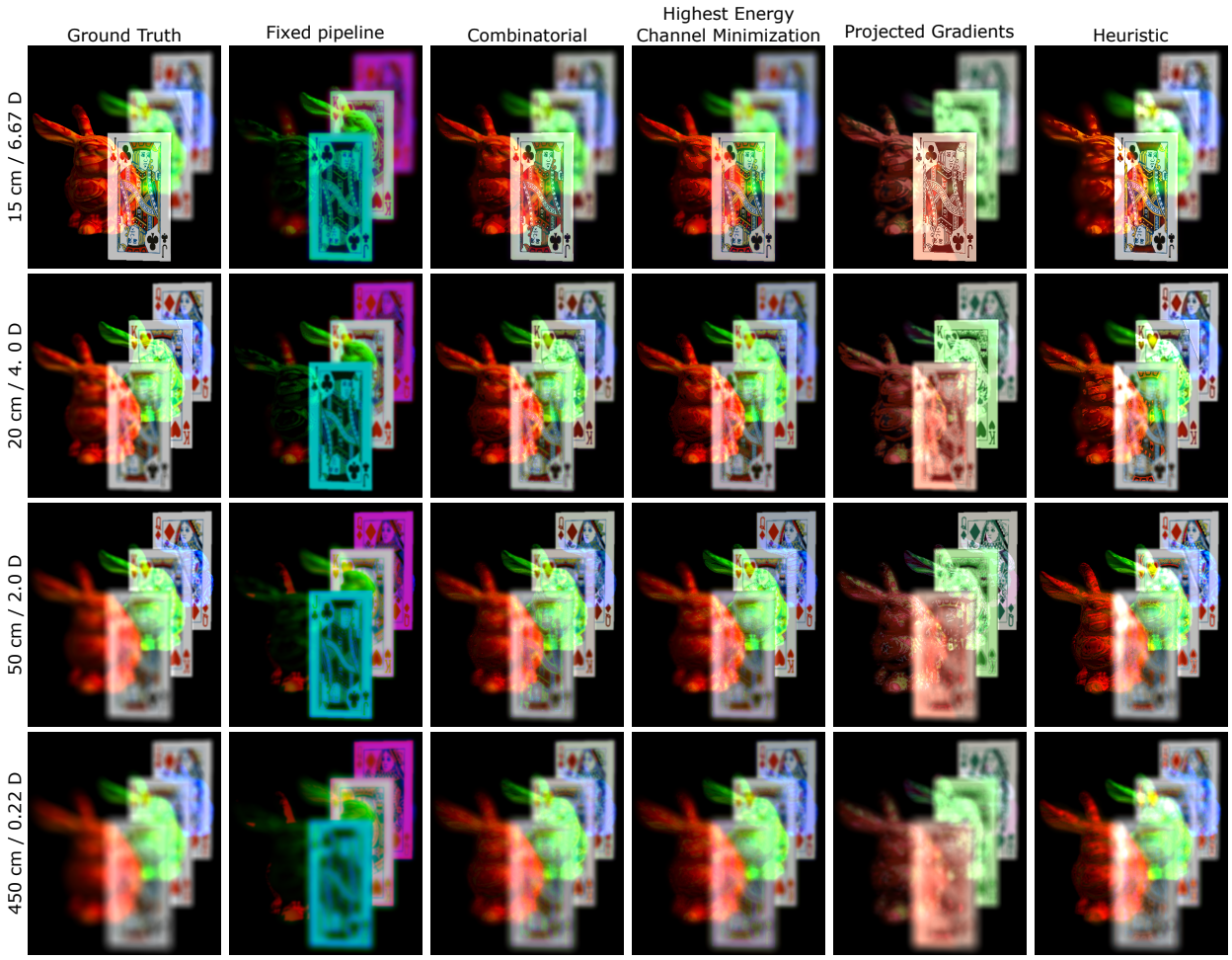

Figure 3.17: Figure shows focal stacks of the binary volume for the different decomposition algorithms. For this set of images, $N_{\text{planes}} = 25$.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	19.29	44.09	45.56	22.29	29.80
15 cm / 6.67 D	19.61	37.42	37.58	24.03	31.23
20 cm / 4.0 D	20.02	38.46	39.44	24.26	32.65
50 cm / 2.0 D	20.33	37.44	38.60	24.14	33.82
450 cm / 0.22 D	20.87	43.15	44.07	25.40	36.18

Table 3.7: Table shows PSNR values for focal stacks of the different algorithms. This table is for a virtual scene composed of transparent objects and 280 binary depth planes.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	17.02	30.96	30.84	22.15	27.94
15 cm / 6.67 D	17.11	29.01	28.58	23.24	26.79
20 cm / 4.0 D	17.46	29.75	28.73	23.84	27.28
50 cm / 2.0 D	17.51	30.14	29.20	23.86	27.44
450 cm / 0.22 D	17.90	32.18	31.41	25.32	30.78

Table 3.8: Table shows PSNR values for focal stacks of the different algorithms. This table is for the cards target image and 25 binary depth planes.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	0.956	1.000	1.000	0.942	0.997
15 cm / 6.67 D	0.953	0.997	0.997	0.935	0.997
20 cm / 4.0 D	0.949	0.997	0.997	0.937	0.998
50 cm / 2.0 D	0.943	0.997	0.998	0.940	0.999
450 cm / 0.22 D	0.939	0.998	0.999	0.937	0.999

Table 3.9: Table shows SSIM values for focal stacks of the different algorithms. This table is a virtual scene composed of transparent objects and 280 binary depth planes.

Focal depth	Fixed pipeline	Combinatorial	Highest-Energy Minimization	Projected Gradients	Heuristic
All (pinhole aperture)	0.912	0.994	0.994	0.941	0.997
15 cm / 6.67 D	0.883	0.979	0.979	0.933	0.981
20 cm / 4.0 D	0.890	0.981	0.977	0.930	0.984
50 cm / 2.0 D	0.885	0.981	0.979	0.931	0.978
450 cm / 0.22 D	0.872	0.983	0.982	0.932	0.983

Table 3.10: Table shows SSIM values for focal stacks of the different algorithms. This table is for the cards target image and 25 binary depth planes.

Summary To summarize, here are the key observations from these results:

1. For a virtual scene composed of only opaque objects and a display with many depth planes like 280, *fixed pipeline* appears to do a good job, especially when we model the perceived image as seen by a pinhole camera. However, when we take into account focus cues such as accommodation or defocus blur, we see that *fixed pipeline* does worse than some other decomposition algorithms.

2. For virtual scenes composed of transparent objects or for a display with few depth planes like 25, *fixed pipeline* shows unacceptable artifacts such as completely missing color voxels or misrepresenting them.

3. *Projected gradient* and *heuristics* algorithms need a significantly fewer number of binary voxels to represent color voxels compare to other algorithms. However, when it comes to image quality, *projected gradients* does much worse than most algorithms in almost all experimental settings, e.g., usually results in artifacts such as grayscale images or incorrect colors.

4. Overall, the *heuristics* approach performs the best. The only scenario where *heuristics* does not outperform all other algorithms is when we consider PSNR values for a virtual scene composed of transparent objects. For the same experimental setting however, *heuristics* shows better SSIM values than other algorithms.

Conclusion In conclusion, by treating the problem as a combination of a combinatorial optimization (in selecting which color channels need to be decomposed at a given depth and thereby calculating the optimum binary pattern) and conventional continuous space optimization (in calculating the optimum LED values) resulted in a heuristic algorithm which strikes a balance between speed and image quality. We also emphasize that it is vital to consider the non-zero aperture and defocus blur effects when assessing these different approaches.

3.8 Towards a Real-Time System

A real-time system with only 8 depth planes was developed as part of this dissertation to demonstrate the future feasibility of a completely developed real-time system. Fig. 3.18 shows an overview of the real-time volumetric display system. In Fig. 3.18, solid lines indicate real-time tasks, and dashed lines indicate

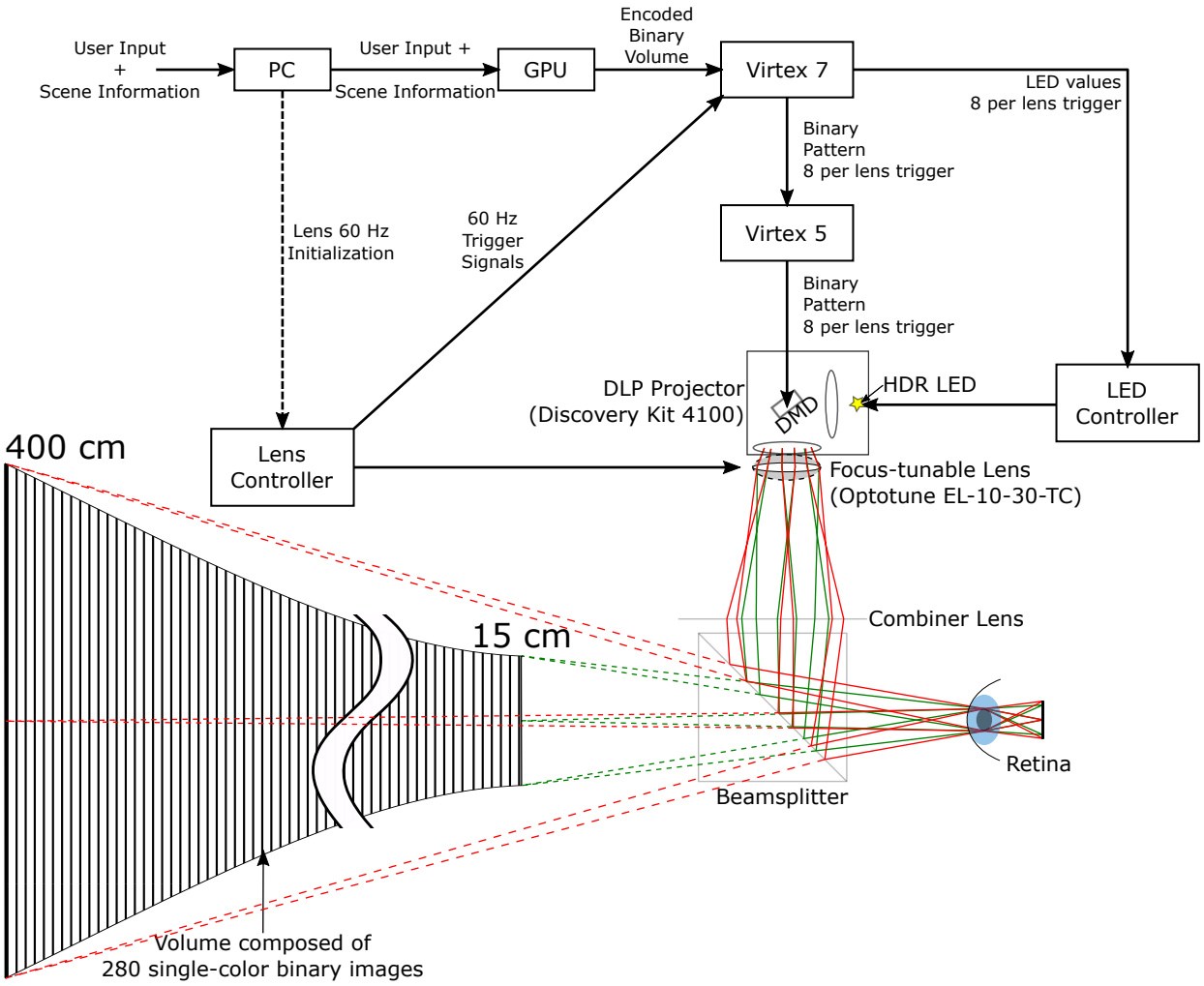

Figure 3.18: Figure shows the overview of the real-time volumetric display system.

off-line tasks. The only off-line task is to initialize the lens driver to oscillate the focal length of the lens at 60 Hz. The computational components of this real-time system and their tasks is summarized below:

1. PC:
    - Initializes the lens driver to oscillate at 60 Hz.
    - Passes scene infomration (3D models, lighting information, camera position) to the GPU.
    - Interprets user mouse movements to modify camera position.
2. GPU (in the order mentioned):
    - Render 3D scene into an RGB image and depth map.
    - Decomposes RGB image and depth map into 8 binary images and 8 LED colors.

- Encodes these 8 binary images into a gray-scale image and copies this gray-scale image into the three color channels of the image being sent over a *digital visual interface* (DVI) to the Virtex-7 FPGA.

3. Virtex-7 FPGA:

   - Receives the DVI image and stores it into an on-board *random access memory* RAM.
   - Copies two color channels of the image on the RAM onto the cache memory.
   - Waits for a trigger signal from the focus-tunable lens driver which indicates the start of a lens cycle.
   - Sends 8 RGB LED values to the custom LED controller. These 8 values are uniformly spaced temporally over the duration of one lens cycle, assuming that the lens is oscillating at 60 Hz.
   - Sends 8 binary patterns to the Virtex 5 DMD controller board. These 8 values are uniformly spaced temporally over the duration of one lens cycle, assuming that the lens is oscillating at 60 Hz.

3.8.1 GPU computation

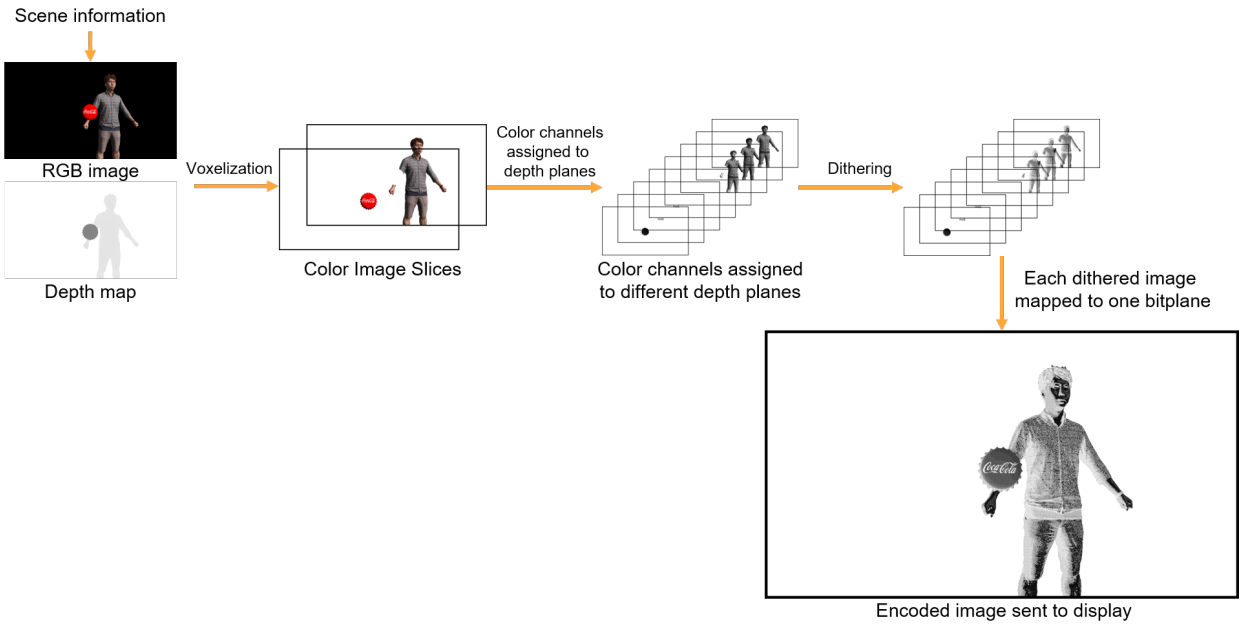

Figure 3.19: Figure shows the GPU computation pipeline for the real-time volumetric display.

For this system, we use the fixed-pipeline decomposition scheme and allocate the RGB LED colors of the 8 depth planes to be $L_1 = \{\alpha, 0, 0\}, L_2 = \{0, \alpha, 0\}, L_3 = \{0, 0, \alpha\}, L_4 = \{0, 0, 0, \}, L_5 = \{0, 0, 0\}, L_6 = \{\alpha, 0, 0\}, L_7 = \{0, \alpha, 0\}, L_8 = \{0, 0, \alpha\}$, where α is some light intensity.

Fig. 3.19 shows the GPU computation pipeline for the real-time volumetric display system. Assuming that we are dealing with only opaque objects, we start with rendering the RGB image and depth map of the scene.

