ABSTRACT

Title of dissertation: TOWARDS EFFECTIVE DISPLAYS FOR VIRTUAL AND AUGMENTED REALITY

Xuetong Sun, Doctor of Philosophy, 2020

Dissertation directed by: Professor Amitabh Varshney
Department of Computer Science

Virtual and augmented reality (VR and AR) are becoming increasingly accessible and useful nowadays. This dissertation focuses on several aspects of designing effective displays for VR and AR.

Compared to conventional desktop displays, VR and AR displays can better engage the human peripheral vision. This provides an opportunity for more information to be perceived. To fully leverage the human visual system, we need to take into account how the human visual system perceives things differently in the periphery than in the fovea. By investigating the relationship of the perception time and eccentricity, we deduce a scaling function which facilitates content in the far periphery to be perceived as efficiently as in the central vision.

AR overlays additional information on the real environment. This is useful in a number of fields, including surgery, where time-critical information is key. We present our medical AR system that visualizes the occluded catheter in the external ventricular drainage (EVD) procedure. We develop an accurate and efficient catheter tracking method that requires minimal changes to the existing medical equipment. The AR display projects a virtual image of the catheter overlaid on the occluded real catheter to depict its real-time position. Our system can make the risky EVD procedure much safer.

Existing VR and AR displays support a limited number of focal distances, leading to vergence-accommodation conflict. Holographic displays can address this issue. In this dissertation, we explore the design and development of nanophotonic phased array (NPA) as a special class of holographic displays. NPAs have the advantage of being compact and support very high refresh rates. However, the use of the thermo-optic effect for phase modulation renders them susceptible to thermal proximity effect. We study how the proximity effect impacts the images formed on NPAs. We then propose several novel algorithms to compensate for the thermal proximity effect on NPAs and compare their effectiveness and computational efficiency.

Computer-generated holography (CGH) has traditionally focused on 2D images and 3D images in the form of meshes and point clouds. However, volumetric data can also benefit from CGH. One of the challenges in the use of volumetric data sources in CGH is the computational complexity needed to calculate the holograms of volumetric data. We propose a new method that achieves a significant speedup compared to existing holographic volume rendering methods.

TOWARDS EFFECTIVE DISPLAYS FOR VIRTUAL AND AUGMENTED REALITY

by

Xuetong Sun

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Professor Amitabh Varshney, Chair/Advisor
Professor Mario Dagenais
Professor David Mount
Professor Martin Peckerar
Professor Matthias Zwicker

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Advances in virtual and augmented reality (VR and AR) allow us to explore new ways to present information more efficiently and effectively. Our research in this area has four foci. First, we explore ways to present information efficiently in wide field-of-view (FOV) VR and AR devices that leverage the characteristics of the far peripheral vision. Second, we explore how surgery can benefit from the use of AR. Specifically, we develop an AR system that tracks a catheter in an external ventricular drain (EVD) procedure accurately in real-time and displays its position directly in a surgeon's field of view. Third, we explore the novel holographic technology for the VR and AR displays. Conventional VR and AR near-eye displays based on stereoscopic vision only provide one or a limited number of focal distances, leading to vergence-accommodation conflict. Holographic displays allow a natural-to-senses perception of objects and scenes by recreating the optical waves from the scene or object. We have designed a nanophotonic phased array (NPA) which modulates the phase using the thermo-optic effect. Our design has the advantage of a fast refresh rate and being compact compared to other phase modulators such as liquid crystal on silicon spatial light modulators (LCoS SLMs). However, the NPA suffers from thermal proximity effect which diminishes the phase modulation accuracy. In this dissertation, we investigate how proximity effect affects the images formed by an NPA. We propose several algorithms to address this challenge. We have carried out numerical simulations to compare and contrast various algorithms in terms of image quality and computational efficiency. Last, we explore ways to generate holograms from volumetric data. Such techniques will be critical for visualization in holographic displays. Computer-generated holography (CGH) has largely focused on 2D images and 3D images in the form of meshes and point clouds. Volumetric data can also benefit from holographic visualization. In order to tackle the computational complexity of calculating the hologram of a volume, we develop a new calculation method that brings the computation time from hours previously to mere seconds. We also validate our method on an actual holographic display.

We introduce the background and contributions of our work in the rest of this chapter. Then in Chapter 2, we introduce our work on perception in the far periphery. In Chapter 3, we present our AR system that depicts the occluded catheter position for EVDs. In Chapter 4, we describe our nanophotonic phased array, and propose several proximity effect correction methods. In Chapter 5, we cover our accelerated holographic volume rendering algorithm. We summarize our contributions and propose the directions for future work in Chapter 6.

1.1 Perception in the Far Periphery

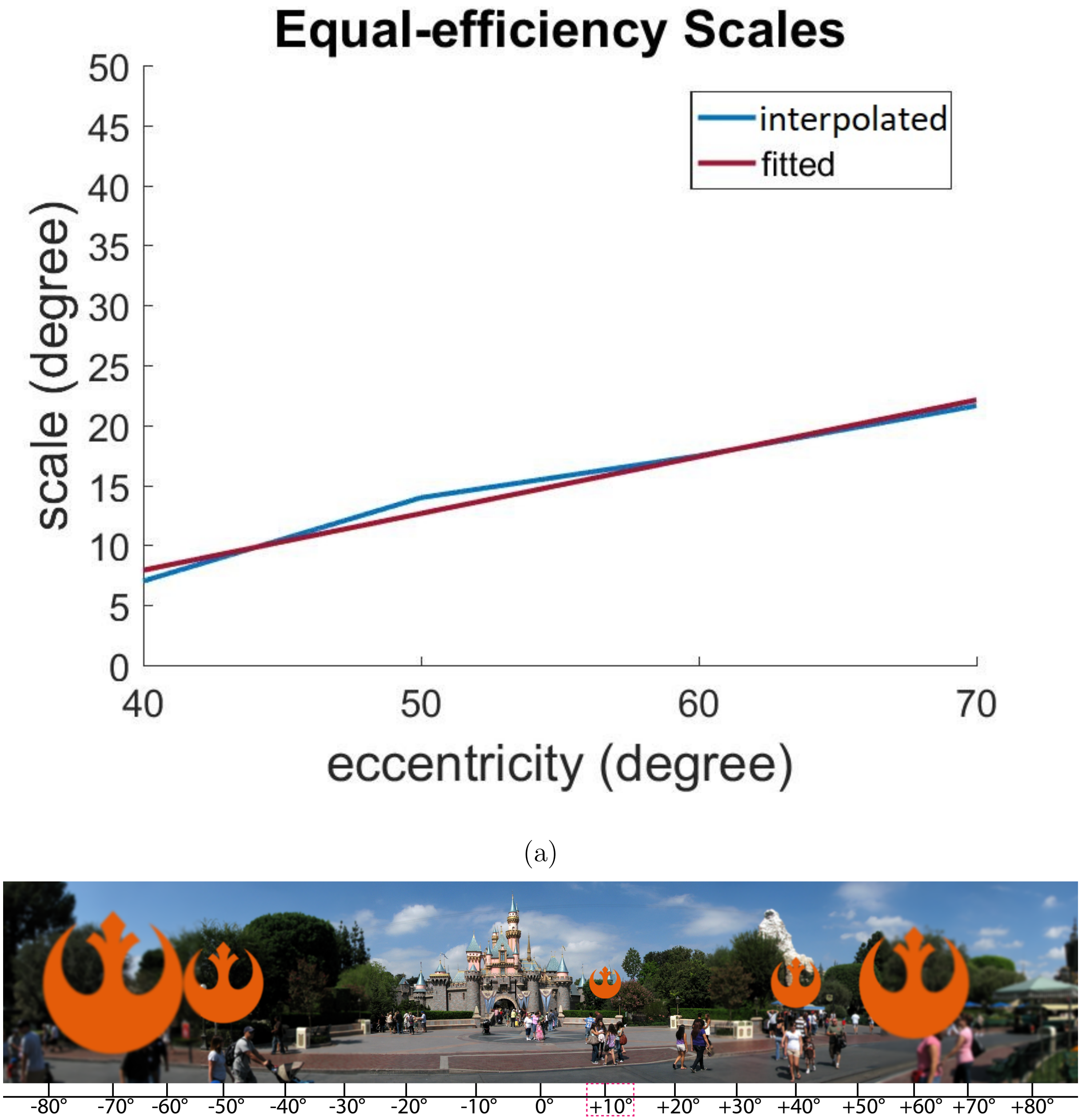
The far peripheral vision (beyond 60° eccentricity) is beginning to be supported in the latest VR and AR headsets. While lacking in visual acuity, the far peripheral vision is capable of, and plays a crucial role for a variety of visual and cognitive functions ([10–12]). Making use of the far peripheral vision benefits the VR and AR experiences by offering a larger display real estate. However, the visual properties of the far periphery are different from those in the central vision because of the physiological differences between the areas on the visual cortex responsible for the respective types of vision.

Our goal is to investigate the visual properties in the far peripheral vision to take advantage of the wider field of view in VR and AR headsets. We conduct a study to explore the relationship between the perception time, which is the time it takes to perceive a certain object, and the eccentricity. Specifically, we conduct a user study to measure participants' perception time of a series of size-scaled simple patterns over several eccentricities. In the study, two types of patterns of different scales (sizes) show up at different locations on a wide FOV display. The task of the participants is to perceive the pattern and respond based on what they perceive. We use an eye-tracker to make sure that the participants' gaze is fixated at the center of the display and record their response time as the perception time. In our study, we had a total of 40 participants, all with normal or corrected-to-normal vision. The collected data shows that it takes longer to perceive a given pattern at a higher eccentricity than in the central vision; however, a larger pattern at a higher eccentricity can be perceived in a similar amount of time as the original in central vision. Our main contribution is deriving scaling functions with which the objects displayed at higher eccentricities could be perceived as efficiently as in lower eccentricities. Our scaling functions are valid from 40° to 70° eccentricity.

Our study is related to M-scaling in psychophysics. Most past work on M-scaling focuses on low-level visual functions such as contrast sensitivity. Rovamo and Virsu [13] show that the contrast sensitivity functions with retinally scaled stimuli are similar at different eccentricities. Grogoric *et al.* [14] conduct a study to find the proper scales for objects to be recognized up to 40° eccentricity.

Making use of the far peripheral vision has several advantages. First, the peripheral vision provides an opportunity for additional information to be displayed. Second, displaying non-critical information in the far periphery can reduce visual clutter in the central vision. Last but not the least, displaying content in the far periphery is less intrusive to the central vision, allowing techniques such as subtle attention management.

We show an example of this scaling function in Figure 1.1. To achieve the same perception time as a 5° pattern at 10° eccentricity, the patterns at higher eccentricities can be scaled according to our equal efficiency scales. Details are available in Chapter 2 and peer-reviewed publication Sun *et al.* [15].



(a)

(b)

Figure 1.1: An example of the equal-efficiency scales. At higher eccentricities (40° and up, patterns can be scaled according to our finding to achieve the same perception time as perceiving a 5° pattern at 10° eccentricity.

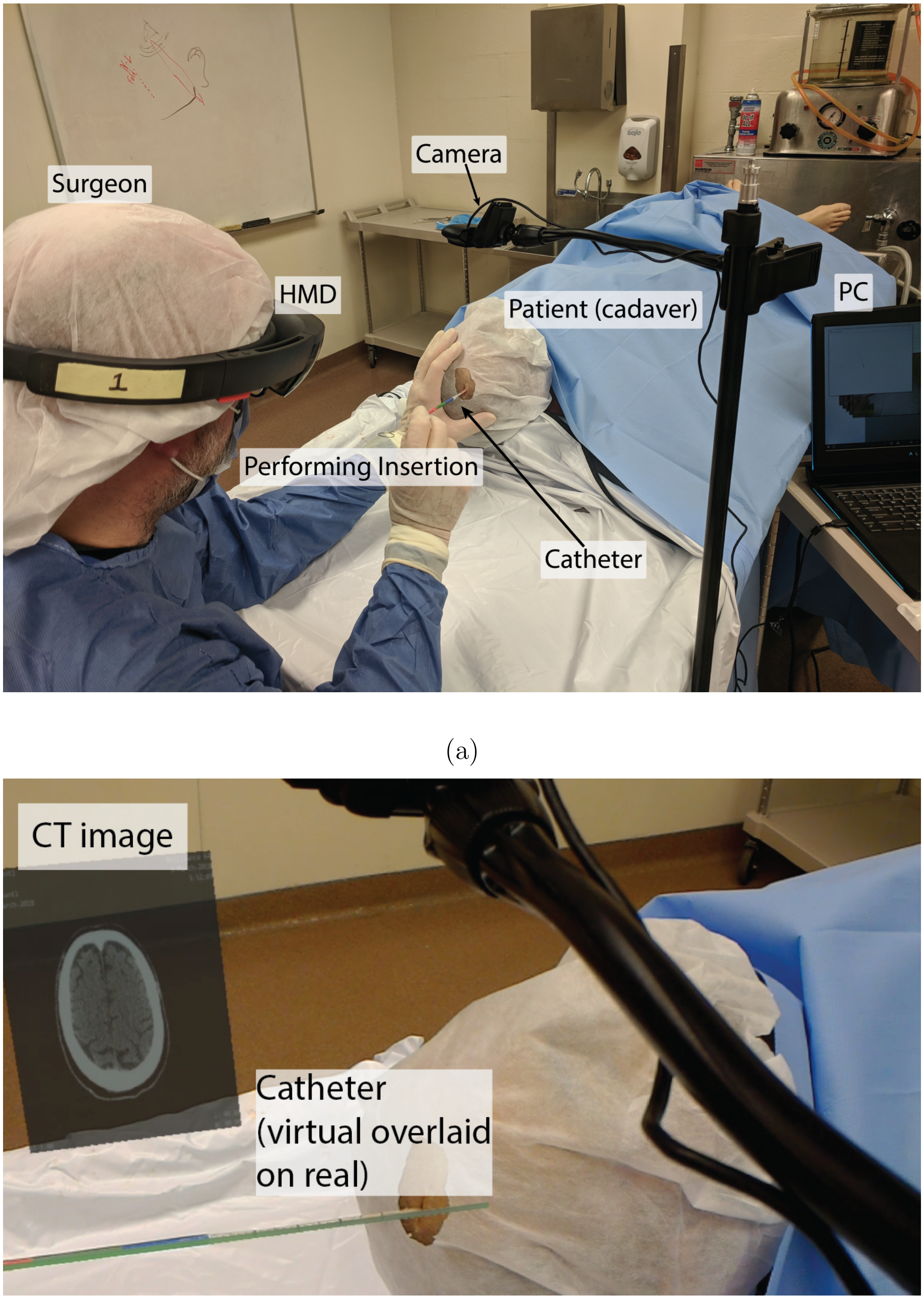
1.2 AR Displays for Surgery

AR has transformative potential for a wide variety of applications, such as entertainment, driving assistance, city planning, and manufacturing assembly guidance. Our focus here is on leveraging AR capabilities for health care and medicine. We have built several prototype medical AR systems in a collaborative effort with the University of Maryland Shock Trauma Center.

The first prototype is a system that displays images from medical imaging machines directly in the healthcare provider's field of view. In many surgical operation scenarios, when the surgeon needs the information from an imaging device, he or she needs to constantly move the sight away from the patient to look at a separate screen. Our prototype can alleviate this psychomotor burden and help health-care professionals to focus on giving the best possible care to the patient.

Our first prototype serves as a proof-of-concept that AR technology can benefit medical procedures. It helped us resolve technical difficulties in developing AR applications with multimedia. We have selected the Microsoft HoloLens AR headset as our platform for this project. It provides a good 6 degree-of-freedom (DOF) inside-out spatial tracking and an optical see-through field of view of $30^\circ \times 17^\circ$. This FOV is small compared to most VR headsets. However, the Hololens offers natural and intuitive user interfaces including hand gesture and voice recognition and is suitable for our see-through AR prototypes. We use a video capture card to convert the output signal from medical imaging machines to be ingested by our system, which sends the images to the AR headset over WiFi. Based on this experience, we proceeded to build another prototype system to assist surgeons in external ventricular drain (EVD) procedures. EVD is a high-risk medical procedure that involves inserting a catheter inside a patient's skull, through the brain, and into a ventricle, to drain cerebrospinal fluid relieving elevated intracranial pressure. Once the catheter has entered the skull its tip cannot be seen. The neurosurgeon has to rely on experience to guess its location inside the cranium and direct it towards the ventricle using only anatomic landmarks.

Our prototype system tracks the catheter using a novel approach that does not change how it feels to operate and projects the catheter's image into the surgeon's field of view, aligned with the real catheter. Our tracking technique requires minimum modification to the catheter. We color portions of the catheter with different colors. By tracking the color bands with known lengths, we can infer the position of the catheter. A simple one-step calibration tells the relationship between the camera space and the display space. The position of the catheter is sent to the display via WiFi and a virtual catheter is rendered to overlay the real (and occluded) catheter to depict its position. We have conducted the preliminary tests in a controlled environment on a cadaveric head. Our collaborating surgeons are pleased with the high accuracy and the low latency that we have been able to achieve. Figure 1.2 shows how our system is used in the test by a neurosurgeon and a see-through view from the surgeon's perspective. Details are available in Chapter 3 and peer-reviewed publication Sun *et al.* [16].



(a)

(b)

Figure 1.2: A surgeon using our EVD assistance system and the surgeon's see-through view.

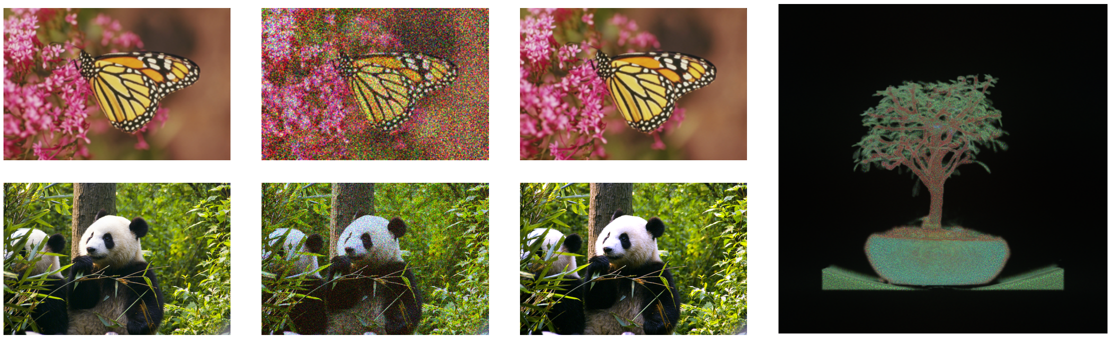
1.3 Correcting Proximity Effect in Nanophotnoic Phased Arrays

Holographic displays recreate the scene by modulating, phase, and sometimes also the amplitude, of the optical wave. A phase modulator that can accurately and independently control the phase of the wavefront is crucial to the successful display of a hologram. While LCoS SLMs have been demonstrated to be useful as phase modulators, they suffer from a relatively low refresh rate (below 200 Hz) and complicated light source configurations.

Nanophotonic phased arrays are phased arrays operating at optical wavelengths. An NPA is composed of an optical power distribution system, a phase modulation mechanism that can control the phase of each pixel individually, and antennas for propagating phase-modulated light into the free space (Raval [17]). Sun *et al.* [18] showed that nanophotonic phased arrays can also be used for generating an optical wavefront which through careful phase modulation in the near field can display complicated images (or 3D scenes) in the far field. One major advantage of the NPAs over SLMs is their fast refresh rate. The thermal simulation with our NPA design shows that it can complete a heat-cool cycle within 10 µs, suggesting a refresh rate up to 100 kHz which is uniquely suitable for dynamic content as well as time-division multiplexing operations such as color-switching or image smoothing. In addition, the NPAs support integrated light sources. Lasers can be coupled into the NPA via an optical fiber, distributed to each pixel and emitted by the on-board antennas. This way, the NPA displays can be made more compact as optical operations such as beam expansion and polarization necessary for the typical LCOS SLM display systems are not necessary.

The successful display of desired 2D or 3D imagery lies in the precise and independent control of the phases of the pixels in the NPAs. Most NPA displays, such as those in Doylend *et al.* [19], Sun *et al.* [18] and our own design, depend on the thermo-optic effect to modulate the phase of a pixel. The phase of a pixel can be changed from 0 to 2π by adjusting the temperature of the pixel. However, the thermal proximity effect, the phenomenon where the temperature of one pixel affects the temperature of neighboring pixels, impacts the accuracy of phase modulation and degrades the observed image. Therefore, to achieve the desired phase on the near-field wavefront, this thermal proximity effect must be taken into consideration. This is further complicated by the fact that most NPAs only have heating mechanisms with no cooling capability, making some phase patterns impossible to realize.

In Chapter 4, we explore the impact of the thermal proximity effect on the observed images of holograms displayed on the NPA. The proximity effect can be modeled as a convolution and we explore correcting the proximity effect with deconvolution methods. Most NPA display systems including our design only have phase modulation capabilities. We develop a novel proximity effect correction (PEC) method based on the proximal optimization method. This method is applicable to both the Fourier holograms usually for displaying images in the far field and the Fresnel holograms for displaying 2D and 3D images at user-specified depths. Our new method is also more effective and efficient than previous proximity effect correction methods based on the iterative phase retrieval algorithm. The results of our



original no PEC proposed PEC proposed PEC

Figure 1.3: The first row of the middle two columns show the simulated observations of the Fourier hologram displayed on the NPA without any proximity effect correction and with our proposed PEC method. The second row show the simulated observations of the Fresnel hologram. Our PEC method can effectively correct the proximity effect correction at the tested proximity effect level ($\sigma = 0.66$ px). The simulated observation of the 3D volume by applying our proposed PEC method to a stack of Fresnel holograms can be seen on the right.

proximity effect correction method can be seen in Figure 1.3. As can be seen, the near-field phase modulation is negatively impacted by the thermal proximity effect. Without proximity effect correction, the image quality of the formed image degrades. Our proposed proximity effect correction method can effectively correct the proximity effect and restore the image at the tested proximity effect level ($\sigma = 0.66$ px). We can even display a 3D volume by applying the PEC method to a stack of Fresnel holograms at their respective depths. Details are available in Chapter 4 and peer-reviewed publication Sun *et al.* [20].

1.4 Fast Holographic Volume Rendering

Light has a wave-particle duality, meaning it could be modeled as a particle traveling from one location to another as well as a wave propagating from the source. Much of the computer graphics rendering techniques have been developed using the particle model, a popular example of which is raytracing [21, 22]. However, as is shown in Dennis Gabor's experiments in the 1940s [23], a scene can be reconstructed by recording and illuminating the interference pattern of an object wave and a reference wave. With the advancement in electro-optics, there are devices that can programmatically control the phase and amplitude of the light wave, such as spatial light modulators (SLMs).

Computer-generated holography (CGH) refers to digitally computing the interference pattern, either from a digital model or from specially captured images. Although in its early stages, this technology promises to solve several significant existing problems in conventional VR and AR displays, such as vergence-accommodation conflict due to the limited number of focal distances. Varifocal and light-field displays also offer a wider range of focal distances. However, the former usually involves moving components while the spatial and angular resolutions of the latter are limited by optical diffraction as the pixel density increases. On the contrary, the quality of CGH improves as the pixel density increases.

CGH has largely focused on 2D images or 3D scenes in the form of meshes or point clouds. The Fourier transform is a common and efficient method to calculate the hologram of a 2D image. Calculating the hologram of a 3D scene is compu-

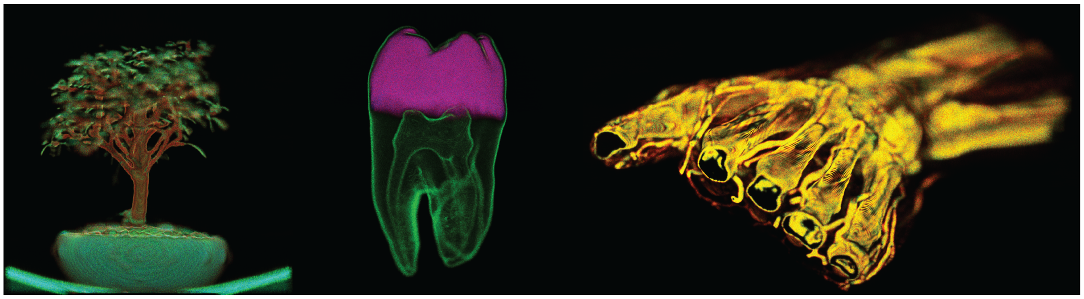


Figure 1.4: Sample results of our holographic volume rendering algorithm.

tationally intensive and usually involves converting the scene into a collection of points directly visible from the hologram and calculating the hologram following the Huygens-Fresnel principle. Modern computational hardware and acceleration methods have enabled real-time calculation of holograms from 3D meshes.

Volumetric data from medical imaging can be visualized to reveal critical information about the organs and tissues. Volumes from physics simulations are also used in the realistic rendering of fog, smoke and so on. Volume visualization can benefit from the advantages of holographic displays. However, calculating the hologram of a volume is an order-of-magnitude more complicated than calculating that of a 3D mesh, as every voxel contributes to the perception of the volume instead of only those closest to the hologram.

We develop a new method to calculate the hologram of volumetric data that achieves significant speedup compared to existing holographic volume rendering methods. We also show through simulations how to choose the light source for the holographic display to strike a balance between the speckle noise and image sharpness for the observation. Some simulated observations of our proposed method can be seen in Figure 1.4. Details are available in Chapter 5.

Chapter 2: Perception in the Far Periphery

2.1 Overview

Computer graphics has thus far primarily focused on high-fidelity rendering that is best suited for the central vision. Rendering in the far peripheral vision has not received much attention because most of the displays till now have only had a narrow field of view (FOV). However, rapid advances in display technologies are enabling head-mounted displays (HMDs) with an ever-increasing FOV. This has necessitated a better understanding of how peripheral vision could assist in the next-generation of VR and AR applications.

In this study, we are interested in the far peripheral vision, which suffers from significantly diminished visual acuity compared to the central vision. Definitions of the peripheral vision differ throughout the literature. In this chapter, the foveal vision (within 2° from the fixation point), together with the perifoveal, is called the *central vision* [2,24]. The central vision covers the region from 0° to 8° eccentricity and corresponds to the eye's *macula* which is responsible for the high-resolution color vision. Beyond the central vision is the peripheral vision. Near, middle, and far periphery span 8° to 30°, 30° to 60°, and above 60° eccentricity, respectively [25]. All eccentricities in this study are horizontal. A visual representation can be seen

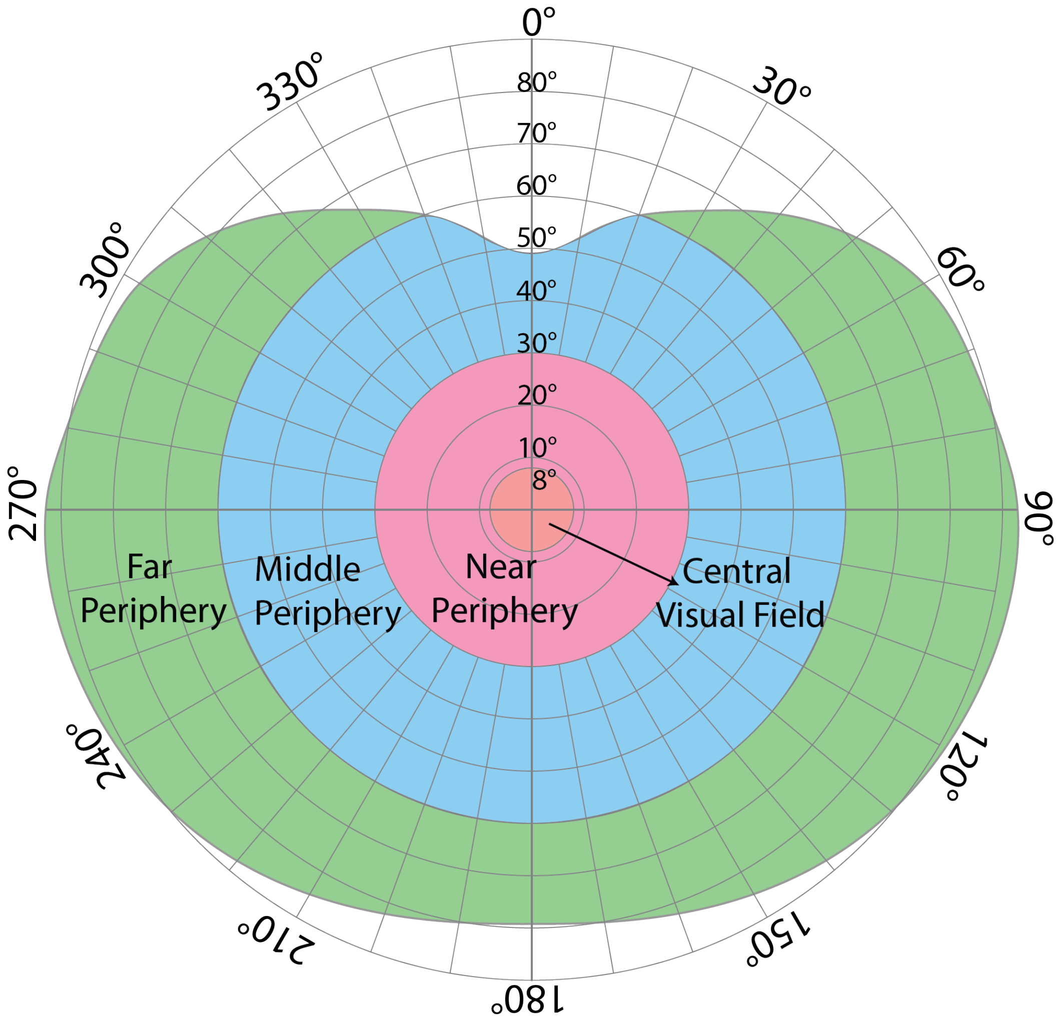


Figure 2.1: Illustration of the human visual field. The central visual vision, near, middle, and far peripheral vision are from 0° to 8°, 8° to 30°, 30° to 60° and beyond 60° respectively. This image is adapted from [2].

in Figure 2.1.

The visual property difference between the central and peripheral vision has been used in foveated rendering such as Meng *et al.* [26–28] to save the computational capabilities. In this chapter, we are more interested in the perception in the far periphery and conduct a user study to investigate the perception time in the far peripheral vision. We wish to find a quantifiable scaling function so the information presented in the far periphery can be perceived as efficiently as in the central vision.

Our study is related to concepts in psychophysics called *cortical magnification* and *M-scaling*. Cortical magnification says that M, the diameter in the primary visual cortex onto which one degree of visual field projects [29], is inversely linear to the eccentricity E. The relationship can be formulated as follows. a is a coefficient [24]

$$M^{-1}(E) = M_0^{-1}(1 + aE) \tag{2.1}$$

Variations of the formula exist in [13, 30–38]. This leads to M-scaling, which says performance variations with eccentricity can be minimized by using appropriately scaled stimuli ([24]). M-scaling can be represented in the following equation. $S(E)$ is the scale at eccentricity E.

$$S(E) = S_0(1 + E/E_2) \tag{2.2}$$

Our work makes the following contributions. First, many studies on M-scaling focus on low-level visual properties such as contrast sensitivity [13], In our study, participants are asked to perform a higher-level visual task by responding to different patterns. We record the time for participants to respond to patterns of different scales at different locations. This facilitates the determination of an appropriate magnification for patterns at higher eccentricities so that they can be perceived in a similar time as those in the central vision. Second, to our knowledge, we are the first to test and report a scaling function in the far peripheral vision, applicable to a field of view of up to 140° horizontally.

2.2 Related Work

In this section, we introduce past studies related to our work on the peripheral vision. These include the peripheral vision and wide FOV displays.

2.2.1 Peripheral Vision

Low-level visual properties like contrast sensitivity have been studied extensively in the central vision (0° to 8° away from the fixation point, or eccentricity). But there has been little work on the far peripheral vision (beyond 60° eccentricity) or on high-level perceptual functions. Below we introduce related perceptual psychology works on the peripheral vision.

Cortical magnification says that M, the diameter in the primary visual cortex onto which one degree of visual field projects (Daniel and Witteridge [29]), is inversely linear to the eccentricity E. Various models have been proposed. Cowey and Rolls [30] first fit the visual cortex electrical stimulation data provided by Brindly and Lewin [31] to the inverse linear model ($M^{-1}(E) = M_0^{-1}(E/E_2)$)with $M_0 = 8.55$ mm/° and E_2=1.746. Through medical imaging of the visual cortex, Horton and Hoyt [32], Schira *et al.* [39] and Larsson and Heeger [33] derive different values for M_0 and E_2. Rovamo and Virsu [13] propose adding an extra term for improved accuracy $M^{-1}(E) = (1 + aE + bE^3)M_0^{-1}$ and compile four equations for the four half meridians separately. There is also the power function used by Tolhurst and Ling [34], which is $M^{-1}(E) = M_0^{-1}(1 + aE)^{1.1}$ with $M_0 = 24.88\,\text{mm}/°$.

This leads to M-scaling, which says performance variations with eccentricity can be minimized by using appropriately scaled stimuli (Strasburger *et al.* [24]). M-scaling can be represented as $S(E) = S_0(1 + E/E_2)$ where $S(E)$ is the scale at eccentricity E. Rovamo and Virsu [13] adjust the stimuli presented in the periphery using the function $M^{-1}(E)$, and the contrast sensitivity functions at different eccentricities are made similar as a result. However, it is worth noting that the cortical magnification and M-scaling concepts are not golden rules to "make everything equally visible independent of eccentricity". There are cases in which scaling does not compensate for the decreasing M factor and even cases where the visual performance is diminished [37, 38]. For more details on cortical magnification and M-scaling, we refer readers to Strasburger *et al.* [24].

High-level visual functions like letter recognition are also peripheral vision study topics. Strasburger *et al.* [24] propose a scaling function $S(E) = 0.022E + \frac{0.25}{\log C + 2.07 - 0.0345E}$ based on data from Strasburger [40] where C is the Michelson threshold contrast. Grogoric *et al.* [14] find a linear scaling function from 0° to 40° eccentricity to make objects presented in a virtual scene perceivable.

Studies have also shown that the peripheral vision is capable of, and sometimes plays an important part in, higher-level perceptual tasks like scene and object recognition. Thorpe *et al.* [10] show that even at 70° eccentricity, an accuracy of 60.6% can be achieved in recognizing animal faces. Boucart *et al.* [11] compare people's ability to identify images of human faces, vehicles, and animals at different eccentricities. The results show that human faces have an advantage over vehicles and animals in terms of response time, and that advantage persists even at extreme eccentricities up to 80°. In scene gist recognition, Loschky *et al.* [12, 41] compare participants' abilities to categorize a scene from a "window" (central portion of the image) and a "scotoma" (blocking central portion, only keeping the peripheral part of the image). They show that the peripheral vision is more important than central vision in scene gist recognition, as recognizing a 30° radius scotoma achieves the same accuracy as recognizing the whole image, and a 5° radius window is significantly harder to recognize. Other studies have shown that peripheral vision allows, though at a reduced level, recognition of facial expressions and emotions ([42, 43]).

Ren *et al.* [44] investigate whether a larger field of view improves searching task performance in augmented reality. Their results show that a larger field of view (108° × 82°) leads to a shorter task completion time, but lowers the correctness compared to a limited field of view (45° × 30°). Kline and Witmer [45] conduct a user study to investigate the effect of field of view on distance judgment in the virtual environment and find that with a smaller field of view (60° × 38.5°), participants tend to overestimate distance while with a larger field of view (140° × 90°) participants tend to underestimate. Jones*et al.* show in [2, 46] that adding stimulation in the far periphery in a VR headset causes more accurate distance and size judgment in a virtual environment and changes people's walking behavior. A good literary review on the effects of the field of view in virtual environments can be found in Jones *et al.* [46].

2.2.2 Wide FOV Displays

Researchers in the graphics community have long used the fact that the human peripheral vision perceives less detail than the foveal vision. Oshima *et al.* [47] analyze user gaze direction and blur objects not in the user's foveal vision to reduce rendering complexity. In an augmented reality application, Ishiguro and Rekimoto [48] display an icon above objects in the periphery of the user's field of view and show the detailed information when the user's gaze is moved to that object. Jones *et al.* [49] project images that are aesthetically matched with the content on the TV. Positive user feedback shows that enriching the peripheral vision can improve the interactive user experience.

The peripheral vision cannot be utilized in VR and AR without wide-FOV displays. Various ways to build wide-FOV VR and AR displays have been proposed. Most of the systems we review here support 100° horizontal FOV or greater. The most straightforward system that extends the viewer's FOV is the *cave automatic virtual environment* (CAVE). Cruz-Neira *et al.* [50] create a $7' \times 7' \times 7'$ cube-shaped CAVE where images were projected to three side walls and the floor. It covers the entire field of view when the viewer is facing the middle of the three projection walls). There have been many iterations and variations of the CAVE system [44,51].

In an effort to study the immersion on desktop VR, Robertson *et al.* [52] use two additional monitors (referred by them as *peripheral lenses*) on either side of the main monitor. The total horizontal FOV using two peripheral lenses is approximately 38°. Attaching additional monitors can possibly cover the entire FOV.

In traditional HMDs, the two most important components are the display panel and optics. Modifying either or both is a common method of enlarging the FOV of HMDs. Nagahara *et al.* [53] build a prototype video see-through HMD that uses a complicated configuration of an ellipsoidal and a hyperboloidal curved mirror to reflect light from the display into the viewer's eyes, covering 180° horizontal FOV. Kiyoshi [54] creates an optical see-through HMD using a hyperbolic mirror instead of a flat one to increase the horizontal FOV to 190°. Dunn *et al.* [55] take the optics design one step further to accommodate depth cues while achieving a large field of view by using a flexible mirror membrane to reflect light from the display. This allows a 90° by 45° binocular FOV. The membrane is also varifocal, controlled by airtight cavities. The focal power can be adjusted with a gaze tracker. Rakkolainen *et al.* [56, 57] use flexible OLED that curved around the viewer's eye and curved optics to reach super-wide FOV. They create proof-of-concept prototypes using Fresnel lenses and achieved 232° × 130° and SI318° × 130° FOV.

Tiling display panels or optics can also increase the field of view. Massof *et al.* [58] achieve 150° × 100° FOV by using a specially-molded array of Fresnel lenses and surrounding each lens with 16 OLED panels. Tiling optics of tiny pinholes has been used by Aksit *et al.* [59]. Given the pupil and aperture size and the aperture-pupil and aperture-display distance, the FOV of a single aperture can be calculated. To widen the FOV of the HMD, an array of pinholes is used. Specially synthesized images are displayed on a cell phone, from which light travels across the apertures to reconstruct the original images. The FOV is easily expandable with larger displays and more apertures. Xiao and Benko [60] fill the periphery of the HMD with sparse arrays of high-contrast LED lights, dubbed *sparse peripheral display*. Their prototypes based on existing commercial VR and AR HMDs can achieve a FOV of 170° and 190° respectively.

Light-field and holographic near-eye HMDs, while in their early development stages, have the potential to support a large FOV and address problems like the vergence-accommodation conflict. A prototype by Huang *et al.* [61] uses a rank 1-factored light field, and has a FOV of 87° by 91°. Maimone *et al.* [62] and Shi *et al.* [1] use spherical light waves to address the limitations of a small diffraction angle. The former demonstrates a prototype with an 80° horizontal FOV.

We compile below a list to include the different approaches and examples of wide FOV displays as well as their resolutions in Table 2.1.

2.3 User Study Design

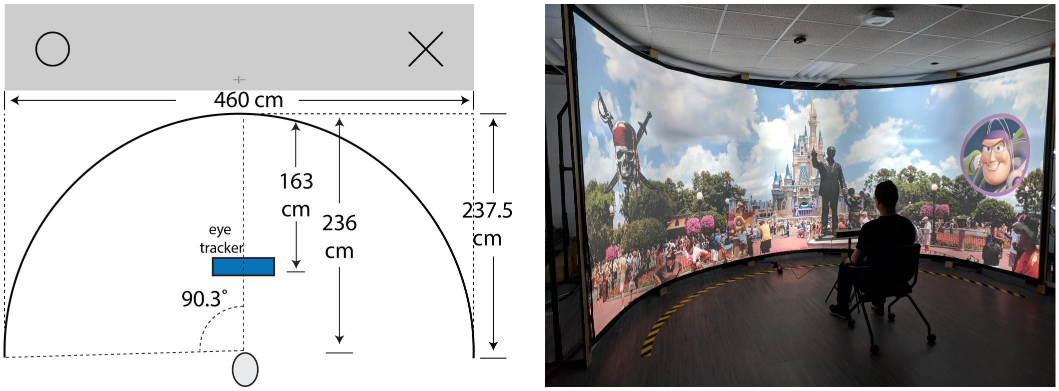


Figure 2.2: Top view and stimuli (left) of our user-study setup. In the experiment, patterns appear one at a time. A photograph of our system used in the study (right). Shown on screen is one of the possible use cases of the system.