From the RGB image and depth map, we calculate the color volume by quantizing the depth values into either the near depth plane or the far depth plane. For each color depth plane, we split the RGB image into its three color components and assign each color component to one depth plane. If C_1 is the near depth plane, then the red channel of C_1 is assigned to the nearest depth plane, the green channel of C_1 is assigned to the second depth plane, and the blue channel of C_1 is assigned to the third depth plane and so on. At this stage, note that each depth plane is still assigned an 8-bit color image. We rather need binary images at each depth plane so that it is compatible with our system. If we had a system with a large number of depth planes like our static volumetric display, each of these color depth planes could be decomposed into 8 binary depth planes. Since we don't have that, we perform spatial dithering to convert each 8-bit color image into a binary image. After this step, we have 8 1-bit images for each depth plane which can be encoded into a single color channel, either red, green, or blue. Due to some limitations in our FPGA implementation (described in Sec. 3.8.3), we copy this encoded image into all the three color channels to make the subsequent parts of the display pipeline agnostic to the color channel it chooses to display.

3.8.2 Results

A video was recorded as a demonstration of this real-time volumetric display system. This video was submitted as part of this dissertation and is also available publicly at this URL: `https://www.youtube.com/watch?v=pUtvBEPkzfA`. The video shows two real objects and two virtual objects. The real objects are a postage stamp at 30 cm and a real person at 300 cm. The virtual objects are a Coca-Cola bottle cap model and a 3D body scan of a person. To demonstrate the real-time nature of the system, the real person was asked to juggle some balls, the AR objects are rotated during the recording, and the camera's focus is changed gradually between far and near focus settings bringing different parts of the scene into focus.

As we can see, the resolution and colors of the AR body model are severely compromised due to our system's limitations which reduces a 3-channel 8-bits-per-channel color volume to a 3-channel 1-bit-per-channel binary volume. The visual quality of the AR scene can be significantly improved by increasing the number of binary planes of the system and by using more sophisticated decomposition schemes, as explored in Section 3.7.

3.8.3 Current limitations

The Virtex-7 FPGA program used here is a slightly modified version of Lincoln et al.'s scene-adaptive low-latency AR display Lincoln et al. (2017). The modifications include (1) listening for the trigger from the lens driver, (2) modified timings to output only 8 binary images in $\frac{1}{60}$-th of a second. As mentioned above, one of the Virtex-7 FPGA's activities is to transfer two color channels of the received image from the RAM to the cache memory. In doing so, it may choose to transfer either red and green, or green and blue, or blue and red. The order did not matter in Lincoln et al. (2017)'s work but it matters for our system because the images shown within one lens cycle's duration need to be in a fixed and pre-determined order. However, we didn't implement the necessary modifications to ensure a deterministic order of copying the color channels and instead chose to make the system color channel-agnostic by copying the same gray-scale image into the three color channels of the image. If this modification were done, the number of binary planes would be 24 instead of 8.

3.9 Optical Distortion Correction[2]

In this section, we develop optical calibration and distortion correction for our volumetric augmented reality display. An unintended property of this display is that the field-of-view of the depth planes changes slightly over depth. This change in field-of-view can cause the following problems:

1. Image distortions: if two digital objects at different depths are expected to line up with each other, they will not. Alternatively, a long object somewhat parallel to the display axis will appear to curve rather than look straight.

[2] This section previously appeared as an article in Emerging Digital Micromirror Device Based Systems and Applications XII. The original citation is as follows: Rathinavel, K., Wang, H., and Fuchs, H. (2020). Optical calibration and distortion correction for a volumetric augmented reality display. In *Emerging Digital Micromirror Device Based Systems and Applications XII*, volume 11294, page 112940M. International Society for Optics and Photonics

2. Reduced spatial resolution: since some of these displays distribute the decomposition of a color voxel over multiple binary voxels at different depths, the slight curve can introduce a blurring effect and lead to a loss in spatial resolution Rathinavel et al. (2018b).

3. Incorrect depth cues: Some depth cues, e.g., motion parallax, perspective, relative density, and relative size, will be slightly incorrect due to this distortions Cutting and Vishton (1995).

To address these issues, we develop an optical calibration method and a distortion correction as a post-processing step to our rendering pipeline.

3.9.1 Approach for calibration and distortion correction

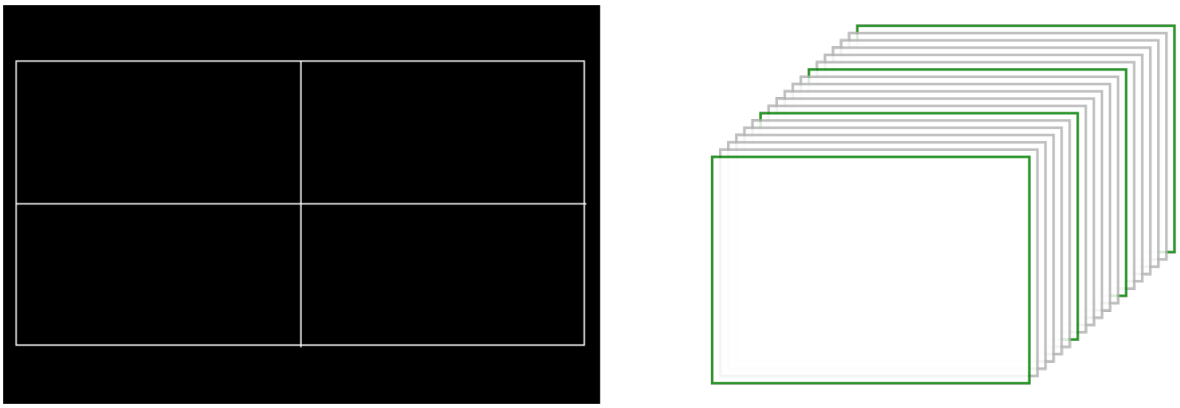

Figure 3.20: (Left) Calibration image that can be placed at multiple depth-planes. (Right) Calibration volume which is mostly composed of fully-black images (depicted by gray border) with a few calibration images (depicted by green border).

Our approach for calibration and distortion correction assumes that the depth-planes are centered on and perpendicular to the display's optical axis. The reason we assume this is because it is difficult to measure the depth of one (or a few nearby) display voxels. To measure the depth, we would need a camera with a very narrow depth-of-field such that it can discern the difference in depth between any two adjacent depth-planes — this is not practical. Furthermore, the human vision's depth of field is 0.3 diopters, and our depth-plane spacing is dense enough to be well within the 0.3 diopters threshold Rathinavel et al. (2018b). Below we discuss each stage of our one-time calibration procedure and our post-rendering distortion correction step.

3.9.1.1 Synthetic volume for calibration

To use in the calibration steps, we generate a synthetic volume composed of mostly black images interspersed with images composed of a centered rectangle, centered horizontal line, and centered vertical line are bright. An example calibration image is shown in Fig. 3.20 (left). This image is placed at a sparse set of depth planes. Fig. 3.20 (right) is a concept diagram explaining the sparse locations of the calibration images — gray bordered depth-planes are fully black images whereas green bordered depth-planes contain the calibration pattern shown in Fig. 3.20 (left). Note that unlike the depiction in Fig. 3.20 (right), our display's volume is much denser (composed of 280 depth-planes).

3.9.1.2 Pre-calibration: Aligning camera's and display's optical axis

To demonstrate our calibration and distortion correction, we first need to align the recording camera's axis to coincide with the display's axis. An explanation for this follows:

Recall that in our display, each color voxel is decomposed into some binary voxels such that these binary voxels will lie on a single perspective projection line such that they get integrated onto the same retinal or camera pixel. Since multifocal plane displays are view-dependent displays, it is necessary to track the pupil position, and the decomposition also needs to be view-dependent. But, in this paper, we do not use an eye-tracker, and we assume that all the depth planes are centered and perpendicular to the display's optical axis. If we assume the eye's axis to be aligned with the display's axis, the display's axis is the only line that will not need calibration or distortion correction. Hence, we need to align the camera's and display's axis.

To align the camera's axis with the display's axis, the stack of synthetic images is displayed and the camera's position was manually adjusted until the centered-horizontal and centered-vertical lines aligned. When aligned properly, the image seen is shown in Fig. 3.22 (left).

3.9.1.3 Calibration

Our calibration approach is to sample the field-of-view for a sparse set of depth-planes and interpolate the scaling factor that needs to be applied to each depth-plane to ensure a constant field-of-view across the entire volume. This happens in these steps:

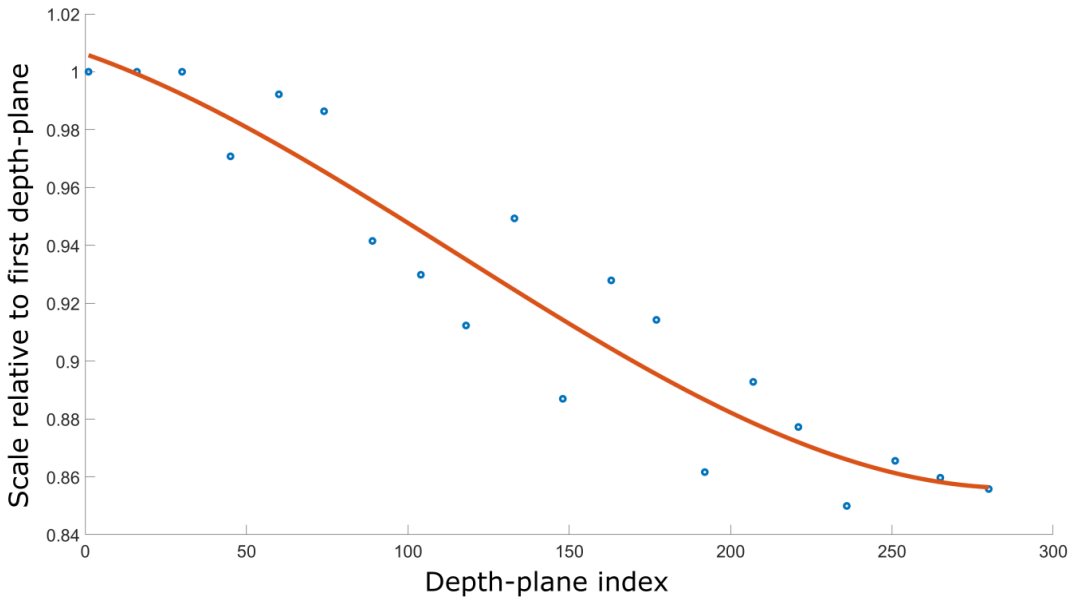

Figure 3.21: Plot of the relative scale as a function of the depth-plane's index, i.e., $s(j)$ mentioned in Sec. 3.9.1.3

1. To sample the field-of-view for each depth-plane, we place the image shown in Fig. 3.20(left) at that depth plane, capture the image seen by a camera, and measure the number of pixels between the left edge of the outer rectangle to the right edge of the outer rectangle. Say this results in a set $\{\theta_i\}$ where θ_i is the field-of-view of the i^{th} depth-plane and $i \in [1, ..., M]$, where M is the number of depth planes where we sample the field-of-view. In our experiment, $M = 10$.

2. We calculate the relative scaling factor for these depth planes as $s(i) = \frac{\theta_i}{\theta_1}$, i.e., the first (nearest) depth-plane is assumed to have a scaling factor of 1 and all other depth-planes are assumed to be scaled relative to this.

3. We estimate the scaling factor for all depth-planes as the function $s(j), j \in [1, 280]$ as a cubic interpolation of the data-points $\{(s(i), i)\}, i \in [1, M]$. This interpolation is shown in Fig. 3.21 where the blue circles are the sparse samples and the red curve is the estimated $s(j)$ function.

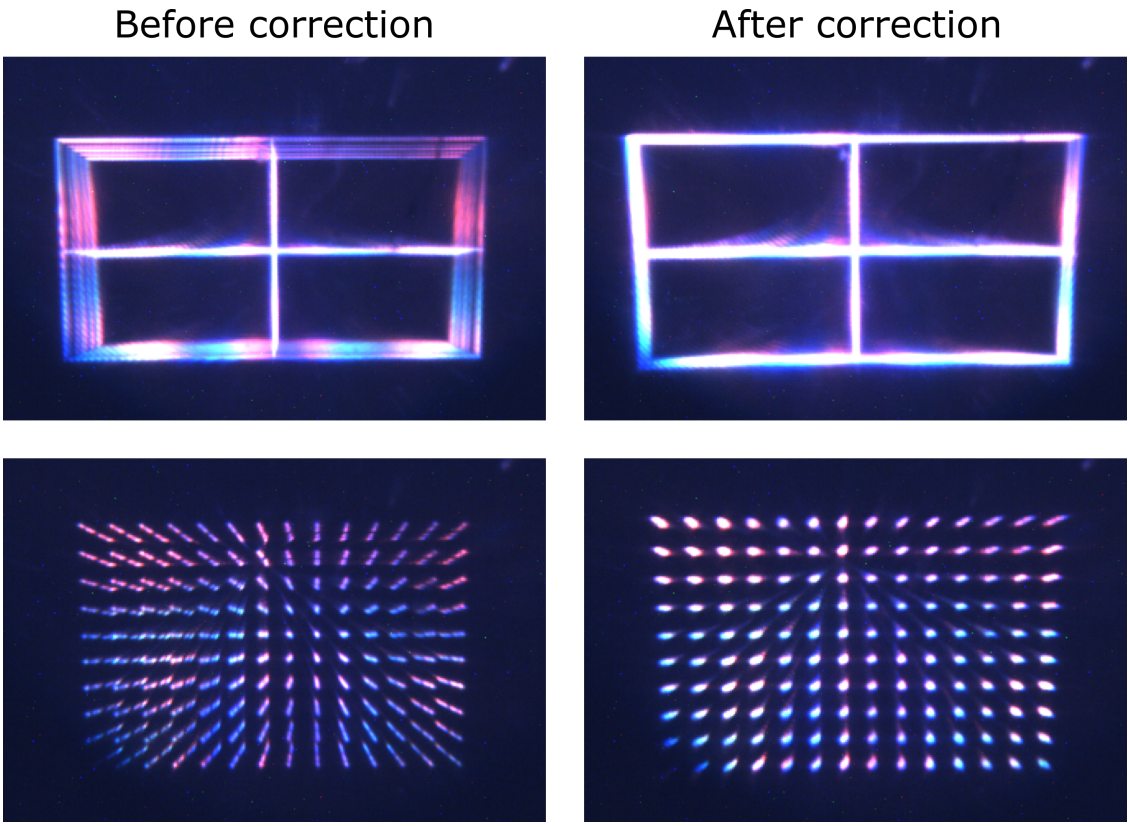

Figure 3.22: Demonstration of our calibration and distortion-correction approach. For these images, the display is displaying a volume across a large depth-range (15 cm to 400 cm). To ensure that all the images in this large depth-range are clearly visible, the aperture of the camera was set to the smallest setting. The chromatic artifacts seem to occur only for this narrow aperture setting.

3.9.1.4 Distortion correction

To correct for the distortion introduced by the changing field-of-view, we need to scale each depth-plane by the inverse of its scaling factor, i.e., by $\frac{1}{s(j)}$.

Fig. 3.22 (Top row) shows the before correction and after correction images for the calibration volume.

3.9.2 Results

Notice how the rectangles do not line up in the before-calibration image, but they line up correctly in the after-calibration image. Another synthetic volume was generated composed of a point grid and placed at different depths. Fig. 3.22 (Bottom row) shows the before and after images for the point-grid volume. Notice how the points appear as lines in the before-calibration image but nearly appear as points in the after-calibration image. Both images of Fig. 3.22 were taken with a very narrow aperture camera so that

the images at the different depths will appear clearly. If we were to take these images with a wider aperture setting, only one of the depth-planes would be in-focus, and the others would be out-of-focus and appear very blurry preventing us from verifying whether our method works. Images in Fig. 3.22 appear to have severe chromatic artifacts, but these are actually diffraction effects because we keep the aperture very small to capture a very large depth range. For even a slightly wider aperture setting, these chromatic artifacts are not present.

3.9.3 Limitation of our approach

Our approach does not address optical distortions that change across different lens cycles. To track optical distortions across lens cycles, we need a sophisticated lens tracking technology. Currently available lens tracking technologies are insufficient because they only track the focal length of the lens.