We build a user study system that uses a large curved screen and eye-tracking to simulate a wide FOV HMD. Stimuli of different sizes are displayed at various

Table 2.1: List of wide-FOV virtual and augmented reality display techniques and devices. The spatial resolution of a holographic display is the same as the SLM pixel pitch when inside the depth of field [1]. The items with * are calculated from measurement and other reported device specifications.

Display	Max reported FOV (°)	Color	Angular Resolution
Spherical wave holographic display(2017) [1,62]	80(H)	RGB	3.74 µm
Stereoscopic factored light field synthesis(2015) [61]	87 x 91	RGB	4.1 arcmin/pix*
Flexible mirror membrane(2017) [55]	90 x 45	RGB	3.4 arcmin/pix*
LCD and optics (Meta 2)	90 x 50	RGB	3 arcmin/pix
Detachable display and optics (Samsung Gear VR, Google Daydream)	100(D)	RGB	2.5 arcmin/pix*
Pinlight(2014) [63]	110(D)	Mono	4.5 arcmin/pix*
Tiling display and Fresnel lenses(2003) [58]	150 x 100	RGB	3 arcmin/pix
LCD and catadioptrical optics(2003) [53]	180(H)	RGB	5.7-10.5 arcmin/pix*
Projector and hyperbolic mirror(2007) [54]	190(H)	RGB	60 arcmin/pix*
sparse peripheral display(2016) [60]	190(H)	RGB	5.5 arcmin/pix*
OLED and Fresnel optics (HTC Vive, Oculus Rift, StarVR)	210 x 130	RGB	5.5 arcmin/pix*
Curve display and optics(2016) [56,57]	318 x 130	RGB	5.4-7.5 arcmin/pix*
CAVE [44,50,51,64]	full	RGB	1.5-3 arcmin/pix*
Tiled desktop displays [52]	full	RGB	3-4 arcmin/pix*

eccentricities. Participants respond to the stimuli by pressing buttons on a controller device. The responses are recorded. Our setup is illustrated in Figure 2.2.

The display we use in this study is a custom-made curve screen using 5 × 3 projector array, with a 6400 × 2100 resolution. This display has a 180° horizontal FOV when viewed 236 cm away from the center of the screen. The system setup can be seen in Figure 2.2 right. The eye-tracking device we use is a Tobii EyeX commercial eye tracker. The eye tracker is placed 163 cm away from the center of the screen, slightly below eye level facing the user.

The system displays various patterns of different sizes at different eccentricities according to a pre-determined randomized order. The response from the user's game controller is recorded at 60 Hz.

2.3.1 Task and Stimuli

We display two types of patterns, crosses and circles, as shown in Figure 2.2 left top. In the user study task, one type of pattern is specified as the *target*, and the other is the *distraction*. The task is to perceive the pattern and respond to the target pattern (by pressing a button on a controller) as soon as it appears and ignore the distraction while keeping fixated at the center of the screen. For the purposes of this study, we define the time from the appearance of the pattern to the button being pressed is recorded as the *perception time*. We recognize that this includes the time for the eye to perceive the pattern, the brain to understand the pattern, and for the user to respond by pressing the button on the controller. If a participant responds incorrectly, the stimulus will turn red. For example, when the cross is specified as the target, and the circle is the distraction, the participant should press a button on the controller as soon as he or she sees a cross on the screen, and do nothing with a circle. Participants are further directed to keep looking at the center of the screen and not move their heads or gaze directions. The cross-hair at the center will turn green if the eye tracker confirms the participant is looking at the center and turns black otherwise.

Since our display supports 180° FOV and considering the space to display the pattern at the outermost eccentricity, the stimuli are tested at eccentricities from 0° to 80°, with a 10° interval. Because we are more interested in higher eccentricities, we test stimuli at higher eccentricities at a wider scale range. Specifically, eccentricities 0°, 10°, 20° and 30° are tested at scales 5°, 10° and 15°, while eccentricities 40°, 50°, 60°, 70° and 80° are tested at scales 5°, 8°, 11°, 14°, 17°, 20°, 23° and 26°. The scale ranges are determined after our pilot study. Each eccentricity-scale combination is repeated 5 times for each pattern as the target and 2 times for each pattern as the distraction (14 times in total).

2.3.2 User Study Procedure

The study is conducted with one participant at a time. We first explain to the participants the purpose of this study. We then ask them to log their basic personal information, including name, age, and whether or not they have vision correction. We then explain how the study is conducted, and perform practice rounds of the study. The experimenter answers any question at this point.

To reduce participant fatigue, the entire experiment is divided into four shorter rounds between which the participants can take a break. Two rounds have the cross as the target pattern; two have the circle as the target. For example, the circle is repeated 5 times for each eccentricity-scale combination and the cross is repeated 2 times for each eccentricity-scale combination. These patterns are shuffled and divided into the 2 circle-target rounds. The opposite is done for the 2 cross-target rounds. The 4 rounds are the same for each participant, but half go over them in a *cross1, circle1, cross2 circle2* order, and the other half in a *circle1, cross1, circle2, cross2* order. During each round, the experimenter stays out of the eye tracker's range. The experiment takes about sixty minutes for each participant. After all the rounds are done, each participant was compensated with $10. During the experiment, we recorded the participants' eye movements and game controller responses. We cross-referenced this data with the stimuli to determine the precise perception time.

For the pilot user study, we had six participants. For the formal user study, we recruited 40 participants (24 male and 16 female). Every participant had a normal (31 participants) or corrected-to-normal vision (6 had eyeglasses and 3 used contact lenses).

2.4 Results

We first visualize the perception time over all eccentricity-scale combinations. Then we characterize the relation among the perception time, the eccentricity and the scale. Finally, we quantify the optimal scaling at high eccentricities to achieve the same perception time as in the central vision.

We first show the perception time over all eccentricity-scale combinations in the form of a 3D surface mesh. The perception time is indicated both as the height and by color. The surface mesh of the median (50^{th} percentile), as well as the 25%

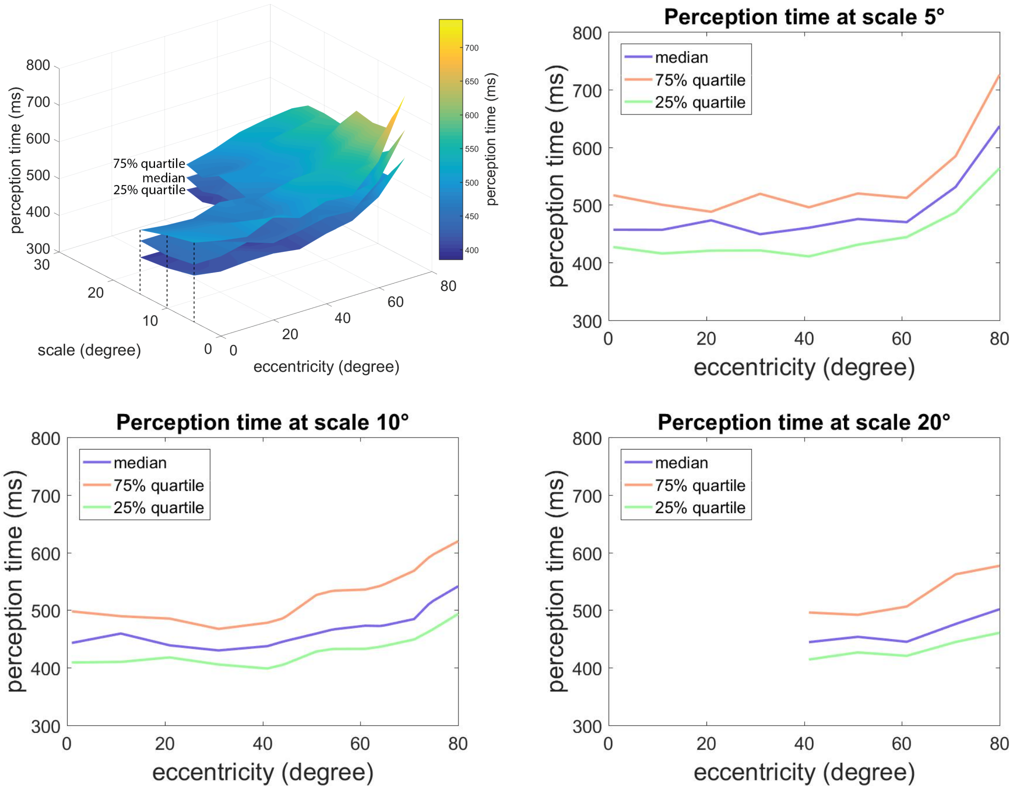


Figure 2.3: Perception time of all eccentricity-scale combinations is presented as a 3D colored mesh (top left). The three planes from top to bottom are the 75% quartile, median (50% quartile), and 25% quartile of all participants. The perception time at three scales (5°, 10° and 20°) is also shown.

and 75% quartiles of all participants, can be seen in Figure 2.3. In Figure 2.4, the median perception time of participants over all eccentricity and scale combinations is shown in greater detail, and we plot the linear relationship between scale and eccentricity that will result in the same perception time (Section 2.4.3). Figures 2.3 and 2.4 show that generally, perception time increases as eccentricity gets higher, and decreases as the scale becomes larger. One-way ANOVA test on the perception time with each eccentricity-scale combination as a group has a p-value of 1.3×10^{-6}, giving us high confidence to reject the null hypothesis that the perception times follow the same distribution.

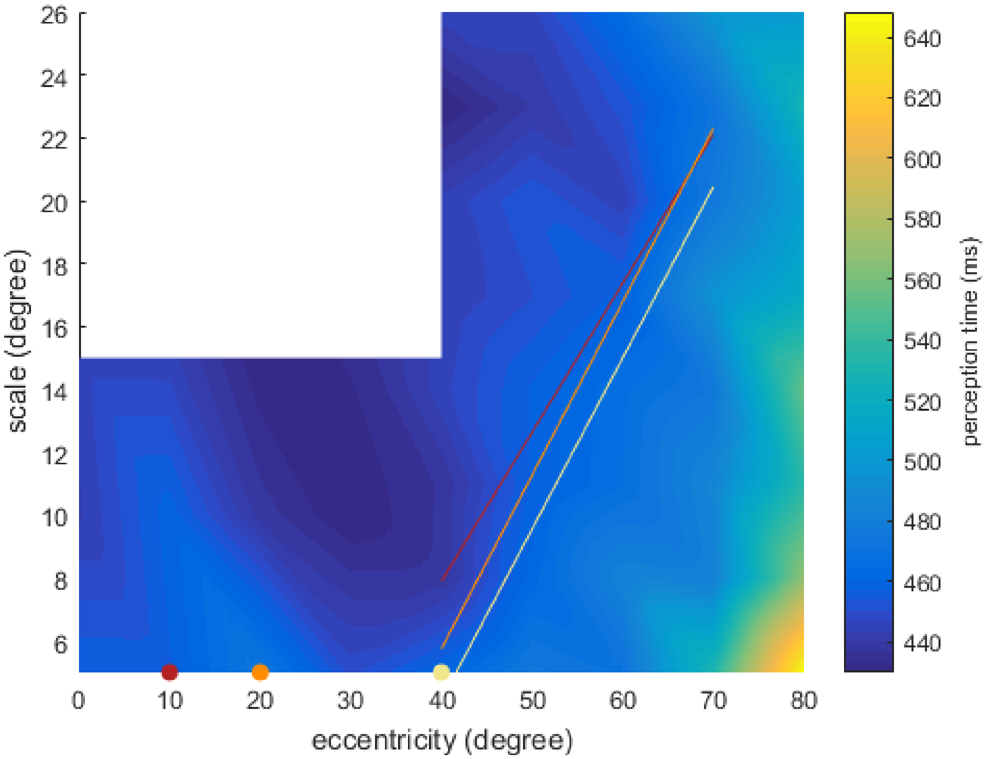


Figure 2.4: A color-coded map of the median perception time for all eccentricity-scale combinations. Using our findings in Section 2.4.3, we plot the scales at higher eccentricities to be perceived in the same time as a 5° pattern at 10° (red), 20° (orange) and 40° (yellow).

2.4.1 Perception Time and Scale

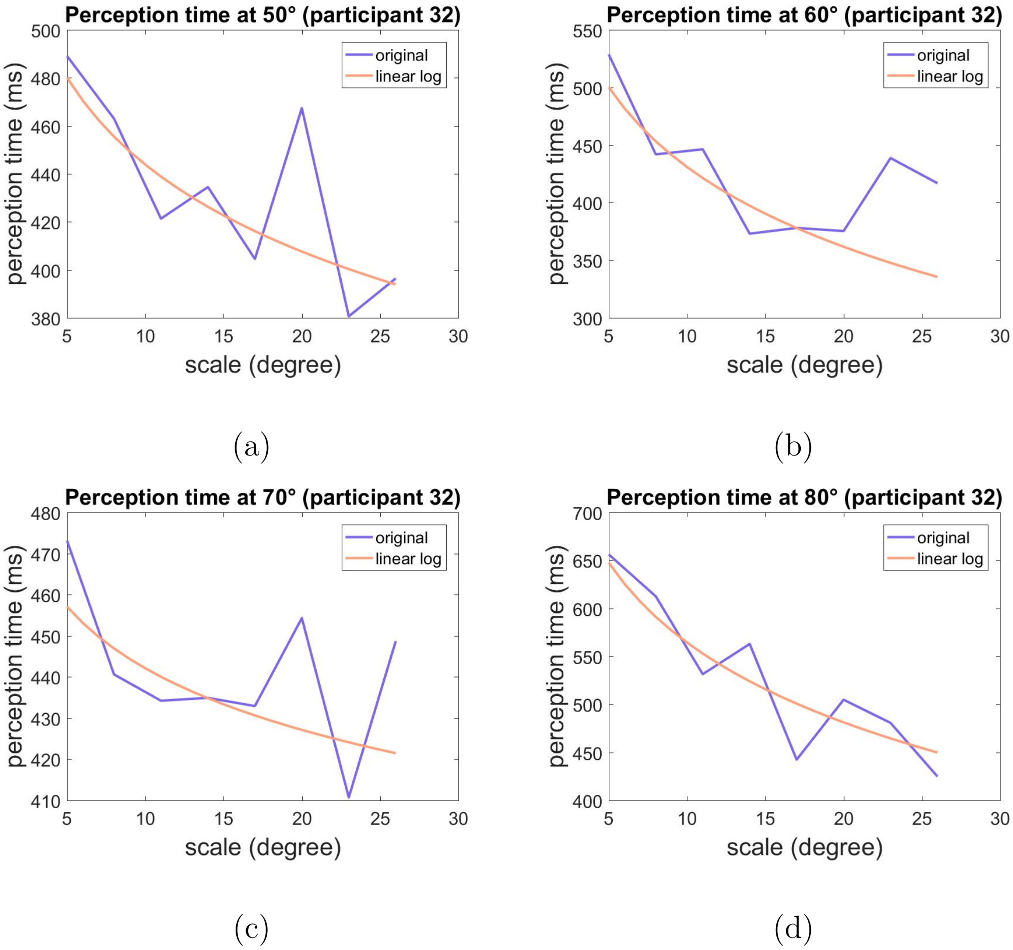
The aim of this study is to determine the optimal scaling of the content displayed at higher eccentricities so that it is perceived in a similar time as in the central vision. Specifically, for a pre-specified perception time T_0, we want to find at each higher eccentricity the scale at which the pattern's perception time is equal to T_0.

First, we derive the relationship between perception time T and scale S at a given eccentricity. We plot the perception time at each tested scale for a given eccentricity directly from the user study data. We wish to represent the perception time as a function of the scale. As can be seen in Figure 2.5, for a given eccentricity, perception time generally decreases as the scale gets larger.

Judging from Figure 2.5, we fit the relation of perception time and scale as a function in the following form.

$$T(S) = a\log(S) + b \tag{2.3}$$

$T(S)$ is the perception time in milliseconds, S is the scale in degrees in the visual field, and a, b are parameters. The function $T(S)$ is calculated for each eccentricity for each participant. We only calculate the perception time vs. scale relation on eccentricities above 40° because we are interested in calculating this relationship for the far peripheral vision.



(a) (b)

(c) (d)

Figure 2.5: Figures (a), (b), (c) and (d) show the relation between perception time and scale measured for a single user (participant 32) at 50°, 60°, 70° and 80° eccentricities, respectively.

2.4.2 Perception Time and Eccentricity

We present the relation between perception time and eccentricity in Figure 2.6. Specifically, we show the median perception time of all participants with 5°, 10° and 20° patterns at higher eccentricities. When the size of the pattern is kept constant, it takes longer for the patterns at higher eccentricities to be perceived. Perception time decreases as the pattern size increases at the same eccentricity (see Figure 2.5). As is shown, there is a clear difference between the time in perceiving the same-sized

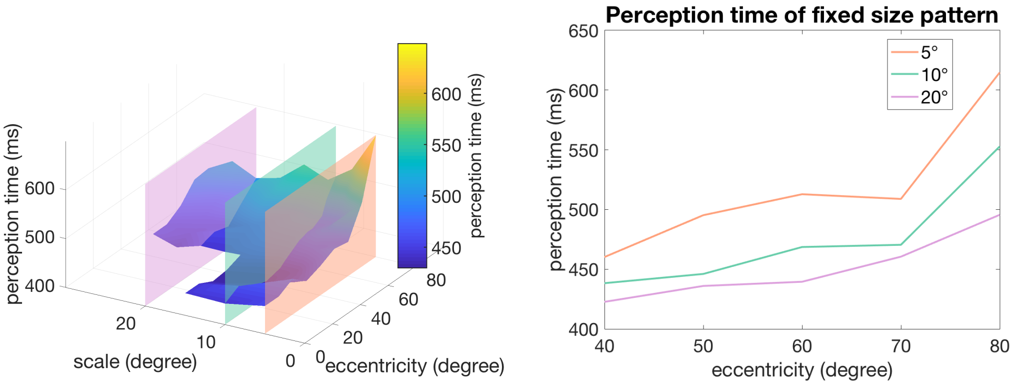


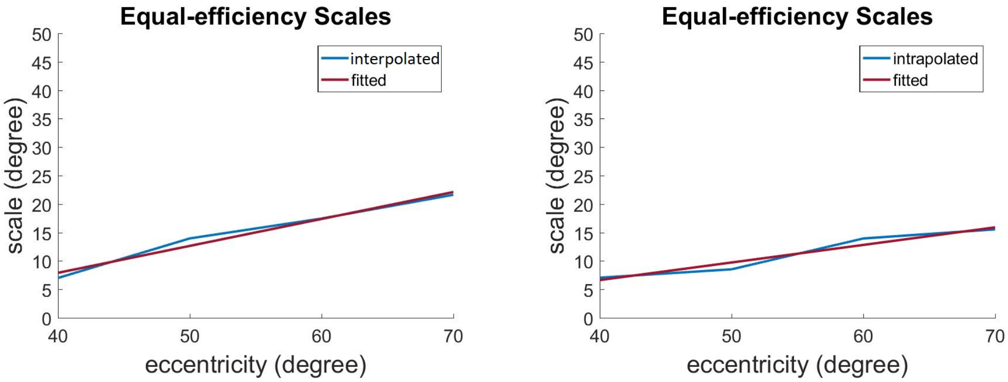
Figure 2.6: The perception time across eccentricities with an s-sized pattern is the intersection of the plane $scale = s$ and the perception time surface in Figure 2.3. The intersection with 5°, 10° and 20° scale planes are shown in (left). The intersection curves are shown in (right).

pattern at a lower eccentricity versus at a higher eccentricity. The 150 ms difference between perceiving a 5° pattern at 40° and 80° eccentricity is noticeable and it is meaningful to characterize the scaling at higher eccentricities to bridge this gap.

2.4.3 Equal-Efficiency Scale at each Eccentricity

With the function $T(S)$, we can calculate for any given eccentricity the scale at which the pattern is perceived within a given amount of time. Therefore, we can plot the relations of eccentricity and scale for a desired perception time. We call this the *equal-efficiency scale*.

Given a target perception time, we calculate the equal-efficiency scale for each participant and present the median of all participants in Figure 2.7. We note that we only test within the scale range of [5,26]°. The equal-efficiency scales at 80° eccentricity for a sizable portion of the participants are outside our tested range and therefore cannot be considered valid. The equal-efficiency scales within the



(a) (b)

Figure 2.7: Equal-efficiency scales that produce the same perception time as (a) a 5° pattern at 10° eccentricity and (b) perceived in 450 ms. These are the median over all participants.

tested range exhibit a linear relationship between scale and eccentricity. $S(E) = \alpha E + \beta, 40 \leq E \leq 70$.

We present the parameters for a few example target perception time choices in Table 2.2. We can either use a fixed perception time, e.g. 450 ms, or we can specify a target pattern, a pattern of a given size at a given eccentricity. The time it takes to perceive this pattern is used as the target perception time. We also conduct one-way ANOVA tests on the equal-efficiency scales of all participants for each high eccentricity with the null hypothesis being that equal-efficiency scales over different eccentricities follow the same distribution.

We illustrate the equal-efficiency scales in a theme park navigation example in Figure 2.8. We choose the target perception time to be the perception time of a 5° sized pattern at 10°, 20° and 40°, respectively. The patterns at higher eccentricities are scaled so they can be perceived within the target perception time. The target perception time increases as the eccentricity goes from 10° to 40°. The target equal-

Table 2.2: Parameters of the equal-efficiency scales fitted to a linear model. The valid range of eccentricity is [40, 70]°. Analysis of variance is conducted with the all participant's equal-efficiency scale at a certain eccentricity as a group. We include the last 5 rows because the equal-efficiency scales at 70° eccentricity are no more than 4° above the tested range.

Target perception time	α	β	dF	F	p-value
5° at ecc 10°	0.47	-11.04	3	7.02	0.0002
5° at ecc 20°	0.55	-16.3	3	11.96	4.46×10^{-7}
5° at ecc 30°	0.54	-15.08	3	7.96	5.75×10^{-5}
5° at ecc 40°	0.54	-17.64	3	12.48	2.45×10^{-7}
10° at ecc 10°	0.52	-13.11	3	4.78	0.0033
10° at ecc 20°	0.58	-14.8	3	9.3	1.13×10^{-5}
10° at ecc 30°	0.65	-16.52	3	7.04	0.0002
15° at ecc 10°	0.65	-18.49	3	4.22	0.0067
15° at ecc 20°	0.51	-6.67	3	5.32	0.0017
15°at ecc 30°	0.67	-18.74	3	5.31	0.0017

efficiency scales decrease as a result. The scales in Figure 2.8 are calculated using the linear equations in Table 2.2.

2.5 Conclusion

Wide FOV VR and AR displays enable information perception in the far periphery. In this work, we conduct a user study to investigate the perception time in the far peripheral vision. We found that patterns can be scaled linearly with the eccentricity it is shown at to reach a constant perception time. Based on these equal-efficiency scales, we can optimally scale the content displayed at higher eccentricities so that it can be perceived efficiently. Our finding is useful for designing VR and AR experiences that are able to leverage the entire human visual system's field of view.

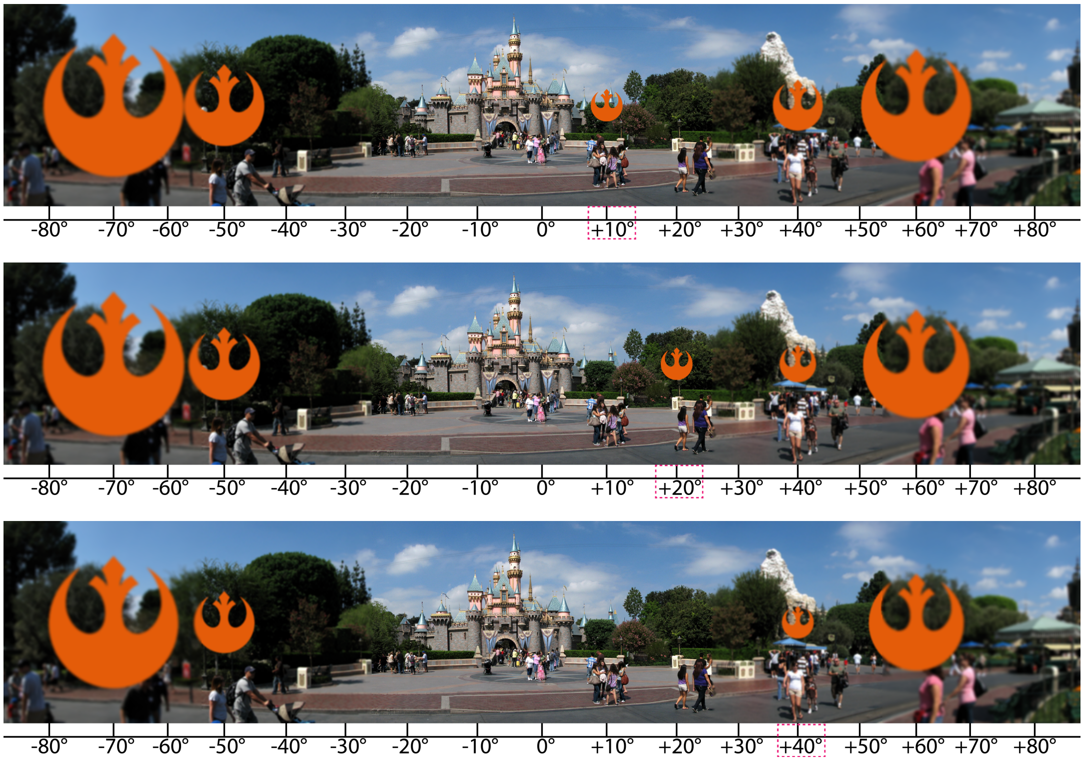


Figure 2.8: Visualization of equal-efficiency scales for a notional theme park navigation. The Star Wars attraction logo is scaled at various eccentricities using equal-efficiency scales. The target perception times are the perception time of a target pattern of size 5°, at eccentricities (top) 10°, (middle) 20°, and (bottom) 40° (these are shown with red dashed boxes). Theme park image reproduced with the permission of Bernie Kelm, https://www.rocket9.net.

Chapter 3: AR Displays for Surgery

3.1 Overview

External ventricular drain (EVD) placement involves inserting a very small clear plastic tube into a lateral ventricle of the brain. Once the catheter is inside the skull, its position can no longer be directly observed, and the neurosurgeon must rely on his or her experience and knowledge of human anatomy to guide placement of the catheter. Huyette *et al.* [65] and Kakarla *et al.* [66] retrospectively assess the accuracy of ventriculostomy catheter placement by comparing postprocedure CT scans with preoperative scans. The former study includes 98 ventriculostomy catheter placements and finds that in the author's institution, it requires on average 2 passes to successfully place the catheter, and 22.4% of those successful placements end up in nonventricular spaces. The latter includes studies of 346 patients and finds that 13% of the catheter placements are suboptimal, resulting sometimes in unfunctional drainage. Furthermore, Woo *et al.* [67] finds that catheter misplacement is a significant independent predictor of catheter-associated hemorrhage. In a radiographic simulation, Robertson *et al.* [68] finds that the traditional method of EVD trajectory selection and placement is predicted to have a 19% chance to impact a cortical vein. These results show that there is still much room for improvement. A method of visualizing the location of the catheter inside the skull would be of great assistance to the surgeon.

A few EVD visualization techniques have been proposed. Cutler *et al.* [69] have developed a system in which an optical marker is attached to the catheter and tracked by the camera on a head-mounted display (HMD). Other medical instrument tracking techniques by Low *et al.* [70], Najafi *et al.* [71, 72], Stolka *et al.* [73] and Fan *et al.* [74] make use of markers of different kinds.

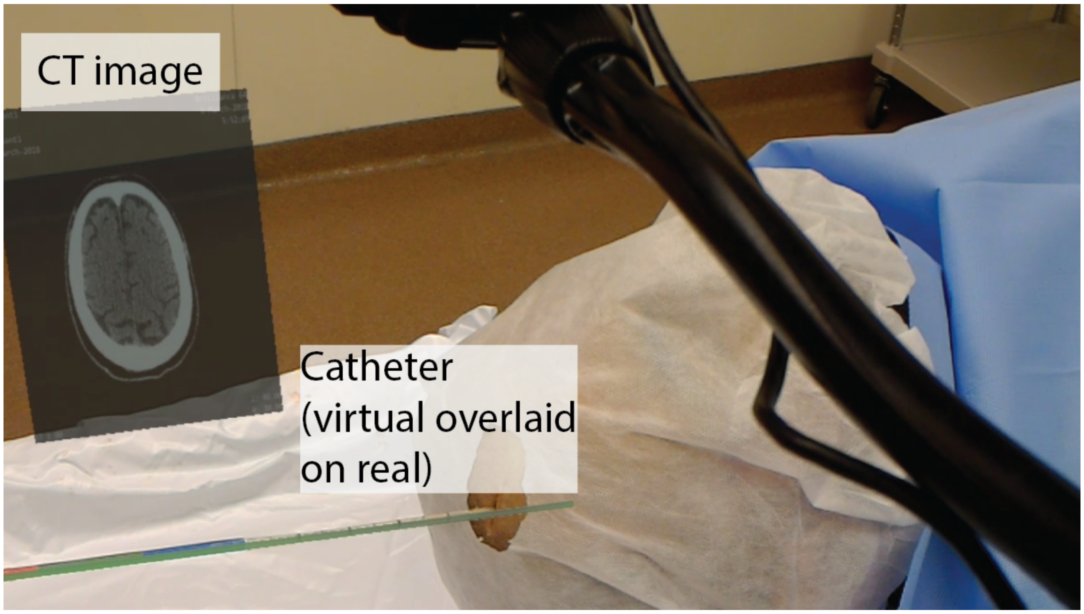


Figure 3.1: The view observed by the surgeon (captured with HoloLens mixed reality capture function). This is a realistic surgical scenario where EVD is performed on a cadaveric head. Additional information could be shown such as the CT slice at the tip of the catheter if a CT volume is registered to the patient. The CT image shown in this figure serves as an illustration of this potential. Other medical imaging visualization techniques such as in Bista *et al.* [3, 4] may also be used.

Preliminary tests with Microsoft Kinect 2.0 showed that the thinness of the catheter makes it hard to detect and track with commodity depth sensors. The shortcomings of infrared depth sensing, in this case, are out of the scope of this chapter. Instead, we devise a new optical marker and tracking technique, suitable for augmented reality. We label a portion of the catheter with three distinct colors. One design principle we have observed after consulting our medical collaborators is making minimal changes to the catheter. Our modification does not change the catheter in any way that would make it feel different to operate. We develop an algorithm to detect the color bands and then use them to calculate the position of the catheter. This way, even if the tip of the needle is occluded, we still know where it is and are able to visualize it, as long as enough of the colored portion remains visible. We display a virtual catheter overlaid on the real catheter directly in the user's field of view using a see-through Microsoft HoloLens, which allows the surgeon to better focus on the patient while still having access to the relevant data and provides better depth perception and understanding of the operating context. We only need to do a one-time calibration to determine the transformation between the HoloLens and the camera. Testing shows our system achieves high accuracy and low latency. We also show the usefulness of our system in a realistic surgical environment on a cadaveric head. An example of the view directly observed by the surgeon is shown in Figure 3.1.

Our system takes a step beyond Cutler *et al.*'s pioneering research [69], and improves on several aspects. First, our tracking technique does not require significant changes to the shape and weight of the catheter. Second, since we use a stationary camera for tracking, it does not require the user of the head-mounted display to be looking at the catheter to track. This allows the surgeon to freely look anywhere without losing the tracking of the catheter. Finally, the capability of our system is not tied to the HMD. We empirically observed a lag of 1.43 s in Cutler *et al.* [69]'s video demonstration[1], which is likely caused by the HMD's inability to keep up with the processing. In our system, the images from the camera, as well as the medical volume are processed separately on another machine, and results are sent to the HMD wirelessly. This allows for faster and more accurate sensing and higher fidelity medical images. Our system can be naturally integrated with existing medical image registration techniques. A proposed example is illustrated in 3.1 where the CT slice at the tip of the catheter is displayed if the CT volume is registered with the patient. Due to the scope of this study, we do not perform the medical image registration but refer interested readers to surveys by Kersten-Oertel *et al.* [75] and Meola *et al.* [76].

In short, we built an augmented reality EVD tracking and visualization system that

- overlays a virtual catheter onto a user's field-of-view, mimicking the position of the real catheter,
- leaves the catheter functionally unchanged, and
- is accurate and responsive.

[1] We observed this at 0:50 and 0:51 in video https://youtu.be/qqykWW9f41Q

3.1.1 Technical Details from Previous AR Ultrasound System

Before developing this AR assistant system for EVD, we developed an AR ultrasound system where we gained a lot of experience in developing AR applications. The AR ultrasound system streams the video from a medical imaging machine live to be viewed in an AR head-mounted display. With this, the surgeons can focus on the patient during a procedure instead of having to look away to look at various monitors. For the sake of completeness, we document these technical details.

The AR platform we use is the Microsoft HoloLens[2]. We develop and deploy the Hololens application (client application) from a Windows machine. The application is designed with Unity[3] and the C-sharp scripting language.

The HoloLens is completely tetherless. The information, both the location of the catheter and the medical images are sent to the Hololens from another program (feeder program) running on a separate machine via WiFi. The feeder program for the AR ultrasound system is developed with Unity and C-sharp. And the feeder program in the AR assistant system for EVD is developed with C++ and OpenCV. We use the User Datagram Protocol (UDP) in our system (DatagramSocket in the .Net framework and Winsock[4]).

To get the live medical images from the imaging machines, we use a portable video grabber (AV.io[5]) to send the output from the imaging machines to the feeder

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

[3] https://unity3d.com/

[4] https://docs.microsoft.com/en-us/windows/desktop/winsock/windows-sockets-start-page-2

[5] https://www.epiphan.com/products/avio-hd/

programs as a webcam signal.

With this setup, it is possible to send the same information to multiple HoloLenses from one instance of the feeder program. We have found success with two HoloLenses.

3.2 Related Work

External ventricular drainage has undergone numerous iterations in materials and techniques since it was first performed in the 1740s. The complete documentation of the evolution of EVD is out of scope for this chapter; readers can refer to Srinivasan *et al.* [77]. In this section, we will review recent developments in EVD catheter placement techniques. In addition, we discuss augmented reality use cases in neurosurgery.

3.2.1 Catheter Placement

Proper catheter placement is key to the success of an EVD procedure. A 2016 Neurocritical Care Society study (Fried *et al.* [78]) found that with a stabilization tripod ("Ghajar guide") or other technical adjuncts such as CT guidance, stereotactic navigation could improve accuracy and/or lower the number of EVD procedures performed on a patient.

The "Ghajar guide" is a catheterization assistance device in current clinical use, invented by Jamshid Ghajar in 1985. It is a tripod-shaped device that can be securely placed on the patient's skull; the catheter travels along the axis of the device and can be stabilized as it goes into the skull. A study by O'Leary *et al.* [79] found that successful catheterization can be achieved either using a Ghajar guide or freehand, but that a smaller distance to target was achieved with the Ghajar guide.

For ventricle exploration and catheter placement training, Phillips and John [80] developed a simple web application that lets trainee surgeons practice simulated catheterization (catheter placement) via a personal computer. Cros *et al.* [81] incorporated haptic feedback into the ventriculostomy simulator; users feel a "pop" sensation when the catheter pierces the ventricular surface.

Krombach *et al.* [82] conducted a free-hand catheterization simulation before performing the actual surgery. In simulations, MRI images were registered onto the patient's head with fiducials. A pointer instrument was tracked by infrared diodes attached to it. The surgeons used the pointer to virtually move the catheter and the position of the catheter was evaluated.

Luciano *et al.* [83] developed the *ImmersiveTouch* ventriculostomy simulation system. Stereoscopic goggles are tracked electromagnetically and represent the head position. A hand stylus is tracked mechanically and can offer different resistance sensations to simulate catheter moving in different types of tissue. This way, the displayed content can change according to the user's point of view and the virtual catheter is visualized based on the location of the hand stylus. Haptic feedback is provided to perceive the interaction between the virtual catheter and the virtual brain model. ImmersiveTouch is used in studies by Banerjee *et al.* [84], Lemole *et al.* [85], Schirmer *et al.* [86] and Yudkowsky *et al.* [87].

Tai *et al.* [88] 3D-printed an actual physical phantom head (manikin head for medical procedure simulations) from the reconstruction of CT images and used the brain model as an EVD training platform. This platform provides a realistic anatomical structure and visual/haptic feedback.

The most recent experimental EVD development is by Cutler *et al.* [69]. In their system, a rigid catheter is attached to an optical marker, in this case, a piece of paper. The marker's, and therefore the catheter's position is tracked by the camera on an HMD, which displays the virtual catheter image as well as a 3D model of the damaged part of the brain. This way, as long as the marker is in the camera's field of view, the catheter can be visually tracked, even when the tip is occluded. The system compellingly demonstrates how the advances in sensing and displays can create intuitive and reliable assistance to neurosurgery. Somewhat similar to Cutler *et al.* [69] but used in biopsy and ablation are Lin *et al.*'s HoloNeedle [89] and Kuzhagaliyev *et al.* [90], where the needle is tracked using the OptiTrack[6] system and a virtual needle is displayed in an AR headset to align with the real one. In Lin *et al.*'s HoloNeedle, the biopsy needle's shape is further obtained with embedded Fiber Bragg Grating sensors to take the needle bending into consideration. All the systems mentioned above make a significant change to the shapes of the instruments to accommodate their tracking techniques which our method avoids (more detail in Section 3.4). Najafi *et al.* [71,72] use a closed-form formula to calculate the trajectory of a needle instrument based on the live footage from a single camera attached to an ultrasound probe and guide the needle angle to reach the target shown in the ultrasound image. Our system is different in that the exact location of the catheter

[6] https://optitrack.com/

tip is calculated visualized directly in the user's field of view via an AR HMD. Stolka *et al.* [73] use stereo cameras to detect the 5DOF information of a needle marked with pseudo-random binary sequence(PRBS) patterns and display the needle in the ultrasound image or project a line on the patient's skin to mimic the needle's location. A generalized version incorporating multiple medical imaging and probing modalities has been proposed in Basafa *et al.* [91]. Our system is different in that it uses perspective foreshortening of color bands on the catheter to estimate the tip location from a single camera with higher precision than the prior art.

3.2.2 Augmented Reality Displays for Neurosurgery

One type of augmented reality technique being used in neurosurgery is overlaying detailed preoperative medical images on intraoperative images. Masutani *et al.* [92] experimented with overlaying preoperative 3D vascular models onto intraoperative X-ray fluoroscopy images via medical fiducial markers. Two clinical test cases showed the registration reprojection displacement to be below 2.6 mm. Lovo *et al.* [93] reconstructed volumes from cerebral MR scans and overlaid the volumes onto intraoperative cerebral images.

One step further is to track the imaging probe (or microscope) during surgery and overlay meaningful information such as anatomy models onto the intraoperative images. Edwards *et al.* [94–96] developed and tested an AR microscope-assisted guided intervention system that tracked the microscope and overlaid 3D medical images on the two eyepieces. Shahidi *et al.* [97], Scholz *et al.* [98] experimented with similar setups using surgical endoscopes. Kockro *et al.* [99] developed a hand-held AR system, *Dex-Ray*, that overlays medical volumes or other images onto a live stream video. Low *et al.* [70] used Dex-Ray integrated with the *Dextroscope* system (an immersive medical image viewing system) and performed meningioma (a type of usually noncancerous brain tumor) excision on five patients. Inoue *et al.* [100] tracked off-the-shelf webcams to overlay anatomical medical structures on the webcam images.

Sauer *et al.* [101] and Maurer *et al.* [102] developed an AR video see-through surgery navigation HMD and demonstrated its use on a phantom head with encouraging results. Two color cameras record the real-world scene, and one additional camera tracks the retroreflective optical markers attached to the phantom head. A medical MR (magnetic resonance) volume aligned with the head phantom is displayed stereoscopically on the HMD. Abe *et al.* [103] develop an AR guidance system for percutaneous vertebroplasty where the insertion location and intended insertion orientation is displayed in an AR HMD. Trials on the phantom show improved accuracy using this system, and successful clinical trials demonstrate the practical usability. Abhari *et al.* [104] engineer a mixed reality system that facilitates the training and planning for a brain tumor resection procedure and demonstrated the greatly improved performance of junior residents and the reduction in time to perform clinically relevant tasks by clinicians. Li *et al.* [105] conduct clinical EVD trial where manually aligned CT holograms are shown in an AR headset to display the catheter placement trajectory and target location. The catheter is not tracked or visualized, but it took the AR-assisted group fewer passes and less time on average to complete the procedure. Azimi *et al.* [106] develop a mixed reality surgical training system that integrates functionalities such as preoperative procedure visualization and real-time performance assessment and demonstrate a catheter placement training session where the system informs the trainee of the orientation error, though the system's tracking accuracy is not reported. Very recently, Meola *et al.* [76] and Guha *et al.* [107] present two thorough surveys on the past work in which AR is used for neurosurgery.

3.3 System

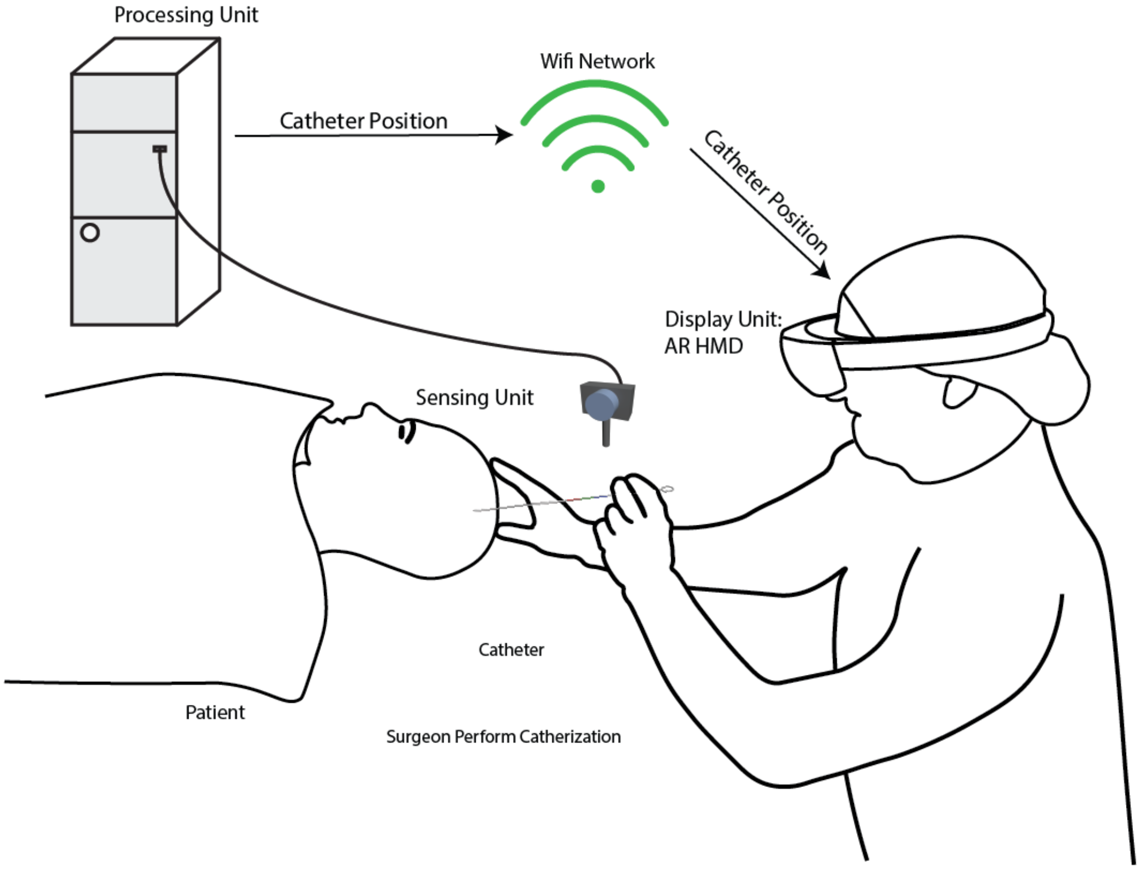


Figure 3.2: System Overview

Our system consists of three components: a sensing unit, a processing unit and a display unit in addition to the modified catheter itself. The functions of each component are described in Section 3.3.1, as well as the actual hardware used for each component. The way the system is operated in our experiment is described in Section 3.3.2. Finally, the calibration between the sensing unit and the display unit is described in detail in Section 3.3.3.

3.3.1 System Components

The sensing unit captures the live footage of the operation area that contains the modified catheter. The processing unit analyzes the footage and calculates the catheter's position in the camera's reference system (camera space), which is then sent via a wireless network to the display unit. The display unit uses a transformation obtained at the calibration step (more detail in Section 3.3.3) to transform the received catheter's location into the HMD's reference system (HMD space) and displays the virtual catheter at that transformed location to overlay the physical catheter. A general illustration of the system, as well as the intended use scenario, can be seen in Figure 3.2. The workflow of our system is depicted in Figure 3.3. An actual view captured using HoloLens's mixed reality capture function is shown in Figure 3.1.

The sensing unit is an off-the-shelf Logitech c920 web camera fixated on a tripod. The reason we use a separate camera, rather than the one integrated into the display, is to avoid being limited by the quality of the integrated camera and the processing capabilities of the display, as well as reduce network latency. The processing unit is an Alienware laptop computer with an Intel i7 7820HK processor running at 2.9GHz. The display unit is a Microsoft Hololens tetherless AR head-mounted display.

The catheter used in our experiment is a Codman® EDS 3™ clear ventricular CSF catheter. This catheter measures 35 cm in length which is typical of catheters used in EVD. Fried *et al.* [78] suggest that, in surgical scenarios, the catheter should not penetrate the skull by more than 6.5 cm, therefore the place the color bands at 7 cm away from the tip. We use three color bands each 3.8 cm in length (see Figure 3.5 for an illustration).

3.3.2 System Operation

When using this system, the user wears the HMD which displays a virtual catheter. The virtual catheter can be dragged around in the HMD space with hand gestures. After the user manually aligns the virtual catheter with the physical one observed through the HMD, the user issues a calibration command which triggers the system to calculate the mapping from the camera space to the HMD space. The system then uses this mapping to transform the catheter's location in the camera space into HMD space as mentioned above for the subsequent frames, until a new calibration command is issued when the users see fit.

In the experiment, the surgeons held the rear portion of the catheter as illustrated in Figure 3.2 to avoid occluding the color bands.

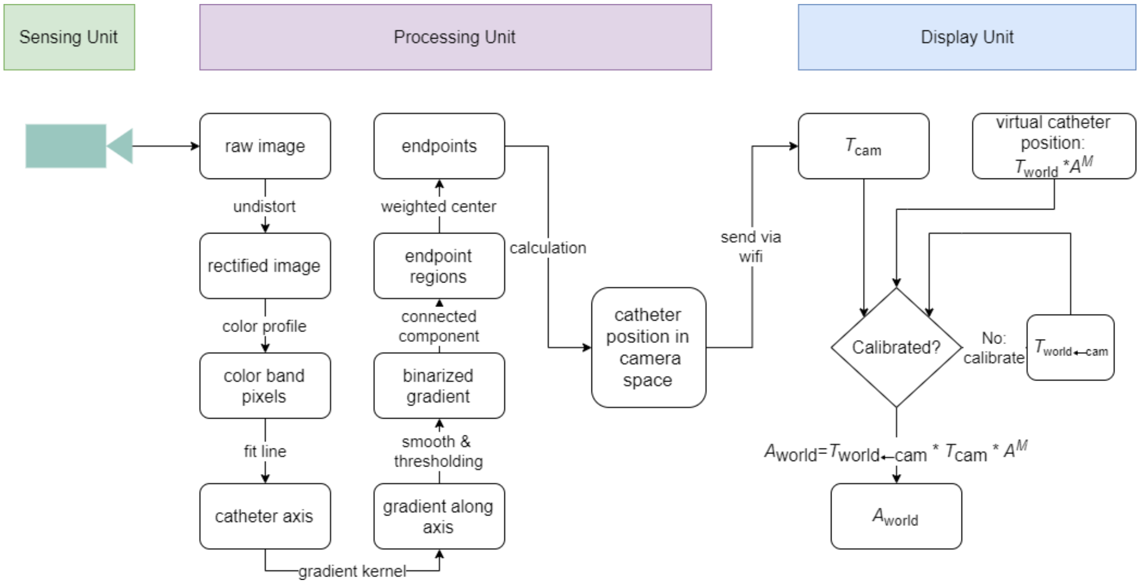


Figure 3.3: The workflow of the system

3.3.3 Reference System mapping

Our algorithm calculates the position of the catheter in the camera's reference system (camera space). To be able to visualize the catheter in the HMD, a way to transform from camera space to the HMD's reference system (HMD space) is required. We achieve this by doing a one-time calibration with the calculated catheter position in camera space and the virtual catheter position in HMD space. Suppose the coordinate of the catheter in its own reference system (model space) is A^M. At a certain frame, T_{cam} and T_{hmd} transform A^M into camera space and HMD space respectively. We want to find a transformation $T_{cam \to hmd}$ that will transform from camera space to HMD space.

$$T_{cam} T_{cam \to hmd} A^M = T_{hmd} A^M \tag{3.1}$$

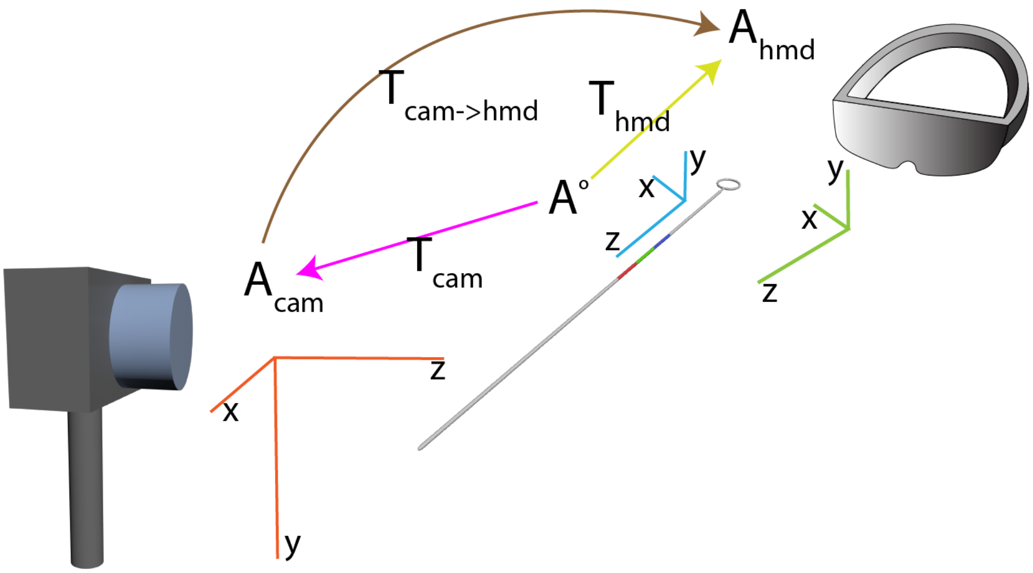


Figure 3.4: Reference system mapping

In order for this to work with every A^M, we simply take

$$T_{cam \to hmd} = T_{cam}^{-1} T_{hmd} \tag{3.2}$$

Then with every frame where a new T_{cam} is calculated from the position of the catheter in the image, we can calculate a corresponding T_{hmd} and display the virtual catheter. Reference system mapping is illustrated in Figure 3.4

3.4 Catheter Tracking

We develop a way of tracking the 3D position of the catheter in real-time, enabled by three color bands painted on the catheter (see Figure 3.5). The three color bands are adjacent, and we know each of their lengths and locations relative to the catheter; the colors only need to be sufficiently distinct to be identified in the image.

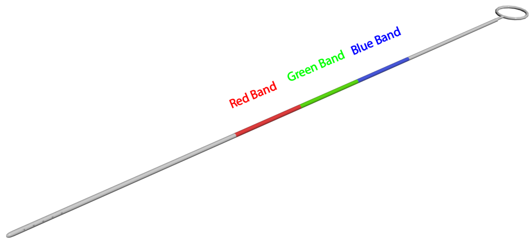


Figure 3.5: A drawing of the catheter and the color bands. The color bands need to be distinct and continuous. The lengths of the three color bands, as well as the uncolored forward portion (from the catheter tip to the beginning of the first color band) of the catheter, are known. With the lengths and the positions of the color band endpoints detected in camera space, we can calculate the 5-DOF (no roll) information of the catheter, and infer the position of the tip.

We use a pinhole model to represent the camera. As seen in Figure 3.6, P is the center of projection of the pinhole model and is the origin point in camera space. We observe in the image the 2D coordinates of the catheter A', B', and C' (Figure 3.6 shows three endpoints of two color bands as an example). Given that we know the lengths of the color bands $|AB|$ and $|BC|$, this becomes a simplified version of the perspective three-point problem (P3P) thoroughly studied and documented in [108–111]. In a general P3P problem, there could be as many as four solutions. In our case, where the three points are collinear, there could be two solutions, of which only one is desired. Determining the correct solution requires finding the angle between the catheter and its image in the image plane, shown in Figure 3.6.

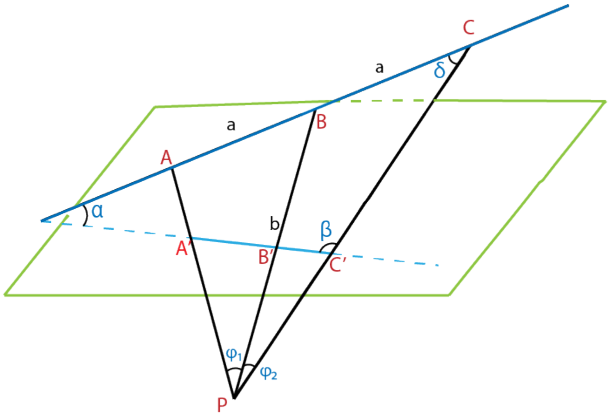


Figure 3.6: We need to find the angle α between the catheter and its image. Let A, B, and C be three endpoints of two consecutive color bands. The color bands are of equal length a. P is the center of projection in the pinhole camera model. The distance between B and P is b

We describe our approach to the P3P problem below. First, we use a geometric method to find α in Section 3.4.1. Then we calculate the position of the catheter in Section 3.4.2. Finally, we extract the segments' endpoints A', B' and C' from the image in Section 3.4.3.

3.4.1 Angle α

We denote the three endpoints of two consecutive color bands as A, B, and C in Fig 3.6, and the center of projection as P. The images of A, B and C are A', B' and C' on the image plane. Given the camera intrinsics which can be obtained by a one-time calibration, we can explicitly represent P, A', B' and C' with exact coordinates in pixels. We can calculate the angles

$$\angle APB = \varphi_1 = \mathrm{acos}\frac{\mathbf{PA'} \cdot \mathbf{PB'}}{|\mathbf{PA'}||\mathbf{PB'}|} \tag{3.3}$$

$$\angle BPC = \varphi_2 = \mathrm{acos}\frac{\mathbf{PB'} \cdot \mathbf{PC'}}{|\mathbf{PB'}||\mathbf{PC'}|} \tag{3.4}$$

Now assume the two color segments are of equal length a (Assume $|AB| = |BC| = a$) and $|PB| = b$. Trigonometric equations give the following relations. δ is the angle $\angle ACP$.

$$\frac{a}{\sin\varphi_1} = \frac{b}{\sin(\pi - \varphi_1 - \varphi_2 - \delta)} \tag{3.5}$$

$$\frac{a}{\sin\varphi_2} = \frac{b}{\sin\delta} \tag{3.6}$$

δ can be solved with the following equation.

$$\delta = \arctan(\frac{\sin\varphi_2\sin(\varphi_1 + \varphi_2)}{(\sin\varphi_1 - \sin\varphi_2\cos(\varphi_1 + \varphi_2))}) \tag{3.7}$$

Then we can calculate $\alpha = \pi - \beta - \delta$. α could be either positive or negative depending on whether A is farther away from the image plane than C or closer. When A is closer than C (Figure 3.6), $\alpha > 0$. When A is farther, $\alpha < 0$.

3.4.2 Position of Catheter

With the 3D orientation, we can calculate the position of the catheter in camera space. In Figure 3.7, AB represents a color band. AT represents the portion of the catheter from the tip to the color band. AT may not be entirely visible from the camera. A', B', and T' are the images of A, B and T on the image plane,

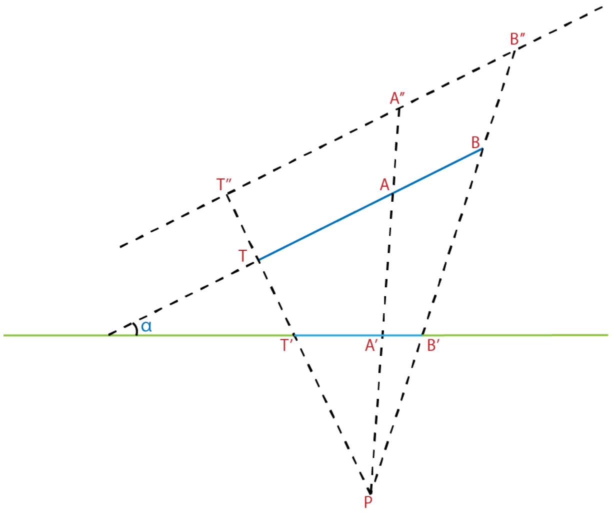


Figure 3.7: Finding the position of the catheter in camera space. P is the center of projection. TA is the uncolored forward portion of the catheter. AB is the first color band on the catheter.

respectively. We know the lengths of $\boldsymbol{A'B'}$ because we have detected them in the image. We also know the lengths of $\boldsymbol{PA'}$ and $\boldsymbol{PB'}$ as well.

The actual catheter is in the plane formed by P and $T'B'$, and we know its orientation. But it could be on any line that is parallel to TB, for example, $T''B''$. We can set $\boldsymbol{PA''} = m\boldsymbol{PA'}$. The actual ratio m does not really matter because a different ratio only leads to a parallel line to BC. We can assume, for example, $m = 10$, and let $\boldsymbol{PB''} = k\boldsymbol{PB'}$. Then $\boldsymbol{A''B''} = \boldsymbol{PB''} - \boldsymbol{PA''} = k\boldsymbol{PB'} - \boldsymbol{PA''}$.

We know the angle formed by the catheter and its image α, which is the angle formed by $A''B''$ and $A'B'$.

$$\boldsymbol{A''B''} \cdot \boldsymbol{A'B'} = |\boldsymbol{A''B''}||\boldsymbol{A'B'}|\text{cos}\alpha \tag{3.8}$$

Equation 3.8 can be further transformed into

$$k\boldsymbol{PB'} \cdot \boldsymbol{A'B'} - \boldsymbol{PA''} \cdot \boldsymbol{A'B'} = \sqrt{k^2\boldsymbol{PB'}^2 - 2k\boldsymbol{PB'}\boldsymbol{PA''} + \boldsymbol{PA''}^2}|\boldsymbol{A'B'}|\text{cos}\alpha \tag{3.9}$$

All the values in Equation 3.9 are known except for k. We can solve k and thus find the position of B''. It is worth noting that Equation 3.9 is a quadratic equation, and there could be two solutions for k. Judging from Figure 3.7, if $\alpha > 0$, B should be further away from the image plane than A. Therefore k should be greater than the assumed ratio of $|\boldsymbol{PA''}|$ over $|\boldsymbol{PA'}|$ and vice versa.

Now that we know the length of the catheter color band AB in the real world, we can calculate the position of the color band AB in the camera space.

$$\boldsymbol{PA} = \frac{|\boldsymbol{AB}|}{|\boldsymbol{A''B''}|}\boldsymbol{PA''} \tag{3.10}$$

$$\boldsymbol{PB} = \frac{|\boldsymbol{PA}|}{|\boldsymbol{PA''}|}\boldsymbol{PB''} \tag{3.11}$$

Since we know where AB is on the catheter, we can calculate the position of the catheter in camera space. For example, we can calculate the position of the tip in camera space since we know the lengths of the uncolored forward portion of the catheter and the color bands.

$$\boldsymbol{PT} = \boldsymbol{PA} - \frac{|\boldsymbol{AB}|}{|\boldsymbol{AT}|}\boldsymbol{AB} \tag{3.12}$$

It is worth noting that we can calculate the position of the catheter with two adjacent color bands. We use three color bands to make our system robust against occlusion and improve accuracy when all three are visible.

3.4.3 Color Band Detection

In the color band detection step, the input is an image I of the catheter ($P_{catheter}$ being all the pixels that belong to the catheter), and the output is the locations of the endpoints of the color bands. The accuracy of the color band detection is the most important factor in finding the catheter tip. To reliably detect the endpoints of the color bands, we develop a new algorithm (see Figure 3.3). Starting from the original image (Figure 3.8 (a)) and its undistorted result (Figure 3.8 (b)), we first use simple thresholds to get the most of the pixels $P^*_{catheter} \subseteq P_{catheter}$ of the catheter in the image. We then fit these pixels to a line, which we call the axis L_{axis} of the catheter (Figure 3.8 (c)). The gradient $G_{L_{axis}}$ along the direction of L_{axis} is calculated using a variation of Sobel filter $K_{L_{axis}}$ that is weighed anisotropically according to L_{axis} (Figure 3.8 (d)). $G_{L_{axis}}$ has high values near the border of two different colors. After smoothing using a median filter, the gradient map is thresholded into binary images $G^*_{L_{axis}}$ with positive values where the borders are (Figure 3.8 (e)). Finally, we take the weighted centers of the connected components of $G^*_{L_{axis}}$ as the endpoints of color bands (Figure 3.8 (f)). Our processing technique is robust against blurry images caused by motion, as seen in Figure 3.8 (g, h, i).

3.5 Results

In this section, we analyze the performance of our proposed algorithm. First, we measure the stability of our tracking algorithm. Then we test tracking accuracy on a 2D planar grid as well as a specially designed and printed 3D grid. Finally, we

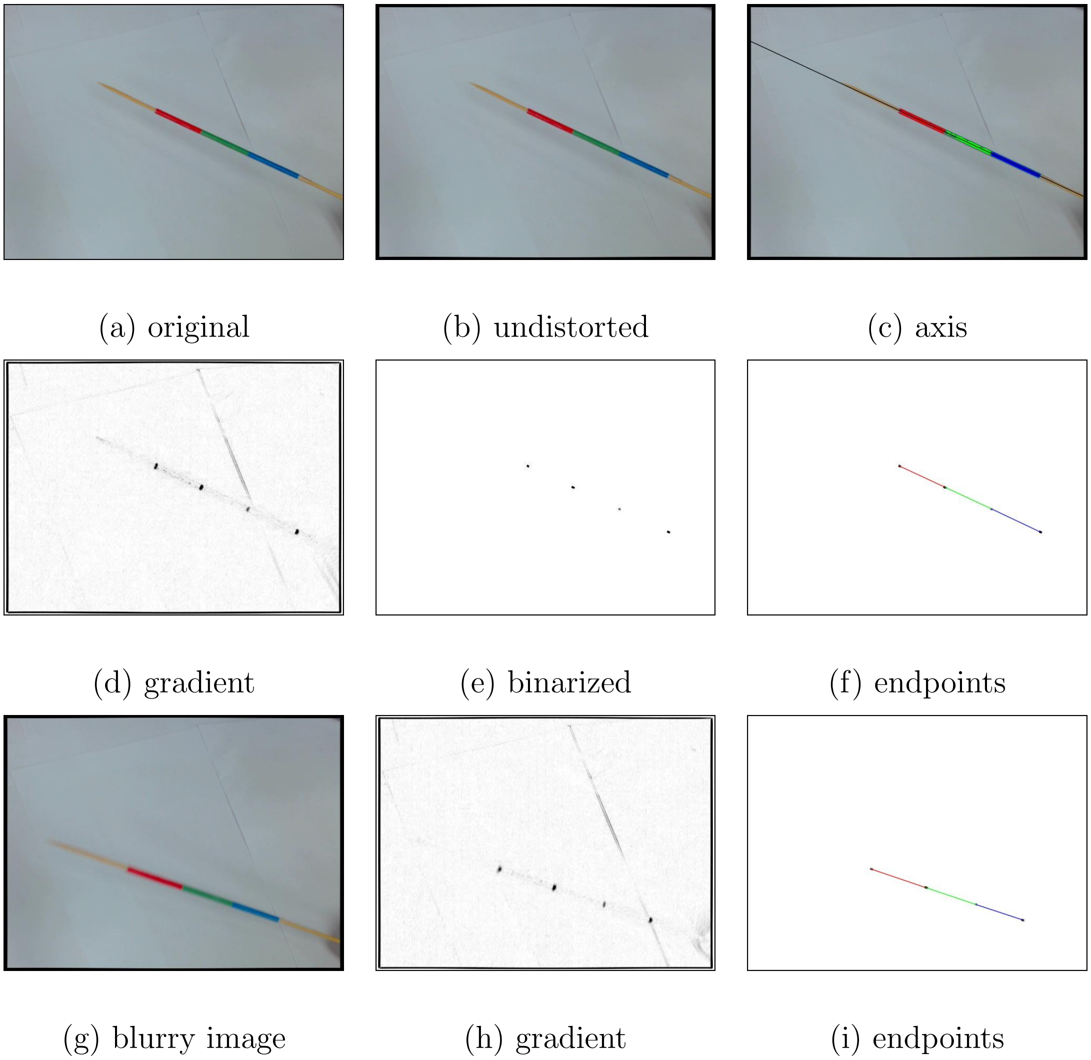


Figure 3.8: Detecting endpoints. (g), (h) and (i) show detecting endpoints in an image where the catheter is blurry because of movement. Our algorithm is able to handle this case.

test the latency compared with a third-party tracker.

3.5.1 Stability

Given the color segment endpoints detected in the image, our tracking algorithm will output the computed catheter position. The instability in the tracking algorithm comes from random noise in each frame. The noise could cause the same endpoint in two frames to be detected a few pixels apart, even when the catheter stays still.

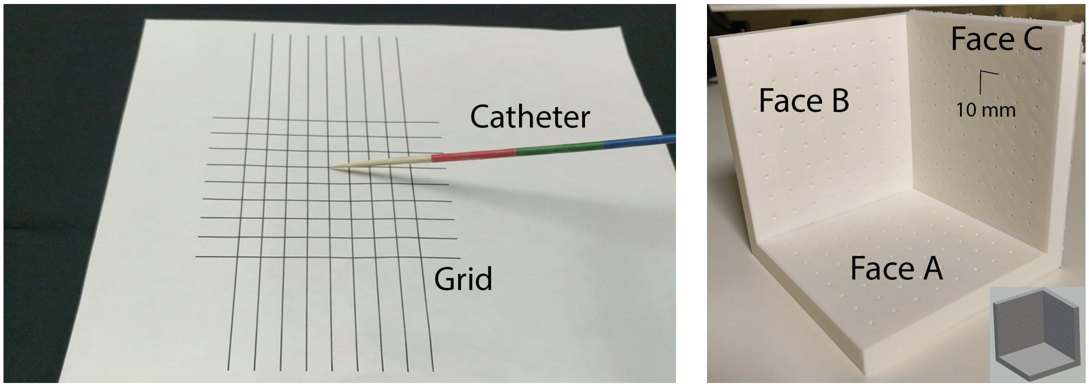
We measure the stability as the root mean square of the change of calculated tip in camera space in two consecutive frames as in Niehorster *et al.* [112] while keeping the catheter still. Given n frames, and the tip of the catheter in frame i as $\boldsymbol{x_i}$, the stability is measured as

$$S_{RMS} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1}||\boldsymbol{x_i} - \boldsymbol{x_{i+1}}||^2} \tag{3.13}$$

Several factors could influence the stability of the tracking algorithm, including lighting conditions, thresholds for color segmentation, and distance from the camera to the catheter. In a typical lab setting, our algorithm achieves a stability of 0.33 mm as measured over 870 frames.

3.5.2 Accuracy over a 2D Grid

To test the accuracy of the tracking algorithm, we move and point the tip of the catheter at the intersections of a grid. The grid, 8.16 cm × 8.16 cm in size, is printed on a white sheet. This provides enough space to test the accuracy of



(a)

(b)

Figure 3.9: Setup for testing (a) the tracking accuracy over a 2D grid, and (b) tracking accuracy over a 3D grid with the 3D CAD model in the lower right

catheter movement, more than the 6.5 cm suggested by Fried *et al.* [78]. The setup for this test can be seen in Figure 3.9 (a).

We assume the center intersection of the grid to be a reference point. Before the test, a 2D marker is placed on the workbench, and its up and right directions (parallel to the edges of the paper sheet the marker is printed on) in the camera space are detected. The sheet with the grid is then placed on top of the marker with the edges aligned. Therefore, the grid has the same up and right direction as the marker, and with its size known, we know the position of each grid intersection relative to the reference point. When moving along the grid, the catheter's orientation is kept relatively fixed and its tip position in the camera space is recorded. Then the tip positions relative to that when the tip is pointed at the reference point is compared to positions of the grid intersections relative to the reference point.

Figure 3.10 shows the catheter tip locations on the grid with the reference point as the origin. Blue circles indicate the grid intersections, and the orange

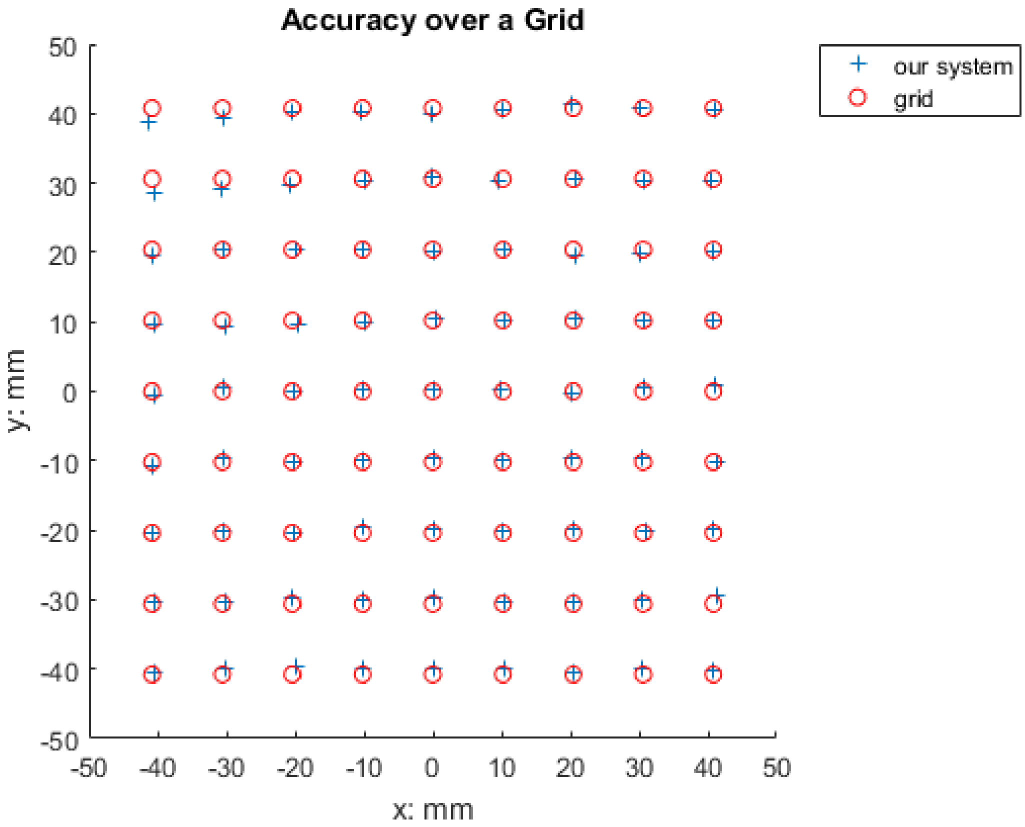


Figure 3.10: The accuracy of our catheter tracking approach over the grid. The red circles indicate the grid intersection. The blue crosses indicate the computed catheter tip position.

crosses are the detected catheter tips. The average distance from the catheter tip to the corresponding grid intersections is 0.58 mm.

3.5.3 Accuracy over a 3D Grid

We also demonstrate the accuracy of our algorithm when the catheter is moving in vertical space. To construct a reliable ground truth, we use a 3D-printed structure where square grids of blind holes are on its three faces (see Figure 3.9).

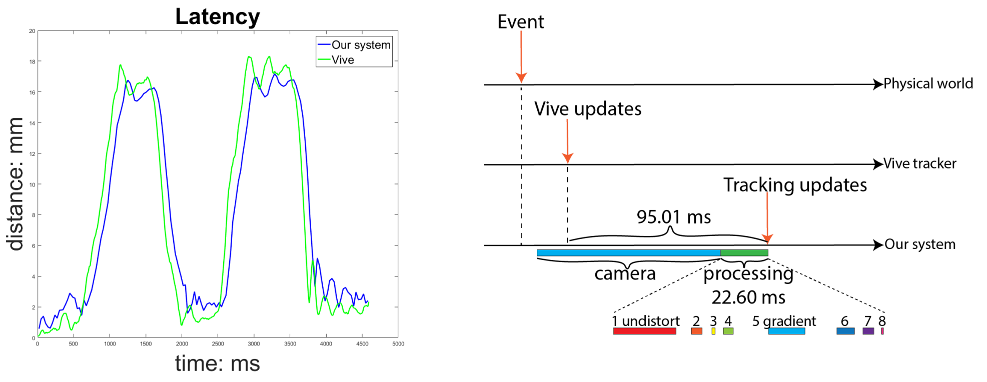
The centers of the blind holes are $d = 10$ mm apart in both directions which we manually verified with a caliper. Each blind hole is designed to be a cone with 1 mm radius and 3 mm height. On each face, there are 8×8 blind holes.

When testing on each face, one of the center blind holes (for example (m, n) on the 8×8 grid) is chosen as the reference point. The location of the tip when the needle tip is inserted into each blind hole (i, j) is recorded as $T(i, j)$. We construct the ground truth $G(i, j)$ as the designed distance of each blind hole to the reference point $G(i, j) = \sqrt{(m-i)^2 + (n-j)^2}d$. The distance between the calculated tip and the reference at each blind hole is $D(i, j) = ||T(i, j) - T(m, n)||_2$. The average accuracy on the face f is then $\text{Acc}_f = \frac{1}{64} \sum_{i,j} \text{abs}(G(i, j) - D(i, j))$.

In this test, $\text{Acc}_A = 0.49$ mm, $\text{Acc}_B = 0.93$ mm, $\text{Acc}_C = 1.12$ mm. The overall average accuracy on all three faces is 0.85 mm.

3.5.4 Latency

We measure our system's latency compared to an HTC Vive tracker. This latency is measured as the time elapsed between the tracker update and our system catheter's location update. In our test, we rapidly move the catheter, which is attached to the tracker, and record for each frame the distance from the catheter tip (tracked with our system and by the HTC Vive tracker) to their original location recorded at the beginning of the test. The distances are then plotted, and we manually measure the average time by which our system lags the tracker. Figure 3.11 (a) shows the plotted distances that we recorded. Manual measurements show an



(a) (b)

Figure 3.11: The latency of our system compared to an HTC Vive tracker (a) and the various components of our system's latency in image capture and processing (b).

average latency of 95.01 ms. A break down of the latency can be seen in Figure 3.11 (b).

There are two factors in the latency with respect to the HTC Vive tracker. It takes about 72 ms for our camera (together with the driver, and OpenCV functions) to capture and store the image. The processing takes an additional 22.60 ms on our machine, which uses an Intel i7 7820HK processor running at 2.9GHz. As we show in Table 3.1, undistorting takes the longest time, 10.05 ms per frame (44.42% of the total processing time). Speed up may be achieved by performing color segmentation first and undistorting only the pixels of the color bands. The next most time-consuming task is calculating the gradient along the catheter axis, (6.81 ms per frame, 30.12% of total processing time). Our current implementation is on the CPU. A GPU is likely to reduce the processing time through parallelization. A professional-grade camera with a low image capture time can greatly reduce the latency.

ID	Function	Avg time per frame (ms)	Percentage in processing (%)
1	Undistort	10.04	44.42
2	Color segmentation	1.05	4.64
3	Erosion	0.23	1.02
4	Get axis	0.93	4.13
5	Gradient	6.81	30.12
6	Connected components	1.15	5.08
7	Weighted centroids	0.92	4.09
8	Calculate catheter position	0.21	0.92
	Other	1.26	5.58
	Total	22.60	100.00

Table 3.1: Time breakdown of functions in our tracking algorithm.

3.6 Discussion

We have demonstrated a novel method of tracking a catheter tip and projecting it onto an augmented reality (AR) display. In combination with CT scan projection into the skull, this could allow for a safer, better, and more accurate way to place an external ventricular drainage into the brain of a sick patient. The first challenge we address is significantly enhancing the accuracy and latency of our system over the state-of-the-art augmented reality systems for this procedure such as Cutler *et al.* [69].

External ventricular drainage is a technically difficult procedure. It involves placing a catheter several centimeters into the brain through tissues of varying density including bone, dura, brain, and cerebral spinal fluid. In addition, it is generally performed in bright light with blue sterile drapes. These realities can potentially affect the accuracy of the AR tip projection in difficult-to-predict ways and needed to be validated in our study. Furthermore, the AR system should not interfere with procedure performance by not hindering the surgeon placing the drain. A purely computational evaluation of our approach would not have been sufficient to assess the efficacy of our augmented-reality approach in such challenging environments.

In order to validate the true feasibility of AR-assisted EVD placement, we have carried out our evaluation in a realistic experimental environment, which included a cadaveric head, using sterile drapes and the existing EVD placement kits, closely mimicking actual EVD placement in a patient. After inserting the catheter tip at the insertion point, the surgeons advanced the catheter in a straight line. During the process, the displayed virtual catheter projection aligned well with the visible portion of the real catheter and the pace at which the real and the virtual catheter moved matched. The surgeons have reported that the handling of our slightly modified catheter feels identical to a traditional catheter. The surgeons, while aware of the HMD, did not find it cumbersome.

Tests show that our system is accurate with sub-millimeter accuracy and low latency. If further improvements in accuracy and latency were needed, we could do so with the following enhancements. First, a professional-grade camera with a higher resolution, higher dynamic range, and lower latency could improve tracking by enhancing our ability to discern the segment endpoints and reduce the overall latency. With a higher resolution, the camera could be farther away from the catheter without compromising tracking quality, thus allowing a larger tracking area. All this comes with a higher computational burden. At the moment, our algorithm is implemented on a CPU but is highly parallelizable. Implementing a parallel GPU version of the algorithm is a high priority in our future work. At present, we observe around $167ms$ end-to-end latency between moving the real catheter and its updating the virtual catheter in the HMD. In our current system, we use an independent camera and a laptop for processing, but it would be desirable to explore an all-in-one setup. This is highly likely with further advances in commodity HMDs. Current-generation AR HMDs often suffer from drift in spatial tracking. In our studies, this has ranged from $2.15mm$ to $6.35mm$, depending on the amount of user head movement. Complementing the 3D spatial tracking with 2D optical tracking like in Frantz *et al.* [113] should be able to improve spatial tracking accuracy.

In our system, the calibration between the camera space and the HMD space is performed manually to make sure the observed virtual catheter aligns with the observed physical virtual catheter. Therefore very little optical misalignment has been observed in our experiment as long as the user does not remove or adjust how the HMD is worn. The integration of eye-tracking technology in most of the next-generation HMDs should further improve optical alignment.

Most catheters used in EVD (such as Integra®, Medtronic®, and Codman®) measure longer than $30cm$ and there is sufficient space to include color bands, as outlined in our approach, on clinical catheters. To further mitigate occlusion-related

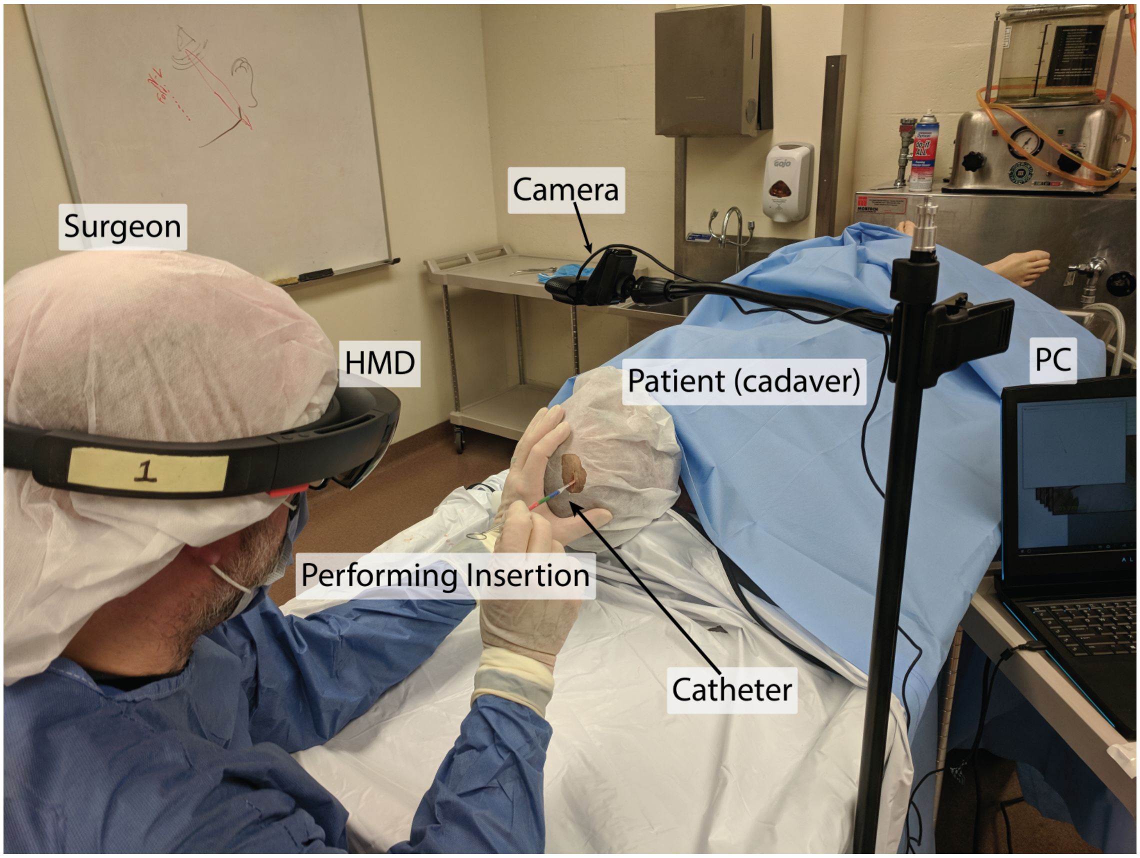


Figure 3.12: Our system being used in a cadaver study

concerns around color bands, one could explore adding additional redundant color bands. Last but not least, attention needs to be paid to the clinical safety of catheter modification. Technologies such as VisiMark®[7] could be used to create medical-grade markings to serve as the color bands. Alternatively, the catheter could be constructed with medical-grade silicone of various colors and go through standard sterilization. Slight changes would need to be made to the color band detection process.