3.10 Discussion

3.10.1 Limitations

Bulky Optics The bulk of the optics is due to the large optical engine of the DLP Discovery 4100 kit, and the tiny aperture of the focus-tunable lens. Other DMD development boards have much smaller optics, and we also note that there is a commercially available AR display that uses a DMD chip Dewald et al. (2016). The small aperture of the focus-tunable lens constrains the optical design and limits the etendue of the system. There are focus-tunable lenses with a wider aperture that could be used, e.g., the focus-tunable lenses presented in Dunn et al. (2017). If we redesigned the optics and used alternative components, our NED could approach moderate form factor.

Bulky electronic components All of the driving electronics (DLP Discovery 4100 kit, custom RGB LED controller, microcontroller) could be reimplemented in a compact ASIC (Application Specific Integrated Circuit) device.

3.10.2 Future Work

Our near-eye display can emulate some other display technologies, such as multifocal and varifocal displays, and is thus suitable as a versatile platform for user studies. The current work could benefit

greatly from a compact, wearable, wide-FoV, binocular, and real-time implementation. Since the hardware platform and application are similar, this work could be integrated with recent low-latency Lincoln et al. (2016), and HDR AR Lincoln et al. (2017) displays work. This would require combining the volumetric rendering pipeline (presented in this chapter) and the low-latency rendering pipeline (presented in Lincoln et al. (2016, 2017); Lincoln (2017)). Another opportunity for research is to investigate if this display can be made entirely independent of eye-tracking requirements. Another avenue for future work is to explore adaptive lens functions. While this dissertation always oscillates the focal length of the lens according to a sinusoidal or triangular waveform at 60 Hz, the lens is capable of following any arbitrary current waveform. While our display demonstrates 280 dioptrically equidistant depth planes, an adaptive lens function can give an adaptive depth distribution of depth planes. Uses of such adaptive depth distribution may be foveation in depth, getting high-quality perceptual performance while using fewer depth planes, etc.

3.11 Conclusion

We have introduced a near-eye volumetric display capable of presenting a large volume over an extended depth-of-field created external to the display's physical volume. We view our system as a hybrid between traditional volumetric displays that create the volume within the confines of the display's physical volume, and view-dependent multifocal near-eye displays. We presented the optical design of our implementation and the rendering pipeline that synthesizes the volume for our display. Our main contribution is the idea that color-to-binary volume decomposition can be performed on a per-voxel-basis rather than an image-basis. We propose multiple decomposition algorithms and compare them with each other. We demonstrate a static display system which shows full-color volumetric display refreshed at 60 Hz and comprising 280 focal planes, each at a unique depth, ranging from 15cm (6.7 diopters) to 4M (0.25 diopters). We also demonstrate a dynamic volumetric display system with 8 depth planes. One of the key advantages of the proposed volumetric display system is the flexibility of the display system itself. It is composed of several components, each of which could be implemented and integrated in different combinations and methods to achieve different results. We hope that this system will inspire future research work in near-eye displays to rethink the rendering pipeline.

CHAPTER 4: VARIFOCAL-OCCLUSION AUGMENTED REALITY DISPLAY[1]

This chapter describes an augmented reality display that presents virtual imagery with support for depth-dependent hard-edge occlusion.

4.1 Introduction

Augmented Reality (AR) systems offer unprecedented experiences and are considered a next-generation computing platform. These wearable displays promise to seamlessly augment the physical world around us with digital content, such as information displays or user interfaces. Providing a seamless, perceptually realistic experience, however, requires the display to accurately support all depth cues of the human visual system Palmer (1999); Howard and Rogers (2002). While current AR displays offer impressive capabilities, they typically do not support the most important depth cue: occlusion Cutting and Vishton (1995).

Providing accurate (i.e., mutually consistent and hard-edge) occlusion between digital and physical objects with optical see-through AR displays is a major challenge. When digital content is located in front of physical objects, the former usually appear semi-transparent and unrealistic (see Fig. 4.1, columns 1 and 2). To adequately render these objects, the light reflected off of the physical object toward the user has to be blocked by the display before impinging on their retina. This occlusion mechanism needs to be programmable to support dynamic scenes and it needs to be perceptually realistic to be effective. The latter implies that occlusion layers are correctly rendered at the distances of the physical objects (see Fig. 4.2), allowing for pixel-precise, or hard-edge, control of the transmitted light rays.

Recent proposals on occlusion-capable optical see-through (OST) displays have only partially addressed this challenge. Global dimming Mori et al. (2018), for example, is successful in controlling the light transmission of the display but without spatial control. Image-forming systems Kiyokawa et al.

[1] This chapter previously appeared as an article in Transactions on Visualization and Computer Graphics. The original citation is as follows: Rathinavel, K., Wetzstein, G., and Fuchs, H. (2019). Varifocal occlusion-capable optical see-through augmented reality display based on focus-tunable optics. *IEEE transactions on visualization and computer graphics*, 25(11):3125–3134

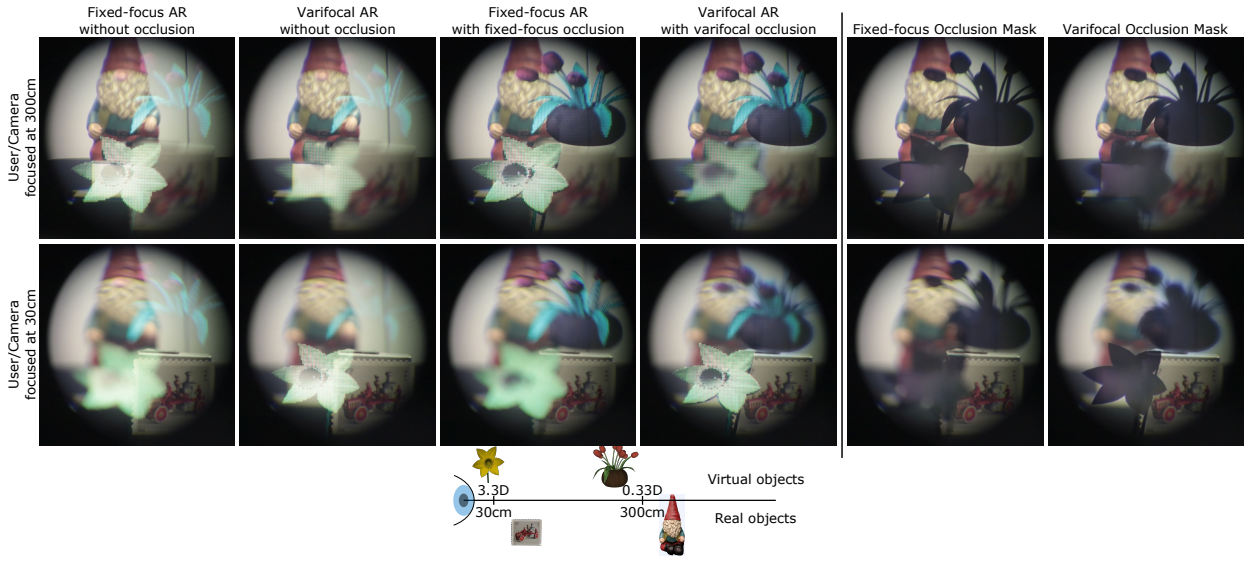

Figure 4.1: **Left of the vertical line:** views through our prototype AR display, which is emulating different AR display technologies for each column. The augmented scene is composed of real-world objects (stamp, motorcycle, and gnome) and virtual objects (ring, teapot, and bull). Objects are distributed at different depths: stamp and ring at 30cm, motorcycle and teapot at 100cm, and gnome and bull at 300cm. *(Column 1)* Commercially available AR displays: a transparent virtual image is presented at a fixed distance. Important depth cues such as occlusion and accommodation are absent. *(Column 2)* Varifocal AR displays: virtual image can be moved to different depths, but images are still transparent. *(Column 3)* Fixed-focus occlusion-capable AR display: Occlusion and virtual images are fixed at a single depth, limiting realism when the user is focused to other depths. Note how all virtual objects, including the nearby ones, are in focus when the camera is focused far, and all virtual objects are defocused when the camera is focused near. *(Column 4)* Varifocal occlusion-capable AR displays: virtual and occlusion image plane can be moved to different depths enabling perceptually correct depth cues for occlusion and accommodation. Note how objects at the same depth, e.g., near objects (stamp and ring) or far objects (gnome and bull), are correctly in focus or defocused depending on the focus state of the user/camera. **Right of the vertical line:** Comparison of occlusion masks between fixed-focus and varifocal occlusion-capable displays.

(2003); Cakmakci et al. (2004); Gao et al. (2012) enable consistent occlusions, but these are only correct at a single distance, severely limiting the image quality at other depths (see Fig. 4.1, column 3) and requiring bulky relay optics. Spatial light modulators (SLMs) for occlusion control can also be used without relay optics Itoh et al. (2017), but these will always be out of focus and require additional compensation techniques. Light field-based occlusion technology Maimone and Fuchs (2013) offers somewhat sharper occlusion control without relay optics. Out-of-focus SLMs Maimone and Fuchs (2013); Itoh et al. (2017) are usually based on liquid crystal displays (LCDs), which introduce diffraction artifacts of the physical world observed in OST displays, thus limiting the perceived image quality.

With this work, we introduce varifocal occlusion-capable optical see-through AR displays. These systems aim at providing a seamless and perceptually realistic experience by providing mutually consistent

occlusions over a large depth range (see Fig. 4.1, column 4). Similar to varifocal near-eye displays, our approach uses focus-tunable lenses to dynamically shift the occlusion SLM to a single, but adaptive, optical distance. We envision this approach to operate in a gaze-contingent mode, where an eye tracker determines the distance of the fixated object and both the digital content and the occlusion system are dynamically focused at this distance.

A unique challenge of varifocal occlusion implemented with focus-tunable optics is precise control of the optical distortion. As lenses change their focal power to align the occlusion SLM with different distance of the physical scene, the latter may also be magnified and its perceived distance altered, because the light of the physical scene and the occlusion SLM must share the same optical path. We derive a formal optimization approach and real-time heuristics to drive the proposed system in a perceptually accurate manner, preventing optical distortions of the physical world.

Specifically, we make the following contributions:

1. We introduce varifocal occlusion as an AR display capability that adaptively changes the focal distance of an occlusion mask to enable hard-edge occlusion over a large depth range.

2. We develop an optimization-based optical design approach for our focus-tunable optical system to achieve varifocal occlusion in a perceptually realistic manner without optically distorting the observed scene.

3. Using insights gained from the optimization approach, we use a ray-transfer matrix approach to derive closed-form solutions for optical designs that allow for varifocal occlusion in real-time.

4. We implement a monocular varifocal occlusion-capable AR display and demonstrate improved realism through depth-dependent occlusion.

4.2 Related Work

Table 4.1 shows an overview of how our new display technology compares with previous display technologies. A detailed discussion of the previous display technology follows.

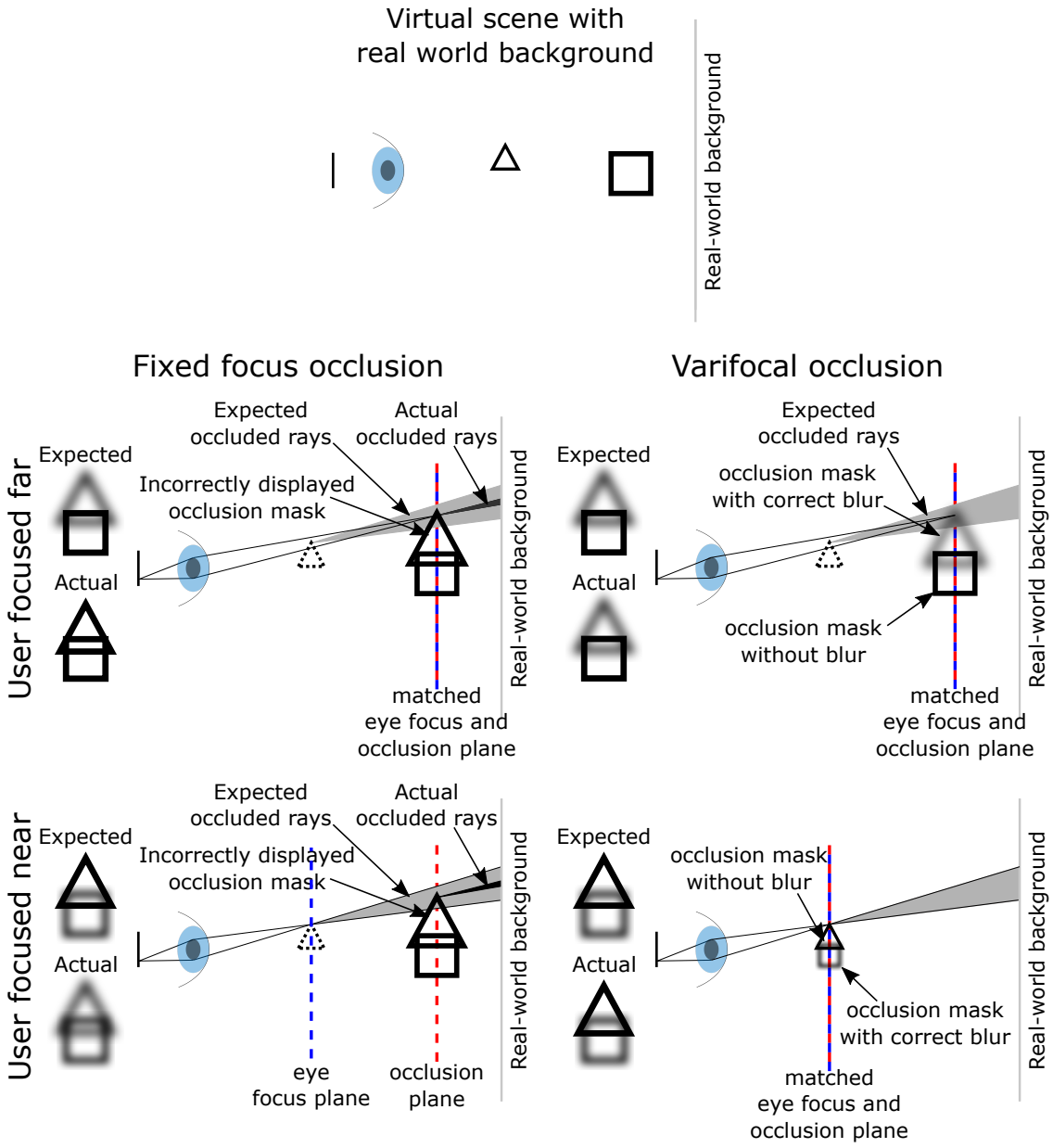

Figure 4.2: *Topmost Row:* A virtual scene composed of one near and one far object placed in front of a real-world background. *Grid of figures:* Comparison of occlusion mechanism only (i.e., ignoring the digital or color image) for fixed-focus and varifocal occlusion displays for the above scene. Dashed blue and red lines indicate the user's focal plane and display's occlusion image plane, respectively. Solid black lines indicate image formation for content placed in the user's focal plane. Images next to the eye show the "Expected" and "Actual" images seen by the user. Note that for fixed-focus occlusion, the occlusion plane is always at the far distance which causes the nearby object's occlusion mask to be seen incorrectly always and the far object's occlusion mask to be seen incorrectly when the eye is focused nearby. Varifocal occlusion-capable displays, on the other hand, move the occlusion plane to the user's focal plane and display an occlusion mask for in-focus objects as it is and a perceptually correct occlusion mask for out-of-focus objects by applying a computational blur.

Products/Prototypes	AR focus mechanism	Occlusion focus mechanism
HoloLens, Meta2, MagicLeap, etc.	Fixed-focus	None
Itoh et al. (2017)	Fixed-focus	Soft-edge
Kiyokawa et al. (2003), Howlett and Smithwick (2017), Cakmakci et al. (2004)	Fixed-focus	Fixed-focus
Dunn et al. (2017), Akşit et al. (2017)	Varifocal	None
Hamasaki and Itoh (2019), This chapter	**Varifocal**	**Varifocal**

Table 4.1: Summary of the type of focus cues that are supported for the virtual imagery and for occlusion by current AR products, previous research prototypes, and this work.

4.2.1 Varifocal Near-eye Displays

Varifocal virtual and augmented reality displays are similar to conventional fixed-focus near-eye displays, but they dynamically adjust the distance of the magnified virtual image. This can be achieved using focus-tunable lenses Liu et al. (2008); Konrad et al. (2016); Johnson et al. (2016); Padmanaban et al. (2017); Laffont et al. (2018); Rathinavel et al. (2018b); Xia et al. (2019), deformable membranes Dunn et al. (2017); Chakravarthula et al. (2018), or by mechanically actuating optical components Shiwa et al. (1996); Padmanaban et al. (2017); Akşit et al. (2017); Rathinavel et al. (2018a); Akşit et al. (2019). Varifocal displays require eye tracking to determine the distance of the fixated object, to which the display is then focused in a gaze-contingent manner.