[7] http://www.surfacesolutionsgroup.com/coatings/visi-mark/

3.7 Conclusion

We present an augmented reality surgical visualization and tracking system that can assist surgeons in EVD catheter placement. Compared to existing similarly-purposed systems, our system does not require changing the shape or weight of the medical instruments. Our main technical contributions include 1) a low-latency, high-performance way to track catheters and other 5DOF thin cylindrical objects, and 2) an image processing algorithm to extract tracking color segment endpoints in an image. Tests show that our system achieves a 0.58 mm accuracy on a 2D plane and an accuracy of 0.85 mm in 3D space. Processing for each frame takes 22.60 ms on a moderately powerful computer. Both the accuracy and speed can be further improved with higher-quality cameras and parallel implementation. We have tested our system in a realistic surgical environment with a cadaveric head. Our color marker and tracking technique can be applied to other medical procedures where the precise location of a very thin medical instrument is needed, such as needle-guided biopsies and minimally invasive surgeries.

Chapter 4: Correcting the Proximity Effect in Nanophotonic Phased Arrays

4.1 Overview

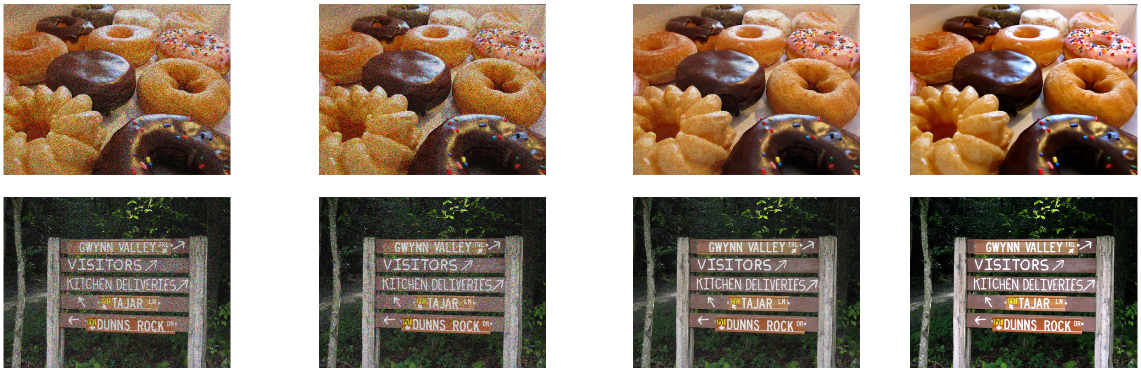
Nanophotonic phased arrays (NPA) are phased arrays operating at optical wavelengths. An NPA is composed of an optical power distribution system, a phase modulation mechanism that can control the phase of each pixel individually, and antennas for propagating phase-modulated light into free space. NPAs can efficiently steer optical waves in the desired direction and do so efficiently without any moving parts. Thus, they have found widespread applications in optical communications (such as Khalighi and Uysal [114]), range-finding (such as Poulton *et al.* [115]), and other fields. Recently, Sun *et al.* [18] and Notaros *et al.* [116] demonstrated the use of the NPA as an augmented reality holographic display. In this application, the NPA generates an optical wavefront. Through careful phase modulation, text and images (or 3D scenes) can be displayed for the observer to see. This use of NPAs is of particular interest for graphics research into near-eye displays (NEDs).

Near-eye displays have become more popular with virtual and augmented reality (VR and AR) going mainstream. In the current VR and AR headsets, users' eyes can only focus at one or a small number of depths. This vergence-accommodation conflict (VAC) can lead to visual discomfort, nausea, and cybersickness. Focus-tunable displays have been shown to provide a preferable viewing experience [117]. Varifocal displays and light field displays have been proposed to address this issue but they are either limited in angular resolution or involve multiple moving components. Holographic displays recreate the optical wave field of 2D images or 3D scenes. Therefore, the light observed by the user looks as if it is coming from the real image or object, allowing viewing from different perspectives. Holographic displays also form images and 3D scenes at the desired depths for eyes to naturally focus on, thereby addressing VAC.

A holographic display modulates the phase and/or amplitude of light at each pixel location to form the desired wavefront to be observed. Previously, Liquid Crystal on Silicon Spatial Light Modulators (LCoS SLMs) are the most common phase modulators for holographic displays. However, NPAs have gained popularity recently. One major advantage of the NPAs over SLMs is their fast refresh rate. One refresh cycle on the NPA includes heating each pixel to a certain temperature corresponding to the desired phase and cooling down to ambient temperature. The simulation with our NPA design shows that it can complete a refresh cycle within 10 µs, suggesting a 100kHz refresh rate which is uniquely suitable for dynamic content as well as time-division multiplexing operations such as color-switching and higher dynamic range. The only other microdisplay capable of kHz-refresh rates is the Digital Micromirror Device (DMD). It has been used for creating near-eye displays in Rathinavel *et al.* [118]. However, the DMD can only modulate binary amplitude patterns which limits the fidelity of its image in a single frame. The NPA is capable of continuous phase modulation. In addition, the NPAs support integrated light sources where a laser is coupled to the NPA via an optical fiber, distributed to each pixel and emitted by the on-board antennas. In this way, the NPA displays can be made more compact as optical operations such as beam expansion and polarization necessary for typical LCoS SLM or DMD display systems are not required. LCoS SLMs also suffer from pixel voltage crosstalk and non-linearity mapping between phase and voltage which hinder accurate phase modulation.

Thermally-modulated NPAs, such as those in Doylend *et al.* [19], Sun *et al.* [18] and our design, depend on the thermo-optical effect of the materials to modulate the phase of a pixel. The phase of a pixel can be changed from 0 to 2π by changing the temperature of the pixel. The precise and independent control of the phase is needed to ensure the successful display of desired 2D or 3D imagery. However, the thermal proximity effect (PE), the phenomenon where the temperature of one pixel affects the temperature of neighboring pixels, impacts the accuracy of phase modulation and degrades the observed image. A simulated example of the proximity effect negatively impacting the formed image of some Fourier holograms can be seen in Figure 4.1. Therefore, in order to form the desirable images, this thermal proximity effect must be taken into consideration. This problem is further complicated by the fact that most NPAs only have heat generation mechanisms, with no cooling capability, making some phase patterns impossible to realize.

There has been limited study on the problem of thermal proximity effect on NPA holographic displays. But the proximity effect is present in many processes



(a) Without PEC (b) Basic Iterative PEC (c) Iterative PEC (d) Proximal PEC

Gemayel *et al.* [119]

Figure 4.1: Nanophotonic phased arrays suffer from thermal proximity effect where one pixel being heated affects the temperature of nearby pixels. This causes inaccurate phase modulation and noise in the formed image of Fourier holograms as shown in (a). The current state-of-the-art methods, as shown in (b), are not able to sufficiently correct the proximity effect for our use case. We propose two proximity effect correction (PEC) methods for Fourier holograms on the NPA which are able to reduce the noise, as shown in (c) and (d). The proximal PEC method has the best correction effectiveness. Images are chosen from the Common Objects in Context dataset [5]. The proximity effect correction for Fresnel holograms is discussed in Section 4.5.

and devices. In beam electron lithography and optical communication using SLMs, the proximity effect is modeled as a convolution, such as in Peckerar *et al.* [120] and Persson *et al.* [121]. In the former case, the correction is conducted as a deconvolution with non-negative constraints on the solution. In the latter case, the Gerchberg-Saxton phase retrieval algorithm is slightly modified to produce a result that is somewhat resistant to the proximity effect. Sun *et al.* [122] also consider the thermal proximity effect on NPAs as a convolution and perform deconvolution on the phase under the assumption of perfect amplitude modulation.

In this chapter, we consider the practical but more challenging case of phase-only holograms on NPAs. Based on the thermo-electric simulations, we also model the proximity effect as a convolution. We first propose two new proximity effect correction methods for the Fourier holograms. The first method works by integrating deconvolution into the phase retrieval algorithm. The second method uses the proximal algorithm to directly optimize the displayed image on the NPA considering the proximity effect. In addition, we propose a proximity effect correction method for Fresnel holograms also based on the proximal algorithm. Fourier holograms are based on the Fraunhofer diffraction approximation where propagating a wavefront to a plane very far away (far field) can be approximated with a Fourier transform. Fourier holograms are fast to compute and are good for reconstructing 2D images in the far field. However, since Fourier holograms do not specify the depths of the formed images, they are not suitable for 3D scenes. Fresnel holograms, on the other hand, are based on the more accurate Fresnel diffraction approximation which is applicable closer to the source than the far field and can specify the depths of the formed imagery. An illustration of the diffraction approximations and hologram types can be seen in Figure 4.2. For more detail, please refer to Goodman [123] and Voelz [6]. The closer range is more suitable for near-eye displays and the ability to specify the depth allows the users to focus on the correct focal distance. This is where holographic displays and computer-generated holography truly shine. Various methods have been proposed to calculate the Fresnel holograms of 3D objects and display them in SLM-based holographic displays, including by Shi *et al.* [1] and Maimone*et al.* [62]. No proximity effect correction method has been proposed for Fresnel holograms on the NPA.

We start by investigating how the proximity effect negatively affects the observed image quality on a phase-only NPA holographic display. We implement the new proximity effect correction methods mentioned above and compare their correction effectiveness with existing solutions. Computational simulations are run with proximity effect of varying severity and both qualitative and quantitative results are presented. The factors affecting the correction effectiveness and processing time of the methods are also investigated.

In summary, the goal of this chapter is to explore the proximity effect on NPA holographic displays and propose new methods of correction. To the best of our knowledge, this is the first detailed analysis of phase-only NPA-based holographic displays with the thermal proximity effect and the first to come up with effective solutions.

The rest of the chapter is organized as follows. In Section 4.2 we briefly summarize the existing literature on several related topics including near-eye displays,

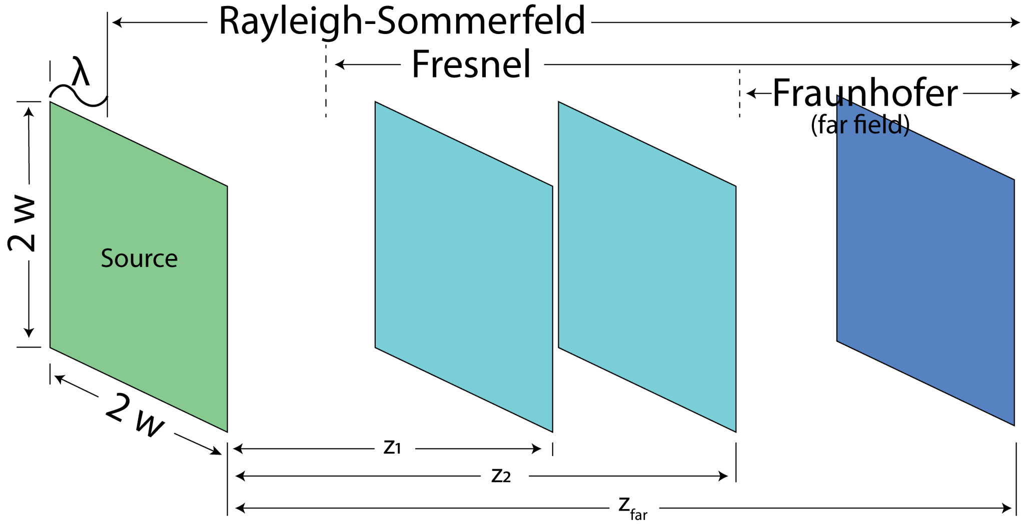


Figure 4.2: The Rayleigh-Sommerfeld model provides a highly accurate, but also computationally expensive, diffraction model. It is applicable one wavelength away from the source. When the propagation distance is farther, the Fresnel diffraction approximation, which is less computationally expensive than Rayleigh-Sommerfeld diffraction may be used. Holograms computed using the Fresnel diffraction are called *Fresnel holograms*. They can be reconstructed at the user-specified depths, such as z_1 and z_2. The Fresnel diffraction approximation is considered acceptable when $N_F = \frac{w^2}{\lambda z} < 30$ ([6]), where w is the radius of the source field, λ is the wavelength and z is the propagation distance. Farther away when $N_F \ll 1$, the Fraunhofer diffraction approximation can be used where the propagation can be approximated with a Fourier transform. Holograms based on the Fraunhofer diffraction are called *Fourier holograms* and are the fastest to compute. In our NPA design, $w = 1\,\text{mm}$. Given a typical green light wavelength $\lambda = 530\,\text{nm}$, the Fresnel diffraction is acceptable for $z > 62.89\,\text{mm}$ and Fraunhofer diffraction is suitable for $z \gg 1886.79\,\text{mm}$.

NPAs, computer-generated holography, and proximity correction. In Section 4.3, we describe how we model the proximity effect on an NPA display. Then in Section 4.4 and 4.5, we describe our proposed proximity effect correction methods for the Fourier holograms and Fresnel holograms, respectively.

4.2 Related Work

We introduce some of the existing work on the various topics related to this chapter. This includes work on NPAs, NEDs, computer-generated holography and the proximity effect correction.

4.2.1 Nanophotonic Phased Arrays

The phased array has been an active area of research since its invention. The microwave version of the phased array was first realized in the mid 1940s [124] and since then it has seen widespread use in applications such as radio detection and ranging (RADAR), broadcasting, and astronomy. The invention of the optical version of the phased arrays, the nanophotonic phased arrays, was made possible due to several advances starting from the invention of the laser in the early 1960s and high-precision silicon photonics fabrication.

Optical beam steering using one-dimensional NPAs allows stable, rapid, and precise beam steering without mechanical motion while providing large scanning angles. Two-dimensional NPAs have several applications, including LiDAR scanning and free-space optical communications. What is of particular interest to us is its use as a holographic display. Sun *et al.* [18] demonstrate a novel NPA architecture capable of integrating an optical directional coupler, a 2π tunable phase modulator and a nanoantenna within a compact $9\,\mu m \times 9\,\mu m$ unit and an infrared image containing text information which can be observed in the far field. Building an NPA on a silicon nitride platform can break its operating wavelength limitation and expand its applications into the visible spectrum. Raval *et al.* [125] show a static image with the horizontal parallax and a viewing angle of 5° generated on an NPA operating at 635 nm wavelength. Notaros *et al.* [116] demonstrate for the first time an NPA-based visible-light near-eye holographic projector where a holographic image encoding methodology has been used and a 3D cube can be observed by the eye. However, in most of the works mentioned above, the phase shift for each pixel has been hard-coded during manufacturing and cannot be dynamically changed. As a result, only static images were shown. A notable exception is the work by Sun *et al.* [18] where a small-scale (8×8) array with active phase tuning is used to demonstrate infrared beam steering. Our NPA design supports active phase tuning for visible light and addresses the challenge of correcting thermal proximity effect that comes with thermally modulated NPAs. A phased array with acousto-optic phase modulation has also been shown by Grinenko *et al.* [126].

4.2.2 Near-eye Displays

Traditionally, near-eye displays use binocular vision to provide the parallax depth cue [127, 128]. While these displays and the modern commercial headsets can provide a very good field-of-view, they can only afford a single focal distance which leads to the vergence-accommodation mismatch. A comprehensive study by Rolland *et al.* [129] provides an engineering specification on the number of focal distances that is required to suppress the vergence-accommodation mismatch. To provide multiple focal distances, Akeley *et al.* [130] use several beamsplitters placed at different distances from the viewer to reflect the content from a single LCD monitor. Dunn *et al.* [55] use a deformable membrane controlled by airtight cavities to modulate the focal distance. A gaze tracker is used to sense the desired focal distance. Commercialized focus-tunable lenses have also been used address VAC and vision correction in near-eye displays in Chakravarthula *et al.* [131] and Xia *et al.* [132]. Akşit *et al.* [133] and Kim *et al.* [134] use a motorized linear stage to move the optical combiner and the MOLED display on the optical axis to adjust the focal distance.

Light field displays encode the radiance as a function of position and angle [135] and have been shown to approximate the accommodation to the extent limited by the angular resolution [136]. Light field displays have been implemented based on a variety of underlying hardware platforms such as microlens arrays [137], attenuating LCDs [63, 138–140], and holographic optical elements combiner layer with projectors [141, 142]. Pamplona *et al.* [143] and Huang *et al.* [144] show that light field displays can correct impaired vision such as myopia and presbyopia and higher-order aberrations. A method to compute the optimal imagery on such displays with multiple focal planes has been described [145]. However, the angular resolution of light field displays is limited by the number of different views which together with the spatial resolution is bound by the device's form factor and pixel density. Attempting to further increase pixel density can lead to undesirable diffraction artifacts. Padmanaban *et al.* [146] solve this problem to some degree with Overlap-Add Stereogram (OLAS) at the cost of more intensive computation. Mi *et al.* [147,148] have shown that the accommodation problem can be circumvented with a type of retinal scanning display that directly projects the image onto the user's retina.

Holographic displays have been built with pre-recorded holographic optical elements [133,134] or with dynamic phase spatial light modulators [1,62] or a combination of both [149]. This technology has the potential to fully address some hard problems faced by the current generation of VR and AR near-eye displays such as vergence-accommodation conflict. However, the refresh rate and form factor of the holographic displays falls short of what the NPA technology can provide.

For more details on near-eye displays we refer the interested reader to the comprehensive survey by Koulieris *et al.* [150].

4.2.3 Computer-Generated Holography and Phase-Only Holograms

Various methods have been proposed for computer-generated holography (CGH). For Fourier holograms, the holographic pattern of an image in the far field can be calculated as its Fourier transform. Lohmann *et al.* [151] show that a binary representation of the Fourier hologram can achieve good reconstruction quality for letters in a 2D image.

Fresnel holograms are calculated by considering the various scene objects, such as points and polygons, as a collection of light sources. Waters [152] synthesized a Gabor-style hologram by calculating the interference pattern from the light emitted by each point on an object and the illumination beam. Directly computing the interference pattern from every point is computationally intensive. Lucente [153] uses a look-up table corresponding to the interference contribution from points at each location in the image space and achieves a 43× speedup. Yoshikawa *et al.* [154] use approximation methods to calculate the distance from a point source to a holographic pixel. Stereograms are small regions of a hologram. They each record the diffraction pattern of a portion of the scene instead of the entire scene and are used in Lucente and Galyean [155], Xu *et al.* [156], and Shi *et al.* [1] to reduce the amount of computation. More recently, Eybposh *et al.* [157] train an unsupervised convolutional neural network that outputs the near-field phase for a given desired image in real-time. Peng *et al.* [158] employ a camera in the hologram calculation process to iteratively optimize the hologram based on captured observation from the camera. These holograms are used to train a neural network that can output high-quality holograms in real-time.

Ziegler *et al.* [159] develop a conversion technique between the light field representation and the holographic representation of a scene. Padmanaban *et al.* [146] use this conversion to develop the OLAS algorithm to escape the trade-off between spatial and angular resolution in traditional light-field displays. Ways to add occlusion or change lighting in a hologram have also been explored in Ziegler *et al.* [160].

While an optical wavefront contains both phase and amplitude, many currently available holographic displays are only capable of modulating either only the amplitude or only the phase. Akşit [161] use a learning-based optimization approach to calculate the amplitude hologram that enhances the spatial resolution by up to four times when illuminated with a fast-refreshing locally-addressable backlight. The NPAs we are investigating here is only capable of phase modulation. Therefore, a method to convert fully complex holograms to phase-only holograms is required. Gerchberg and Saxton [162] proposed the first of a family of iterative phase retrieval algorithms that generate a near-field phase pattern which coupled with a predefined near-field amplitude pattern can propagate to form the desired far-field amplitude. We discuss this algorithm and our modifications in Section 4.4.1. The original Gerchberg-Saxton (GS) algorithm uses Fourier transform to approximate the Fraunhofer diffraction over a long distance. Fresnel diffraction is also frequently used in a similar way when the propagation distances are shorter.

Several researchers have made modifications to the iterative phase-retrieval algorithm. Masuda *et al.* [163] added a binarization step after the propagation from the object plane to the hologram plane to extract a binary phase. Similarly, Persson *et al.* [121] and Gemayel *et al.* [119] add a convolution step to counter the pixel voltage crosstalk on SLMs. However, these methods require heavy padding of the image to be effective and in our opinion do not make the full of the limited resolution on holographic displays. Nevertheless, we include these methods as the most recent existing proximity effect correction method even though the cause of proximity effect in our case is different.

A recent work on phase retrieval is Chakravarthula *et al.* [164] using non-convex optimization based on Wirtinger derivative. Our proposed proximal proximity effect correction method also uses non-convex optimization and shows that it can effectively and efficiently correct thermal proximity effect in NPAs.

4.2.4 Proximity Effect Correction

Parikh [165] introduces a self-consistent technique that provides a mathematically unique solution to the PEC problem given the proximity function by solving a set of linear equations. This technique is accompanied by a second technique that compensates for the unaddressed regions by solving an overdetermined system of equations. Carroll [166] expresses the PEC problem as a linear programming problem with constraints derived from the addressed and unaddressed pixels.

Chow *et al.* [167] use an image deconvolution approach based on the convolution theorem to efficiently solve the PEC problem. Similarly, a matrix inversion approach to deconvolution is tested by Peckerar *et al.* [168]. These deconvolution-based methods are likely to produce negative entries in the result, which is not physically acceptable. Simple techniques have been proposed to address this problem such as setting them to zero (chopping) and adding a background value equal to the absolute of the minimum entry (shifting) in Marrian *et al.* [169]. A more advanced method to avoid negative entries is to add a regularizer to the cost function in the gradient-descent method. One such regularizer used in [168, 169] is proportional to the Shannon entropy of the pattern. Alternatively, Peckerar *et al.* [170] express the non-negative requirement as well as the incident dose as constraints in a linear programming problem that minimizes the total dose. Sun *et al.* [122]

try a number of PEC methods including quadratic programming on an NPA model that suffers from proximity effect in phase modulation but assumes independent and ideal amplitude modulation. Our work addresses the more practical, but also more challenging case of thermal proximity effect correction on a phase-only NPA.

There are many more PEC methods in E-beam lithography that we are not able to cover in the entirety here. We refer interested readers to Li [171] for a thorough review.

4.3 Nanophotonic Phased Array

This section describes the basic structure and operation of an NPA display. The description is based on an NPA of our design currently under development. The differences between this NPA design and previous NPAs such as those in Sun *et al.* [18] and Raval *et al.* [125] are out of the scope of this chapter. The focus of this chapter is the thermal proximity effect which is common in all thermal-modulated NPAs and its correction methods. Therefore, we first explain the phase modulation in Section 4.3.1, followed by how we model and quantify the thermal proximity effect in Section 4.3.2.

4.3.1 Phase Modulation

Figure 4.3(a) shows the schematics of the NPA we have developed. Each array unit (a pixel) consists of a tunable thermo-optical phase shifter that is coupled to an optical antenna. The refractive index of the phase shifter changes with temperature.

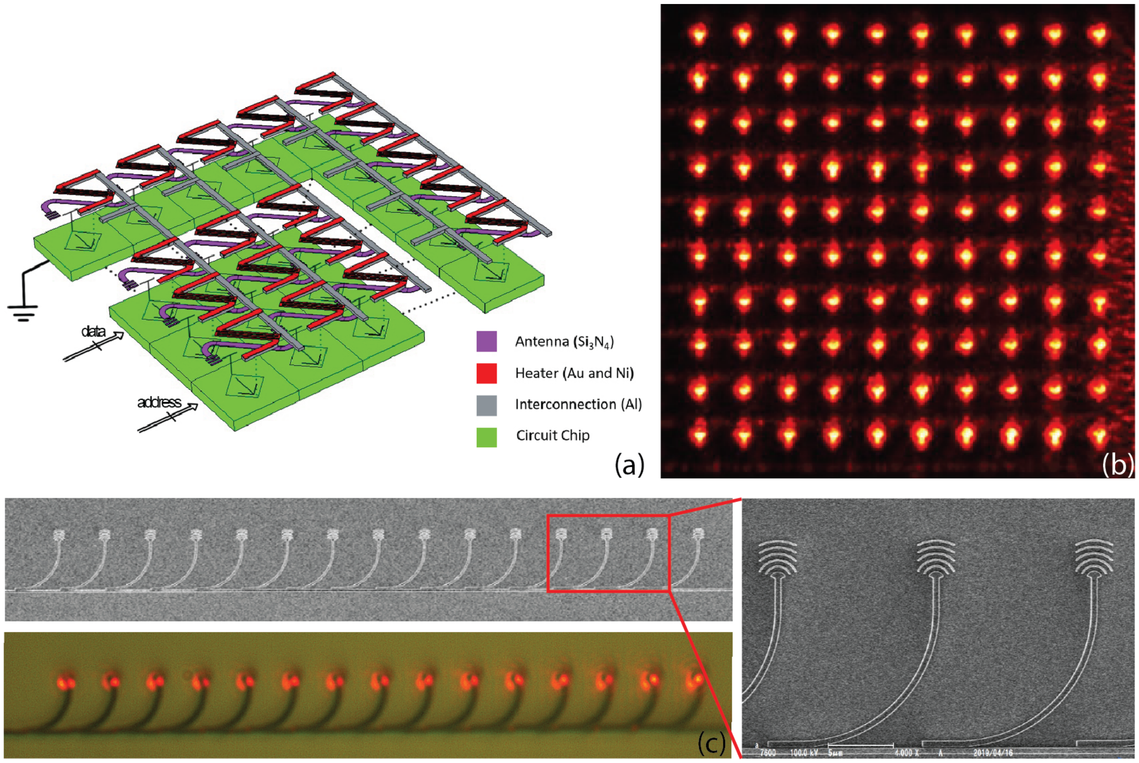


Figure 4.3: The NPA display with electronic and optical layers is shown in (a). The optical power is evenly distributed into each pixel. Therefore, the near-field (b) has uniform amplitude. The antennas (c) propagate the near-field wavefront to form the image to be observed at the desired depth.

Therefore, phase modulation can be achieved by heating each pixel. Special efforts have been put into reducing the physical size of the phase shifter while maintaining its sensitivity to temperature change. This way, the array can be small and a complete 2π phase shift can be achieved with a relatively small temperature range, thus achieving high energy efficiency. The optical power is evenly distributed into each pixel by a precise design of the directional couplers. Therefore, a uniform amplitude is achieved in the near field, as shown in Figure 4.3(b). The phase of the NPA holographic display pixel is modulated with the temperature. An electronic chip (green layer in Figure 4.3(a)) is used to control the heating of each pixel.

There is a linear relationship between the phase ϕ_j of a pixel j and its temperature T_j (above room temperature), expressed in Equation 4.1.

$$\phi_j = \gamma T_j \tag{4.1}$$

Note that T_j must be non-negative for physical feasibility. From the simulations run on our design, γ is inferred to be $\frac{\pi}{175}$ rad °C^{-1} for 630 nm red laser. With precise phase shifter design, γ is proportional to the wavelength, i.e. $\gamma = \frac{\pi}{147}$ rad °C^{-1} for green (530 nm) and $\gamma = \frac{\pi}{124}$ rad °C^{-1} for blue (445 nm) lasers. Given the desired 2D phase pattern of a hologram, we can find the desired 2D temperature profile $\tilde{T}$ with Equation 4.1.

4.3.2 Thermal Proximity Effect

The thermal proximity effect in the NPAs is the phenomenon where one pixel affects the neighboring pixel's temperature when it is being heated. Similar proximity effects are present in beam electron lithography and LCoS SLMs. In Peckerar *et al.* [120] and Persson *et al.* [121], the relation between the energy deposited and energy absorbed and the pixel voltage crosstalk are both modeled as convolutions. We also model the thermal proximity effect in the NPAs as a convolution. To quantify the proximity effect, we simulated a 5×5 array where we supplied the center pixel with power and measured the temperature on all pixels. An illustration of this thermal proximity effect can be seen in Figure 4.4. The data for this figure is obtained from fitting a Gaussian to the 25 values from the simulation on a 5×5

array which is described below.

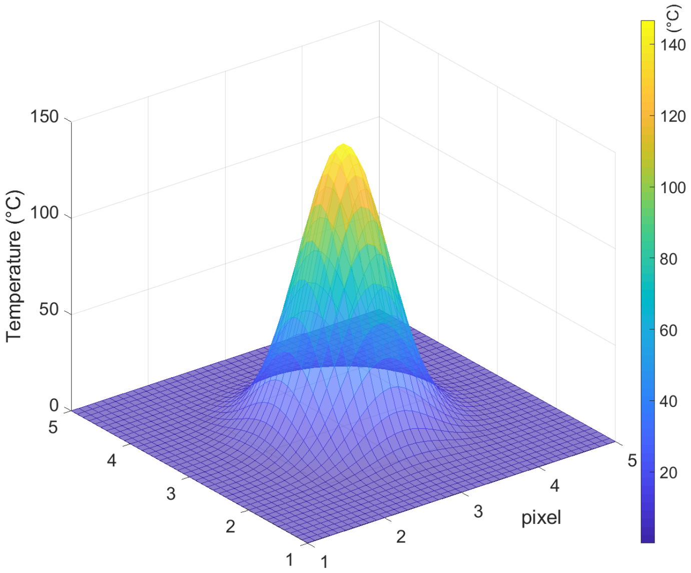


Figure 4.4: Thermal crosstalk arising from power being supplied to the center pixel on a 5 × 5 array.

Using the measurement, we derive the following relationship:

$$T_{k\to j} = T_k e^{-\frac{d(j,k)^2}{\sigma^2}} \tag{4.2}$$

$T_{k\to j}$ refers to the temperature rise in pixel j caused by pixel k, T_j and T_k are the temperatures of pixels j and k before considering the proximity effect and $d(j,k)$ is the distance between pixels j and k measured in pixels (px). The temperature $\hat{T}_j$ of pixel j considering the proximity effect is the sum of T_j and the contributions from all surrounding pixels, expressed in Equation 4.3. The temperature of the entire NPA can be expressed as the convolution in Equation 4.4 where T and $\hat{T}$ are the 2D temperature profiles before and after considering the proximity effect and $*$ is the convolution operation.

$$\hat{T}_j = T_j + \sum_{k \neq j} T_{k \rightarrow j} \tag{4.3}$$

$$\hat{T} = T * K_\sigma \tag{4.4}$$

We derive that $\sigma = 0.66$ px based on the data obtained from thermal-electrical simulations. The convolution kernel K_σ in Equation 4.4 can be constructed using σ. We have used kernels of size 7×7 in our simulations.

The severity of the thermal proximity effect is quantified by the value of σ. Even though we derived that in our current design $\sigma = 0.66$ px, factors such as the thermal conductivity of the materials of the NPA and the pixel pitch could affect this value. For example, if the pixel pitch is reduced, heating one pixel would affect the temperature of its neighboring pixels more as they are closer to the pixel being heated. The impacts of the thermal proximity effect and the effectiveness of the correction methods are presented with varying levels of the thermal proximity effect in Section 4.4.3.

4.4 Proximity Effect Correction for Fourier Holograms

In this section, we describe our novel proximity effect correction methods for Fourier holograms in phase-only NPA holographic displays. The first method is based on the Gerchberg-Saxton iterative phase retrieval algorithm. In this method, a deconvolution step is introduced to the source amplitude constraint in each iteration. We refer to this method as the Iterative Proximity Effect Correction (IPEC) method. The second method which we call the Proximal Proximity Effect Correction (PPEC) method, is based on the proximal algorithm. We use the proximal algorithm to directly find a near-field phase pattern that minimizes the difference between the far-field intensity it forms under the proximity effect and the desired image.

4.4.1 Iterative Proximity Effect Correction

The Gerchberg-Saxton iterative phase retrieval algorithm is a classic and well-known algorithm for calculating phase-only holograms. Here we focus on how we integrate proximity effect correction into it. For completeness, we present a detailed description of this algorithm in Section 4.4.1.1.

The pseudocode for the IPEC method is presented in Algorithm 1. The inputs to this algorithm are $\mathcal{S}, \mathcal{T} \in \mathbb{R}^{M \times N}$, the source and target amplitude patterns each of size $M \times N$. The source amplitude $\mathcal{S}$ is the near-field amplitude on the NPA. According to Section 4.3, $\mathcal{S}$ is uniform in our setting. The target amplitude $\mathcal{T}$ is the amplitude we wish to form in the far field. The phase retrieval algorithm finds the near-field phase that coupled with $\mathcal{S}$ produces the far-field amplitude $\mathcal{T}$. A random initial phase $\phi_0 \in \mathbb{R}^{M \times N}$ may be used to initialize the algorithm. The function "phs" returns the phase of a complex wavefront. We use the Fourier hologram model where the propagation of the near-field wavefront to the observation plane is approximated with a Fourier transform. $\mathcal{F}$ represents the discrete Fourier transform. Let $P(\phi)$ be the function that applies the proximity effect to the phase pattern ϕ. P^{-1} is the reverse process.

We integrate the proximity effect correction into the phase retrieval algorithm

Algorithm 1: Iterative Proximity Effect Correction

```
input  : $\mathcal{S}$, $\mathcal{T}$, $\phi_0$
output: $\phi_r$
$A = \mathcal{F}^{-1}(\mathcal{T} \cdot e^{i\phi_0})$;
while Within maximum iteration do
    $\phi_r = P^{-1}(\text{phs}(A))$;
    $B = \mathcal{S} \cdot e^{iP(\phi_r)}$ (source amplitude constraint);
    $C = \mathcal{F}(B)$;
    if Termination condition met then
        break;
    end
    $D = \mathcal{T} \cdot e^{i\text{phs}(C)}$ (target amplitude constraint);
    $A = \mathcal{F}^{-1}(D)$;
end
```

by modifying the step of the source amplitude constraint. Originally, the source amplitude constraint is applied by having B take the amplitude $\mathcal{S}$ and the phase of A. In our iterative proximity effect correction method, B takes the same amplitude $\mathcal{S}$ but the phase after taking the proximity effect and its correction into account. Specifically, the phase is $P(P^{-1}(\text{phs}(A)))$. P^{-1} takes into consideration the non-negative constraint. Therefore, in most cases, $P(P^{-1}(\phi)) \neq \phi$.

A good termination condition can be the difference, $\|\mathcal{T} - |C|\|_{\text{F}}$, between the amplitude of C and the target $\mathcal{T}$ becoming small enough. While this works well with the original phase retrieval algorithm, we find it hard to define a threshold when P^{-1} is introduced. Instead, we terminate when the minimum and maximum values of $|\mathcal{T} - |C||$ are not updated for several consecutive iterations.

In our setting, P is a convolution, i.e. $P(\phi) = K * \phi$ where ϕ is a 2D phase pattern. P^{-1} is therefore a deconvolution with non-negative constraints. We can express P^{-1} in the following form. $M_K \boldsymbol{x} = \boldsymbol{\phi}$ is the matrix form of the convolution. $M_K \in \mathbb{R}^{MN \times MN}$. $\boldsymbol{\phi}, \boldsymbol{x} \in \mathbb{R}^{MN \times 1}$.

$$\begin{aligned} P^{-1}(\boldsymbol{\phi}) &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \, \|M_K \boldsymbol{x} - \boldsymbol{\phi}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{x} \geq 0 \end{aligned} \tag{4.5}$$

A basic deconvolution method is matrix inversion. If M_K is non-singular, which it is in our case, we can simply take $\boldsymbol{x} = M_K^{-1} \boldsymbol{\phi}$. In practice, it is faster and more robust to solve the linear system $M_K \boldsymbol{x} = \boldsymbol{\phi}$. However, this method is likely to produce a solution that contains negative entries corresponding to a temperature lower than the room temperature. This is physically impossible to achieve on the current design of the NPA.

A more robust deconvolution method is quadratic programming, which is formulated in Equation 4.6. Traditional quadratic programming algorithms such as the interior-point algorithm can be used. Alternatively, we can use gradient descent to minimize the cost function in Equation 4.6 and use a regularizer to ensure the solution is non-negative.

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(M_K^{\mathrm{T}}M_K)\boldsymbol{x} + (-\phi^{\mathrm{T}}M_K)\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x} \geq 0 \end{aligned} \tag{4.6}$$

In the simulations, we use the quadratic programming deconvolution method described above.

4.4.1.1 Gerchberg-Saxton Iterative Phase Retrieval Algorithm

The iterative proximity effect correction method integrates proximity effect correction into the classic Gerchberg-Saxton iterative phase retrieval algorithm. For completeness, we describe the original GS algorithm here.

We have a target amplitude pattern $\mathcal{T}$ that we want to form in the far field and a fixed source amplitude pattern $\mathcal{S}$ in the near field. We wish to find the near-field phase that when coupled with $\mathcal{S}$ can be propagated to the far field to form $\mathcal{T}$. In the traditional iterative phase-retrieval algorithm, there are four complex wavefronts, A, B, C, and D (shown in Figure 4.5). A and B are located in the near field while C and D are located in the far field. In the initialization step, a common way to initialize A is to set it as the inverse Fourier transform of the target amplitude with the random initial phase ϕ_0.

Then in each iteration, B is formed by applying source amplitude constraint on the hologram plane, expressed as $B = \mathcal{S} \cdot e^{i\phi_A}$. B is then propagated to the object plane to form C. If the termination condition is met here, A's phase (same as B's phase) is the retrieved phase of the algorithm and C is the expected wavefront in

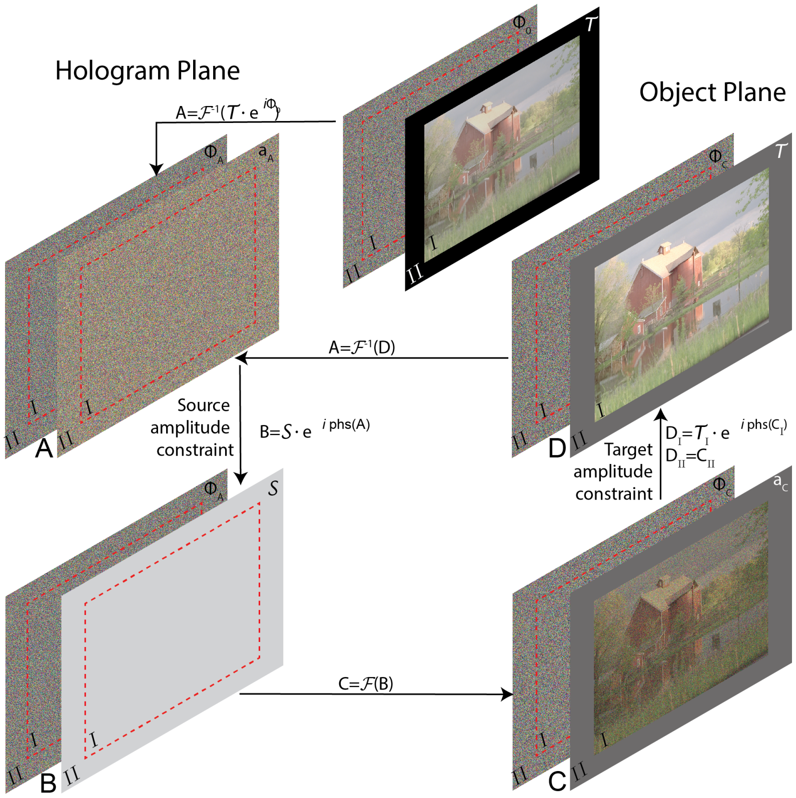

Hologram Plane
Object Plane
$A=\mathcal{F}^{-1}(\mathcal{T}\cdot e^{i\Phi_0})$
Φ_0
$\mathcal{T}$
Φ_A
a_A
$A=\mathcal{F}^{-1}(D)$
Φ_C
$\mathcal{T}$
Source amplitude constraint
$B=\mathcal{S}\cdot e^{i\,\mathrm{phs}(A)}$
$D_I=\mathcal{T}_I\cdot e^{i\,\mathrm{phs}(C_I)}$
$D_{II}=C_{II}$
Target amplitude constraint
A
D
Φ_A
$\mathcal{S}$
Φ_C
a_C
$C=\mathcal{F}(B)$
B
C
I
II

Figure 4.5: Iterative phase retrieval algorithm

the far field. Otherwise, the iteration continues as the target amplitude constraint is applied on the object plane, expressed as $D = \mathcal{T} \cdot e^{i\phi_C}$. A is then formed by propagating D to the hologram plane.

The phase retrieval algorithm may also specify a *signal* region. In this case, we only want to ensure that the farfield amplitude matches $\mathcal{T}$ only in the signal region. We specify the signal region as I, and the rest of the region as background II. We denote the signal and background region on a 2D field A as A_I and A_{II} respectively. $A_I \cup A_{II} = A$. The target amplitude constraint will be applied separately for the signal and background regions.

$$D_I = \mathcal{T}_I \cdot e^{i\mathrm{phs}(C_I)} \tag{4.7}$$

$$D_{II} = C_{II} \tag{4.8}$$

4.4.2 Proximal Proximity Effect Correction

In the proximal proximity effect correction method, we set out to find the near-field phase pattern that minimizes the difference between the far-field intensity it forms under the proximity effect and the desired image I. Namely, given the image $I \in \mathbb{R}^{M\times N}$, we want to find the non-negative near-field phase pattern $X \in \mathbb{R}^{+M\times N}$ that minimizes $g(X)$ in Equation 4.9. $U(X)$ is the intensity of the wavefront the NPA with input phase X forms in the far field, which can be calculated in Equation 4.10. Again, $P(X)$ is the result of the proximity effect on the input phase X. It should be noted that the proximity effect should be applied to the temperature instead of the phase. However in our case, since there is a relationship between the phase and temperature on the NPA, we directly use the phase X for simplicity in the problem formulation.

$$g(X) = \|U(X) - I\|_{\mathrm{F}}^2 \tag{4.9}$$

$$U(X) = |\mathcal{F}(e^{iP(X)})|^2 \tag{4.10}$$

Proximal algorithms are generally used for solving convex optimization problems [172]. However, as is shown in Heide *et al.* [173] and will be shown in Section 4.4.3, the proximal algorithms can also work well on our non-convex problems.

The proximal algorithms work by repeatedly evaluating the *proximal operator*. The proximal operator of a function f is defined as follows ([172]). λ is the step size parameter.

$$\operatorname{prox}_{\lambda f}(v) = \operatorname*{argmin}_{x}(f(x) + \frac{1}{2\lambda}\|x - v\|_2^2) \tag{4.11}$$

To enforce the non-negative constraint, we add an indicator function to the cost function $f(X)$ in Equation 4.12.

$$f(X) = g(X) + c(X) \tag{4.12}$$

$$c(X) = \mathcal{I}_{[0,+\inf)}(X) \tag{4.13}$$

Various proximal algorithms have been proposed. In this chapter, we use the accelerated proximal gradient algorithm. This algorithm splits the cost function into two parts, one of which is differentiable. In our case, $f(X)$ is the differentiable term. Let X^k and λ^k be the solution and the step size in iteration k. Y^k is the extrapolated solution in iteration k whose purpose is to move the solution a step further towards convergence. $g'_\tau(Z, Y) = g(Y) + \operatorname{tr}(\nabla g(Y)(Z - Y))^T + \frac{1}{2\tau}\|Z - Y\|_\mathrm{F}^2$ and $\beta \in (0, 1)$ are used in the line search method (Beck and Teboulle [174]) to determine the step size τ. The algorithm is carried out iteratively according to Equation 4.14.

$$\begin{aligned} Y^{k+1} &= X^k + \omega^k(X^k - X^{k-1}) \\ X^{k+1} &= \operatorname{prox}_{\lambda^k c}(Y^{k+1} - \lambda^k \nabla g(Y^{k+1})) \end{aligned} \tag{4.14}$$

A simple choice for ω^k is $\frac{k}{k+3}$ [172]. The proximal operator of the indicator function $\text{prox}_{\lambda c}(v)$ projects the input v into the non-negative space. The gradient of the function $g(X)$, $\nabla g(X)$, has been derived in Section 4.4.2.1. The function "Re" takes the real components of a complex value. $\overline{X}$ is the complex conjugate of X. K is the convolution kernel. The pseudocode for this algorithm is shown in Algorithm 2.

$$\nabla g(X) = 4K * \text{Re}\{[ie^{iP(X)} \cdot \mathcal{F}((U(X) - I) \cdot \overline{e^{iP(X)}})]\} \tag{4.15}$$

4.4.2.1 Derivation of PPEC Gradient

Here we document the derivation of $\nabla g(X)$. To reiterate, $g(X)$ is the difference between the far-field image $U(X)$ formed from near-field phase X under proximity effect and the desired image I expressed in Equations 4.9 and 4.10.

We use the notation M_{ij} to indicate the element at index $[i, j]$ in a matrix M. For simplicity, we may also use U or P to represent $U(X)$ or $P(X)$.

$$\frac{\partial g(X)}{\partial X_{ij}} = \text{Tr}[(\frac{\partial g}{\partial U})^{\mathrm{T}} \frac{\partial U}{\partial X_{ij}}] \tag{4.16}$$

$$= \text{Tr}[2(U - I)^{\mathrm{T}} \frac{\partial U}{\partial X_{ij}}] \tag{4.17}$$

Let $Q = U - I$, and $\frac{\partial g(X)}{\partial X_{ij}}$ becomse the following.

$$\frac{\partial g(X)}{\partial X_{ij}} = 2 \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} Q_{mn} [\frac{\partial U_{mn}}{\partial X_{ij}}] \tag{4.18}$$

Algorithm 2: Proximal Proximity Effect Correction

```
input  : $X_0$, $I$, $\tau$, $\beta$
output: $X$
$X = X_0$;
$X_{\text{prev}} = X$;
while Within maximum iteration do
    $Y = X + \omega(X - X_{\text{prev}})$;
    while true do
        $Z = \text{prox}_{\tau c}(Y - \tau \nabla g(Y))$;
        if $g(Z) < g'_\tau(Z, Y)$ then
            break;
        else
            $\tau = \beta\tau$;
        end
    end
    $X_{\text{prev}} = X$;
    $X = Z$;
    if Termination condition met then
        break;
    end
end
```

Now we take a look at U_{mn}.

$$\begin{aligned} U_{mn} &= |\mathcal{F}(e^{iP(X)})|^2_{mn} \\ &= [\mathcal{F}(e^{iP(X)})]_{mn}\overline{[\mathcal{F}(e^{iP(X)})]_{mn}} && (4.19) \\ &= (\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} e^{iP_{kl}} e^{-i2\pi m\frac{k}{M}} e^{-i2\pi n\frac{l}{N}})\cdot \\ &\quad (\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} e^{-iP_{kl}} e^{i2\pi m\frac{k}{M}} e^{i2\pi n\frac{l}{N}}) && (4.20) \\ \frac{\partial U_{mn}}{\partial X_{ij}} &= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \frac{\partial U_{mn}}{\partial P_{kl}} \frac{\partial P_{kl}}{\partial X_{ij}} && (4.21) \\ \frac{\partial U_{mn}}{\partial P_{kl}} &= ie^{iP_{kl}} e^{-i2\pi m\frac{k}{M}} e^{-i2\pi n\frac{l}{N}} \overline{[\mathcal{F}(e^{iP})]_{mn}} \;+ \\ &\quad (-ie^{-iP_{kl}}) e^{i2\pi m\frac{k}{M}} e^{i2\pi n\frac{l}{N}}) [\mathcal{F}(e^{iP})]_{mn} && (4.22) \end{aligned}$$

Now let $H^{mn}_{kl} = ie^{iP_{kl}} e^{-i2\pi m\frac{k}{M}} e^{-i2\pi n\frac{l}{N}} \overline{[\mathcal{F}(e^{iP})]_{mn}}$, Equation 4.22 becomes the following.

$$\frac{\partial U_{mn}}{\partial P_{kl}} = H^{mn}_{kl} + \overline{H^{mn}_{kl}} \tag{4.23}$$

Take Equation 4.23 and 4.21 into Equation 4.18, we get the following.

$$\frac{\partial g(X)}{\partial X_{ij}} = 2\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Q_{mn} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} (H^{mn}_{kl} + \overline{H^{mn}_{kl}}) \frac{\partial P_{kl}}{\partial X_{ij}} \tag{4.24}$$

Since Q_{mn} and $\frac{\partial P_{kl}}{\partial X_{ij}}$ are real, we know that $\frac{\partial g(X)}{\partial X_{ij}}$ is real, which is what we expect.

$$\frac{\partial g(X)}{\partial X_{ij}} = 4\mathrm{Re}(S_{ij}) \tag{4.25}$$

$$\begin{aligned}
S_{ij} &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Q_{mn} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} H_{kl}^{mn} \frac{\partial P_{kl}}{\partial X_{ij}} \qquad (4.26)\\
&= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Q_{mn} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} ie^{iP_{kl}} e^{-i2\pi m\frac{k}{M}} e^{-i2\pi n\frac{l}{N}} \overline{[\mathcal{F}(e^{iP})]_{mn}} \frac{\partial P_{kl}}{\partial X_{ij}} \qquad (4.27)\\
&= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \{Q_{mn}\overline{[\mathcal{F}(e^{iP})]_{mn}} e^{-i2\pi k\frac{m}{M}} e^{-i2\pi l\frac{n}{N}}\} ie^{iP_{kl}} \frac{\partial P_{kl}}{\partial X_{ij}} \qquad (4.28)\\
&= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \{[\mathcal{F}(Q\cdot\overline{\mathcal{F}(e^{iP})})]_{kl} \quad ie^{iP_{kl}}\} \frac{\partial P_{kl}}{\partial X_{ij}} \qquad (4.29)
\end{aligned}$$

It is easy to follow that as long as $\frac{\partial P_{kl}}{\partial X_{ij}}$ is constant (P being a convolution is one such case), S and therefore ∇g can be calculated as follows.

$$S = P\{\mathcal{F}[Q\cdot\overline{\mathcal{F}(e^{iP})}]\cdot ie^{iP}\} \tag{4.30}$$

$$\nabla g = 4\mathrm{Re}(S) \tag{4.31}$$

4.4.3 Simulations

We have carried out our computational simulations on a dataset containing 29 real-life images (Sheikh *et al.* [7,8]) and an additional newspaper image containing text from the CHRONIC dataset (Smits and Faber [9]). Our software implementation uses Matlab running on an Intel 9900K CPU. We present the simulation results in color as our proximity effect correction methods are applicable to all wavelengths.

Multi-color displays can be easily implemented by coupling three lasers into the NPA.

4.4.3.1 Implementation and Test

As mentioned before in Section 4.4.1, we perform the deconvolution by minimizing the residual $\|M_K\boldsymbol{x} - \boldsymbol{\phi}\|_2^2$ while maintaining $\boldsymbol{x} \geq 0$. Specifically, the cost function $f(\boldsymbol{x})$ to minimize in the proximal algorithm is the following.

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + c(\boldsymbol{x}) \tag{4.32}$$

$$g(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(M_K^{\mathrm{T}}M_K)\boldsymbol{x} + (-\phi^{\mathrm{T}}M_K)\boldsymbol{x} \tag{4.33}$$

$$c(\boldsymbol{x}) = \mathcal{I}_{[0,\mathrm{inf}]}(\boldsymbol{x}) \tag{4.34}$$

We use the accelerated proximal gradient algorithm. The gradient of $g(\boldsymbol{x})$ is evaluated as the following.

$$\nabla g(\boldsymbol{x}) = M_K^{\mathrm{T}}M_K - \boldsymbol{\phi}^{\mathrm{T}}M_K \tag{4.35}$$

For the PPEC method, we use the accelerated proximal algorithm exactly like described in Section 4.4.2. We use the residual change to determine when to terminate the algorithm. Let X^k be the solution in iteration k. The residual is $g(X^k)$ from Equation 4.9. Let r^k be the change in iteration k compared to the previous iteration. We terminate the algorithm when r_k drops below 0.001 for 10 consecutive frames.

$$r_k = \frac{\text{abs}(g(X)^k - g(X)^{k-1})}{g(X)^{k-1}} \tag{4.36}$$

.

In addition to showing the simulated reconstructions of the far-field images, we also quantify the quality of these images with *structural similarity index* (SSIM) from Wang *et al.* [175]. The power of the luminance, contrast, and structure terms we use in the SSIM are all 1. SSIM(I, R) measures the similarity between the input image I and the reference image R. When I and R are identical, SSIM$(I, R) = 1$. As the similarity decreases, so does the SSIM. We have also considered two other image quality metrics, the classic Peak Signal-to-Noise Ratio (PSNR) and a neural network-based perceptual similarity by Zhang *et al.* [176]. We did not find any contradictions between these image metrics and direct observation in our tests. The larger the proximity effect level, the more noise is present and the lower the quality metric score gets. More detail on these image quality metrics in our test can be seen in Section 4.4.3.2.

4.4.3.2 Image Quality Metrics

In order to quantitatively assess the negative impacts of proximity effect and the effectiveness of the correction, a metric for image quality is needed. We look at three image quality metrics and pick one to use in our simulations. The three are Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) from Wang *et al.* [175] and a neural network-based perceptual image quality metric from Zhang *et al.* [176].

The first two are well-known image quality metrics. Given the reference image R and the image in question I whose quality we want to quantify, the PSNR is defined as follows. The images R and I have the same size $M \times N$.

$$\mathrm{PSNR}_R(I) = 10\log_{10}(\frac{\max(R)^2}{\mathrm{MSE}(R,I)}) \tag{4.37}$$

$$\mathrm{MSE}(R,I) = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(R_{ij} - I_{ij})^2 \tag{4.38}$$

The PSNR treats each pixel separately. Even though it is a good metric for reporting the ℓ_2 difference, it is not suitable for structured signals such as images. The SSIM is a perceptually inspired image metric and is defined as follows.

$$\mathrm{SSIM}_R(I) = L(I,R)^\alpha + C(I,R)^\beta + S(I,R)^\gamma \tag{4.39}$$

$$L(I,R) = \frac{2\mu_I\mu_R + C_1}{\mu_I^2 + \mu_R^2 + C1} \tag{4.40}$$

$$C(I,R) = \frac{2\sigma_I\sigma_R + C_2}{\sigma_I^2 + \sigma_R^2 + C_2} \tag{4.41}$$

$$S(I,R) = \frac{\sigma_{I,R} + C_3}{\sigma_I\sigma_R + C_3} \tag{4.42}$$

$L(I,R)$, $C(I,R)$ and $S(I,R)$ are the luminance, contrast and structural terms, with μ, σ being the mean and standard deviation of the respective images. $\sigma_{I,R}$ is the cross-covariance for I and R. α, β, γ and C_1, C_2 and C_3 are parameters. We use the default parameters, i.e. $\alpha = \beta = \gamma = 1$, $C_1 = (0.01D)^2$, $C_2 = (0.03D)^2$, $C_3 = \frac{C_2}{2}$. D is the dynamic range of the images, which is 255 for the uint8 images we use.

Zhang *et al.* [176] find that some of the loss functions based on the deep features of image synthesis applications are uniquely suitable to express the quality of the image. In their study, there are cases where the perceptual similarity's choices align more with humans while the images PSNR and SSIM think are better are actually considered worse by humans. It has also been shown that the perceptual similarity also performs well on images outside the domain of training image datasets. The metric value d produced by their network[1] represents the "perceptual difference". To keep up with the PSNR and SSIM where larger metric value means better image quality, we use $s = e^{-d}$ to express the "perceptual similarity".

We want the image quality metric to reflect the severity of the proximity effect and treat every image equally. Specifically, we want to see the increasing severity of proximity effect reflected in the metric for every image. Additionally, when the severity of the proximity effect is fixed, we want the variance of how the metric scores every image to be as small as possible. Therefore, we assess the three image quality metrics in two ways. First, we apply increasing levels of proximity effect to one image and observe how the metrics change. The results can be seen in Figure 4.6.

Next, we fix the proximity effect and plot the metrics on the entire image dataset of 29 images. We want the variance over the images to be as small as possible. The results can be seen in Figure 4.7.

It can be seen that all three metrics satisfy our requirements for the image quality metric. We choose to use SSIM as the image quality metric in our simulations because there is no significant disagreement among between the SSIM and the

[1] Their code is publicly available at `https://github.com/richzhang/PerceptualSimilarity`

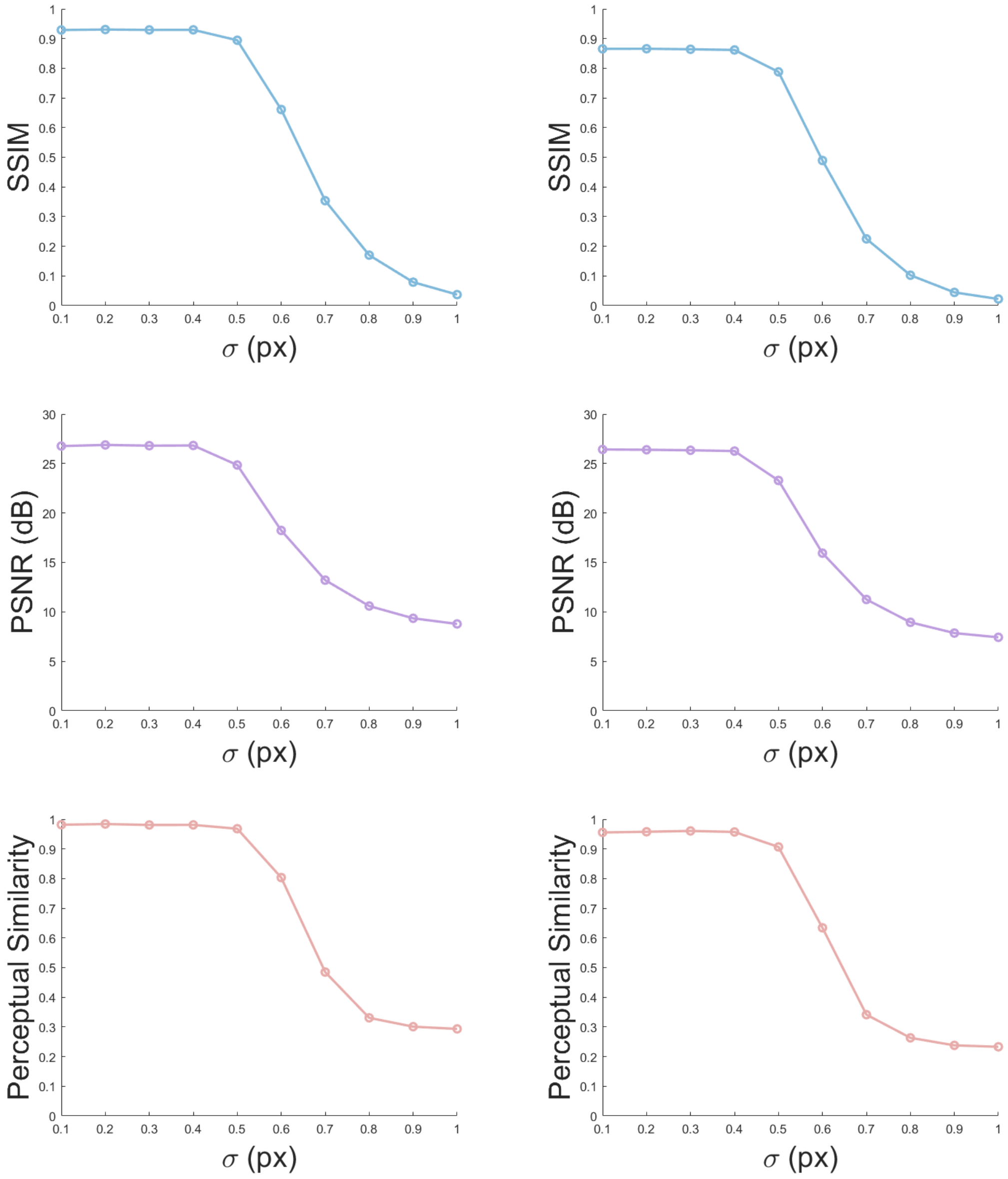
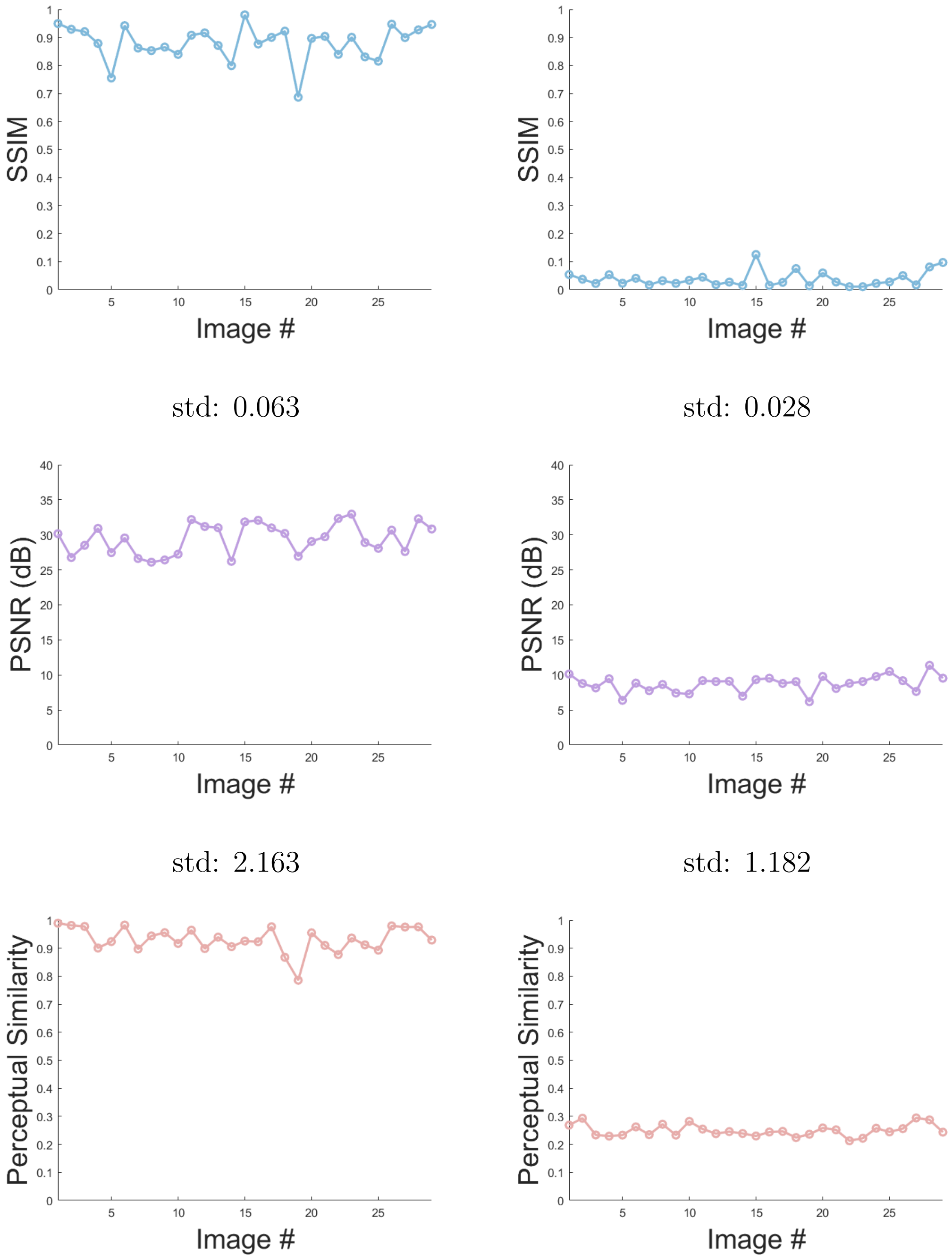


Figure 4.6: How the three image quality metrics change with increasing levels of proximity effect on a single image. The results on two example images are shown (left and right column). These results are representative of the entire image dataset.



std: 0.063

std: 0.028

std: 2.163

std: 1.182

std: 0.050

std: 0.084

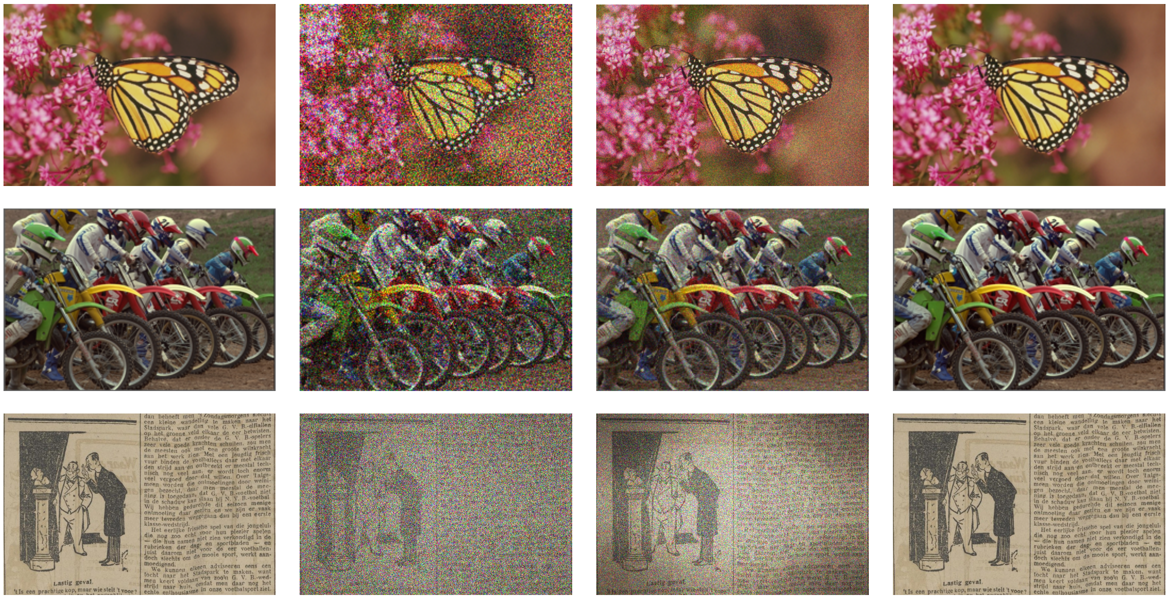
Figure 4.7: The image quality metrics over the entire dataset when the proximity effect level is fixed. A small proximity effect ($\sigma = 0.1$ px, left) and a large one ($\sigma = 1.0$ px, right) are shown. The standard deviations of the metrics over the dataset are also shown under each image.

perceptual similarity in this simulation and the SSIM value drops to around zero when the proximity effect level is high and the result image is completely noise.

4.4.3.3 Correction Effectiveness

We first present the qualitative visual results in Figure 4.8. The original images are shown on the left as references. We also include the method in Persson *et al.* [121] and Gemayel *et al.* [119] in the second column. We refer to this method as the Basic Iterative Proximity Effect Correction. The basic IPEC method skips the deconvolution step when applying the source amplitude constraint, namely using $B = \mathcal{S} \cdot e^{iP(\text{phs}(A))}$ in Algorithm 1. The results of the IPEC method and the PPEC method are presented in the remaining two columns, respectively. The proximity effect level in the first two rows is $\sigma = 0.66$ (px) corresponding to our sample device. To show the impact of a greater proximity effect level especially on text-rich images, we use $\sigma = 0.8$ (px) in the third row with the image of a newspaper.

It can be seen from Figure 4.8, the IPEC method performs much better than the basic IPEC method and is able to produce reasonable results. However, it is still not able to effectively correct high levels of proximity effect. The PPEC method has the best correction effectiveness of all methods compared and is the only method that can recover text-rich images when the proximity effect level is high. At high proximity effect levels such as $\sigma = 0.80$ px, a vignette effect (where the center of the image is brighter than the border region) can be observed in the results of the basic IPEC as well as IPEC. This is because the proximity effect is modeled as a



Original

Basic IPEC Gemayel *et al.* [119]

Iterative PEC

Proximal PEC

Figure 4.8: This figure compares the various Proximity Error Correction (PEC) methods. The proximity effect level used in the first two rows is $\sigma = 0.66$ px. At this level, the IPEC results are noisy but still reasonable. The proximity effect level for the bottom row is $\sigma = 0.8$ px. At this level, the results from the IPEC methods are incomprehensible but the Proximal PEC method provides readable results. The images in the first two rows are from the LIVE dataset (Sheikh *et al.* [7, 8]) and the newspaper image in the bottom row is from the CHRONIC dataset (Smits and Faber [9]).

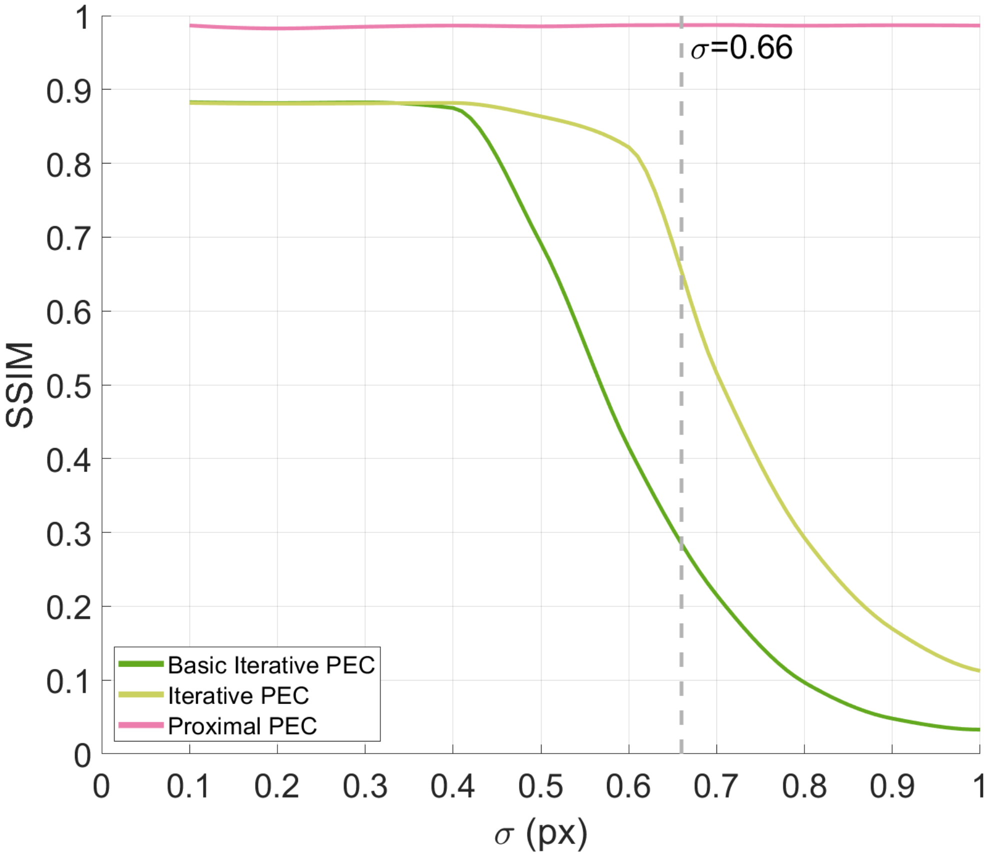


Figure 4.9: Image quality degradation for various methods. This figure is plotted with interpolated data from simulations where σ ranges from 0.10 to 1.00 px with a 0.10 px interval.

Gaussian convolution. When the proximity effect is high, the Gaussian kernel gets flatter and the near-field phase becomes more uniform. Since the Fourier transform is used to approximate the propagation, more energy gets focused around the center in the far field.

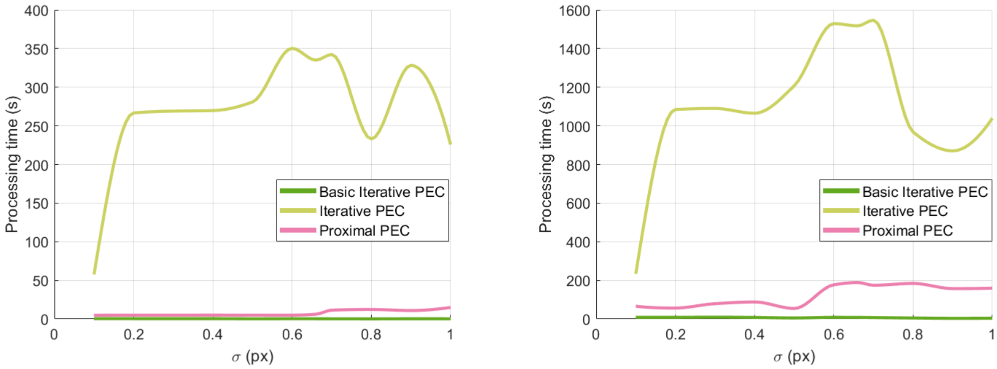
Figure 4.9 shows how the average image quality degrades with increasing amounts of proximity effect for the methods. We observe that the image quality degrades with increasing proximity effect σ for the IPEC method. The PPEC method retains the best image quality at all proximity effect levels. At $\sigma = 0.66$ px, the IPEC and PPEC methods improve the SSIM image quality metric by 121.26% and 249.98% compared to the basic IPEC method, respectively.

4.4.3.4 Processing Time

We also look at the processing time of the proposed proximity effect correction methods. Two factors may impact the processing time, the severity of the proximity effect and the resolution of the image.

The processing time of the proposed methods with increasing thermal proximity effect levels is presented in Fig 4.10. It is clear at first glance that the IPEC method is very time-consuming. It is to be expected since IPEC is an iterative method and at the same time, the deconvolution in each iteration is also implemented in an iterative manner. The PPEC method is able to maintain a much faster processing speed than the IPEC method. The processing time for the PPEC method also increases with the proximity effect level. The basic iterative PEC method is the fastest of the three methods tested. However, as shown in Figure 4.8 and 4.9, this method does not provide sufficient proximity effect correction.

The processing time of the methods with increasing image resolution is shown in Figure 4.11. The processing time increases with image resolution for both the IPEC and PPEC methods.



384 × 256

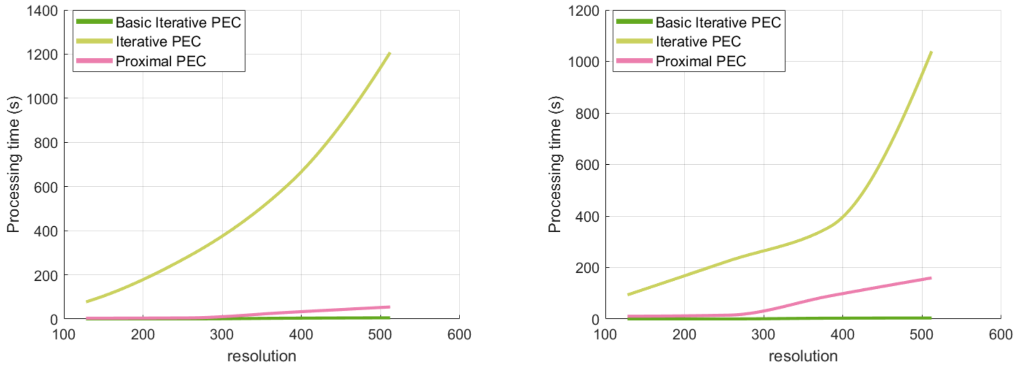
768 × 512

Figure 4.10: The processing time of the proposed proximity effect correction methods with increasing levels of proximity effect at respective resolution.

4.4.3.5 Spatial Frequency of Images

Since the Gaussian filter used in the convolution modeling of the thermal proximity effect is a low-pass filter, we want to investigate whether the spatial frequency of the image affects the image quality of the proximity effect correction methods.

To test this, we prepare images filtered by low-pass filters with several frequency cutoff thresholds. We use the Butterworth low-pass filter to avoid the ringing artifacts in the filtered images. The low-pass filter we use is expressed in Equation 4.43. $f(u, v)$ is the normalized distance from (u, v) to the center of the spectrum. $f(u, v) = \sqrt{(\frac{2u}{M} - 0.5)^2 + (\frac{2v}{N} - 0.5)^2}$. f_c is the frequency cutoff threshold. We use frequency cutoff thresholds $f_c = 0.1, 0.2, 0.3, 0.4, 0.5$. The results of the PEC methods on the low-pass-filtered images are shown in Figure 4.12.



$\sigma = 0.5$ px

$\sigma = 1.0$ px

Figure 4.11: The processing time of the proposed proximity effect correction methods on images of increasing resolution. The image aspect ratio is kept at 3 : 2. The resolution on the x-axis indicates the vertical resolution of the image.

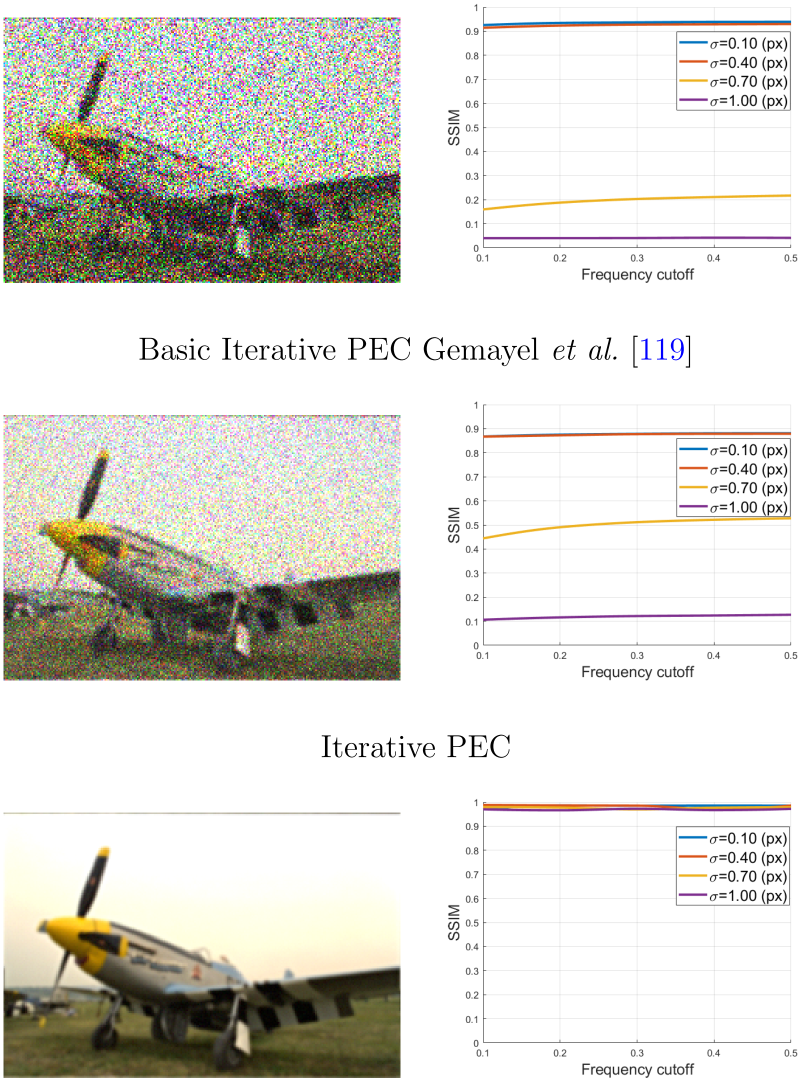
$$\mathrm{BLPF}(I) = \mathcal{F}^{-1}[\mathcal{F}(I) \cdot H] \tag{4.43}$$

$$H(u, v) = \frac{1}{1 + [f(u, v)/f_c]^4} \tag{4.44}$$

As can be seen in Figure 4.12, there is no significant change of image quality with different image spatial frequencies. This may be because the thermal proximity effect mainly causes noise artifacts in the reconstructed images rather than blurring.

4.4.4 Discussion

Here we propose the iterative proximity effect correction which integrates deconvolution into the iterative phase retrieval algorithm and the proximal proximity effect correction which directly finds the phase that optimizes the formed image under thermal proximity effect. Here we discuss some details of the proposed methods.