Previous work on varifocal near-eye displays has primarily sought to adjust the virtual image of the digitally displayed content, primarily to mitigate the vergence-accommodation conflict Kooi and Toet (2004); Lambooij et al. (2009).

In this work, we extend the concept of varifocal displays to the problem of mutually consistent occlusion in AR, where the focal distance of an occlusion SLM is dynamically updated with the goal of improving perceptual realism. We discuss optical design strategies and demonstrate a varifocal occlusion-capable AR display that dynamically adjusts the focus of both digital image and occlusion SLM.

4.2.2 Occlusion-capable AR displays

4.2.2.1 Projection-based Lighting

Projection displays can be used to control the lighting of a scene in a spatially varying manner. Using such controlled illumination, mutually consistent occlusions, shading effects, and shadows in projector-based AR systems can be synthesized Bimber and Fröhlich (2002); Bimber et al. (2003); Maimone et al. (2013); Avveduto et al. (2017). The primary disadvantages of these systems are that projectors are required for the AR experience, which are not necessarily portable or wearable, and that they lack sufficient contrast in the presence of ambient illumination. We aim for a fully integrated occlusion-capable AR display that does not require additional projectors.

4.2.2.2 Global Dimming

Commercial AR displays (e.g., Microsoft HoloLens, Magic Leap) often use a neutral density filter placed on the outside of the display module to reduce ambient light uniformly across the entire field of view. An adaptive version of global dimming was recently proposed by Mori et al. (2018), where the amount of dimming is controlled by a single liquid crystal cell and responsive to its physical environment. While these approaches may be useful in some scenarios, they do not provide spatial control of the occlusion layer.

4.2.2.3 Fixed-focus Occlusion

The physical scene can be focused onto an occlusion SLM which selectively blocks its transmission in a spatially varying manner before it reaches the user's eye. This idea was first proposed by the seminal work of Kiyokawa et al. (2000, 2001, 2003). Improvements of related systems were later demonstrated Cakmakci et al. (2004, 2005); Wilson and Hua (2017); Howlett and Smithwick (2017); Wetzstein et al. (2010); Gao et al. (2012, 2013).

Unfortunately, focusing a scene on an SLM usually requires a bulky optical system, first to focus it to the SLM, then to negate the effect of the first lens, and then to flip the resulting image the right way up. Moreover, as this approach only focuses a single distance of the scene on the occlusion SLM, hard-edge occlusion is only achieved at this fixed focus distance. This limitation is similar to the characteristics of

fixed-focus near-eye displays, which has been alleviated by varifocal displays. In this work, we propose an extension of the concept of varifocal displays to occlusion.

Two key challenges for fixed-focus occlusion-capable displays are: (1) to ensure unit magnification of the see-through scene and (2) to ensure zero viewpoint offset between the see-through scene and the real-scene as seen without the display, so that the images of the real-world objects are at the correct distance. Both of these considerations are significantly more challenging for varifocal occlusion displays because unit magnification and zero viewpoint offset needs to be ensured while adjusting the focus of the SLM, which shares the optical path with the physical scene.

Kiyokawa et al. (2003) derive optical design parameters that satisfy unit magnification for all real-world object distances and also propose an interesting geometric configuration of the optical components that make the offset between the real-world objects and their images equal to zero. Cakmakci et al. (2004) propose a compact optical design that satisfies the magnification requirements, but it does not achieve zero offset between the real viewpoint and the virtual viewpoint; however, the offset is small (5 cm). Howlett and Smithwick (2017) propose an optical design approach based on ray-transfer matrices to achieve unit magnification and zero viewpoint offset, which is in turn inspired by optical cloaking Choi and Howell (2014). We extend the optical design approach based on ray-transfer matrices to varifocal occlusion displays and generalize the theory to asymmetrical optical designs.

4.2.2.4 Soft-edge Occlusion

To avoid a bulky optical system, a single LCD can be placed directly in front of the user's eyes Wetzstein et al. (2010); Itoh et al. (2017). However, due to the fact that the occlusion LCD is out of focus, it always appears blurred. Itoh et al. (2017) recently proposed to compensate for this blur by modifying the digitally displayed image. Such an approach could be interpreted as a hybrid optical see-through and video see-through AR display. Calibrating such a system requires extremely precise alignment, and the mismatch in resolution (spatial and angular), latency, brightness, contrast, and color fidelity between the digital display and physical world may contribute to perceived inconsistency and reduced perceptual realism in such a system Rolland and Fuchs (2000). Maimone et al. (2014) also used an out-of-focus LCD, where the occlusion mask is calculated as the silhouette of the virtual object. None of these approaches achieves hard-edge occlusion, which severely limits perceptual realism.

4.2.2.5 Light Field Occlusion

Maimone and Fuchs (2013) propose a 4D light field occlusion mask using stacked LCD layers placed out of focus in front of the eye, where the occluding patterns are calculated by light field factorization algorithms Lanman et al. (2010); Wetzstein et al. (2012). The advantage of light field occlusion is that depth-dependent occlusion can be presented for virtual content at different depths simultaneously in a compact form factor. In practice, see-through LCDs mounted close to the eye are light inefficient and result in significant diffraction artifacts, which are due to the electronic components in each pixel as well as the wiring of the display panel. This effect significantly degrades the observed image quality of any soft-edge or light field occlusion system.

Another approach for light field occlusion is presented in Yamaguchi and Takaki (2016) using concepts of integral imaging systems. This system has a very narrow field of view (4.3 degrees) and is fundamentally limited by the spatio-angular resolution tradeoff as well as diffraction.

As opposed to any of these methods, the proposed varifocal occlusion approach achieves hard-edge occlusion at varying distances in the scene at high resolution, with better light efficiency, and using technology components that make it easily compatible with emerging varifocal near-eye display.

4.2.2.6 Varifocal Occlusion

Concurrently and independently of our work, Hamasaki and Itoh (2019) also developed a strategy for a varifocal occlusion-capable AR display. Unlike our approach that builds on focus-tunable optics to dynamically adjust the depth of the occlusion layer, their approach requires mechanical motion of the occlusion SLM. Each approach has certain benefits and limitations. For example, robust calibration of the mechanically moving parts in their approach can be challenging, especially in a wearable display form factor. Our approach, on the other hand, requires focus-tunable optics, such as liquid lenses or Alvarez lenses (see Sec. 4.2.1).

4.2.3 Consistent Colors, Shading, and Shadows in AR

Spatial AR systems and optical see-through AR display often aim at providing radiometrically consistent, color-corrected or even color-stylized imagery Bimber et al. (2008); Wetzstein et al. (2010); Langlotz et al. (2016); Langlotz et al. (2018); Itoh et al. (2019). All of these approaches are successful in enhancing

the viewing experience in AR, but none of them tackle the problem of mutually consistent occlusions in optical see-through AR displays.

4.3 Optical Design

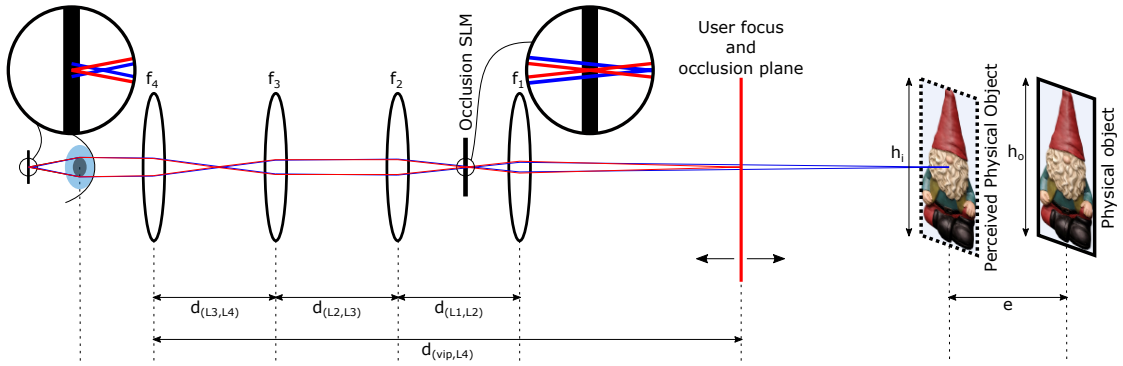

Figure 4.3: Illustration of the unfolded optical path of a 4-lens system for image-forming occlusion-capable AR displays. With a varifocal display, the distance of the virtual image and occlusion mask matches the user's focus distance, indicated by the thick vertical red line. Red and blue lines going from points in the scene through the optics onto the retina indicate ray diagrams for the image formation of the virtual/occlusion image and physical objects, respectively. Enlarged inset at the occlusion SLM shows that the physical world at the user's focal plane is brought into focus at the SLM where portions of the real world can be occluded. Enlarged inset at the retina shows that the same rays (red) that are in-focus at the occlusion SLM are also in-focus at the retina – this property is utilized to also depict a perceptually correct occlusion mask for out-of-focus virtual objects by applying a computational blur. Finally, the image of real-world objects seen through the display should ideally have the same magnification and distances from the eye as compared to seeing the real world without the display, i.e., $\frac{h_i}{h_o} = 1$ and $e = 0$. In our implementation, we can match the magnification, but not the distance.

Our goal is to design a varifocal occlusion-capable OST AR display that satisfies several key requirements. These include

1. The virtual image of the occlusion SLM, i.e., the occlusion mask, and the digital image should be optically placed together in the scene and their distance be dynamically adjustable.

2. The lateral and longitudinal magnification of the physical scene seen through the display should be equal to one, such that the experience is similar to viewing the scene without any optical elements.

3. No mechanical motion should be introduced to any component (lenses, SLM, etc.) to adjust the distance of its virtual image. Instead, the virtual image should be moved by changing the focal powers of the employed lenses.

In the following, we first provide an overview of the optical design we consider, introduce a ray-transfer matrix analysis of prior work on fixed-focus occlusion-capable AR displays (see Sec. 4.2.2.3), and finally introduce our focus-tunable varifocal occlusion approach.

Overview of the optical design We consider an optical design composed of four lenses (see Fig. 4.3), whose respective functions are: The first lens brings the real world at a particular depth into focus at the SLM. This image is always flipped, similar to how the image of the real world that is formed on our retina inside our eyes is always flipped. The next two lenses re-invert the in-focus image at the SLM, similar to a 4f system. The last lens finally places the image back into the appropriate depth for comfortable viewing. Let us denote these lenses by L_1, L_2, L_3, L_4 (see Fig. 4.3).

The occlusion SLM can be placed in either of the image planes of the optical system. One is between L_1 and L_2, and the other is between L_3 and L_4. We place the occlusion SLM between L_1 and L_2 because it simplifies Eq. (4.13). The digital image SLM can also be placed in any of the image planes of the optical system. We choose to place it between L_1 and L_2 because in this case, we can treat both the occlusion SLM and the virtual SLM to be optically equivalent and derive just one set of conditions for both of them.

4.3.1 Modeling Fixed-focus Occlusion Masks

The light transport through optical components can be modeled using ray-transfer matrices. In this approach, a light ray is represented by a column vector composed of lateral distance (x) and angle of propagation (θ) with respect to the optical axis. The propagation of paraxial light rays through an optical component is modeled as the multiplication of the ray vector with a 2×2 ray-transfer matrix. Ray-transfer matrices are known for standard optical components, e.g. let us denote the ray-transfer matrix for a lens with focal length f by $\mathbf{M}$ and the ray-transfer matrix for free-space propagation with a distance d by $\mathbf{S}$. Then, $\mathbf{M}$ and $\mathbf{S}$ are given by:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix}, \qquad \mathbf{S} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix}. \tag{4.1}$$

The composite ray transfer matrix that models the propagation of light rays through a series of optical components is simply the multiplication of the various individual ray transfer matrices of each optical component.

For our optical design (Fig. 4.3), the composite ray transfer matrix is represented as:

$$\mathbf{T} = \mathbf{M}_4\mathbf{S}_{(L_3,L_4)}\mathbf{M}_3\mathbf{S}_{(L_2,L_3)}\mathbf{M}_2\mathbf{S}_{(L_1,L_2)}\mathbf{M}_1, \tag{4.2}$$

where $\mathbf{M}_i$ is the ray-transfer matrix describing L_i and $\mathbf{S}_{(L_i,L_j)}$ describes the free-space propagation between lenses L_i and L_j.

The above linear system of equations is composed of four equations and seven unknowns (four unknown focal lengths and three unknown distances). This is an ill-posed inverse problem. Instead of attempting to solve it directly, previous works relied on symmetry constraints, such that $f_1 = f_4$, $f_2 = f_3$, and $d_{(L_1,L_2)} = d_{(L_3,L_4)}$.

Previous works have explored mainly two choices for the composite ray transfer matrix.

Shifted Perspective. In this configuration, the virtual viewpoint is shifted to the front of the optical system. In other words, the first lens and the last lens form conjugate aperture planes. Another way to think of it is that the light field entering the optical system and the light field exiting the optical system are equivalent. Mathematically, this condition represents $\mathbf{T} = \mathbf{I}$, where $\mathbf{I}$ is the 2×2 identity matrix. Some of the earlier prototypes of Kiyokawa et al. (2000, 2001) and Cakmakci et al. (2004) had a shifted perspective.

Correct Perspective. In this configuration, a user looking through the optical system should see the exact same image of a physical scene behind it as if the optical system was absent. There is no shift in the viewpoint. Kiyokawa et al. (2003) proposed a folded optics design for achieving correct perspective. This condition was analyzed formally with ray-transfer matrix equations for the first time in the context of optical cloaking Choi and Howell (2014) and later applied to the problem of occlusion in AR displays Howlett and Smithwick (2017). Mathematically, this is represented via the ray-transfer matrix $\mathbf{T} = \mathbf{S}_{(L_1,L_4)}$.

While an OST AR display should ideally be made to satisfy the correct perspective constraint, the disadvantage in doing so is that the field of view of the optical system is much smaller, being at most equal to the field of view seen through the first lens' aperture from a viewing distance of the length of the optical system. This limitation is exacerbated in our implementation by the small aperture (1 cm) of our focus-tunable lenses. For this reason, we design and implement an optical system that satisfies the shifted-perspective constraint.

4.3.2 Modeling Varifocal Occlusion Masks

Consider the general system of linear equations for image-forming occlusion optical designs (Eq. (4.2)). Recall that solving this is not possible by simply analyzing the ray-transfer matrix equations because there are more unknowns than equations. Our approach is to apply an optimization approach to this problem. Gaining some insights from the optimization approach, we then revisit the ray-transfer matrices approach to derive closed-form solutions.

Both our approaches aim to satisfy these requirements:

1. the virtual image should be placed at a desired (but movable) distance.

2. the magnification of the see-through image of the real-world should be unity irrespective of the virtual image plane distance.

4.3.2.1 Optimization approach

The optimization approach needs to calculate the set of focal lengths that minimize the error in the magnification of the see-through view and the error in the virtual/occlusion image plane's depth.

To do this, we define an image formation model for OST occlusion-capable displays, a cost function for the errors, and apply known methods to minimize the error iteratively. We start off by assuming that all lenses are focus-tunable lenses.

Image Formation The image formation for the virtual and real-world is modeled by successive application of the Gaussian thin lens equations:

$$i = \frac{of}{o - f}, \tag{4.3}$$

where i is the image distance, o is the object distance, and f is the focal length of the lens.

For an optical system composed of multiple lenses, the object of the subsequent lens (L_{j+1}) is the image of the previous lens (L_j). So, the object distance for L_{j+1} is: $o_{j+1} = d_{(L_j, L_{j+1})} - i_j$.