Basic Iterative PEC Gemayel *et al.* [119]

Iterative PEC

Proximal PEC

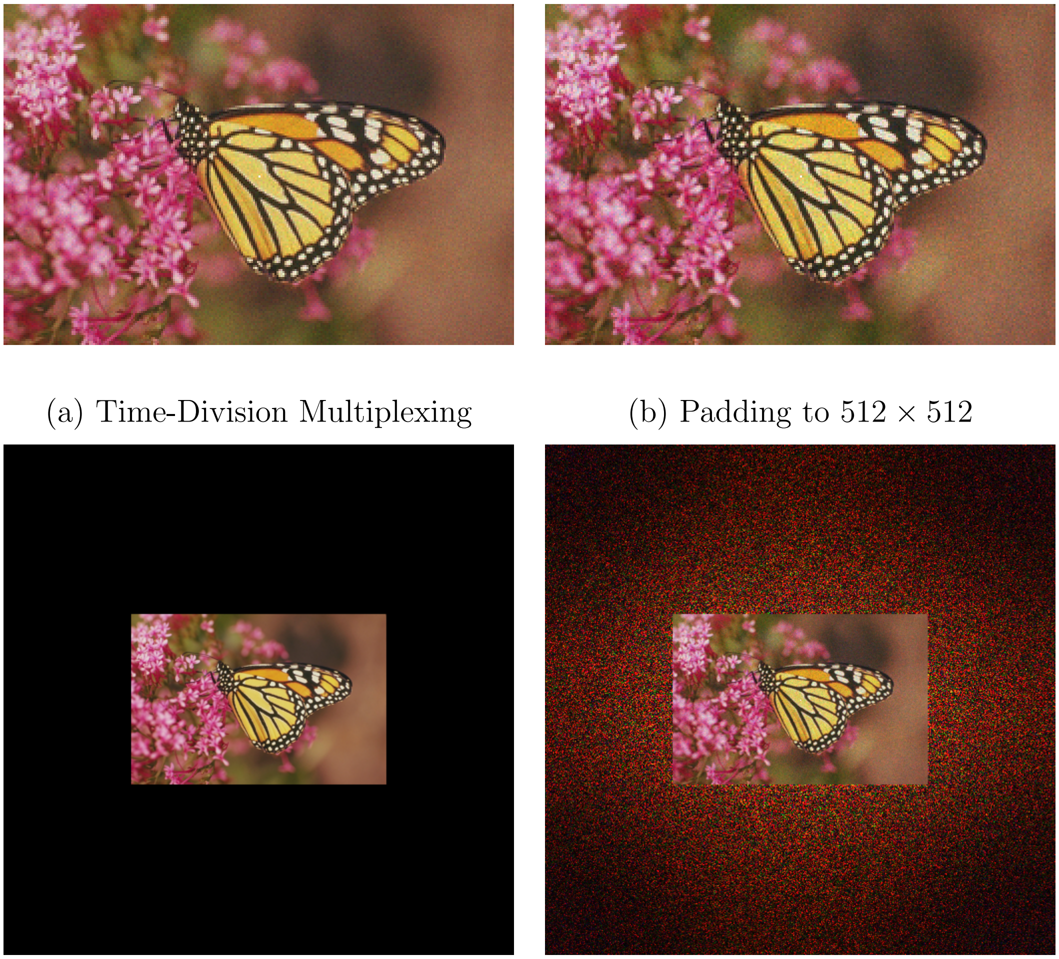
Figure 4.12: The results of the basic iterative PEC method (top row) and the iterative PEC method (middle row) and the proximal PEC method (bottom row) on low-pass filtered images. The images (left column) are the results of the respective PEC methods applied to a low-pass filtered image with frequency cutoff threshold $f_c = 0.2$ at proximity effect level $\sigma = 0.7$ px. The image qualities of each PEC method at different proximity effect levels on images with different frequency cutoff thresholds are shown in the right column.

4.4.4.1 Improving IPEC

As can be seen in Figure 4.9, the PPEC method achieves better image quality at all the thermal proximity effect levels tested. In fact, the PPEC method is able to maintain the image quality with increasing thermal proximity effect levels, albeit at the cost of slightly longer processing time (see Figure 4.10). IPEC methods perform reasonably well at lower proximity effect levels. But at high proximity effect levels, the IPEC method inevitably breaks down. An example with proximity effect level $\sigma = 0.8$ px can be seen in Figure 4.8 bottom row. However, some techniques can be applied to the IPEC method to further improve image quality.

The first technique is time-division multiplexing (TDM). We observe that the artifacts in the IPEC results are noise artifacts. We can run the IPEC method multiple times with different initial phase ϕ_0 and display the different resulting phase patterns in consecutive frames. This would increase the processing time of the IPEC method. The TDM technique uniquely takes advantage of the fast refreshing nature of the NPA. We can simulate the TDM by averaging the resulting far-field images, which are shown in Figure 4.13(a). Obviously, if IPEC is run for n times, the TDM technique would take n times the processing time of the IPEC method.

The other technique is to pad the desired image with a background region. In the iterative phase retrieval algorithm, a signal region and a background region on the wavefront can be specified. The two regions have no intersection. We want the signal region to display the desired amplitude pattern therefore the target amplitude constraint is only applied in the signal region. The background region offers more



(a) Time-Division Multiplexing

(b) Padding to 512 × 512

(c) Initial padded image

(d) Result padded image

Figure 4.13: (a) shows the result of TDM application to the IPEC method with 10 different initial phase. (b) shows the of result of the padding technique applied to IPEC method. The center signal region is shown. The image is padded from 171 × 256 to 512 × 512. The initial and result padded images are shown in (c) and (d), respectively. The thermal proximity effect level is $\sigma = 0.66$ px for both techniques.

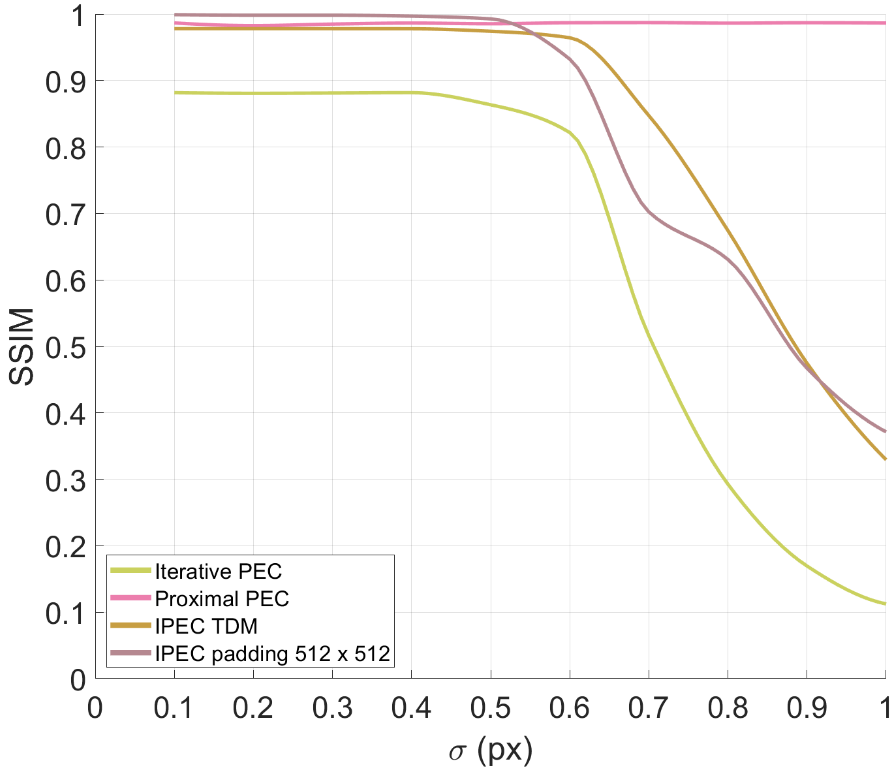


Figure 4.14: IPEC with TDM and padding techniques compared to the original IPEC method and the PPEC method.

freedom to phase manipulation. We have included more detail on the iterative phase retrieval algorithm in the supplemental material. An example of this technique is shown in Figure 4.13(b). The padding technique uses more pixels for a given image than PPEC and wastes a portion of the display. Padding also increases the processing time.

A comparison of the two techniques with the IPEC and PPEC methods is shown in Figure 4.14. As can be seen, both techniques perform much better than the original IPEC method without the techniques. The two techniques achieve comparable image qualities to the PPEC method at lower proximity effect ($\sigma < 0.7$

px). This indicates that the IPEC method has the potential to achieve comparable correction effectiveness as the PPEC method. However, it is necessary to find fast deconvolution methods for the IPEC method to be efficient.

4.4.4.2 Pixel Pitch and Thermal Proximity Effect

Reducing the pixel pitch is one of the most effective ways of improving the quality of a holographic display. Given the form factor of the display, reducing the pixel pitch not only increases the number of pixels but also improves the field-of-view of the display. The importance of pixel pitch has been discussed in a number of studies on LCoS based holographic displays such as Shi *et al.* [1] and Jang *et al.* [149]. The same principle also applies to NPA holographic displays. It is easy to see that when the pixel pitch decreases, the pixels are more densely packed and each pixel is affected more by the heat from neighboring pixels. This is highly likely to lead to a more severe proximity effect. The PPEC method is the only effective correction method at the moment to address this problem.

4.4.4.3 Choice of Proximal Algorithm

There are multiple variations of the proximal algorithm. Two examples are the proximal gradient algorithm and the ADMM (alternating direction method of multipliers). Both algorithms can be used to minimize the function $f(X) = g(X) + c(X)$ in Equation 4.12. In the proximal gradient algorithm, let X^k be the solution in iteration k. The update in iteration $k + 1$ is the following.

$$X^{k+1} = \mathrm{prox}_{\lambda c}(X_k) \tag{4.45}$$

In the ADMM algorithm, the updates in each iteration are the following. X^k and Z^k converge to each other and are in the domains of f and g respectively.

$$X^{k+1} = \mathrm{prox}_{\lambda f}(Z^k - U^k) \tag{4.46}$$

$$Z^{k+1} = \mathrm{prox}_{\lambda g}(X^{k+1} + U^k) \tag{4.47}$$

$$U^{k+1} = U^k + X^{k+1} - Z^{k+1} \tag{4.48}$$

The main difference between the two algorithms is the evaluation of the proximal operator of the function $g(X)$. $g(X)$ is a non-linear function, and minimizing it is the core of the proximal proximity effect correction problem. Since it is difficult to evaluate $\mathrm{prox}_{\lambda g}$, we use the proximal gradient algorithm which only involves calculating the gradient of $g(X)$. In practice, we use the accelerated version of the proximal gradient algorithm. Its update steps are described in Section 4.4.2. The accelerated proximal gradient algorithm has an extrapolation step that can be interpreted as moving the solution a step faster towards the optimal.

In the implementation of the quadratic programming deconvolution in the IPEC method, we also use the proximal algorithm. Even though there exists closed-form evaluation of the function $g(\boldsymbol{x})$ in Equation 4.33 (see Parikh and Boyd [172]), it still involves solving a linear system with tens of thousands of variables. We have found that the accelerated proximal gradient implementation is faster than the ADMM implementation.

4.4.4.4 Iterations of PPEC

Here we show how the image quality improves over the iterations in the PPEC method. An example with proximity effect level $\sigma = 0.66$ px is shown in Figure 4.15.

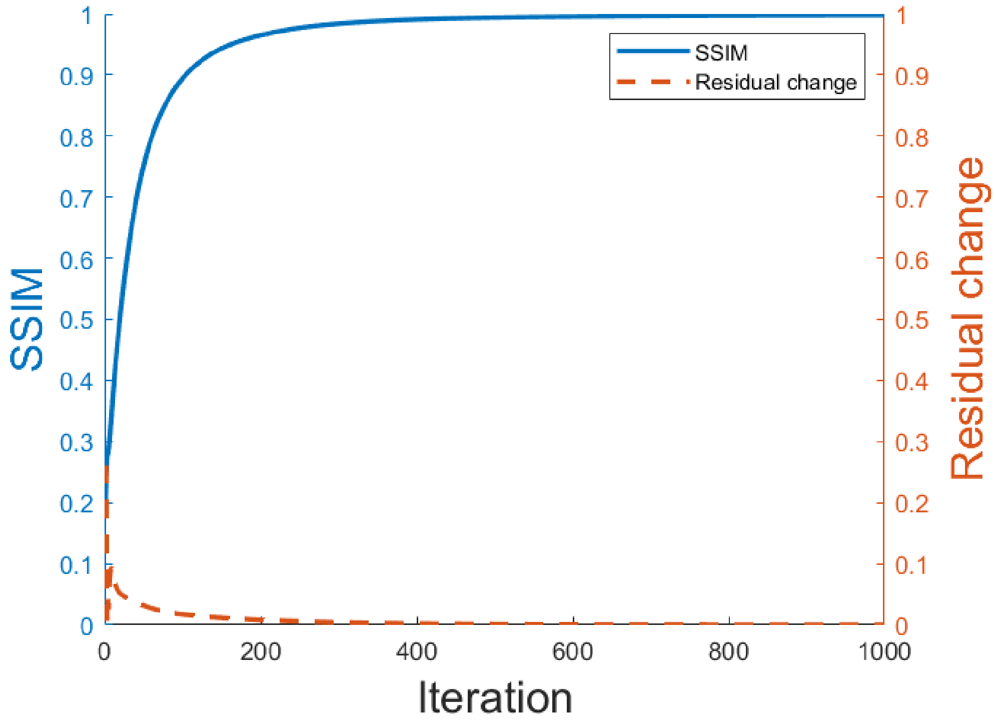


Figure 4.15: Image quality improves over the iterations. The orange dashed line indicates the change in the residual compare to the previous iteration. $\sigma = 0.66$ px in this case.

The image quality improves over the iterations. We use the residual change r^k in Equation 4.36 to determine when to terminate the algorithm. We terminate the algorithm when r^k drops below 0.001 for 10 consecutive frames at which point the image quality changes very little.

4.4.5 Limitations

Our simulations have shown that the proposed PPEC method achieves very good correction effectiveness across different proximity effect levels. However, there are still limitations to this approach. The NPA's properties can help addresses some of the limitations. Some of the properties may pose problems as well. Here we list some of the limitations and propose potential solutions.

4.4.5.1 Amplitude Modulation and Diminished Reality

NPAs have the advantage of fast refresh rate and integrated light source. However, NPAs also have limitations. The integrated light source makes it difficult for NPAs to modify the light from the real scene in augmented reality. Reflective or transmissive SLMs can modify the wavefront from the real scene that reaches them, achieving diminished reality or occlusion of real objects in AR, such as in Rathinavel *et al.* [177]. Combining LCoS SLMs can expand the capabilities of NPA holographic displays.

4.4.5.2 Choice of Initial Phase

The initial phase value of X in the PPEC method has an effect on the result. We find that a constant value often leads to poor image quality or an excessively long time to convergence. We use random values in our implementation and about 1% of the PPEC results suffer from slight speckle noise. We currently deal with this by detecting the noise and rerun PPEC with a different initial value. Heide *et* *al.* [173] propose to use the result from the iterative phase retrieval algorithm as the initial value for the phase retrieval using the proximal algorithm. We do not think it is a viable choice for us because the iterative phase retrieval would add to the processing time of the method. Still, the intermediate result of the iterative method after only a few iterations may offer better robustness for our method.

4.4.5.3 Larger Proximity Effect for PPEC

We have shown that the PPEC method effectively corrects the proximity effect within the tested range (from $\sigma = 0.10$ px to 1.00 px). While this is a realistic range in which our NPA design lies, the PPEC method may not suffice at higher proximity effect levels. For example, at $\sigma = 1.50$ px, the average SSIM of the PPEC method is 0.48, and at $\sigma = 2.00$ px, the average SSIM drops to 0.30.

4.4.5.4 3D Holograms

In this section, we have explored proximity effect correction on NPAs to form 2D images. Holographic displays truly shine at displaying realistic 3D scenes at the correct depth. The most naive approach would be to generate the phase-only hologram of a 3D scene either with double-phase encoding (such as in Maimone *et al.* [62]) or multi-plane phase retrieval (such as Makowski *et al.* [178]) and apply deconvolution to the phase-only hologram. Or we can integrate deconvolution into the multi-plane phase retrieval just like the proposed IPEC method. Noise and discontinuity in the depth can be expected from these approaches. Another approach that takes advantage of the fast refresh rate of the NPA is time-division multiplexing. Similar to what is suggested in Chakravarthula *et al.* [164], we can divide the 3D scene into a stack of images at different depth and apply PPEC to each image. The results are displayed in consecutive frames. The PEC methods need to adapt to a different wave propagation model that specifies the propagation distance such as the Fresnel propagation or Fraunhofer propagation. We also plan to look at direct optimization methods that optimize the 3D scene as a whole. I try to address 3D holograms in Section 4.5.

4.5 Proximity Effect Correction for Fresnel Holograms

Here, we present a novel proximity effect correction method for Fresnel holograms on the NPAs. We refer to this method as the Fresnel Proximity Effect Correction (FPEC). This method uses the proximal algorithm to directly find the phase of a phase-only hologram that minimizes the difference between the desired image and the image formed at a specific distance under the proximity effect.

Since our method focuses on Fresnel holograms and reconstructing 2D images or 3D scenes, we consider the case of Fresnel diffraction approximation. Assuming a wavefront U_1 formed on the NPA, the wavefront U_2 formed by propagating U_1 by a distance z can be calculated as in Equation 4.49. The two wavefronts U_1 and U_2 are parallel to each other. $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier transform and its inverse. $H_{z\lambda}(f_X, f_Y) = e^{ikz} \cdot e^{-i\pi\lambda z(f_X^2+f_Y^2)}$ is the transfer function in the frequency domain where $k = \frac{2\pi}{\lambda}$ is the wave number for wavelength λ. We refer interested readers to

Goodman [123] for the detailed derivation of the Fresnel diffraction.

$$U_2(x, y) = \mathcal{F}^{-1}\{\mathcal{F}[U_1(x, y)]H_{z\lambda}(f_X, f_Y)\} \tag{4.49}$$

The rest of the FPEC method is very similar to the PPEC method. Given the desired image $I \in \mathbb{R}^{+M\times N}$, we wish to find the non-negative phase $X \in \mathbb{R}^{+M\times N}$ that minimizes the difference between the desired image I and the formed image under the proximity effect, expressed as $g(X)$ in Equation 4.9. Let $U(X)$ be the intensity of the wavefront which the NPA with input phase X forms at a specific distance z from the NPA. Let $P(X)$ be the function that applies the proximity effect to the phase pattern, i.e. $P(X) = X * K$ where K is the convolution kernel in Equation 4.4.

$$g(X) = \|U(X) - I\|_{\mathrm{F}}^2 \tag{4.50}$$

$$U(X) = |\mathcal{F}^{-1}[\mathcal{F}(e^{iP(X)}H_{z\lambda}]|^2 \tag{4.51}$$

To enforce the non-negative constraint, we add an indicator function $c(X)$. The FPEC method minimizes the function $f(X)$ in Equation 4.12 using the proximal algorithm.

$$f(X) = g(X) + c(X) \tag{4.52}$$

$$c(X) = \mathcal{I}_{[0,+\inf)}(X) \tag{4.53}$$

The same pseudocode in Algorithm 2 can be used for the FPEC method. We document the derivation of the gradient for the FPEC method in Section 4.5.0.1.

4.5.0.1 Derivation of FPEC Gradient

Here we present the derivation of the gradient ∇g of the function $g(X)$. Like before, we use the notation M_{ij} to indicate the element at index $[i, j]$ in a matrix M. For simplicity, we use U or P to represent $U(X)$ or $P(X)$, respectively.

$$\frac{\partial g(X)}{\partial X_{ij}} = \mathrm{Tr}[(\frac{\partial g}{\partial U})^{\mathrm{T}} \frac{\partial U}{\partial X_{ij}}] \tag{4.54}$$

$$= \mathrm{Tr}[2(U - I)^{\mathrm{T}} \frac{\partial U}{\partial X_{ij}}] \tag{4.55}$$

Let $Q = U - I$, and we can represent $\frac{\partial g(X)}{\partial X_{ij}}$ as follows:

$$\frac{\partial g(X)}{\partial X_{ij}} = 2 \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} Q_{mn} [\frac{\partial U_{mn}}{\partial X_{ij}}] \tag{4.56}$$

Next, we consider the derivation of U_{mn}. Let F be the complex wavefront in the observation plane. For simplicity, we omit mentioning the subscripts z and λ with the transfer function $H_{z\lambda}$ in Equation 4.49. To closely relate to the implementation, $\mathcal{F}$ and $\mathcal{F}^{-1}$ are written as the discrete Fourier transform and its inverse. $\overline{X}$ is the complex conjugate of X.

$$F(X) = \mathcal{F}^{-1}[\mathcal{F}(e^{iP(X)}H]$$

$$F_{mn} = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} H_{pq}(\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} e^{iP_{kl}} e^{-i2\pi p\frac{k}{M}} e^{-i2\pi q\frac{l}{N}}) e^{i2\pi m\frac{p}{M}} e^{i2\pi n\frac{q}{N}}$$

$$\overline{F_{mn}} = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} H_{pq}(\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} e^{-iP_{kl}} e^{i2\pi p\frac{k}{M}} e^{i2\pi q\frac{l}{N}}) e^{-i2\pi m\frac{p}{M}} e^{-i2\pi n\frac{q}{N}}$$

$$U_{mn} = F_{mn} \cdot \overline{F_{mn}}$$

$$\frac{\partial U_{mn}}{\partial X_{ij}} = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \frac{\partial U_{mn}}{\partial P_{kl}} \frac{\partial P_{kl}}{\partial X_{ij}} \tag{4.57}$$

$$\begin{aligned}\frac{\partial U_{mn}}{\partial P_{kl}} &= \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} H_{pq} i e^{iP_{kl}} e^{-i2\pi p\frac{k}{M}} e^{-i2\pi q\frac{l}{N}} e^{i2\pi m\frac{p}{M}} e^{i2\pi n\frac{q}{N}} \cdot \overline{F_{mn}} \\ &+ \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} H_{pq}(-ie^{-iP_{kl}}) e^{i2\pi p\frac{k}{M}} e^{i2\pi q\frac{l}{N}} e^{-i2\pi m\frac{p}{M}} e^{-i2\pi n\frac{q}{N}} \cdot F_{mn}\end{aligned} \tag{4.58}$$

Since the two summation terms in Equation 4.58 are complex conjugates, it easy to see that $\frac{\partial U_{mn}}{\partial P_{kl}}$ is real and can be simplified as shown next. The function "Re" takes the real components of a complex value.

$$\frac{\partial U_{mn}}{\partial P_{kl}} = 2\mathrm{Re}\{\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} H_{pq} i e^{iP_{kl}} e^{-i2\pi p\frac{k}{M}} e^{-i2\pi q\frac{l}{N}} e^{i2\pi m\frac{p}{M}} e^{i2\pi n\frac{q}{N}} \cdot \overline{F_{mn}}\} \tag{4.59}$$

Using Equation 4.59 and 4.57 with Equation 4.56, we get the following:

$$\frac{\partial g(X)}{\partial X_{ij}} = 4\mathrm{Re}(S_{ij})$$

$$\begin{aligned}S_{ij} &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Q_{mn} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} H_{pq} \cdot ie^{iP_{kl}} \\ &\quad \cdot e^{-i2\pi p\frac{k}{M}} e^{-i2\pi q\frac{l}{N}} e^{i2\pi m\frac{p}{M}} e^{i2\pi n\frac{q}{N}} \cdot \overline{F_{mn}} \cdot \frac{\partial P_{kl}}{\partial X_{ij}} \\ &= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} \frac{\partial P_{kl}}{\partial X_{ij}} H_{pq} ie^{iP_{kl}} e^{-i2\pi p\frac{k}{M}} e^{-i2\pi q\frac{l}{N}} \\ &\quad \cdot \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Q_{mn}\overline{F_{mn}} e^{i2\pi p\frac{m}{M}} e^{i2\pi q\frac{n}{N}} \\ &= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} \frac{\partial P_{kl}}{\partial X_{ij}} ie^{iP_{kl}} H_{pq} e^{-i2\pi p\frac{k}{M}} e^{-i2\pi q\frac{l}{N}} [\mathcal{F}^{-1}(Q\cdot F)]_{pq} \\ &= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \frac{\partial P_{kl}}{\partial X_{ij}} \cdot ie^{iP_{kl}} \cdot \mathcal{F}\{H\cdot[\mathcal{F}^{-1}(Q\cdot F)]\}_{kl}\end{aligned}$$

Finally, we can calculate the gradient of $g(X)$ as follows:

$$\begin{aligned}\nabla g &= 4\mathrm{Re}(S) \\ S &= P(ie^{iP}\cdot\mathcal{F}\{H\cdot[\mathcal{F}^{-1}(Q\cdot F)]\})\end{aligned}$$

4.5.1 Simulations

We present computational simulations of our novel Fresnel PEC method and compare it with the case without proximity effect correction where the proximity effect is applied to the phase of the phase-only hologram. We conduct the simulations on Matlab on a PC with an Intel 9900K CPU. We present the simulation results in color as our FPEC method is applicable to all wavelengths. Multi-color displays can be easily implemented by coupling three lasers with the NPA. In our simulations, we use wavelengths 630 nm, 530 nm, and 445 nm for the red, green, and the blue channels. The NPA is of size 2 mm × 2 mm and the images are placed 100 mm away from NPA.

We use the same termination condition as in the PPEC method: when the residual change r in Equation 4.36 remains below 0.001 for 10 iterations.

We compare the proposed Fresnel PEC method with two baselines. The first is the case without proximity effect correction where the proximity effect is applied to the phase of the phase-only hologram. The second is the IPEC method in Chapter 4.4.1 adapted to the Fresnel holograms. Here we replace the wavefront propagation with the Fresnel propagation approximation. We refer to this method as the *Iterative Fresnel Proximity Effect Correction* (IFPEC) method.

4.5.1.1 2D Fresnel Holograms

The qualitative results of the Fresnel PEC on 2D images are shown in Fig. 4.16. As can be seen, the proximity effect results in appreciable noise in the simulated images. The proximity effect levels in Fig. 4.16 are σ = 0.70 px, 0.80 px, 0.90 px, and 1.00 px in the four rows from top to bottom, respectively. For the two baseline methods, the higher the proximity effect, the noisier the images become. Our proposed Fresnel PEC method is able to effectively reduce the noise caused by the proximity effect even at very high proximity effect levels. Compared with the iterative Fresnel PEC method, our method is able to better correct the proximity effect and restore the images.

In addition to qualitative reconstructions of the images, we also quantify our

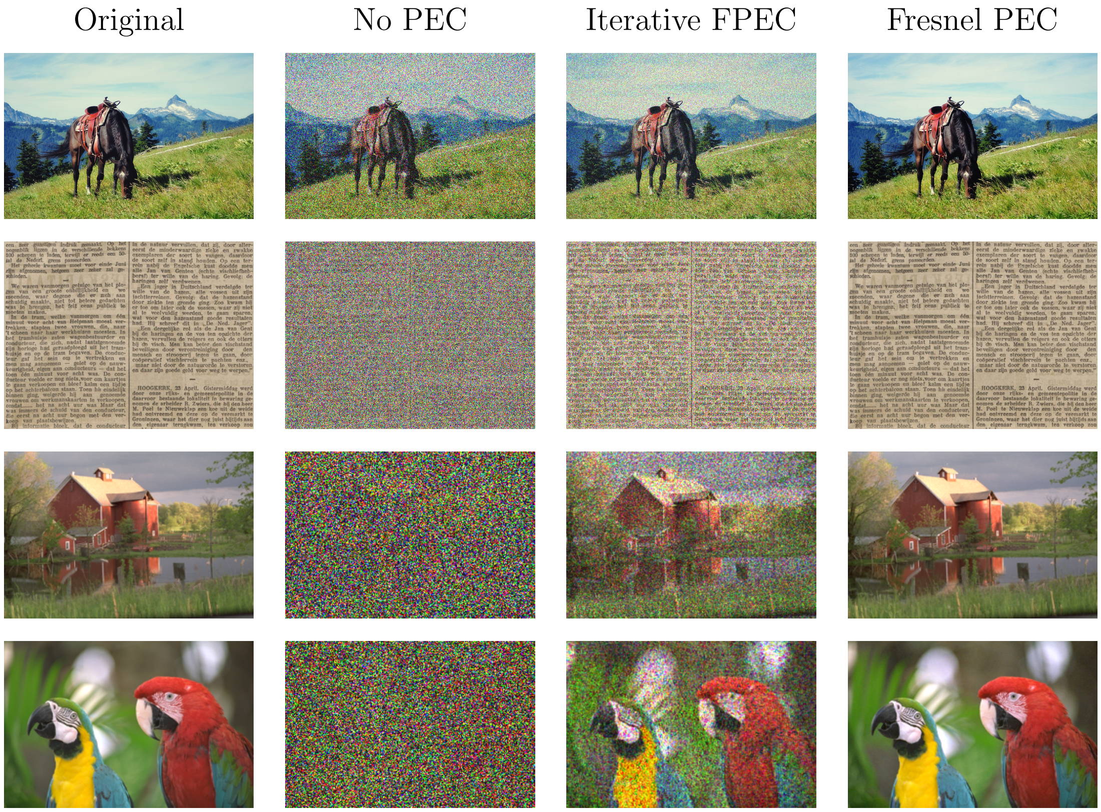
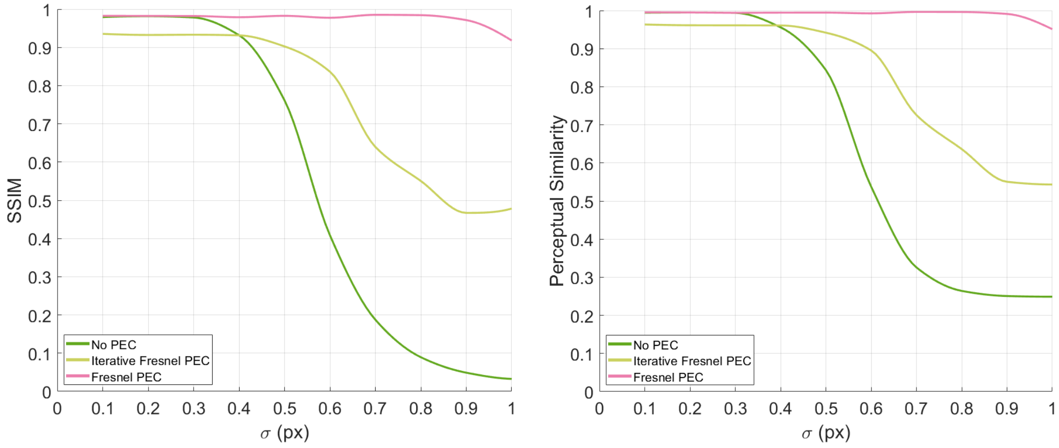


Figure 4.16: Qualitative comparison of the Fresnel PEC method with the case without proximity effect correction and the iterative Fresnel PEC method. The original image is presented in the left column for reference. The thermal proximity effect levels for the images from top to bottom are $\sigma = 0.70$ px, 0.80 px, 0.90 px and 1.00 px, respectively. At $\sigma = 0.70$ px, the proximity effect causes noise but the images are still recognizable. However, for more severe proximity effect, the noise overwhelms the images. Our Fresnel PEC method is able to correct the proximity effect and restore the images close to the original even at such high proximity effect levels. The images in the first and second rows are from the Common Objects in Context dataset [5] and the CHRONIC dataset (Smits and Faber [9]), respectively. The remaining images are from the LIVE dataset (Sheikh *et al.* [7,8]).



SSIM

Perceptual Similarity

Figure 4.17: The SSIM and perceptual similarity of the Fresnel PEC results compared to the case without proximity effect correction and the iterative Fresnel PEC (IFPEC) method. When there is no PEC, the image quality degrades quickly with the proximity effect level beyond $\sigma = 0.30$ px. The image quality of the IFPEC method starts to degrade starting from $\sigma = 0.40$ px. The Fresnel PEC method can effectively correct the proximity effect at almost all proximity effect levels tested, only to see image quality degrade slightly at extreme proximity effect level $\sigma = 1.00$ px.

results with image quality metrics. We present both the *structural similarity index* (SSIM) from Wang *et al.* [175] and the *perceptual similarity* from Zhang *et al.* [176]. More details on these metric can be seen in Section 4.4.3.2.

We present the average image metrics of the images from the LIVE dataset from Sheikh *et al.* [7, 8] at each proximity effect level from $\sigma = 0.10$ px to $\sigma = 1.00$ px. As can be seen, the proposed Fresnel PEC method can effectively correct the proximity effect at proximity effect levels from $\sigma = 0.10$ px to 0.90 px, with only a slight dip in the image quality at $\sigma = 1.00$ px. As a comparison, the case without PEC starts to see rapid image quality degradation starting from $\sigma = 0.30$ px, and the iterative Fresnel PEC method suffers from noise starting from $\sigma = 0.40$ px. The two image metrics are largely consistent with each other with the exception of the iterative Fresnel PEC method at proximity effect levels at $\sigma = 0.90$ px and 1.00 px. In this case, we think that the perceptual similarity represents the image quality better. Some examples of such mismatch between the two image metrics can be seen in the discussion in Section 4.5.2.1.

4.5.1.2 3D Holograms

The Fresnel PEC method can effectively correct 2D Fresnel holograms formed at desired depths. This can be used in combination with the fast refresh rate of the NPA to display 3D holograms. A 3D scene can be divided into a stack of 2D images at different depths. The 2D Fresnel hologram of each image can be calculated and the proximity effect corrected with the Fresnel PEC method at the desired depth. The stack of holograms can be displayed on the NPA in consecutive frames so that the 3D scene can be observed.

We present the simulations of such 3D holograms[2] in Fig. 4.18. A 3D volume is divided into a stack of images whose 2D Fresnel holograms are calculated at their respective depths with the Fresnel PEC method. Those holograms are reconstructed by propagating the wavefronts formed at the holograms to a certain distance from

[2] Volumes are from `https://klacansky.com/open-scivis-datasets` and `http://www.sci.utah.edu/download/IV3DData.html`

the NPA that corresponds to the user's focal plane. The average of those reconstructions is taken as the observation. Holograms of 3D meshes and point clouds can be computed and displayed in a similar way. The user can focus on different parts of the 3D scene. We simulate this by adjusting the focal plane accordingly.

4.5.2 Discussions

Here we discuss some limitations of our proposed FPEC method and propose several directions for future research.

4.5.2.1 Disagreement in Image Metrics

We observe in Figure 4.17 that there is some disagreement between the SSIM and the perceptual similarity (PS) at $\sigma = 0.9$ px and 1.0 px. Here we show in Figure 4.19 some examples where we think that the perceptual similarity is a more accurate representation of image quality than SSIM. These are the results of the Iterative Fresnel PEC method. We believe the images formed with the proximity effect $\sigma = 0.9$ px are better than those formed with more severe proximity effect $\sigma = 1.0$ px. Even though a substantial amount of noise is present with both proximity effect levels, the noise in images with $\sigma = 0.9$ px is more granular, making those images relatively more desirable in our opinion. The PS metrics are in agreement with the our observation while the SSIM metrics are not.

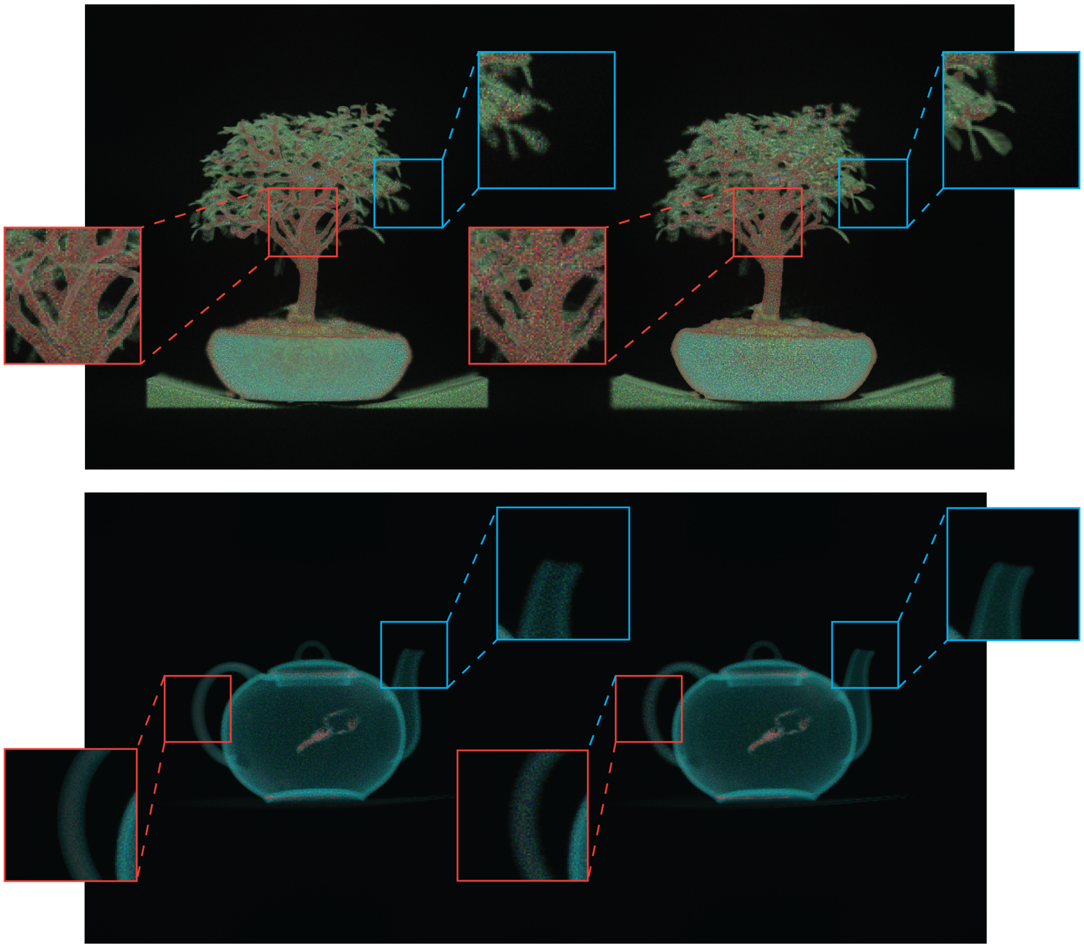
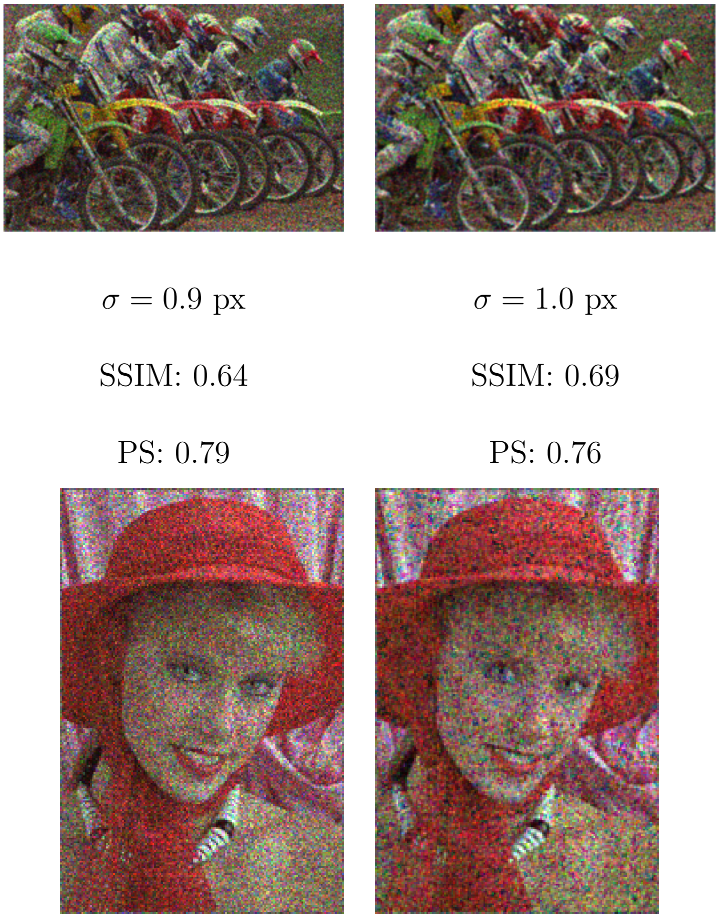


Figure 4.18: The 3D volume is divided into a stack of images at different depths. The Fresnel holograms of all the images are calculated with the Fresnel PEC method at their respective depths and displayed on the NPA in consecutive frames so the 3D scene can be observed. We simulate the observation by reconstructing the holograms at a certain distance that corresponds to the user's focal plane. By adjusting the focal plane (for example, a closer focal plane in the left images and a farther focal plane in the right images), different parts of the scene (red or blue squares) are in focus. The top and bottom volumes are divided into 232 and 253 equidistant slices, respectively. The proximity effect level used is $\sigma = 0.66$ px.



$\sigma = 0.9$ px $\sigma = 1.0$ px

SSIM: 0.64 SSIM: 0.69

PS: 0.79 PS: 0.76

$\sigma = 0.9$ px $\sigma = 1.0$ px

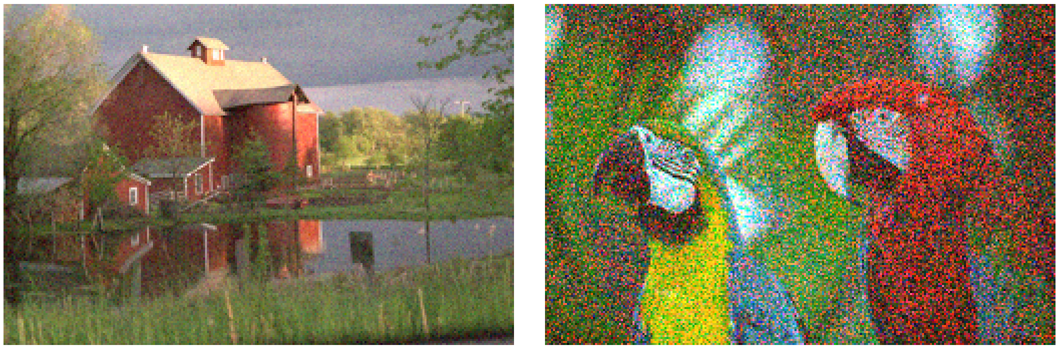
SSIM: 0.60 SSIM: 0.62

PS: 0.52 PS: 0.47

Figure 4.19: Two examples where perceptual similarity disagrees with SSIM. We observe that the results of the Iterative Fresnel PEC method at proximity effect $\sigma = 0.9$ px are of better quality than the results of the same method at proximity effect $\sigma = 1.0$ px. The perceptual similarity metric agrees with this observation while the SSIM disagrees.