Occlusion and Virtual Image Formation For the occlusion and virtual image, the objects are the occlusion and virtual SLMs which are optically placed together by design. Only the lenses between these SLMs and the eye (L_2, L_3, L_4) contribute to the virtual/occlusion image formation. So, the object distance

($d_{(\text{SLM},L_2)}$) is propagated through lenses L_2, L_3, and L_4, to obtain the distance to the perceived occlusion/virtual image plane from lens 4 ($d_{(\text{vip},L_4)}$). Let us denote this image formation function by:

$$[d_{(\text{vip},L_4)}] = I_V(f_2, f_3, f_4, d_{(\text{SLM},L_2)}, d_{(L_2,L_3)}, d_{(L_3,L_4)}), \tag{4.4}$$

where I_V is composed of the successive application of Eq. (4.3), beginning with

$$i_2 = \frac{d_{(\text{SLM},L_2)} f_2}{d_{(\text{SLM},L_2)} - f_2}, \qquad o_3 = d_{(L_2,L_3)} - i_2, \tag{4.5}$$

and ending with:

$$d_{(\text{vip},L_4)} = \frac{o_4 f_4}{o_4 - f_4}. \tag{4.6}$$

See-Through Image Formation For the real-world, we first discretize the real-world into N real-world depth planes, where the number N is chosen such that the system samples the real-world denser than the human eye's depth-of-field which has been measured to be 0.3 diopters Campbell (1957); Watt et al. (2005). So, for a display whose nearest and farthest depth planes are at $D_{(R_{\text{near}},L_1)}$ diopters and $D_{(R_{\text{far}},L_1)}$ diopters respectively, the minimum number of discretized real-world depth planes should be:

$$N > \frac{D_{(R_{\text{near}},L_1)} - D_{(R_{\text{far}},L_1)}}{0.3}. \tag{4.7}$$

Each real-world depth ($d_{(R_j,L_1)}$) is propagated through lenses L_1, L_2, L_3, L_4 from which we get the see-through image depth from L_4 ($d_{(V_j,L_4)}$):

$$\begin{aligned}&[d_{(V_1,L_4)}, d_{(V_2,L_4)}, ..., d_{(V_N,L_4)}] = \\ &I_R(f_1, f_2, f_3, f_4, d_{(L_1,L_2)}, d_{(L_2,L_3)}, d_{(L_3,L_4)}, d_{(R_1,L_1)}, d_{(R_2,L_1)}, ..., d_{(R_N,L_1)}),\end{aligned} \tag{4.8}$$

where I_R is a successive application of Eq. (4.3) for each discretized real-world depth plane beginning with:

$$i_1 = \frac{d_{(R_j,L_1)} f_1}{d_{(R_j,L_1)} - f_1}, \qquad o_2 = d_{(L_1,L_2)} - i_1, \tag{4.9}$$

and ending with:

$$d_{(V_j,L_4)} = \frac{o_4 f_4}{o_4 - f_4}. \tag{4.10}$$

Error function The error associated with the occlusion/virtual image is the difference between desired occlusion/virtual image plane depth (d_{in}) and actual occlusion/virtual image plane depth ($d_{(\text{vip},L_4)}$) calculated as: $d_{in} - d_{(\text{vip},L_4)}$.

The error associated with the magnification of the physical scene is the difference between one and the magnification of the see-through image, where magnification is calculated as $m = -\frac{\text{image distance}}{\text{object distance}}$. However, in calculating the magnification, we need to be careful about what we consider as the object distance: Recall that in the see-through image formation function (Eq. (4.8)), we've defined the real-world object distances with respect to the first lens ($d_{(R_j,L_1)}$), whereas the final image distance is calculated with respect to the last lens ($d_{(V_j,L_4)}$). This discrepancy is alright when the optical system is designed to satisfy the shifted-perspective constraint. However, for the correct-perspective constraint, the object distance should be modified to $d_{(R_j,L_1)} + d_{(L_1,L_4)}$. For our display, where the correct object distance is $d_{(R_j,L_1)}$ and the magnification is given by $m_j = -\frac{d_{(V_j,L_4)}}{d_{(R_j,L_1)}}$

The combined error vector is given below:

$$E = \begin{bmatrix} d_{\text{in}} - d_{(\text{vip},L_4)} \\ 1 - m_1 \\ 1 - m_2 \\ \dots \\ 1 - m_N \end{bmatrix}. \tag{4.11}$$

The optimization problem is to find a set of focal lengths (f_1, f_2, f_3, f_4) that minimize the above error:

$$\underset{f_1,f_2,f_3,f_4}{\text{argmin}} \; ||E||^2. \tag{4.12}$$

Our implementation of this indicates that the set of focal lengths that minimizes the above error function always has a fixed f_2 and f_3.

Unfortunately, the execution time of this optimization is not real-time. We could calculate the dynamic values of f_1 and f_4 for different occlusion mask distances and use the calculated values in a look-up table to get real-time performance. Alternatively, we could use the new information that a fixed f_2 and f_3 can satisfy all the requirements to calculate closed-form solutions, as discussed below.

4.3.2.2 Closed-form solutions

Consider the same 4-lens optical design for a varifocal occlusion-capable display composed of the following parameters: $f_1^{(t)}, f_2, f_3, f_4^{(t)}, d_{(L_1,L_2)}, d_{(L_2,L_3)}, d_{(L_3,L_4)}, d_{(\text{SLM},L_1)}$, where the superscript $.^{(t)}$ indicates a dynamically changing parameter.

Using the Gaussian thin lens equation (Eq. (4.3)), $f_1^{(t)}$ is calculated based on the desired virtual image plane distance ($d_{(\text{vip},L_1)}^{(t)}$) and the distance between L_1 and the occluding SLM ($d_{(\text{SLM},L_1)}$):

$$f_1^{(t)} = \frac{d_{(\text{vip},L_1)}^{(t)} d_{(\text{SLM},L_1)}}{d_{(\text{vip},L_1)}^{(t)} + d_{(\text{SLM},L_1)}}. \tag{4.13}$$

Solving for the rest of the parameters needs an analysis of the ray-transfer matrix equation. To satisfy the shifted-perspective condition, the ray-transfer matrix needs to satisfy:

$$\mathbf{I} = \mathbf{M}_4^{(t)} \mathbf{S}_{(L_3,L_4)} \mathbf{M}_3 \mathbf{S}_{(L_2,L_3)} \mathbf{M}_2 \mathbf{S}_{(L_1,L_2)} \mathbf{M}_1^{(t)}. \tag{4.14}$$

Finding optical parameter values that satisfy the above equation automatically ensures that the requirements listed in the beginning of Sec. 4.3.2 will be satisfied. Since we have learned from our optimization experiments that solutions exists where L_2 and L_3 are fixed-focal length lenses, we solve Eq. (4.14) for $\mathbf{M}_4^{(t)}$ and analyze the conditions that ensure that the constants of matrix $\mathbf{M}_4^{(t)}$ (i.e., the ones and zero of $\mathbf{M}_4^{(t)}$) are their appropriate values:

$$\mathbf{M}_4^{(t)} \overset{a}{=} \begin{bmatrix} \frac{1+BC}{\frac{C}{f_1^{(t)}}+A} & C \\ B & \frac{C}{f_1^{(t)}} + A \end{bmatrix} \overset{b}{=} \begin{bmatrix} 1 & 0 \\ -\frac{1}{f_4^{(t)}} & 1 \end{bmatrix}, \tag{4.15}$$

where $\overset{a}{=}$ is obtained by solving Eq. (4.14) for $\mathbf{M}_4^{(t)}$ and $\overset{b}{=}$ is obtained because $\mathbf{M}_4^{(t)}$ should have the ray-trasfer matrix for a lens, and where A, B, C are the following:

$$A = 1 - \frac{d_{(L_3,L_4)} + d_{(L_2,L_3)} \left(1 - \frac{d_{(L_3,L_4)}}{f_3}\right)}{f_2} - \frac{d_{(L_3,L_4)}}{f_3}, \tag{4.16}$$

$$B = \frac{1 - \frac{d_{(L_2,L_3)}}{f_3} - d_{(L_1,L_2)} \left(\frac{1 - \frac{d_{(L_2,L_3)}}{f_3}}{f_2} + \frac{1}{f_3} \right)}{f_1^{(t)}} + \frac{1 - \frac{d_{(L_2,L_3)}}{f_3}}{f_2} + \frac{1}{f_3}, \tag{4.17}$$

$$C = d_{(L_2,L_3)} \left(1 - \frac{d_{(L_3,L_4)}}{f_3} \right) + d_{(L_3,L_4)} + d_{(L_1,L_2)} A. \tag{4.18}$$

From Eq. (4.15), we can infer that $C = 0$, and thereby, we can derive that $A = 1$ by substituting $C = 0$ in:

$$1 = \frac{C}{f_1^{(t)}} + A. \tag{4.19}$$

Re-arranging Eq. (4.16) by substituting $A = 1$:

$$-\frac{f_2}{f_3} = 1 + \frac{d_{(L_2,L_3)}}{d_{(L_3,L_4)}} \left(1 - \frac{d_{(L_3,L_4)}}{f_3} \right). \tag{4.20}$$

Re-arranging Eq. (4.18) by substituting $C = 0$ and $A = 1$:

$$-\frac{d_{(L_1,L_2)}}{d_{(L_3,L_4)}} = 1 + \frac{d_{(L_2,L_3)}}{d_{(L_3,L_4)}} \left(1 - \frac{d_{(L_3,L_4)}}{f_3} \right). \tag{4.21}$$

This gives us the condition that:

$$\frac{d_{(L_1,L_2)}}{d_{(L_3,L_4)}} = \frac{f_2}{f_3}. \tag{4.22}$$

$d_{(L_2,L_3)}$ can be derived by re-arranging Eq. (4.18) and substituting $d_{(L_1,L_2)} = \frac{d_{(L_3,L_4)} f_2}{f_3}$, $A = 1$, and $C = 0$:

$$\begin{aligned} -\frac{f_2 d_{(L_3,L_4)}}{f_3} &= d_{(L_3,L_4)} + d_{(L_2,L_3)} \left(1 - \frac{d_{(L_3,L_4)}}{f_3} \right) \\ \implies d_{(L_2,L_3)} &= \frac{d_{(L_3,L_4)} \left(1 + \frac{f_2}{f_3} \right)}{\frac{d_{(L_3,L_4)}}{f_3} - 1}. \end{aligned} \tag{4.23}$$

$d_{(L_2,L_3)}$ has to be positive. This gives us an improved version of the condition in Eq. (4.22):

$$\frac{d_{(L_3,L_4)}}{f_3} = \frac{d_{(L_1,L_2)}}{f_2} > 1. \tag{4.24}$$

$f_4^{(t)}$ is primary calculated from Equations (4.15) and (4.17):

$$f_4{}^{(t)} = -\frac{1}{B}. \tag{4.25}$$

Summary Here are steps that can be taken to arrive at the static parameters of the optical design:

1. Using Eq. (4.24), choose any three among $d_{(L_1,L_2)}, d_{(L_3,L_4)}, f_2, f_3$ and calculate for the fourth parameter. This choice can be based on the available fixed-focus lenses for f_2 and f_3 or based on constraints placed upon $d_{(L_1,L_2)}$ and $d_{(L_3,L_4)}$ by the hardware prototype. Although $d_{(\text{SLM},L_1)}$ doesn't feature in any of the conditions that we've derived, it should also be considered carefully in this step because it influences $d_{(L_1,L_2)}$ in that $d_{(L_1,L_2)} > d_{(\text{SLM},L_1)}$.

2. $d_{(L_2,L_3)}$ can now be calculated using Eq. (4.23).

During the operation of the display, the dynamic parameters ($f_1^{(t)}$ and $f_4^{(t)}$) are calculated using Equations (4.13) and (4.25) which are in turn dependent on only one dynamic value which is the virtual image distance ($d^{(t)}_{(\text{vip},L_1)}$).

Again, these equations ensure Eq. (4.14) which means that the see-through image of the real world would have unit magnification, although with a longitudinal shift which is equal to the length of the optical system from L_1 to L_4.

4.4 Implementation

We demonstrate varifocal occlusion with a monocular benchtop prototype (see Fig. 4.4 (A)). Optical design details and components details are discussed in the following.

Optical Design. To minimize distortion and chromatic aberrations in the prototype, all fixed-focus lenses (L_2, L_3) in our prototype are Nikon Nikkor 35-mm f/2 camera lenses. We use a 30-mm cage polarizing beamsplitter cube (ThorLabs CCM1-PBS251) to combine the real-world view after occlusion and the digital image. This design choice and the bulkiness of the Nikon imaging lenses constrains $d_{(L_1,L_2)}$ to a minimum of 10 cm. With this choice of parameters, and for an augmented scene whose minimum and maximum occlusion/virtual image plane depths are 30 cm and 300 cm, respectively, we obtain $f_1^{(t)}$ to lie in the range 25–28.5 diopters and $f_4^{(t)}$ in the range 2.64–5.67 diopters by using our closed-form solutions (Sec. 4.3.2.2). However, neither of these ranges of optical powers is directly supported by the

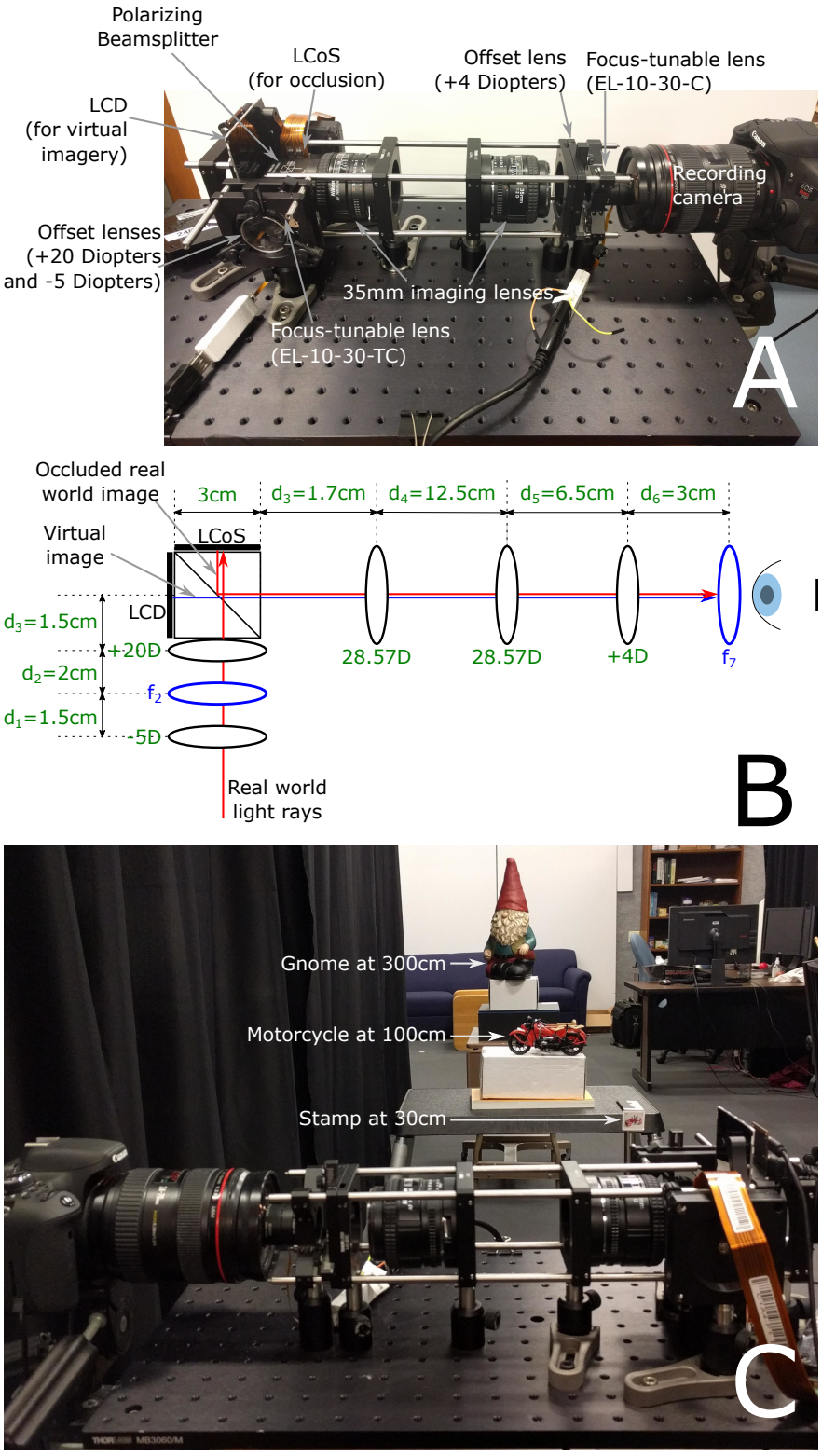

Figure 4.4: *(A)* Photo of our varifocal occlusion-capable AR display *(B)* Optical design of the prototype. Static design parameters are denoted in green. Propagation of real-world light through the system is depicted with red arrows. Propagation of the virtual image is depicted with blue arrows. The arrows are only representative of the general direction of propagation and do not depict the exact path taken by the light rays. *(C)* Photo of lab set-up which shows the prototype and the three real objects: stamp at 30cm, motorcycle at 100cm, and gnome at 300cm.

d_{om}	3.33	3.03	2.73	2.43	2.13	1.83	1.53	1.23	0.93	0.63	0.33
f_2	17.9	17.7	17.5	17.3	17.0	16.8	16.5	16.3	16.0	15.8	15.5
f_7	6.47	6.77	7.06	7.36	7.66	7.96	8.26	8.55	8.85	9.15	9.45

Table 4.2: Focus settings of the focus-tunable lenses for each setting of the occlusion mask distance (d_{om}) modeled in our optimization routine for the prototype display shown in Fig. 4.4. All values are in units of diopters.

focus-tunable lenses. The focus-tunable lenses in our prototype are Optotune EL-10-30-TC whose focal range is 8.3–20 diopters and Optotune EL-10-30-C whose focal range is 5–10 diopters.

Additional offset lenses are necessary to bring the operating range of optical powers into the supported range. The combined lens power (D_{combined}) of a focus-tunable lens ($D^{(t)}_{\text{tunable}}$) and an offset lens (D_{offset}) is theoretically $D_{\text{combined}} = D_{\text{offset}} + D^{(t)}_{\text{tunable}}$. In practice, however, we cannot place the offset lens exactly on top of the focus-tunable lens, so it is necessary to modify the composite ray-transfer matrix equations to additionally model the free-space propagation between offset and focus-tunable lenses.

Adding offset lenses changes the composite ray-transfer matrix and solving the equations analytically is tedious. Instead, we used the optimization based method (Sec. 4.3.2.1) because it is easy to introduce additional offset lenses in Eqs. 4.4 and 4.8 rerun the optimization. The resulting optical design is shown in Fig. 4.4.

Optimization. Our display's nearest depth plane is $D_{R_{\text{near}},L_1} = 3.33$ diopters and the farthest distance is $D_{R_{\text{far}},L_1} = 0.33$ diopters. The number of discretized real-world depth planes (N) considered for optimization can be calculated using Eq. (4.7) to be at-least 11 planes. The software for our optimization framework is implemented in Python using the package SciPy and the optimization function used is *differential_evolution*. The best optimization result out of 10 trials is chosen as the final optimization result. Tables 4.2 shows the focal lengths of the focus-tunable lenses calculated using our optimization approach. The optimization for each virtual image plane distance (i.e. each column of Table 4.2) takes about 4 seconds.

Displays. For the occlusion SLM, we use a reflection mode *liquid crystal on silicon* (LCoS) modulator (Silicon Micro Display ST1080) with a resolution of 1,920 $\times$ 1,080 and a screen diagonal of 0.74". For the digitally superimposed imagery, we use a liquid crystal display (LCD, Topfoison TF60010A) with a resolution of 2,560 $\times$ 1,440 pixels and a screen diagonal of 5.98". Both of these displays are placed at the same optical distance with respect to the user/camera. The pixel density of the LCD is much lower

than that of the LCoS panel, which results in pixelated virtual imagery, observed in Figures 4.1, 4.5. An additional polarizer was placed on top of the virtual image's LCD panel and manually adjusted to reduce its brightness enough to match with the real world's brightness.

Real-time System. The software for real-time rendering of the occlusion and virtual images is implemented in C++ using OpenGL/GLSL. Multi-pass shaders implement rendering of the RGB image and linearized depth map of the scene, which is used to calculate the depth-dependent computational blur for the occlusion and virtual image. The PC controlling the displays and the focus-tunable lenses uses an Intel Xeon E5-2630 2.4 GHz processor with an NVIDIA GeForce GTX 980 running Windows 7.

Recording Setup. An augmented reality scene was set up as shown in Figure 4.4 (C) and it is composed of three real objects: a stamp placed at 30 cm, a toy motorcycle placed at 100 cm, and a garden gnome placed at 300 cm. The scene seen through the display includes several digitally superimposed objects, i.e. one virtual object placed adjacent to each physical object. A Canon T6i Rebel camera with a Canon 24-70 mm f/2.8 lens is used to capture photographs through the display. For each see-through view presented in this chapter (Figs. 4.1, 4.5, 4.6), the camera settings were: 70 mm, f/14, ISO-1600, 0.6 s exposure time.

Emulating different AR and occlusion displays. In addition to demonstrating varifocal occlusion, our display is capable of emulating previous AR display technologies that differ from each other in terms of whether or not they provide accommodation support or occlusion support. We utilize this to compare different AR technologies. Here are the four major types of previous AR displays we compare, and the method by which these technologies are emulated:

- **Fixed-focus AR without occlusion**: Current commercially available AR displays present a fixed-focus virtual image without support for occlusion. These displays are emulated by setting our prototype to always present an image at the farthest virtual image plane distance and by setting the occlusion image to full white (reflects as much of the incident light as possible).

- **Varifocal AR without occlusion**: These displays are emulated by dynamically adjusting the focal lengths of the focus-tunable lenses for the given virtual image plane distance, and by applying a computational blur that mimics the perceived defocus blur to the virtual objects that are supposed to be defocused. Occlusion support is turned off by setting the occlusion image to full white.

- **Fixed-focus AR with fixed-focus occlusion**: Previous prototypes of hard-edge occlusion always present the occlusion and virtual imagery at a far distance. These displays are emulated by setting our prototype to always present the image at a far distance while displaying a silhouette of the virtual objects as the occlusion mask.

- **Varifocal AR with varifocal occlusion**: Our proposed display technology dynamically adjusts the focal lengths of the focus-tunable lenses for the given virtual image plane distance, and by applying a computational blur to the virtual objects that are supposed to be defocused. The varifocal occlusion mask is computed by applying a similar computational blur to the silhouette of the virtual image.

4.5 Results

4.5.1 See-through images

Figures 4.1 and 4.5 show a comparison of the see-through view of different AR and occlusion technologies. In each of these figures, the augmented scene is composed of real-world objects and virtual objects placed at different distances. At each distance, one virtual object is placed slightly in front of the real-world object to demonstrate our display's ability to occlude real-world objects. The mechanism by which the different occlusion and AR displays are emulated is explained in Sec. 4.4. The see-through view for the different AR and occlusion technologies are shown column-wise:

- **Column One:** Emulates commercially available AR displays. In these displays, the virtual imagery looks transparent and is placed at a fixed distance, which does not provide the user with important depth cues like occlusion or accommodation.

- **Column Two:** Emulates varifocal AR displays. The virtual image plane is movable in these displays and should be designed to match the user's focal distance. A computational blur can be applied optionally to virtual content that is out-of-focus with the focal distance. The improvement over commercially available AR displays is that accommodation cues are provided in a perceptually correct manner, but these displays still lack the ability to show the most important depth cue, namely occlusion Cutting and Vishton (1995).

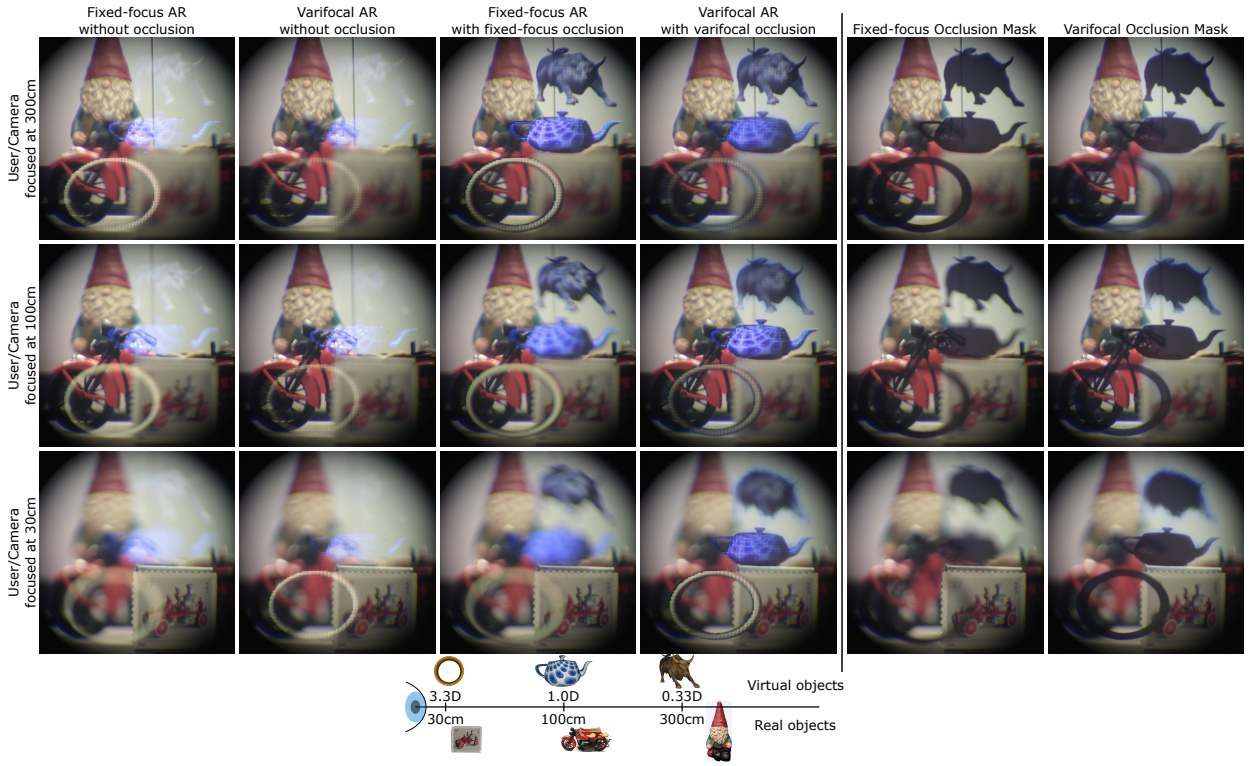

Figure 4.5: **Left of the vertical line:** views through our prototype AR display, which is emulating different AR display technologies for each column. The augmented scene is composed of real-world objects (stamp, motorcycle, and gnome) and virtual objects (ring, teapot, and bull). Objects are distributed at different depths: stamp and ring at 30cm, motorcycle and teapot at 100cm, and gnome and bull at 300cm. *(Column 1)* Commercially available AR displays: a transparent virtual image is presented at a fixed distance. Important depth cues such as occlusion and accommodation are absent. *(Column 2)* Varifocal AR displays: virtual image can be moved to different depths, but images are still transparent. *(Column 3)* Fixed-focus occlusion-capable AR display: Occlusion and virtual images are fixed at a single depth, limiting realism when the user is focused to other depths. Note how all virtual objects, including the nearby ones, are in focus when the camera is focused far, and all virtual objects are defocused when the camera is focused near. *(Column 4)* Varifocal occlusion-capable AR displays: virtual and occlusion image plane can be moved to different depths enabling perceptually correct depth cues for occlusion and accommodation. Note how objects at the same depth, e.g., near objects (stamp and ring) or far objects (gnome and bull), are correctly in focus or defocused depending on the focus state of the user/camera. **Right of the vertical line:** Comparison of occlusion masks between fixed-focus and varifocal occlusion-capable displays.

- **Column Three:** Emulates fixed-focus occlusion-capable AR displays. In these displays, occlusion of real objects by virtual objects can be displayed, but the occlusion mask and virtual image are always displayed at a fixed depth, which reduces the realism for virtual objects located at other depths. Note how in Fig. 4.5, all three virtual objects, namely ring, teapot, and bull are in-focus when the camera is focused far, and all three objects are defocused when the camera is focused at other distances.

- **Column Four:** Demonstrates our varifocal occlusion-capable AR display. Our display is able to move the occlusion and virtual image planes to different distances, and hence, is able to provide depth-dependent occlusion and accommodation depth cues. Note how in Fig. 4.5, the camera correctly records only one virtual and one real object in-focus at each focus setting.

- **Columns Five and Six:** Comparison of only the occlusion masks for fixed-focus and varifocal occlusion displays.

4.5.2 Quality of real world magnification

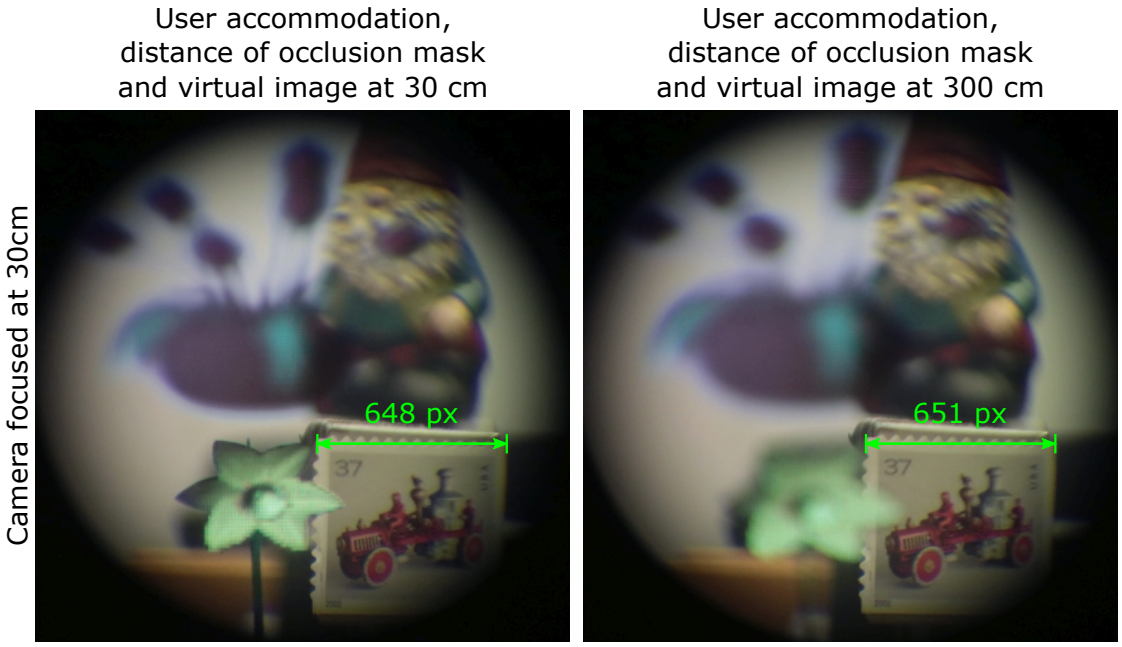

Figure 4.6: View through our prototype occlusion-capable AR display for different settings of occlusion/virtual image plane depth with camera focus fixed on foreground. The user fixates the foreground objects (left) and background objects (right) and the virtual image distance and occlusion mask are following their fixation distance. The camera remains focused on the foreground object, demonstrating that changing optical settings of the display do not change the magnification of the physical scene, as indicated by the stamp's size.

For any AR display, whether occlusion-capable or not, the magnification of see-through images of the real world should be unity irrespective of the virtual image plane distance. Ensuring this property is particularly challenging for a varifocal occlusion-capable AR display. Section 4.3.2.1 and 4.3.2.2 discuss complementary strategies to ensure this. Our prototype display shown in Fig. 4.4 was designed using the optimization approach (Sec. 4.3.2.1). For different settings of the occlusion or virtual image plane distance,

d_{rw} \ d_{om}	3.33	3.03	2.73	2.43	2.13	1.83	1.53	1.23	0.93	0.63	0.33
3.33	0.93	0.94	0.95	0.95	0.96	0.97	0.98	0.99	1	1.01	1.02
3.03	0.93	0.94	0.95	0.96	0.96	0.97	0.98	0.99	1	1.01	1.02
2.73	0.93	0.94	0.95	0.96	0.96	0.97	0.98	0.99	1	1.01	1.02
2.43	0.93	0.94	0.95	0.96	0.97	0.97	0.98	0.99	1	1.01	1.02
2.13	0.94	0.94	0.95	0.96	0.97	0.98	0.98	0.99	1	1.01	1.01
1.83	0.94	0.95	0.96	0.96	0.97	0.98	0.98	0.99	1	1.01	1.01
1.53	0.95	0.95	0.96	0.97	0.97	0.98	0.99	0.99	1	1.01	1.01
1.23	0.95	0.96	0.97	0.97	0.98	0.98	0.99	0.99	1	1	1.01
0.93	0.97	0.97	0.98	0.98	0.98	0.99	0.99	1	1	1	1.01
0.63	0.99	0.99	1	1	1	1	1	1	1	1	1
0.33	1.07	1.07	1.06	1.05	1.04	1.03	1.02	1.01	1	0.99	0.98

Table 4.3: Magnification predicted by our optimization routine for each real world distance (d_{rw}) propagated through the optical system for each setting of the occlusion mask distance (d_{om}) for the prototype display shown in Fig. 4.4. Distances (d_{om} and d_{rw}) are in diopters. Note that all magnification values are close to 1.0, indicating good optimization quality.