4.5.2.2 Larger Proximity Effect

We have shown in our simulations that the proposed FPEC method is effective within the tested proximity effect level range ($\sigma = 0.10$ px to 1.00 px). While this is a large range within which our NPA design lies, it is not difficult to imagine cases where a more severe proximity effect may arise. For example, as the pixels on the NPA are more densely packed, it can be expected that each pixel is affected more by nearby pixels. In Figure 4.17, we see that the image quality of the FPEC results start to decrease at $\sigma = 1.00$ px. At higher proximity effect levels, the image quality would drop more. At proximity effect levels $\sigma = 1.50$ px and 2.00 px, the average SSIMs of the results of FPEC are 0.57 and 0.34 while the average perceptual similarities are 0.69 and 0.48. Examples of the FPEC method at higher proximity effect levels can be seen in Figure 4.20.



$\sigma = 1.50$ px

$\sigma = 2.00$ px

Figure 4.20: The FPEC method at higher proximity effect levels where much noise is observed.

4.6 Conclusion

Nanophotonic phased arrays have several advantages in its use as a holographic display. The integrated light source enables small-factor integration and the fast refresh rate is ideal for dynamic content. The use of NPA in a near-eye display has been demonstrated in Notaros *et al.* [116] and other works. However, the use of a large scale dynamically modulated NPA as a holographic display has not been achieved. We propose two proximity effect correction methods to address the thermal proximity effect issue of Fourier holograms in NPAs. The PPEC method is the obvious choice as it is faster and more effective in correcting proximity effect at all proximity effect levels. The iterative proximity effect correction is effective at low proximity effect levels and the correction effectiveness can be improved with TDM or padding techniques. The IPEC method is also readily applicable to multiplane scenes. However, more efficient deconvolution methods with non-negative constraints are required for the IPEC method to be practical. We also propose the Fresnel proximity effect correction method that can effectively correct the thermal proximity effect for 2D Fresnel holograms on the NPAs when realistic levels of proximity effect are present. We show that we can use the FPEC method to display 3D holograms so that different parts of the 3D scene can be focused on. We believe that our approaches make a meaningful contribution to the emerging field of NPAs. We hope that it brings this highly interesting avenue for holographic displays closer to practical use.

Chapter 5: Fast Holographic Volume Rendering

5.1 Overview

Holographic displays and computer-generated holography (CGH) have been an active area of research in recent years. Holographic displays recreate the optical wave of the light coming from a scene, naturally addressing the vergence-accommodation conflict faced by the conventional stereoscopic displays with a single or a limited number of focal planes. Varifocal and light-field displays also offer a wider range of focal distances. However, the former usually involves moving components while the spatial and angular resolutions of the latter are limited by optical diffraction as the pixel density increases. On the contrary, the quality of CGH improves as the pixel density increases. Recent advancement has also proposed solutions to problems faced by holographic displays such as limited field of view (FOV) and miniaturization.

CGH has largely focused on 2D images or 3D scenes in the form of meshes or point clouds. The Fourier transform is a common and efficient method to calculate the hologram of a 2D image. Calculating the hologram of a 3D scene is computationally intensive and usually involves converting the scene into a collection of point light sources directly visible from the hologram and calculating the hologram following

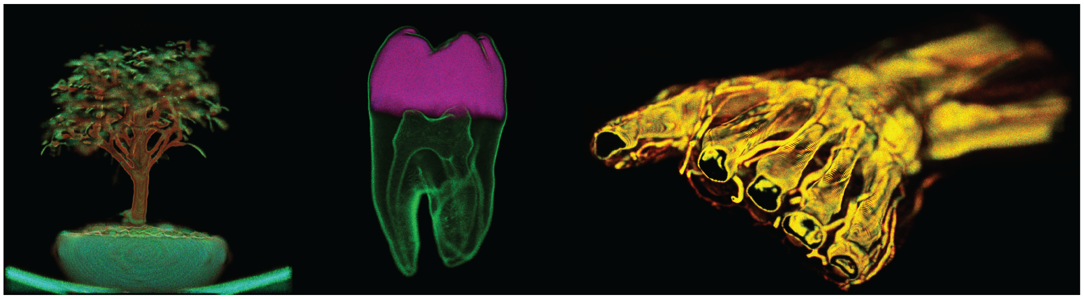


Figure 5.1: We propose an efficient way to compute the hologram of a 3D volume. We use the raycasting direct volume rendering method to sample a collection of point light sources and use the convolution theorem to accelerate hologram computation. The digital reconstructions of several such holograms are shown. Holographic volume rendering is ideal for visualizing volumetric data such as 3D medical imaging data or physics simulations on a holographic display.

the Huygens-Fresnel principle. Modern computational hardware and acceleration methods have enabled real-time calculation of holograms from 3D meshes.

Volumetric data is another type of data suitable for visualization. Volumetric data from medical imaging such as computed tomography (CT) and magnetic resonance imaging (MRI) can be rendered to reveal critical information about the organs and tissues. Volumes from physics simulations are also used in the realistic rendering of fog, smoke and, so on. Volume visualization can benefit from the advantages of holographic displays. For simplicity, we shall refer to the holograms of volumetric data as *volumetric holograms* and the practice of generating volumetric holograms and displaying them on holographic displays as *holographic volume rendering.* A few challenges have prevented CGH from being applied to volumetric data. First, calculating the volumetric hologram is an order-of-magnitude more complicated than calculating that of a 3D mesh, as every voxel contributes to the perception of the volume instead of only those closest to the hologram. In the latest work on this topic, Lu and Sakamoto [179] find the color of each voxel and calculate the hologram using the positions and colors of those voxels. Every voxel contributes to each hologram pixel and the computation takes more than three hours for a 1280×768 hologram. Second, the small FOV is a problem suffered by all digital holographic displays. This is caused by the limited diffraction angle of the currently existing light modulators. Various methods to increase the FOV have been proposed for different holographic displays such as in Maimone *et al.* [62], Shi *et al.* [1] and Jang *et al.* [149]. But the methods to increase the FOV need to be considered in accordance with the hologram computation, as is with the case of volumetric holograms. Finally, the choice of the illuminating light source is also critical to the observed image of a holographic display. A light source with a narrow spectrum such as a laser can produce images of high resolution but also leads to heavy speckles while a light source with a wider spectrum such as an LED reduces speckles at the cost of lower resolution.

In this chapter, we address each of the three challenges faced by holographic volume rendering mentioned above. To address the first challenge, we propose an accelerated method to compute volumetric holograms. This is primarily achieved by treating the summation in the Huygens-Fresnel principle as a convolution and accelerate it with the convolution theorem. Maimone *et al.* [62] recently noted that convolution can be used to calculate the hologram of a 2D image. We take this idea one step further and combine it with an elegant way to sample the volume based on raycasting direct volume rendering (DVR) to achieve significant acceleration compared to the brute-force method. To address the second challenge, we propose a holographic display configuration that increases the FOV. Special attention is paid to making sure the display is compatible with the accelerated volumetric hologram computation method. And finally, we investigate what kind of light source strikes the balance between speckles and image resolution for holographic volume rendering.

To validate the holograms computed using our method, we present the digital reconstruction of the volumetric holograms. We also realistically simulate the observed image on our display configuration with different light sources to identify the light source that best balances the speckles and resolution. The light source customization is implemented on the optics bench and validated experimentally.

In summary, we improve the holographic volume rendering in following ways:

- We develop an accelerated hologram computation method for volumetric data. Our method samples the volume based on raycasting direct volume rendering and uses the convolution theorem to significantly improve the computation speed.

- We propose a configuration that increases the FOV of the holographic display while maintaining the compatibility with the accelerated hologram computation method.

- We identify the appropriate light source for holographic volume rendering with realistic simulation.

5.2 Related Work

In this section, we give an overview of the existing literature on several topics related to this work. These include volume visualization and computer-generated holography based on 2D images, 3D meshes, and volumetric data.

5.2.1 Volume Visualization

Traditional volume rendering methods can be classified into two groups: surface fitting and direct volume rendering. We briefly cover some of the basics and the advantages and disadvantages of both groups. For a more thorough review, we refer readers to a survey by Elvins [180].

Surface fitting methods find a surface or some other geometric element that is representative of a meaningful object in the volume and visualize it. One such method is contour-connecting [181,182] where a contour matching a specified threshold value is selected on each slice of the volume and the contours from neighboring slices are connected to form a mesh. This mesh is then rendered. The marching cubes algorithm [183] is another famous method in the surface fitting group. In this method, the voxels whose values straddle a specified threshold are converted into a mesh to be rendered. Improvements have been made to reduce the ambiguity in how the corners of the voxels are connected to form triangles [184, 185].

While the surface fitting methods are fast and able to provide good visualization for medical data, one major disadvantage is that they depend on a pre-specified threshold value and only the voxels matching that value are visualized. Direct volume rendering, on the other hand, can make use of all the information in the volume. Raycasting [186–188] is the most widely-used direct volume rendering technique. In raycasting, rays are shot from each image pixel into the volume. The color and opacity associated with the ray are composited at each sampling point along the ray. The point light source sampling process in our proposed hologram calculation methods is also based on raycasting. A detailed description and our modification are presented in Section 5.3.1. Another direct volume rendering method is splatting, where all the voxels in the volume are traversed with the voxels closer to the image plane being processed before those farther away. When processing a voxel, a region around the projection of the voxel onto the image plane is determined where the contribution of this voxel to the image is defined. Different ways for voxels to contribute to the image (aka footprints) have been proposed [189, 190]. The colors and opacities of the voxels are blended into the image in their respective contribution regions.

Recently, a generative model for volume rendering has been proposed by Berger *et al.* [191] where a generative adversarial network (GAN) can interactively help the user craft the transfer function. With the advancement of graphics processors and raytracing techniques, photorealism can be achieved with physics-based volume rendering [192].

5.2.2 Computer-Generated Holography

Various methods have been proposed to calculate the hologram in CGH. For 2D images, Lohmann *et al.* [151] show that a binary representation of the Fourier hologram can achieve good reconstruction quality. Iterative multi-plane methods (Makowski *et al.* [193]) have been proposed to allow images to be reconstructed at different depth locations.

To generate holograms based on 3D objects, the object is often converted to a collection of point light sources and the complex wavefronts from all the light sources are summed up. Directly computing the summation from every point is computationally intensive. Lucente [153] uses a look-up table corresponding to the interference contribution from points in each location in the image space and achieves a 43× speedup. Masuda *et al.* [194] build graphics hardware specifically for calculating holograms. Matsushima *et al.* [195] calculate the approximated hologram by reusing the distance between a light source and a hologram pixel for nearby hologram pixels. The 3D scene can be seen as a collection of planar light sources as well. Matsushima *et al.* [196] create high-definition holograms of complicated 3D objects using a polygon-based CGH method to simplify the calculation.

Stereograms are small regions of a hologram. They each record the diffraction pattern of a portion of the scene instead of the entire scene and are used in Lucente and Galyean [155], Xu *et al.* [156], and Shi *et al.* [1] to reduce the amount of computation.

In addition, Ziegler *et al.* [159] develop a conversion technique between the light-field representation and the holographic representation of a scene. Padmanaban *et al.* [146] use this conversion to develop the overlap-add stereogram (OLAS) algorithm to overcome the trade-off between spatial and angular resolution in traditional light-field displays. Ways to add occlusion or change lighting in a hologram have also been explored in Ziegler *et al.* [160].

CGH based on volumetric data is a less-studied area. Sakamoto *et al.* [197] calculate the hologram of a volume by propagating the slices one by one to the hologram plane using Fresnel propagation. Lu and Sakamoto [198] use two methods to calculate holograms from a volume. In the first, maximum intensity projection is used to find a surface of point light sources from which a Fresnel hologram is calculated. In the second, the volume is converted into a polygonal model using the marching cubes algorithm and the hologram is calculated based on the polygonal model. Both of these methods have the same disadvantage as the surface-fitting method, that they only visualize a surface in the volume instead of its entirety. In their later work, Lu and Sakamoto [179] find the color of each voxel considering the intensity decay as light travels through other voxels. While this method does not require the hologram to be parallel to one of the volume's faces like in [197], it is based on brute-force summation and may take hours to compute. Our proposed method significantly accelerates the calculation, offering over 2000× speedup.

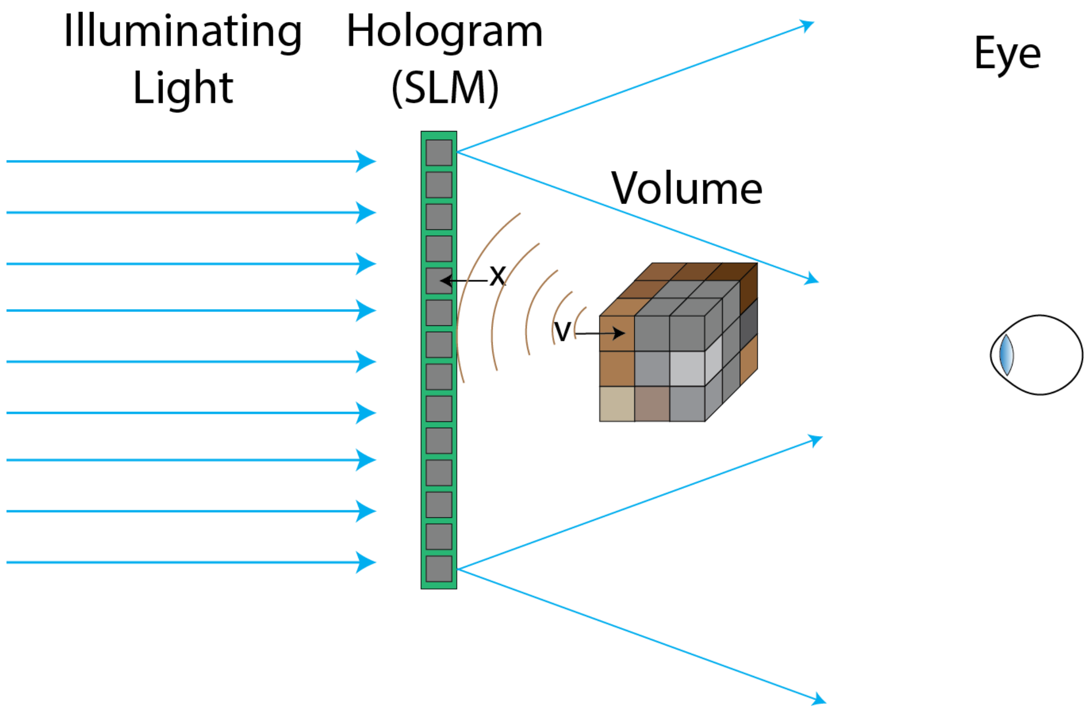


Figure 5.2: The general placement of the volume, the hologram, the illuminating light (planar wave), and the observer. The hologram is calculated as the summation of the wavefronts from the collection of spherical point light sources in the volume.

5.3 Algorithm and Display

We base our holographic volume rendering on the heuristics of the Huygens-Fresnel principle, which treats the object (the volume in our case) as a collection of spherical point light sources. Let $\boldsymbol{x}$ be a pixel on the hologram, $\boldsymbol{v}$ be a point light source and $r(\boldsymbol{x}, \boldsymbol{v})$ be the distance between them. $A(\boldsymbol{v})$ and $\phi(\boldsymbol{v})$ are the amplitude and initial phase of the light emitted from $\boldsymbol{v}$, λ is the wavelength, and $k = \frac{2\pi}{\lambda}$ is the wave number.

With the placement of the volume, the hologram, the illuminating light, and the observer arranged in Figure 5.2, the electric field[1] $E(\boldsymbol{x})$ at pixel $\boldsymbol{x}$ can be expressed in Equation 5.1.

$$E(\boldsymbol{x}) = \sum_{\boldsymbol{v}} \frac{A(\boldsymbol{v})}{r(\boldsymbol{x}, \boldsymbol{v})} e^{i(kr(\boldsymbol{x},\boldsymbol{v})+\phi(\boldsymbol{v}))} \tag{5.1}$$

The sign of the phase term $kr(\boldsymbol{x}, \boldsymbol{v})$ in Equation 5.1 is irrelevant as long as a consistent coordinate system is used throughout the calculation.

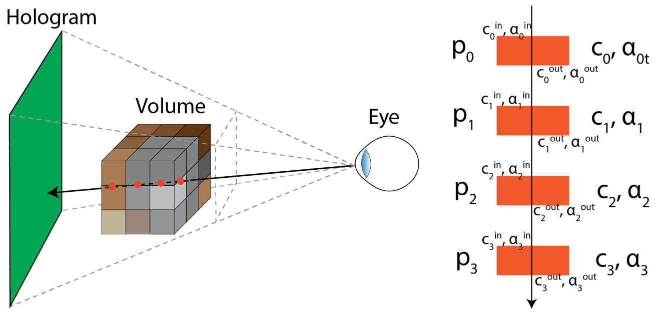
Given the Huygen-Fresnel principle-based hologram computation, steps must be taken to generate the point light sources and find out their amplitude and initial phase values before the hologram computation can proceed. We cover below in Section 5.3.1 how we use the direct volume rendering to sample those point light sources. In Section 5.3.2, we describe how to convert the summation in Equation 5.1 to a convolution to significantly accelerate the hologram calculation.

5.3.1 Direct Volume Rendering Sampling

We use the raycasting direct volume rendering and front-to-back compositing to generate the position as well as the color of the point light sources. An illustration can be seen in Figure 5.3(a).

Rays are cast from the observer's eye position into the volume. Compositing is performed inside the volume along the rays' directions (front to back). At compositing step i (illustrated as p_0 to p_3 in Figure 5.3(b)), a pre-specified transfer function

[1] The magnetic part of the electromagnetic wave is usually ignored in computer-generated holography. Detail can be seen in [123].


Hologram
Volume
Eye
p_0
c_0^{in}, α_0^{in}
c_0, α_{0t}
$c_0^{out}, \alpha_0^{out}$
p_1
c_1^{in}, α_1^{in}
c_1, α_1
$c_1^{out}, \alpha_1^{out}$
p_2
c_2^{in}, α_2^{in}
c_2, α_2
$c_2^{out}, \alpha_2^{out}$
p_3
c_3^{in}, α_3^{in}
c_3, α_3
$c_3^{out}, \alpha_3^{out}$

(a) (b)

Figure 5.3: (a) illustrates the direct volume rendering method used for sampling the point light sources. Front-to-back compositing is used in our proposed method (b). The position and color at each compositing step in the direct volume rendering are used for the hologram calculation.

is used to derive a color c_i and opacity value α_i associated with the volume scalar value at location p_i. Assuming the composited color and opacity before compositing step i are c_i^{in} and α_i^{in}, the composited color and opacity after the compositing step i are c_i^{out} and α_i^{out}. Their relations are as follows.

$$c_i^{\text{out}} = c_i^{\text{in}} + (1 - \alpha_i^{\text{in}})\alpha_i c_i \tag{5.2}$$

$$\alpha_i^{\text{out}} = \alpha_i^{\text{in}} + (1 - \alpha_i^{\text{in}})\alpha_i \tag{5.3}$$

$$c_{i+1}^{\text{in}} = c_i^{\text{out}} \tag{5.4}$$

$$\alpha_{i+1}^{\text{in}} = \alpha_i^{\text{out}} \tag{5.5}$$

The compositing usually stops as the ray exits the volume, or when the termination condition is satisfied (such as the composited opacity is larger than a certain value). In traditional volume rendering, we are interested in the final composited color c_{n-1}^{out}, assuming n compositing steps. As shown in Equation 5.6, the final composited color is a linear combination of the color at each compositing step.

$$c^{\text{out}}_{n-1} = \sum_{i=0}^{n-1} T_i \alpha_i c_i \tag{5.6}$$

$$T_i = T_{i-1}(1 - \alpha_{i-1}) = \prod_{j=1}^{i}(1 - \alpha_{j-1}) \tag{5.7}$$

The pseudocode for the raycasting direct volume rendering is presented in Algorithm 3. The pseudocode shows the compositing steps for the ray r with its origin at $\boldsymbol{o}^r$, direction $\boldsymbol{d}^r$ and step size s^r. t^r_0 indicates the distance from the ray's origin at which the compositing starts. D is the maximum number of compositing steps. $\mathcal{C}$ and $\mathcal{A}$ are the color and opacity textures derived from the volume and its transfer function. We use "Sampler3D" to sample $\mathcal{C}$ and $\mathcal{A}$ and find the color and opacity at each compositing step. The outputs are the color contribution C^r_j and the position $\boldsymbol{p}^r_j$ at each compositing step.

For our holographic volume rendering, the position p_i and the composited color $T_i \alpha_i c_i$ of each composition step of each ray are used to calculate the hologram in Equation 5.1. Each ray is associated with a random initial phase and it is used for all the point light sources generated on this ray. We refer to the brute-force way of calculating the hologram according to Equation 5.1 as the *direct* method.

5.3.2 Acceleration using Convolution Theorem

Assuming a total of $R_x \times R_y$ rays are cast and each ray has a maximum of D compositing steps, direct volume rendering generates $O(DR_xR_y)$ point light sources. Given the hologram dimension of $H_x \times H_y$, the complexity of the direct calculation method is $O(DR_xR_yH_xH_y)$. The pseudocode of the direct calculation method is

Algorithm 3: Raycasting Direct Volume Rendering Sampling

```
input  : $\boldsymbol{o}^r$, $\boldsymbol{d}^r$, $s^r$, $t_0^r$, $D$, $\mathcal{C}$, $\mathcal{A}$
output: $C_j^r$, $p_j^r$, $j = 0, 1, ...D - 1$
$\boldsymbol{p} = \boldsymbol{o}^r + t_0^r \cdot \boldsymbol{d}^r$;
$\alpha^r = 0$;
for $j \leftarrow 0$ to $D - 1$ do
    $\boldsymbol{p}_j^r = \boldsymbol{p}$;
    $c_p = \text{Sampler3D}(\mathcal{C}, p)$;
    $\alpha_p = \text{Sampler3D}(\mathcal{A}, p)$;
    $C_j^r = (1 - \alpha^r)\alpha_p c_p$;
    $\alpha^r = \max(\alpha^r + (1 - \alpha^r)\alpha_p, 1)$;
    $\boldsymbol{p}$=$\boldsymbol{p}$ + $s^r \cdot \boldsymbol{d}^r$;
end
```

shown in Algorithm 4. ϕ_0^r is the random initial phase assigned to each ray r.

The three color channels of the hologram are calculated separately with different wavelengths λ. The function $\mathrm{HgPos}(x, y)$ finds the 3D position of the hologram pixel (x, y) and the function $\mathrm{dist}(\boldsymbol{p}, \boldsymbol{q})$ finds the distance between the two points $\boldsymbol{p}$ and $\boldsymbol{q}$. The output is the complex hologram $E(x, y)$. As we will show in Section 5.5, this complexity leads to hologram calculation times that span hours with holograms that match the resolution of state-of-the-art SLMs. We next present a method to significantly accelerate this calculation.

Algorithm 4: Direct Holographic Volume Rendering

```
input  : $C_j^r$, $\boldsymbol{p}_j^r$, $\phi_0^r$, $R_x$, $R_y$, $D$, $\lambda$, $H_x$, $H_y$
output: $E(x, y)$
for $x \leftarrow 0$ to $H_x - 1$ do
    for $y \leftarrow 0$ to $H_y - 1$ do
        $E(x, y) = 0$;
        $\boldsymbol{p}_{x,y} = \mathrm{HgPos}(x, y)$;
        for $r \leftarrow 0$ to $R_x \times R_y - 1$ do
            for $j \leftarrow 0$ to $D$ do
                $E(x, y) = E(x, y) + \sqrt{C_j^r} \frac{1}{\mathrm{dist}(\boldsymbol{p}_{x,y}, \boldsymbol{p}_j^r)} e^{i(\frac{2\pi}{\lambda} \mathrm{dist}(\boldsymbol{p}_{x,y}, \boldsymbol{p}_j^r) + \phi_0^r)}$
            end
        end
    end
end
```

Equation 5.1 can be written as the following.

$$E(\boldsymbol{x}) = \sum_{\boldsymbol{v}} A(\boldsymbol{v}) e^{i\phi(\boldsymbol{v})} \frac{e^{ikr(\boldsymbol{x},\boldsymbol{v})}}{r(\boldsymbol{x},\boldsymbol{v})} \tag{5.8}$$

If the point light sources are sampled on a plane parallel to the hologram, $E(\boldsymbol{x})$ can be seen as the summation of a series of integrals across the collection of sampling distances z.

$$E(x, y) = \sum_{z} E_z(x, y) \tag{5.9}$$

$$E_z(x, y) = \iint_{\xi,\eta} A_z(\xi, \eta) e^{i\phi_z(\xi,\eta)} \frac{e^{ikr_z(x-\xi,y-\eta)}}{r_z(x-\xi, y-\eta)} d\xi d\eta \tag{5.10}$$

$\boldsymbol{x} = (x, y)$ refers to the hologram pixel position. $A_z(\xi, \eta)$ and $\phi_z(\xi, \eta)$ are the amplitude and initial phase of the sampled point light source at position (ξ, η) at sampling distance z. $r_z(m, n) = \sqrt{z^2 + m^2 + n^2}$. The integral in Equation 5.10 can be seen as a convolution shown below and this convolution can be calculated using the convolution theorem for a significant speedup. $*$ is the convolution operator.

$$E_z = C_z * K_z \tag{5.11}$$

$$C_z(\xi, \eta) = A_z(\xi, \eta) e^{i\phi_z(\xi,\eta)} \tag{5.12}$$

$$K_z(m, n) = \frac{e^{ikr_z(m,n)}}{r_z(m, n)} \tag{5.13}$$

To take advantage of this speedup, two changes need to be made to the direct volume rendering sampling process. First, the compositing step sizes along each ray must be adjusted so that the jth compositing steps for all the rays are on the same plane (referred to as *compositing plane*) parallel to the hologram and of distance $z = d_j$ away from the hologram. As a result, the compositing step sizes are non-uniform for the rays, and opacity correction is needed. Second, a resampling step is needed to make sure that the amplitude $A_z(\xi, \eta)$ and phase $\phi_z(\xi, \eta)$ at each compositing plane have the same sampling pitch as the hologram.

We can use the convolution theorem and fast Fourier transform to calculate the convolution in Equation 5.11 efficiently. $\mathcal{F}$ in Equation 5.14 is the Fourier transform. We refer to this way of calculation the *convolution theorem* (CT) method.

$$\mathcal{F}(E_z) = \mathcal{F}(C_z) \cdot \mathcal{F}(K_z) \tag{5.14}$$

The pseudocode for calculating the hologram using the CT method is presented in Algorithm 5. The function Resample$(C_j^r, \phi_0^r, \rho, H_x, H_y)$ resamples the C_j^r and ϕ_0^r of all rays at compositing step j into complex C_{d_j} that has the same spacing (ρ) as the hologram. The function ConvolutionKernel calculates the convolution kernel K_{d_j} based on the distance d_j between the jth compositing plane, the wavelength and the hologram. Both these functions have complexity $O(H_x H_y)$. The total complexity is $O(DH_xH_y\log(H_xH_y))$ assuming that the fast Fourier transform is used and that both C_{d_j} and K_{d_j} have dimensions in the order of $H_x \times H_y$.

5.4 Increase the FOV

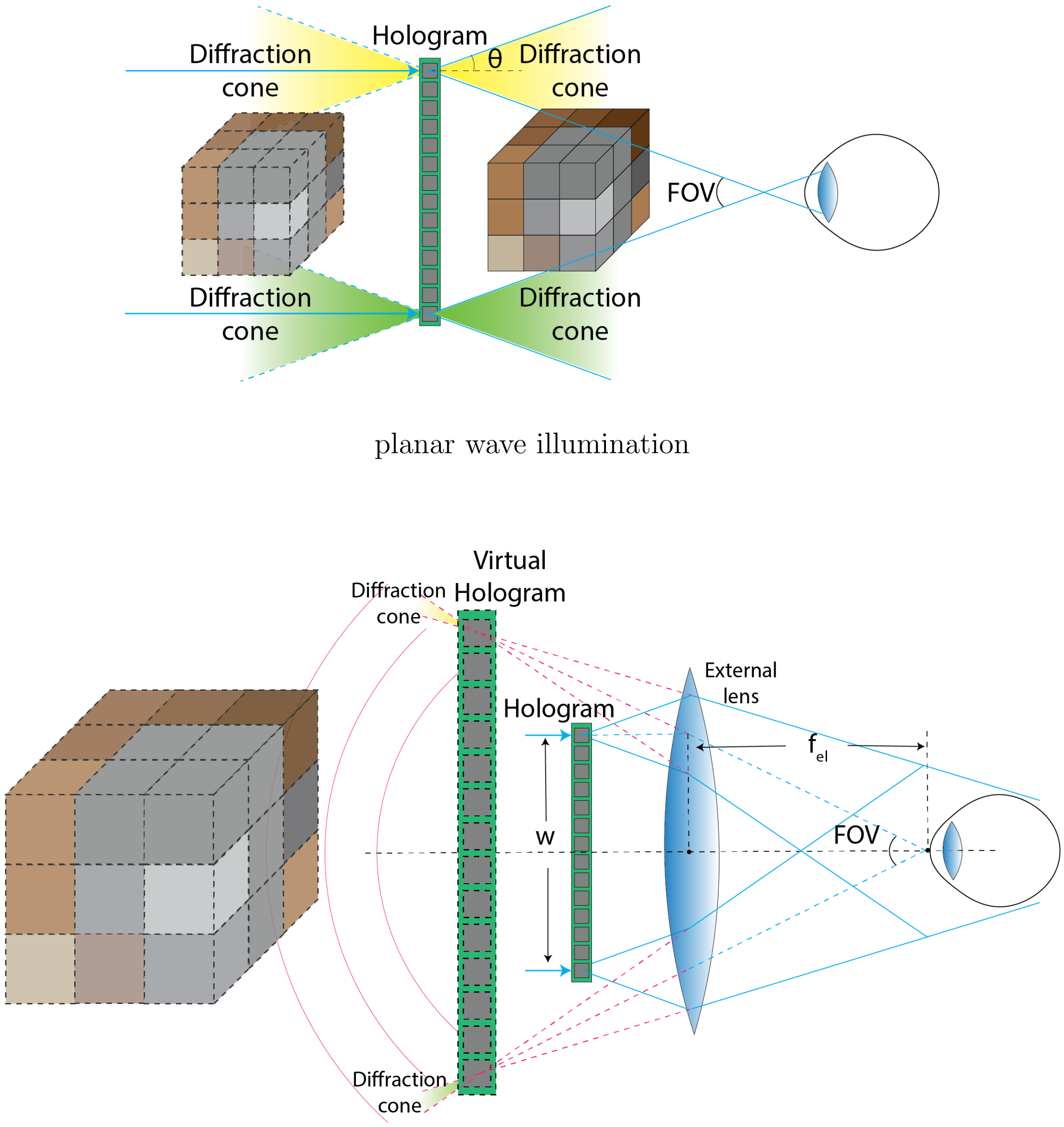
If we are to illuminate the SLM with a planar wave as illustrated in Figure 5.2, the FOV the observer can expect is dependent on the maximum diffraction angle

Algorithm 5: Convolution Theorem Holographic Volume Rendering

```
input  : $C^r_j$, $\phi^r_0$, $D$, $\lambda$, $\rho$, $H_x$, $H_y$
output: $E$
$E = 0$;
for $j \leftarrow 0$ to $D - 1$ do
    $C_{d_j} = \text{Resample}(C^r_j, \phi^r_0, \rho, H_x, H_y)$;
    $K_{d_j} = \text{ConvolutionKernel}(d_j, \lambda, \rho, H_x, H_y)$;
    $E = E + \mathcal{F}^{-1}(\mathcal{F}(C_{d_j}) \cdot \mathcal{F}(K_{d_j}))$;
end
```

θ assuming that the illumination's incident angle is 0°. In this case, $\theta = \text{asin}(\frac{\lambda}{2\rho})$, where ρ is the SLM's pixel pitch and λ is the wavelength. The FOV is therefore 2θ, as illustrated in Figure 5.4. Assuming one of the state-of-the-art SLMs' pixel pitch of 3.74 µm and a red laser wavelength of 636 nm, the FOV is 9.76°.

A common method to increase the FOV is to use the spherical wave illumination. Sakamoto *et al.* [198, 199] and Shi *et al.* [1] place a lens behind the hologram to form the spherical wave illumination. The hologram is within the focal length f_{el} of the external lens, as illustrated in Figure 5.4 bottom. When the planar wave illuminates the hologram and converges around the focal point of the external lens, it can be seen as a spherical wave centered at the focal point of the external lens illuminating a magnified virtual hologram. In this configuration, the FOV is enlarged to $2\text{atan}(\frac{w}{2f})$ with the viewer's eye placed at the focal point of the external lens. We refer to this configuration as *spherical wave illumination*.



planar wave illumination

spherical wave illumination

Figure 5.4: The FOV under planar wave is $2\text{asin}(\frac{\lambda}{2\rho})$. Spherical wave illumination can be used to increase the FOV by placing a lens closely behind the hologram. While the FOV can be increased to $2\text{atan}(\frac{w}{2f_{\text{el}}})$ where w is the side length of the hologram and f_{el} is the focal length of the external lens. This configuration is not compatible with our proposed method of hologram computation because the diffractions are oriented in different directions.

However, the spherical wave illumination is not compatible with our proposed convolution theorem method of calculating the hologram. This is because when calculating the hologram using the Huygens-Fresnel principle, the point light sources $\boldsymbol{v}$ that contribute to the hologram position $\boldsymbol{x}$ must lie within the diffraction cone of the hologram at $\boldsymbol{x}$ under the specific illumination, or it would lead to aliasing noise. This is true for both the real image light sources to the right of the hologram and the virtual image light sources to the left of the hologram in Figure 5.4. The aliasing artifacts of sampling outside the diffraction cones have been discussed in [1]. Under planar illumination, the diffraction cones are the same everywhere on the hologram, therefore allowing the integral in Equation 5.10 to be treated as the convolution in Equation 5.11. However, under spherical wave illumination, the diffraction cones are different at different positions on the hologram. An example can be seen in Figure 5.4 bottom where the diffraction cones of the top and bottom pixels on the virtual SLM are oriented in different directions. Diverging spherical wave illumination is also used in Maimone *et al.* [62] to increase the FOV and is incompatible with our proposed method for similar reasons.

Alternatively, we use a different display configuration to improve the FOV. As illustrated in Figure 5.5, the hologram is farther away from the external lens than its focal length f_{el}. The viewer's eye is placed at or to the right of the focal point on the other side of the external lens. Similar to the spherical wave illumination, the FOV is $2\mathrm{atan}(\frac{w}{2f_{\mathrm{el}}})$. The diffraction cones are the same across the hologram, allowing us to use the convolution theorem method. The volume is placed within the focal length from the external lens on the same side as the hologram. Therefore,

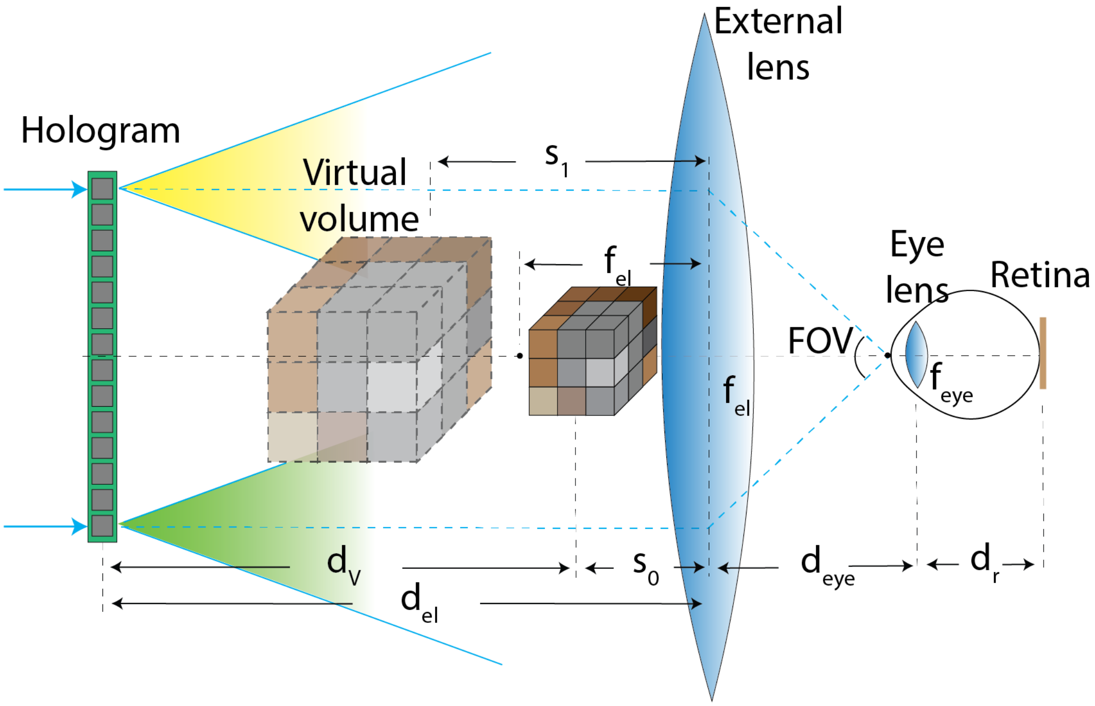


Figure 5.5: We increase the FOV while also making sure the configuration is compatible with our accelerated hologram computation method by placing the external lens farther away from the hologram than its focal length. The volume is placed within the focal length from the external lens. The magnified virtual image of the volume (bound in dashed lines) is observed.

a magnified virtual image of the volume is observed by the viewer.

In addition to the compatibility with our proposed calculation method, another reason that makes the volume magnification method work better in practice is that there is a limit on how close the external lens can be to the hologram. Due to the reflective nature of most of the commercially available high-resolution SLMs in use, a beam splitter (often cube-shaped) is usually inserted between the SLM and the external lens. The external lens itself also has a thickness that cannot be ignored.

As a result, the focal length of the external lens in the spherical wave illumination method is larger than that can be used in our configuration.

We have used the spherical lens to represent the external lens in our illustrations. It is usually difficult to manufacture spherical lenses with a focal length smaller than 10 mm and the smaller the focal length, the thicker the lens will be. A growing trend in optical displays is to use holographic optical elements (HOEs). A holographic plate can be created which mimics the function of a regular lens for the wavelengths of light used in the display. This way, smaller focal lengths can be achieved.

Finally, it is important to note that the FOV discussed in this section refers to the theoretical maximum FOV in which light diffracted from the hologram enters the eye. The FOV spanned by the scene also depends on the size of the scene and the distance between the viewer and the image of the scene. We discuss this in detail in Section 5.6.2.

5.5 Results

In this section, we present both the simulated results of our proposed holographic volume rendering and the realistic simulation of the observed image with different types of light sources. First, the digital reconstruction of the holograms is carried out in Section 5.5.1 to validate our proposed convolution theorem method. Realistic simulations based on our proposed display configuration with different light sources are shown in Section 5.5.2. Finally, we report and analyze the speedup of our proposed hologram calculation method in Section 5.5.3.