Tables 4.2 and 4.3 show the focal length settings of the focus-tunable lenses and the magnification of the see-through image respectively.

Note that the optimization approach (Sec. 4.3.2.1) requires a discretization of only the real-world distances, but accepts continuously changing values for the occlusion mask. Tables 4.2 and 4.3 are calculated for a finite set of occlusion mask distances only to indicate the performance of the display for different occlusion mask distance settings.

Table 4.3 shows that the optimization predicts that the see-through image magnification values are close to unity, but not exactly equal to unity. Using the closed-form approach would have ensured exact unit magnification for all combinations of real-world distance and virtual image plane distance, however, as discussed in Sec. 4.4, due to some hardware constraints, the focal range predicted by the closed-form solutions is unattainable with the focus-tunable lenses at our disposal. Hence, the best we can do currently is the solution predicted by the optimization routine. A similar table could be shown for the other error considered in the optimization approach, i.e., the error in the occlusion or virtual image plane distance (see Eq. 4.11), however, we omit this because these errors are negligible (always less than one centimeter).

We verify the quality of real-world magnification of our prototype by capturing see-through images of our display for different display focus settings for a fixed camera focus distance (see Fig. 4.6). In the left subfigure, the user is assumed to fixate the daffodil in the foreground. In this setting, the other flower pot is blurred due to the computational blur that emulates perceived defocus blur. The camera is also focused on the foreground objects. In the right subfigure, the user now fixates at an object at the farther distance, and the virtual image distance along with the occlusion mask are updated to the farther distance. We intentionally kept the camera focus on the foreground object to highlight the fact that refocusing the virtual image and the occlusion mask does not change the magnification of the physical scene noticeably. This is highlighted by the size of the stamp being roughly constant. Note that the user would never see the camera image shown in the right subfigure because, in a varifocal display, the distance of the object they fixate is the same as the virtual image distance. Nevertheless, this experiment demonstrates our prototype display's capability to maintain constant magnification of the real-world independent of the virtual image distance.

4.5.3 Display specifications

The display's field of view is 15.3°. The supported occlusion/virtual image plane depth is from optical infinity to 30 cm. In our results, we do not include real or virtual objects beyond 300 cm because 300 cm seems equivalent to optical infinity for the display. The eyebox is about 1 cm, equal to the aperture of the last lens in the system.

4.6 Discussion

In summary, we introduce varifocal-occlusion capable AR displays based on focus-tunable optics. This approach improves the realism of optical see-through displays by enabling mutually consistent occlusions between digital and physical objects over a large depth range. We derive a formal optimization approach and real-time heuristics to tune the optical settings of our system to avoid distortions of the physical scene and demonstrate improved realism with a prototype AR display.

4.6.1 Limitations

Similar to other varifocal-type displays, ours requires eye-tracking to determine where to focus the display. Our current prototype does not include an eye tracker, although this capability has been demonstrated with previous varifocal VR displays Padmanaban et al. (2017). The field of view of our prototype is limited by the size of commercially available focus-tunable lenses, although these are steadily increasing Padmanaban et al. (2019). Finally, our prototype shares limitations of other, fixed-focus occlusion-capable AR displays in being implemented as a benchtop system. The main limitation for image-forming occlusion-capable augmented reality display remains their bulky form-factors. Optical path folding, and new methods of fabricating thin lenses using analog holograms and nanophotonics may reduce form-factors of future prototypes.

4.6.2 Future Work

First and foremost, the device form factor of this and other occlusion-capable displays should be reduced to enable wearable occlusion-capable displays. This is a major optical design challenge beyond the scope of this chapter. Eye-tracking should be incorporated into such a wearable system. While most occlusion-capable displays aim at computing a binary occlusion mask, one could also envision the attenuation pattern to be optimized to enable consistent illumination, shading, and shadows of digital and physical objects along with consistent occlusion Bimber et al. (2003) or enable other types of optical image processing capabilities Wetzstein et al. (2010).

4.6.3 Conclusion

To enable seamless experiences with AR displays, hard-edge occlusion control is critical. With this work, we take steps towards improving the realism of optical see-through displays with varifocal occlusion capabilities. Yet, many challenges remain to design and build small, light-weight AR glasses that offer perceptually realistic and seamless experiences.

CHAPTER 5: SUMMARY AND CONCLUSIONS

5.1 Summary

This dissertation was motivated by the lack of perceptually realistic depth cues in the current generation commercial augmented reality displays and research prototypes. This dissertation focuses on three particularly important depth cues, namely, accommodation, defocus blur, and mutual occlusion. We presented two augmented reality displays which present high-quality accommodation, defocus blur, and occlusion across a large depth-range.

Volumetric AR display, is a multifocal display with 280 single-color binary image planes – a significant improvement upon previous multifocal displays. The volumetric AR display can present full-color imagery (24 or higher bit-depth) spanning a large volume (15 cm to 400 cm with 45° Field-of-View) composed of 280 binary images, each of which has the native resolution of the display (1024×768). This dissertation discusses the optical design, synchronization electronics, and the graphics rendering pipeline. One of the stages of the graphics rendering pipeline is the decomposition of the color-volume to the binary-volume. This dissertation develops multiple decomposition schemes — one fixed-pipeline decomposition and multiple optimization-based methods. While most of the results were obtained using an offline implementation of the graphics rendering pipeline, a simple real-time system composed of 8 single-color binary image planes was implemented and was demonstrated with a video recording.

Varifocal occlusion display, is an extension of fixed-focus occlusion displays, and enables a single occlusion image plane to be moveable in depth. This dissertation asserts that extending fixed-focus occlu-

Depth-cue	Volumetric AR display	Varifocal-occlusion AR display
Accommodation	All depths	Selectable depth
Defocus blur	Natural	Synthetic
Occlusion	None	Depth-dependent and Hard-edge

Table 5.1: Summary of contributions

sion displays to varifocal occlusion displays requires a solution to the following problem: that the tunable optics needed to move the occlusion/virtual image plane in depth also needs to transmit the image of the real-world undistorted. To solve this problem, this dissertation analyses the problem using concepts of light fields and uses ray-transfer matrix equations to derive optical designs using optimization and analytical derivation. A real-time system was built using off-the-shelf components and used to compare the proposed technology to previous AR display technologies.

Table 5.1 quantitatively compares the nature and performance of this dissertation's displays against the monocular depth cues considered here.

5.2 Future Work

The immediate next research steps for the volumetric NED could be the real-time implementation of the proposed rendering pipeline. This is not a trivial improvement because of the large computation and communication demands that the system needs to address. However, addressing this large computation and communication demand should be possible with this NED because its components were originally designed for a low-latency AR display Lincoln et al. (2016). So, in addition to a real-time volumetric NED, future work could include developing a low-latency volumetric NED.

The ability of our varifocal occlusion-capable AR display to attenuate real-world light can also be used to depict consistent global-illumination in the AR scene and depict interesting effects such as shadows cast by virtual objects onto the real world and vice-versa, or to relight the real-world to match the virtual scene. The bulky form-factor of image-forming occlusion displays remains the key limitation, and addressing this is definitely an area for future work.

Both of the presented display technologies can also emulate multiple other AR displays, e.g., the volumetric NED can also emulate previous varifocal NEDs and previous multifocal NEDs, and the varifocal occlusion-capable NED can also emulate fixed-focus occlusion displays and occlusion incapable varifocal AR displays. Hence, this dissertation's displays could be used as test-beds to conduct perceptual experiments to understand the human visual system better and to come up with strategies for future NEDs.

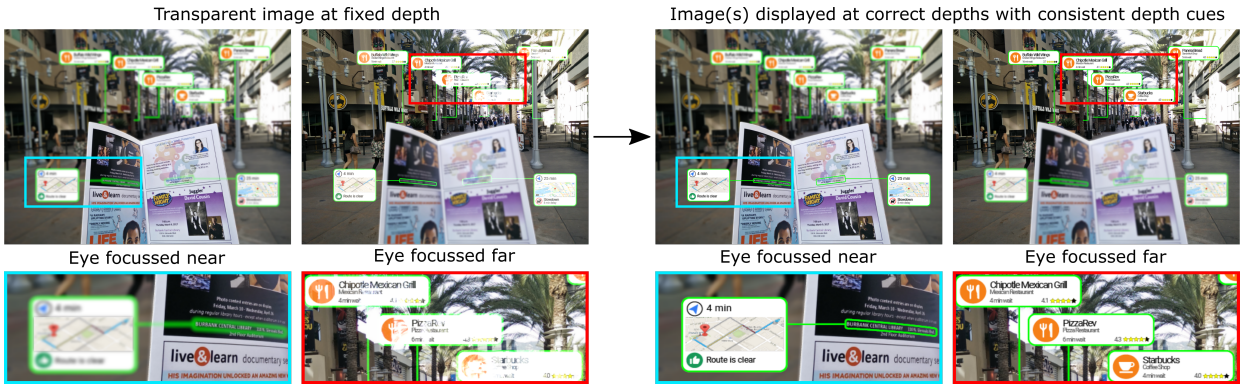

Figure 5.1: Figures shows the contributions of this dissertation with a concept diagram. We've take small steps to go from the current state-of-the-art (on the left) to the envisioned future for augmented reality displays (on the right). Insests at the bottom row show enlarged portions of the regions of interest in the concept image above them. Image credit: Adapted from David Dunn.

5.3 Conclusion

Fig. 5.1 shows a concept augmented reality scene. In this concept scene, the real scene is composed of a pamphlet in the foreground and shops and restaurants in the background. To this scene, the current generation augmented reality displays can superimpose a transparent image at a fixed distance. However, the ultimate display as envisioned for augmented reality will be able to display multiple images at their correct depths with perceptually consistent depths. This dissertation takes a few steps towards realizing this vision.

> *' The screen is a window through which one sees a virtual world. The challenge is to make that world look real, act real, sound real, feel real. '* Sutherland (1965)

Although the above quote is intended for Virtual Reality and considers multiple modalities (sight, hearing, haptics), it helps to convey the vision for Augmented Reality that I subscribe to. The ultimate Augmented Reality display would combine the real and the virtual worlds in a visually convincing manner—with consistent depth cues, latency, resolution, color fidelity, lighting, shadows, reflections, etc. Towards realizing this vision, this dissertation develops methods to improve monocular depth cues for Augmented Reality displays.

REFERENCES

Akşit, K., Lopes, W., Kim, J., Shirley, P., and Luebke, D. (2017). Near-eye Varifocal Augmented Reality Display Using See-through Screens. *ACM Trans. Graph.*, 36(6):189:1–189:13.

Akeley, K., Watt, S. J., Girshick, A. R., and Banks, M. S. (2004). A Stereo Display Prototype with Multiple Focal Distances. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 804–813, New York, NY, USA. ACM.

Akşit, K., Chakravarthula, P., Rathinavel, K., Jeong, Y., Albert, R., Fuchs, H., and Luebke, D. (2019). Manufacturing application-driven foveated near-eye displays. *IEEE transactions on visualization and computer graphics*, 25(5):1928–1939.

Avveduto, G., Tecchia, F., and Fuchs, H. (2017). Real-world occlusion in optical see-through ar displays. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, page 29. ACM.

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47.

Bimber, O. and Fröhlich, B. (2002). Occlusion shadows: Using projected light to generate realistic occlusion effects for view-dependent optical see-through displays. In *Proc. IEEE ISMAR*.

Bimber, O., Grundhöfer, A., Wetzstein, G., and Knödel, S. (2003). Consistent illumination within optical see-through augmented environments. In *Proc. IEEE ISMAR*, pages 198–207.

Bimber, O., Iwai, D., Wetzstein, G., and Grundhoefer, A. (2008). The Visual Computing of Projector-Camera Systems. *Computer Graphics Forum*.

Cakmakci, O., Ha, Y., and Rolland, J. (2005). Design of a compact optical see-through head-worn display with mutual occlusion capability. In *Proc. SPIE 5875*.

Cakmakci, O., Ha, Y., and Rolland, J. P. (2004). A compact optical see-through head-worn display with occlusion support. In *Proc. IEEE ISMAR*, pages 16–25.

Campbell, F. W. (1957). The depth of field of the human eye. *Optica Acta: International Journal of Optics*, 4(4):157–164.

Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., and Ivkovic, M. (2011). Augmented reality technologies, systems and applications. *Multimedia tools and applications*, 51(1):341–377.

Chakravarthula, P., Dunn, D., Akşit, K., and Fuchs, H. (2018). Focusar: Auto-focus augmented reality eyeglasses for both real world and virtual imagery. *IEEE transactions on visualization and computer graphics*, 24(11):2906–2916.

Choi, J. S. and Howell, J. C. (2014). Paraxial ray optics cloaking. *OSA Opt. Express*, 22(24):29465–29478.

Cholewiak, S. A., Love, G. D., Srinivasan, P. P., Ng, R., and Banks, M. S. (2017). Chromablur: Rendering chromatic eye aberration improves accommodation and realism. *ACM Transactions on Graphics (TOG)*, 36(6):1–12.

Cmglee (2019). Human photoreceptor distribution. https://commons.wikimedia.org/w/index.php?curid=29924570. Created: 09-July, 2019. Last checked: Feb-01, 2020.

Cossairt, O. S., Napoli, J., Hill, S. L., Dorval, R. K., and Favalora, G. E. (2007). Occlusion-capable multi-view volumetric three-dimensional display. *Appl. Opt.*, 46(8):1244–1250.

Cutting, J. E. and Vishton, P. M. (1995). Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In *Perception of space and motion*, pages 69–117. Elsevier.

Dewald, D. S., Evans, A. T., Welch, N., Gross, A., and Hill, G. (2016). The Avegant Glyph: Optical Design Considerations and Approach to Near-eye Display. *SID Symposium Digest of Technical Papers*, 47(1):69–71.

Dunn, D., Tippets, C., Torell, K., Kellnhofer, P., Akşit, K., Didyk, P., Myszkowski, K., Luebke, D., and Fuchs, H. (2017). Wide Field Of View Varifocal Near-Eye Display Using See-Through Deformable Membrane Mirrors. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1322–1331.

Favalora, G. E., Napoli, J., Hall, D. M., Dorval, R. K., Giovinco, M., Richmond, M. J., and Chun, W. S. (2002). 100-million-voxel volumetric display. In *Cockpit Displays IX: Displays for Defense Applications*, volume 4712, pages 300–313. International Society for Optics and Photonics.

Gao, C., Lin, Y., and Hua, H. (2012). Occlusion capable optical see-through head-mounted display using freeform optics. In *Proc. IEEE ISMAR*, pages 281–282.

Gao, C., Lin, Y., and Hua, H. (2013). Optical see-through head-mounted display with occlusion capability. In *Proc. SPIE 8735*.

Geng, J. (2013). Three-dimensional display technologies. *Advances in optics and photonics*, 5(4):456–535.

Hamasaki, T. and Itoh, Y. (2019). Varifocal occlusion for optical see-through head-mounted displays using a slide occlusion mask. *IEEE TVCG*, 25(5):1961–1969.

Hoffman, D. M., Girshick, A. R., Akeley, K., and Banks, M. S. (2008). Vergence – accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3):33.

Holliman, N. S., Dodgson, N. A., Favalora, G. E., and Pockett, L. (2011). Three-dimensional displays: a review and applications analysis. *IEEE transactions on Broadcasting*, 57(2):362–371.

Howard, I. P. and Rogers, B. J. (2002). *Seeing in Depth*. Oxford University Press.

Howlett, I. D. and Smithwick, Q. (2017). Perspective correct occlusion-capable augmented reality displays using cloaking optics constraints. *Journal of the Society for Information Display*, 25(3):185–193.

Hu, X. and Hua, H. (2014a). Design and Assessment of a Depth-Fused Multi-Focal-Plane Display Prototype. *Journal of Display Technology*, 10(4):308–316.

Hu, X. and Hua, H. (2014b). High-resolution optical see-through multi-focal-plane head-mounted display using freeform optics. *Opt. Express*, 22(11):13896–13903.

Hu, X. and Hua, H. (2015). Design and tolerance of a free-form optical system for an optical see-through multi-focal-plane display. *Appl. Opt.*, 54(33):9990–9999.

Hua, H. (2017). Enabling Focus Cues in Head-Mounted Displays. *Proceedings of the IEEE*, 105(5):805–824.

Hua, H. and Javidi, B. (2014). A 3D integral imaging optical see-through head-mounted display. *Opt. Express*, 22(11):13484–13491.

Huang, F.-C., Chen, K., and Wetzstein, G. (2015). The Light Field Stereoscope: Immersive Computer Graphics via Factored Near-eye Light Field Displays with Focus Cues. *ACM Trans. Graph.*, 34(4):60:1–60:12.

Huang, F.-C., Pajak, D., Kim, J., Kautz, J., and Luebke, D. (2017). Mixed-primary factorization for dual-frame computational displays. *ACM Trans. Graph.*, 36(4):149–1.

Itoh, Y., Hamasaki, T., and Sugimoto, M. (2017). Occlusion leak compensation for optical see-through displays using a single-layer transmissive spatial light modulator. *IEEE TVCG*, 23(11):2463–2473.

Itoh, Y., Langlotz, T., Iwai, D., Kiyokawa, K., and Amano, T. (2019). Light attenuation display: Subtractive see-through near-eye display via spatial color filtering. *IEEE TVCG*, 25(5):1951–1960.

Jang, C., Bang, K., Moon, S., Kim, J., Lee, S., and Lee, B. (2017). Retinal 3D: Augmented Reality Near-eye Display via Pupil-tracked Light Field Projection on Retina. *ACM Trans. Graph.*, 36(6):190:1–190:13.

Johnson, P. V., Parnell, J. A., Kim, J., Saunter, C. D., Love, G. D., and Banks, M. S. (2016). Dynamic lens and monovision 3d displays to improve viewer comfort. *OSA Opt. Express*, 24(11):11808–11827.

Jones, A., McDowall, I., Yamada, H., Bolas, M., and Debevec, P. (2007). Rendering for an Interactive 360° Light Field Display. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA. ACM.

Jones, B. R., Benko, H., Ofek, E., and Wilson, A. D. (2013). Illumiroom: peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 869–878.

Kanbara, M., Okuma, T., Takemura, H., and Yokoya, N. (2000). A stereoscopic video see-through augmented reality system based on real-time vision-based registration. In *Proceedings IEEE Virtual Reality 2000 (Cat. No. 00CB37048)*, pages 255–262. IEEE.

Kiyokawa, K., Billinghurst, M., Campbell, B., and Woods, E. (2003). An occlusion-capable optical see-through head mount display for supporting co-located collaboration. In *Proc. IEEE ISMAR*.

Kiyokawa, K., Kurata, Y., and Ohno, H. (2000). An optical see-through display for mutual occlusion of real and virtual environments. In *Proc. ISAR*, pages 60–67.

Kiyokawa, K., Kurata, Y., and Ohno, H. (2001). An optical see-through display for mutual occlusion with a real-time stereovision system. *Computers & Graphics*, 25(5):765–779.

Konrad, R., Angelopoulos, A., and Wetzstein, G. (2020). Gaze-contingent ocular parallax rendering for virtual reality. *ACM Trans. Graph.*, 39.

Konrad, R., Cooper, E. A., and Wetzstein, G. (2016). Novel optical configurations for virtual reality: Evaluating user preference and performance with focus-tunable and monovision near-eye displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1211–1220, New York, NY, USA. ACM.

Konrad, R., Padmanaban, N., Molner, K., Cooper, E. A., and Wetzstein, G. (2017). Accommodation-invariant computational near-eye displays. *ACM Trans. Graph.*, 36(4):88:1–88:12.

Kooi, F. L. and Toet, A. (2004). Visual comfort of binocular and 3d displays. *Displays*, 25(2-3):99–108.

Laffont, P.-Y., Hasnain, A., Guillemet, P.-Y., Wirajaya, S., Khoo, J., Teng, D., and Bazin, J.-C. (2018). Verifocal: A platform for vision correction and accommodation in head-mounted displays. In *ACM SIGGRAPH 2018 Emerging Technologies*, pages 21:1–21:2.

Lambooij, M., Fortuin, M., Heynderickx, I., and IJsselsteijn, W. (2009). Visual discomfort and visual fatigue of stereoscopic displays: A review. *Journal of Imaging Science and Technology*, 53(3):30201–1.

Langlotz, T., Cook, M., and Regenbrecht, H. (2016). Real-time radiometric compensation for optical see-through head-mounted displays. *IEEE TVCG*, 22(11):2385–2394.

Langlotz, T., Sutton, J., Zollmann, S., Itoh, Y., and Regenbrecht, H. (2018). Chromaglasses: Computational glasses for compensating colour blindness. In *Proc. SIGCHI*, pages 390:1–390:12.

Lanier, J., Mateevitsi, V., Rathinavel, K., Shapira, L., Menke, J., Therien, P., Hudman, J., Speiginer, G., Won, A. S., Banburski, A., et al. (2016). The realitymashers: Augmented reality wide field-of-view optical see-through head mounted displays. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 141–146. IEEE.

Lanier, J., Rathinavel, K., and Raghuvanshi, N. (2018). Virtually visualizing energy. US Patent 9,922,463.

Lanman, D., Hirsch, M., Kim, Y., and Raskar, R. (2010). Content-adaptive parallax barriers: Optimizing dual-layer 3d displays using low-rank light field factorization. In *ACM SIGGRAPH Asia*, pages 163:1–163:10.

Lanman, D. and Luebke, D. (2013). Near-eye Light Field Displays. *ACM Trans. Graph.*, 32(6):220:1–220:10.

Lee, S., Cho, J., Lee, B., Jo, Y., Jang, C., Kim, D., and Lee, B. (2018a). Foveated Retinal Optimization for See-Through Near-Eye Multi-Layer Displays. *IEEE Access*, 6:2170–2180.

Lee, S., Jo, Y., Yoo, D., Cho, J., Lee, D., and Lee, B. (2018b). Shape scanning displays: tomographic decomposition of 3d scenes. In *Digital Optics for Immersive Displays*, volume 10676, page 1067617. International Society for Optics and Photonics.

Lee, S., Jo, Y., Yoo, D., Cho, J., Lee, D., and Lee, B. (2018c). Tomoreal: Tomographic displays. *arXiv preprint arXiv:1804.04619*.

Lincoln, P. (2017). Low latency displays for augmented reality.

Lincoln, P., Blate, A., Singh, M., State, A., Whitton, M. C., Whitted, T., and Fuchs, H. (2017). Scene-adaptive High Dynamic Range Display for Low Latency Augmented Reality. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '17, pages 15:1–15:7, New York, NY, USA. ACM.

Lincoln, P., Blate, A., Singh, M., Whitted, T., State, A., Lastra, A., and Fuchs, H. (2016). From Motion to Photons in 80 Microseconds: Towards Minimal Latency for Virtual and Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 22(4):1367–1376.

Liu, S., Cheng, D., and Hua, H. (2008). An optical see-through head mounted display with addressable focal planes. In *Proc. IEEE ISMAR*, pages 33–42.

Liu, S. and Hua, H. (2009). Time-multiplexed dual-focal plane head-mounted display with a liquid lens. *Optics letters*, 34(11):1642–1644.

Liu, S. and Hua, H. (2010). A systematic method for designing depth-fused multi-focal plane three-dimensional displays. *Opt. Express*, 18(11):11562–11573.

Liu, S., Hua, H., and Cheng, D. (2010). A Novel Prototype for an Optical See-Through Head-Mounted Display with Addressable Focus Cues. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):381–393.

Love, G. D., Hoffman, D. M., Hands, P. J., Gao, J., Kirby, A. K., and Banks, M. S. (2009). High-speed switchable lens enables the development of a volumetric stereoscopic display. *Opt. Express*, 17(18):15716–15725.

MacKenzie, K. J., Hoffman, D. M., and Watt, S. J. (2010). Accommodation to multiple-focal-plane displays: Implications for improving stereoscopic displays and for accommodation control. *Journal of Vision*, 10(8):22.

Maimone, A. and Fuchs, H. (2013). Computational augmented reality eyeglasses. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 29–38. IEEE.

Maimone, A., Georgiou, A., and Kollin, J. S. (2017). Holographic Near-eye Displays for Virtual and Augmented Reality. *ACM Trans. Graph.*, 36(4):85:1–85:16.

Maimone, A., Lanman, D., Rathinavel, K., Keller, K., Luebke, D., and Fuchs, H. (2014). Pinlight Displays: Wide Field of View Augmented Reality Eyeglasses Using Defocused Point Light Sources. In *ACM SIGGRAPH 2014 Emerging Technologies*, SIGGRAPH '14, pages 20:1–20:1, New York, NY, USA. ACM.

Maimone, A., Yang, X., Dierk, N., State, A., Dou, M., and Fuchs, H. (2013). General-purpose telepresence with head-worn optical see-through displays and projector-based lighting. In *2013 IEEE Virtual Reality (VR)*, pages 23–26. IEEE.

Matsuda, N., Fix, A., and Lanman, D. (2017). Focal Surface Displays. *ACM Trans. Graph.*, 36(4):86:1–86:14.

Mercier, O., Sulai, Y., Mackenzie, K., Zannoli, M., Hillis, J., Nowrouzezahrai, D., and Lanman, D. (2017). Fast Gaze-contingent Optimal Decompositions for Multifocal Displays. *ACM Trans. Graph.*, 36(6):237:1–237:15.

Mori, S., Ikeda, S., Plopski, A., and Sandor, C. (2018). Brightview: Increasing perceived brightness of optical see-through head-mounted displays through unnoticeable incident light reduction. In *Proc. IEEE VR*, pages 251–258.

Narain, R., Albert, R. A., Bulbul, A., Ward, G. J., Banks, M. S., and O'Brien, J. F. (2015). Optimal Presentation of Imagery with Focus Cues on Multi-plane Displays. *ACM Trans. Graph.*, 34(4):59:1–59:12.

Ochiai, Y., Kumagai, K., Hoshi, T., Rekimoto, J., Hasegawa, S., and Hayasaki, Y. (2016). Fairy lights in femtoseconds: Aerial and volumetric graphics rendered by focused femtosecond laser combined with computational holographic fields. *ACM Trans. Graph.*, 35(2):17:1–17:14.

Padmanaban, N., Konrad, R., Stramer, T., Cooper, E. A., and Wetzstein, G. (2017). Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays. *Proceedings of the National Academy of Sciences*.

Padmanaban, N., Konrad, R., and Wetzstein, G. (2019). Autofocals: Evaluating gaze-contingent eyeglasses for presbyopes. *Science Advances*.

Palmer, S. E. (1999). *Vision Science - Photons to Phenomenology*. MIT Press.

Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. (1998). The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188.

Rathinavel, K., Chakravarthula, P., Akşit, K., Spjut, J., Boudaoud, B., Whitted, T., Luebke, D., and Fuchs, H. (2018a). Steerable application-adaptive near eye displays. pages 1–2.

Rathinavel, K., Wang, H., Blate, A., and Fuchs, H. (2018b). An extended depth-at-field volumetric near-eye augmented reality display. *IEEE transactions on visualization and computer graphics*, 24(11):2857–2866.

Rathinavel, K., Wang, H., and Fuchs, H. (2020). Optical calibration and distortion correction for a volumetric augmented reality display. In *Emerging Digital Micromirror Device Based Systems and Applications XII*, volume 11294, page 112940M. International Society for Optics and Photonics.

Rathinavel, K., Wetzstein, G., and Fuchs, H. (2019). Varifocal occlusion-capable optical see-through augmented reality display based on focus-tunable optics. *IEEE transactions on visualization and computer graphics*, 25(11):3125–3134.

Refai, H. H. (2009). Static Volumetric Three-Dimensional Display. *J. Display Technol.*, 5(10):391–397.

Rhcastilhos. and Jmarchn. (2007). Schematic diagram of the human eye. `https://commons.wikimedia.org/w/index.php?curid=1597930`. Created: Jan-24, 2007. Last checked: Feb-01, 2020.

Rolland, J. P. and Fuchs, H. (2000). Optical versus video see-through head-mounted displays in medical visualization. *Presence: Teleoperators and Virtual Environments*, 9(3):287–309.

Rolland, J. P., Holloway, R. L., and Fuchs, H. (1995). Comparison of optical and video see-through, head-mounted displays. In *Telemanipulator and Telepresence Technologies*, volume 2351, pages 293–307. International Society for Optics and Photonics.

Rolland, J. P., Krueger, M. W., and Goon, A. A. (1999). Dynamic focusing in head-mounted displays. In *Stereoscopic Displays and Virtual Reality Systems VI*, volume 3639, pages 463–471. International Society for Optics and Photonics.

Shi, L., Huang, F.-C., Lopes, W., Matusik, W., and Luebke, D. (2017). Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3d computer graphics. *ACM Trans. Graph.*, 36(6):236:1–236:17.

Shibata, T., Kim, J., Hoffman, D. M., and Banks, M. S. (2011). The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of vision*, 11(8):11–11.

Shiwa, S., Omura, K., and Kishino, F. (1996). Proposal for a 3-d display with accommodative compensation: 3ddac. *Journal of the Society for Information Display*, 4(4):255–261.

Simon J. Watt, Kevin J. MacKenzie, L. R. (2012). Real-world stereoscopic performance in multiple-focal-plane displays: How far apart should the image planes be?

Smalley, D., Nygaard, E., Squire, K., Van Wagoner, J., Rasmussen, J., Gneiting, S., Qaderi, K., Goodsell, J., Rogers, W., Lindsey, M., et al. (2018). A photophoretic-trap volumetric display. *Nature*, 553(7689):486.

State, A., Keller, K. P., and Fuchs, H. (2005). Simulation-based design and rapid prototyping of a parallax-free, orthoscopic video see-through head-mounted display. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, pages 28–31. IEEE.

Sullivan, A. (2004). DepthCube solid-state 3D volumetric display.

Sutherland, I. E. (1965). The ultimate display. *In Proceedings of IFIPS Congress (New York City, NY, May 1965)*, vol. 2, pp. 506-508.

Wang, X., Qin, Y., Hua, H., Lee, Y.-H., and Wu, S.-T. (2018). Digitally switchable multi-focal lens using freeform optics. *Optics express*, 26(8):11007–11017.

Wann, J. P., Rushton, S., and Mon-Williams, M. (1995). Natural problems for stereoscopic depth perception in virtual environments. *Vision research*, 35(19):2731–2736.

Watt, S. J., Akeley, K., Ernst, M. O., and Banks, M. S. (2005). Focus cues affect perceived depth. *Journal of vision*, 5(10):7–7.

Westheimer, G. (1966). The maxwellian view. *Vision research*, 6(11-12):669–682.

Wetzstein, G., Heidrich, W., and Luebke, D. (2010). Optical image processing using light modulation displays. *Computer Graphics Forum*, 29(6):1934–1944.

Wetzstein, G., Lanman, D., Hirsch, M., and Raskar, R. (2012). Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Trans. Graph. (SIGGRAPH)*, 31(4):80:1–80:11.

Wilson, A. and Hua, H. (2017). Design and prototype of an augmented reality display with per-pixel mutual occlusion capability. *OSA Opt. Express*, 25(24):30539–30549.

Xia, X., Guan, Y., State, A., Chakravarthula, P., Rathinavel, K., Cham, T.-J., and Fuchs, H. (2019). Towards a switchable ar/vr near-eye display with accommodation-vergence and eyeglass prescription support. *IEEE transactions on visualization and computer graphics*, 25(11):3114–3124.

Yamaguchi, Y. and Takaki, Y. (2016). See-through integral imaging display with background occlusion capability. *OSA Appl. Opt.*, 55(3):A144–A149.

Yan, Y., Chen, K., Xie, Y., Song, Y., and Liu, Y. (2018). The effects of weight on comfort of virtual reality devices. In *International Conference on Applied Human Factors and Ergonomics*, pages 239–248. Springer.

Zannoli, M., Love, G. D., Narain, R., and Banks, M. S. (2016). Blur and the perception of depth at occlusions. *Journal of Vision*, 16(6):17–17.