5.5.1 Digital Reconstruction

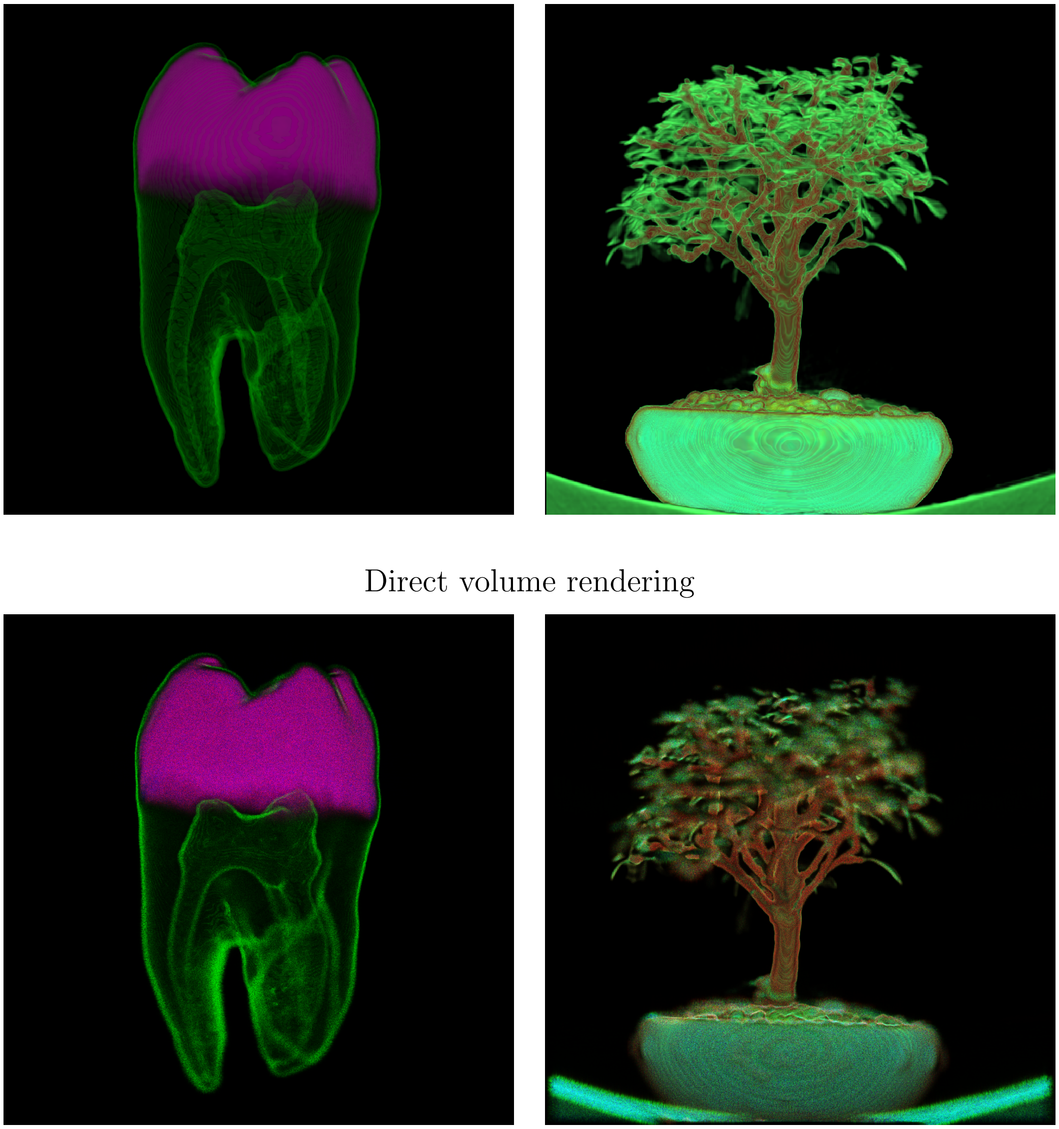
First, we show some of the volumes[2] used in our tests and the digital reconstructions from their holograms in Figure 5.6. These datasets are visualized with raycasting direct volume rendering and front-to-back compositing. 1024 × 1024 rays are cast ($R_x = R_y = 1024$). The holograms' dimensions are 2048 × 2048 ($H_x = H_y = 2048$). The pixel pitch used is $\rho = 3.74\,\mu\text{m}$. The digital reconstruction is done by propagating the complex wavefront from the position of the hologram back to the position of the volume. The propagation distance determines what depth in the volume is in focus. The propagation of the hologram is performed with the wavelengths the hologram is computed with. The speckles are reduced by taking the average of the digital reconstructions of holograms calculated with different initial phase. Realistic simulations of the observed images are presented in Section 5.5.2.

It can be seen that the holograms calculated using the convolution theorem method can successfully be digitally reconstructed to visualize the volume, indicating that the convolution theorem method correctly encodes the volume into a hologram.

5.5.2 Realistic Simulation

While the digital reconstruction is useful in showing that the generated hologram correctly encodes the spatial and color information of the volume, it does not

[2] We tested on six volumetric datasets selected from https://klacansky.com/open-scivis-datasets and http://www.sci.utah.edu/download/IV3DData.html



Direct volume rendering

Convolution theorem hologram reconstruction

Figure 5.6: Direct volume rendering results of two volumetric datasets and the digital reconstruction of the holograms calculated with the convolution theorem methods. We use orthogonal raycasting in the direct volume rendering because it most resembles the digital reconstruction by wavefront propagation.

represent what the viewer actually sees. The configuration of the display and the light source used to illuminate the hologram both affect the viewer's observation. We take these factors into consideration in a realistic simulation of the viewer's observation. We also show simulations of what the viewer would see when he or she focuses at different depths of the volume or changes the perspective.

5.5.2.1 Configuration of the Display

The configuration of the display is shown in Figure 5.5. The volume is placed d_{V} to the right of the SLM. Without specification, we refer to the position of the volume as the position of the center of the volume. The external lens is placed d_{el} to the right of the SLM, which makes it $s_0 = d_{\mathrm{el}} - d_{\mathrm{V}}$ to the right of the volume. As mentioned before, $s_0 < f_{\mathrm{el}}$ so a magnified virtual image of the volume can be observed. The observer's eye is placed to the right of the external lens. The lens of the eye which we will refer to as the *eye lens* is d_{eye} away from the external lens. The retina is d_r away from the eye lens. Depending on the depth in the virtual volume the observer is focusing at, the focal length f_{eye} of the lens in the eye is chosen so that $\frac{1}{f_{\mathrm{eye}}} = \frac{1}{s_1 + d_{\mathrm{eye}}} + \frac{1}{d_r}$ where $s_{\mathrm{eye}} = s_1 + d_{\mathrm{eye}}$ is the focus distance of the eye lens. We can adjust d_{eye} so s_{eye} is a comfortable distance for the user to focus at.

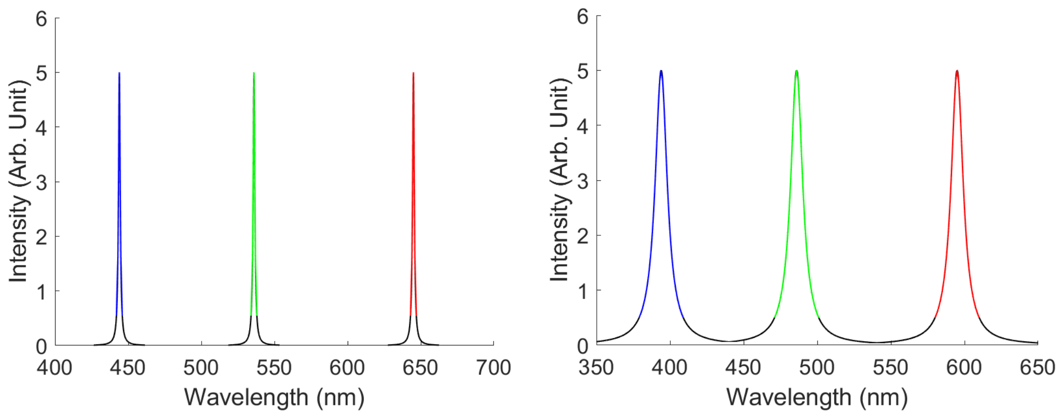
The realistic simulation is performed following the display configuration in Figure 5.5. The wavefront E at the hologram is propagated by a distance of d_{el} to the external lens. The wavefront then goes through the external lens and propagates by a distance of d_{eye} to the eye lens. Finally, the wavefront goes through the eye lens and propagates to the retina. The intensity of the wavefront at the retina is taken as the result. The pseudocode is shown in Algorithm 6. Prop(E, λ, ρ, d) propagates the wavefront E with wavelength λ and sampling pitch ρ by a distance of d. Lens(E, λ, ρ, f, r) applies the transfer function of a lens with focal length f and radius r to the wavefront E with wavelength λ and sampling pitch ρ. In practice, λ is the wavelength of the illuminating light source. ρ is the pixel pitch of the hologram. r_{el} and r_{eye} are the radius of the external lens and the iris of the eye, respectively. The implementations of Prop and Lens follow the Rayleigh-Sommerfeld diffraction solution and the lens function described in Voelz [6].

Algorithm 6: Realistic Simulation

```
input  : $E$, $\lambda$, $\rho$, $d_{\text{el}}$, $f_{\text{el}}$, $r_{\text{el}}$ $d_{\text{eye}}$, $f_{\text{eye}}$, $r_{\text{eye}}$, $d_{\text{r}}$
output: $I$
$E_1 = \text{Prop}(E, \lambda, \rho, d_{\text{el}})$;
$E_2 = \text{Lens}(E_1, \lambda, \rho, f_{\text{el}}, r_{\text{el}})$;
$E_3 = \text{Prop}(E_2, \lambda, \rho, d_{\text{eye}})$;
$E_4 = \text{Lens}(E_3, \lambda, \rho, f_{\text{eye}}, r_{\text{eye}})$;
$E_5 = \text{Prop}(E_4, \lambda, \rho, d_{\text{r}})$;
$I = \text{abs}(E_5)^2$;
```

5.5.2.2 Light Source

The spectrum of the light source is commonly expressed as a Cauchy distribution in Equation 5.15. $I(\lambda; \lambda_0, \gamma, I_0)$ is the intensity of the light source at wavelength



laser

SLED

Figure 5.7: The spectra of the laser and SLED light sources.

λ given the parameters of the distribution. λ_0 is the center wavelength of the spectrum at which the intensity is at the maximum I_0. The intensity is half the maximum intensity at a wavelength of γ away from the center wavelength. γ is the full-width half-max (FWHM) which represents the spectral width of the light source.

$$I(\lambda; \lambda_0, \gamma, I_0) = \frac{I_0}{[1 + (\frac{\lambda - \lambda_0}{\gamma})^2]} \tag{5.15}$$

Different light sources have different spectra. A diode laser typically has an FWHM near 1.4 nm when it is functioning as the light source for a holographic display. Commercially available SLEDs have FWHM ranging from 3 nm to 15 nm. LEDs have even larger FWHM. An example of the RGB laser and SLED light sources can be seen in Figure 5.7.

To find out what kind of light source works well for holographic volume rendering, we perform realistic simulation using 4 different kinds of light sources that correspond to a laser, a narrow-spectrum SLED, a SLED, and a LED. The FWHM of these light sources can be seen in Table 5.1.

Light Source	FWHM nm	Light Source	FWHM nm
Laser	1.4	Narrow SLED	5
SLED	10	LED	20

Table 5.1: Spectra of light sources used in the realistic simulation.

We take the range of the wavelengths with intensity greater than 10% of the peak intensity as the wavelengths used in the realistic simulation for each color. The ranges are marked in respective colors in the examples in Figure 5.7. We sample the spectrum at a regular interval $\Delta\lambda = 1.0\,\text{nm}$ within each range.

In the realistic simulation, the wave propagation from the hologram to the retina described in Algorithm 6 is performed with each sampled wavelength. The weighted average of the intensity is taken as the result of the realistic simulation with the weights being the intensities for each wavelength.

5.5.2.3 Simulations

The realistic simulations of three holograms are shown in Figure 5.8. The holograms are calculated with $R_x = R_y = 1024$, $H_x = H_y = 2048$ and the pixel pitch $\rho = 3.74\,\mu\text{m}$. In the realistic simulation, $d_{\text{el}} = 45.56\,\text{mm}$, $f_{\text{el}} = 20\,\text{mm}$, $d_{\text{eye}} = 50\,\text{mm}$. This way, the focus distance $s_{\text{eye}} = 120\,\text{mm}$. The iris radius $r_{\text{eye}} = 1.5\,\text{mm}$ is typical of an eye in an area with normal illumination. The retina distance $d_{\text{r}} = 24\,\text{mm}$ is typical of an adult.

We can see that with increasing spectral width, the speckle noise becomes less

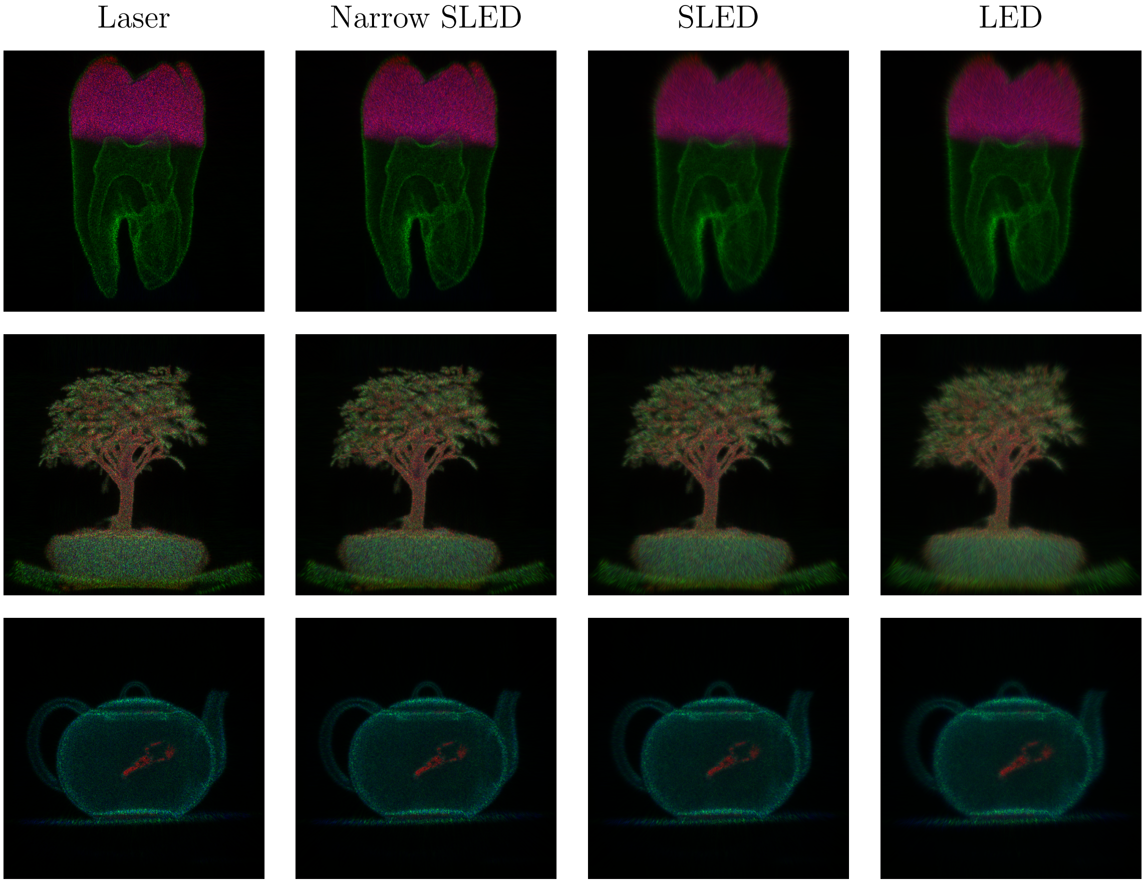


Figure 5.8: The realistic simulations of three holograms with different light sources.

severe. However, the images also become less sharp. Judging from direct observation of Figure 5.8, we conclude that the spectral width of a narrow SLED (FWHM of 5 nm) offers the best balance between speckle noise and image sharpness.

The weighted intensity from the realistic simulation is a 2D real matrix with arbitrarily large maximum values. To show the realistic simulations as images in Figure 5.8, we normalize the weighted intensity to [0, 1] and then perform gamma correction. Let the weighted intensity be I_{w}, the image S is effectively the following.

$$S = (\frac{I_{\mathrm{w}}}{\max(I_{\mathrm{w}})})^{0.5} \tag{5.16}$$

Due to speckle noise, there are a small number of bright spots in the results of realistic simulations. As the speckle noise becomes less severe with increasing spectral width, the max value in I_{w} decreases. As a result, the images obtained from Equation 5.16 become increasingly brighter with increasing spectral width. In reality, the power of the light source determines the brightness of the observed image, and the power of the light source is usually limited by the integrated intensity which can safely enter the observer's eyes. Therefore, assuming the same light source power, the brightness of the observation should be similar to each other. We scale the intensities of the images in Figure 5.8 so they have the same total intensity as the narrow SLED for each channel. As a result, the images for SLED and LED are undersaturated while the images for laser are oversaturated by no more than 25%.

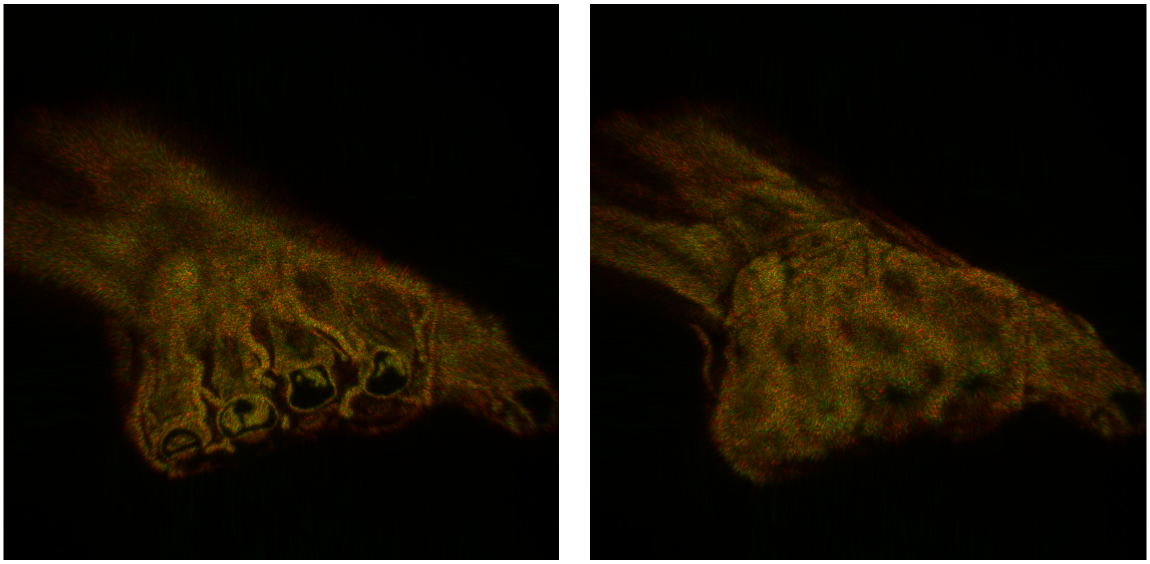


Figure 5.9: By controlling the focal length of the eye lens the viewer is able to focus at different depths of the volume.

5.5.2.4 Focus at different depths

Next, we show the realistic simulation of focusing at different depths in the volume in Figure 5.9. As can be seen, by adjusting the focal length of the eye lens, parts of the volume at different depths appear to be in focus.

5.5.3 Speedup

As mentioned in Section 5.3.2, our proposed hologram calculation method using the convolution theorem reduces the complexity from $O(R_xR_yDH_xH_Y)$ of the direct method to a more manageable $O(DH_xH_y\log(H_xH_y))$. All calculations are performed on a PC with an Intel 9900K CPU and an Nvidia RTX 2080Ti GPU. The implementation is in C++ and CUDA. Here we show the speedup of using the convolution theorem to accelerate the hologram of several volumes in Table 5.2.

The hologram computation time mentioned in this work includes both the point light source sampling using raycasting direct volume rendering and the calculation of Equation 5.1 using the sampled light sources either with the direct method or the convolution theorem method.

volume	d_V	D	direct	CT	speedup
Tooth	20 mm	90	2.89 h	5.09 s	2049×
C60	30 mm	58	1.80 h	11.56 s	561×
Teapot	40 mm	253	7.87 h	64.20 s	441×
Engine	50 mm	116	3.64 h	27.73 s	472×
Hand	60 mm	290	9.30 h	82.95 s	403×
Bonsai	70 mm	232	7.14 h	129.93 s	198×

Table 5.2: Hologram computation time and speedup for several volumetric datasets. These holograms are calculated with $H_x = H_y = 2048$ and $R_x = R_y = 1024$.

As can be seen, our proposed method achieves from hundreds to thousands times speedup compared to the brute-force direct calculation method when calculated on the same hardware. In a simulated comparison with the previous work, it takes our proposed method 5.04 s, compared to 3.6 h in [179], to calculate the hologram of the same dimension with the same number of point light sources. This is roughly a 2500× speedup including the hardware difference.

It can also be seen that the speedup is not uniform in all six test cases. Generally speaking, the closer the volume is to the hologram, the larger the speedup. We will explain the factors of the processing time in Section 5.6.1.

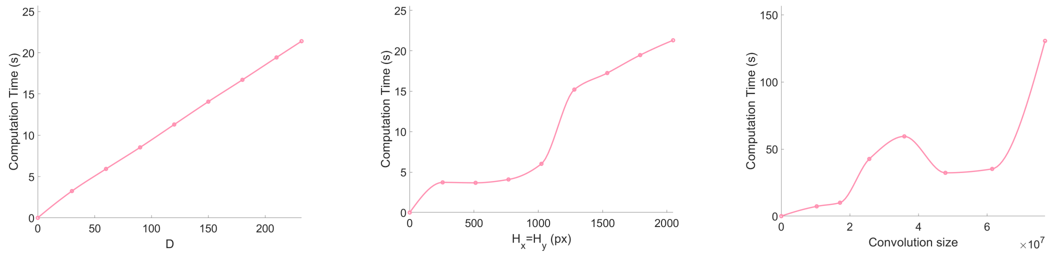


Figure 5.10: The left shows that the computation time increases linearly with the number of compositing steps D with a fixed hologram dimensions ($H_x = H_y = 2048$). The middle row shows the computation time increases with increasing hologram dimensions with $D = 232$. d_V =30 mm for both tests. The right shows how the computation time changes with the convolution size. The convolution size is increased by placing the volume farther away from the hologram thus increasing the sizes of the K_z and C_z.

5.6 Discussion

Here we discuss several details about our holographic volume rendering method including the hologram computation time, observed scene FOV, the use of elemental holograms, and the wavelength-specific interference artifacts.

5.6.1 Factors of Hologram Computation Time

Here, we closely examine the different factors that affect the computation time of our proposed method. The complexity to calculate the hologram using the CT method is $O(DH_xH_y\log(H_xH_y))$ assuming the fast Fourier transform (FFT) implementation and that the arrays in the convolution are both of dimensions $H_x \times$

H_y. We use the cuFFT library[3] in CUDA, which is based on the Cooley-Tukey and Bluestein algorithms. We show how the computation time changes with the increasing number of compositing steps D and hologram dimensions (H_x and H_y) in Figure 5.10 top and middle row. As can be seen, when all the other parameters are kept the same, the computation time increases linearly with the number of compositing steps. Even though the computation time does not scale exactly as the complexity suggests with the increasing hologram dimensions, the computation time does increase with the hologram dimensions.

Another factor that affects the computation time of our method, as can be seen in Table 5.2, is the distance between the hologram and the volume d_{V}. In our implementation, the 2D signal undergoing the FFT has the combined dimensions of K_z and C_z for the compositing plane z away from the hologram. In Section 5.3.2, we assume that both C_z and K_z have dimensions of the same order of magnitude as the hologram. This is generally true for the area where we place the volume. In practice, the sizes of C_z and K_z both increase with z illustrated in Figure 5.11. We sample the point light sources in the space the wavefront from the SLM can diffract to. This space is the union of all the diffraction cones of the SLM pixels. The computation time with increasing convolution size (the number of elements undergoing FFT) can be seen in Figure 5.10 bottom row. A large convolution size would also lead to heavy usage of the VRAM in this implementation. There are several outliers where the computation time does not scale with the convolution size. This may be because we did not pad the arrays undergoing FFT to the size of a

[3] https://developer.nvidia.com/cufft

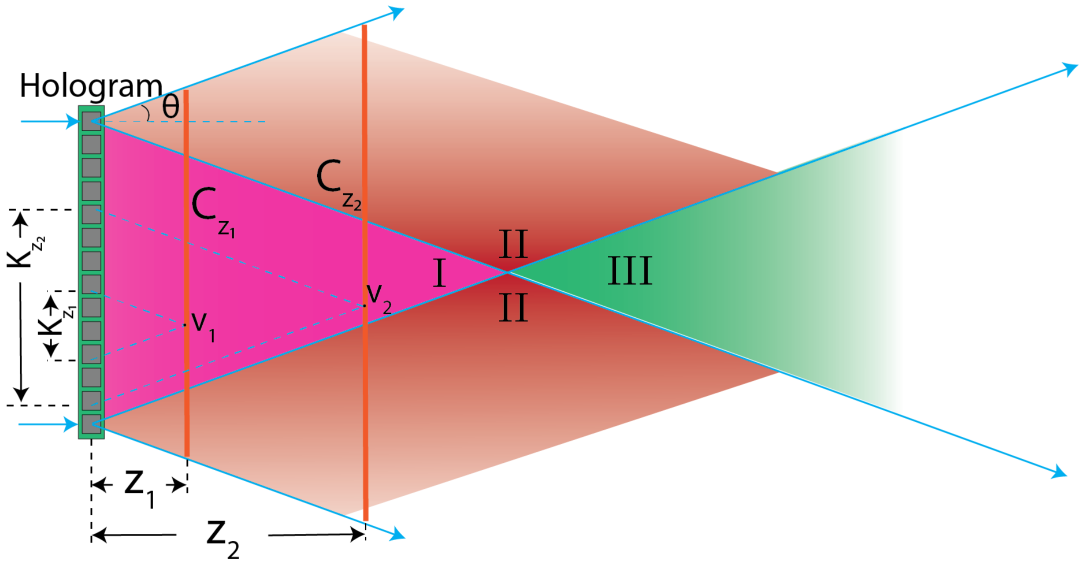


Figure 5.11: The sizes of C_z and K_z of compositing planes z_1 and z_2 away from the hologram are shown. Given the hologram size and the diffraction angle θ, the reconstruction of point light sources in region I has the finest resolution (smallest spot size). The resolution of the reconstruction in region II drops off (the spot size increases) from the middle towards top and bottom. The resolution of the reconstruction in region III drops off horizontally as it gets farther away from the hologram.

power of two. The different convolution sizes and underlying VRAM organization may have caused different variations of the FFT implementation to be used by the cuFFT library, leading to the outliers.

However, we can find an upper bound for the hologram computation time. If the size of the volume is known, the dimension of K_z does not need to increase endlessly with z. Given that the size of the volume is fixed and the center of the volume is placed on the optical axis of the hologram, K_z does not need to be larger than what covers the entirety of the hologram, as illustrated in Figure 5.12 with red dashed lines. In this case, the upper bound of the size of K_z is the combined size of the hologram and the size of the volume at distance z. Let the maximum size of the volume at all compositing planes be V_{max} and the distance between the hologram and the farthest compositing plane be z_{max}. The dimension of C_z is bound by $\frac{V_{\text{max}}}{\rho} \times \frac{V_{\text{max}}}{\rho}$. The dimension of K_z is $\frac{2z\text{tan}(\theta)}{\rho} \times \frac{2z\text{tan}(\theta)}{\rho}$ and is bound by $(H_x + \frac{V_{\text{max}}}{\rho}) \times (H_y + \frac{V_{\text{max}}}{\rho})$. Therefore, the signal undergoing the FFT has dimensions $\frac{V_{\text{max}}+2z_{\text{max}} \cdot \text{tan}(\theta)}{\rho} \times \frac{V_{\text{max}}+2z_{\text{max}} \cdot \text{tan}(\theta)}{\rho}$ and is bound by $(H_x + 2\frac{V_{\text{max}}}{\rho}) \times (H_y + 2\frac{V_{\text{max}}}{\rho})$. Another potential bound is that z_{max} cannot be larger than the form factor of the display. This is usually within 200 mm for a head-mounted near-eye display.

5.6.2 Scene FOV and Volume Placement

The FOV described in Section 5.4 refers to the theoretical maximum FOV where the light diffracted from the hologram enters the observer's eye. The FOV spanned by the scene, which we refer to as the scene FOV, is also dependent on the size of the scene and the distance between the scene image and the eye.

Without loss of generality, we assume that the intersections of the volume and the compositing planes are at their largest V_{max} at the center of the volume. This plane is at a distance of s_0 away from the external lens in Fig 5.5. The magnified virtual image of this plane is at a distance of $s_1 = \frac{f_{\text{el}} s_0}{f_{\text{el}} - s_0}$ away from the external lens. The magnified size of the volume is $L_1 = V_{\text{max}} \frac{f_{\text{el}}}{f_{\text{el}} - s_0}$. The scene FOV is therefore $2\text{atan}(\frac{V_{\text{max}} f_{\text{el}}}{2(f_{\text{el}} - s_0) \cdot (s_1 + d_{\text{eye}})})$.

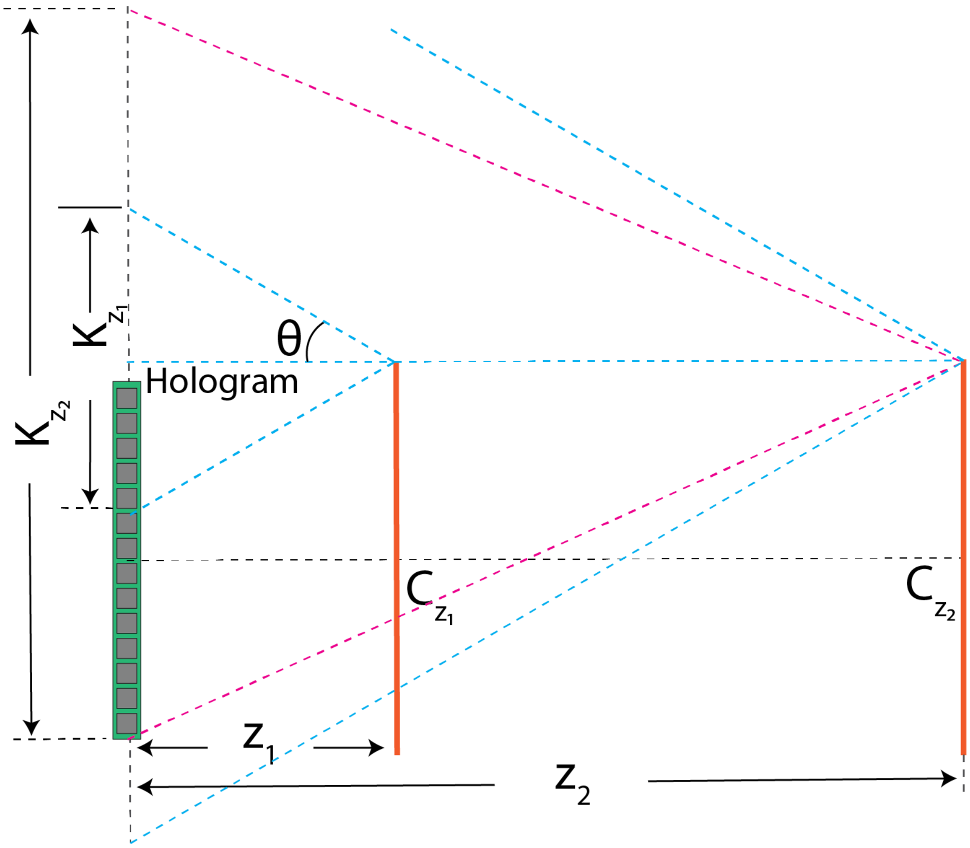


Figure 5.12: Given the volume size, the size of K_z does not need to increase endlessly with distance from the hologram.

The focal distance $s_1 + d_{\text{eye}}$ needs to be large enough for the observer to comfortably focus on the image. Therefore, to achieve a large scene FOV, a large volume is required, leading to longer computation time according to the discussion in Section 5.6.1. For example, if we want the observer to focus on a volume at a distance of 120 mm that spans 40° through an external lens of focal length 10 mm, the volume needs to be of size $V_{\max} = 10.91\,\text{mm}$.

The placement of the volume relative to the hologram also affects the quality of the hologram. According to the Rayleigh criterion [200], the size of the smallest spot that the hologram can create as part of its image is proportional to the distance to that spot divided by the size of the area of the hologram devoted to making that spot. Therefore, the larger the section of the hologram devoted to any particular spot in the image, the finer resolution that spot can have. On the other hand, only those pixels on the hologram for which the diffraction cone contains the image spot can be devoted to making any spot in the image without increasing noise in the image [1]. The area on the hologram plane that provides the full bandwidth to ensure the finest spot size for a given image point source $\boldsymbol{v}$ can be thought of as the base of a cone whose vertex is at $\boldsymbol{v}$, the base is on the hologram plane, and the vertex angle is the same as the diffraction cone. An example is in Figure 5.11 where the areas that provide the full bandwidth for point sources v_1 and v_2 are the areas labeled as K_{z_1} and K_{z_2}. If the hologram cannot cover the area that provides the full bandwidth for some point sources, the reconstructions of those point sources have larger spot sizes and poorer resolution.

We can divide the space to the right of the hologram into three types of regions to place the volume, as illustrated in Figure 5.11. The hologram is able to provide all the bandwidth needed to resolve point light sources in region I. Therefore, reconstruction of these point light sources is of the finest resolution. The bandwidths required to resolve point light sources in regions II and III are both partially provided by a portion of the hologram. Specifically, the resolution of the reconstruction in region II drops off from its border with region I vertically towards the top and bottom. The resolution of reconstruction in region III drops off horizontally as it gets farther away from the hologram.

A small volume can be placed within region I. For larger volumes, it is preferred to place them in region III for better quality. Using the example above, the volume needs to be placed at a distance of 157.81 mm away from the hologram.

5.6.3 Elemental Hologram

Another popular method to accelerate holographic calculation is to use elemental holograms such as in [1] and [179]. However, it does not help much with our hologram computation method with the convolution theorem . If we are to divide the hologram into $M \times N$ elemental holograms each with $\frac{H_x}{M} \times \frac{H_y}{N}$ pixels, the complexity to calculate elemental hologram is $O(\frac{H_xH_y}{MN}\log\frac{H_xH_y}{MN})$. The complexity to calculate the entire hologram is $O(DMN\frac{H_xH_y}{MN}\log\frac{H_xH_y}{MN}) = O(DH_xH_y\log H_xH_y)$. It is not much different from the original complexity of our proposed method, but will potentially add more CUDA kernel calls and increase the computation time. Therefore, we do not use the approach of elemental holograms.

5.7 Conclusion

In this Chapter, we improve the holographic volume rendering by addressing three challenges it faces. First, we develop a new hologram computation method that uses direct volume rendering to generate the collection of point light sources and uses the convolution theorem to significantly accelerate the computation. Compared with the brute-force methods, this accelerated hologram computation method achieves over 2000 speedup. Second, we explore several holographic display configurations and find one that increases the FOV while also maintaining the compatibility with the accelerated hologram computation method. Third, we realistically simulate the observed images under different kinds of light sources and identify the appropriate light source that strikes the balance between speckles and image resolution for holographic volume rendering. These improvements pave the way for bringing volume visualization to holographic displays.

Chapter 6: Conclusions and Future Work

6.1 Perception in the Far Periphery

First, in Chapter 2, we investigate the perception in the far peripheral vision in VR and AR experiences. We conduct a user study in which we record the perception time of 40 participants reacting to patterns of varying sizes appearing at different eccentricities in their peripheral vision. Based on the data collected, we find that patterns can be scaled linearly with the eccentricity it is shown at to reach a constant perception time. Based on these equal-efficiency scales, we can optimally scale the content displayed at higher eccentricities so that it can be perceived efficiently. Our finding is useful for designing VR and AR experiences that are able to leverage the entire human visual system's field of view for the emerging displays capable of supporting a wide FOV. Future work on this topic could explore if the scaling function applies to even higher eccentricities. With a larger field of view display (for example StarVR [1]), we can examine the visual properties up to 105° eccentricity, based on the findings in Chapter 2. In addition to visual acuity, peripheral vision also differs from central vision in several other attributes such as contrast sensitivity in color, intensity, orientation, motion, and texture detail. It will be important

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

to investigate how these attributes affect the perception of information in the far periphery. Finally, future work could use the findings of Chapter 2 to design an actual experience and show how far peripheral vision can enhance the user experience in an immersive VR or AR scenario. Some examples include the VR exploration and surveillance applications by Du*et al.* [201–205] and the virtual memory palace created by Krokos *et al.* [206].

6.2 AR Displays for Surgery

Second, in Chapter 3, we implement an AR application and show through a study how AR could help improve the safety of a dangerous and invasive medical procedure. We present an AR surgical visualization and tracking system that can assist surgeons catheter placement in external ventricular drain procedure. Compared to existing similarly-purposed systems, our system does not require changing the shape or weight of the medical instruments. The catheter tracking in our system has high accuracy and low latency. Tests show that our system achieves a 0.58 mm accuracy on a 2D plane and an accuracy of 0.85 mm in 3D space. Processing for each frame takes 22.60 ms on a moderately powerful computer. Both the accuracy and speed can be further improved with higher-quality cameras and parallel implementation. In the study that took place in a realistic surgical environment with a cadaveric head, the displayed virtual catheter projection aligned well with the visible portion of the real catheter and the pace at which the real and the virtual catheter moved matched. The surgeon participants also reported that the handling of our slightly modified catheter feels identical to a traditional catheter. In the future, we expect to adapt our system to comply with surgical safety standards and improve the accuracy and speed of our system with better hardware and software implementation. We could also make the system more robust to occlusion of the color markers by using more redundant markers.

6.3 Correcting Proximity Effect in Nanophotnoic Phased Arrays

Third, in Chapter 4, we look at the infrastructure of VR and AR displays. Holographic displays recreate the optical wave field of 2D images or 3D scenes, addressing the vergence-accommodation conflicts present in the current VR and AR headsets that rely on stereoscopic vision for depth perception. Nanophotonic phased arrays (NPAs) is a new type of holographic display that has the advantage of high refresh rate and small form-factor compared the LCoS SLM holographic displays. However, the thermally-modulated NPAs suffer from thermal proximity effect as they depend on the thermo-optic effects for phase modulation. We propose novel proximity effect correction methods for both the Fourier holograms and the Fresnel holograms for the NPAs. Computational simulations show that our proposed methods are more effective and efficient in correcting the proximity effect than existing methods. Future work on this topic includes acceleration of the methods and correcting the proximity effect of 3D Fresnel holograms.

6.3.1 Acceleration

Both the Proximal PEC and the Fresnel PEC method are iterative algorithms. For example, it currently takes our implementation of the Fresnel PEC method 4.28 s to process a 256 × 256 image at proximity effect level $\sigma = 0.66$ px. However, the Fresnel PEC method is on average 36.18 times faster than the Iterative Fresnel PEC method while also offering a superior correction effectiveness. Since the Fresnel PEC method heavily uses the Fourier transform, we believe that hardware implementation and massive parallelization can significantly accelerate it. Special hardware for the Fourier transform such as the on-chip integrated all-optical FFT with a 26 Tbps data throughput by Nejadriahi and Sorger [207] may also significantly increase the speed of this method. Alternatively, it is desirable to investigate how to properly adjust the convergence condition and the step size for maximum efficiency.

6.3.2 3D Fresnel Holograms

We have demonstrated displaying 3D holograms by using time-division multiplexing to display multiple 2D Fresnel holograms processed with the FPEC method at different depths in consecutive frames. Proximity effect correction on 3D Fresnel holograms where one hologram encodes the entire 3D scene will be an interesting direction for future work. A naive solution would be to calculate the fully complex or phase-only 3D Fresnel hologram similar to Shi *et al.* [1] and Maimone*et al.* [62] and perform deconvolution on the phase signal. The non-negative constraint needs to be enforced in the deconvolution. An alternative is to optimize the observed 3D scene similar to the proposed FPEC method. It remains to be seen how to define the cost function similar to Equation 4.50 for a 3D scene.

6.4 Fast Holographic Volume Rendering

Last,in Chapter 5, we propose a way to significantly accelerate the calculation of the holograms based on volumetric data. Our proposed method achieves a 2500× speedup compared to the previous state-of-the-art method and reduces the computation time from hours previously to mere seconds. We propose a display configuration that improves the field-of-view of the display while maintaining the compatibility to the speedup. We also show through simulations how to choose the light source for the holographic display to strike a balance between the speckle noise and image sharpness for the observation.

Going forward, there are several directions to be taken.

6.4.1 Phase Shift on the SLM

In our realistic simulation, we simulated the image as the weighted average of the wavefront propagated from the hologram to the retina when the hologram is illuminated with light of different wavelengths sampled from a spectrum. In this simulation, the same electric field parameters (amplitude and phase) are assumed to be present at each wavelength. In reality, when the SLM is illuminated by light of a wavelength other than the one it is calibrated with, the dispersion of the liquid crystal in the SLM alters the phase delay of the light as it is traveling through the liquid crystal. For phase modulation, that means that the phase of the electric field will be slightly different for wavelengths different than the calibration wavelength. Similarly, one would expect slight differences in the amplitude for amplitude modulation. It is worthwhile to investigate how to simulate this effect in order to perform more realistic simulations. Manufacturers may not be forthcoming with information about the dispersion of their liquid crystals; but this is also measurable as a function of wavelength by doing interferometric measurements similar to those commonly used to calibrate SLMs.

6.4.2 Quantifying the Speckles

We investigate the choice of light sources in the realistic simulations. With direct observation, we have found that a light source with an FWHM of 5 nm strikes the balance between the speckles and resolution among the four sample light sources tested. Altering the initial phase of the point light sources in the volume should allow one to generate statistically independent speckle patterns in the observation simulations. By averaging over successive runs, one could, in principle, get an image without speckles. This would allow one to characterize the speckle and find a metric that could be used to optimize the trade-off between speckle and resolution.

To quantify the speckles in a simulated observation image, it is necessary to first identify the speckles. This can be done by subtracting the image without any speckles from the speckled image. The severity, granularity, and anisotropy of the speckles can be quantified with methods such as the power spectral density in Ong *et al.* [208]. Given the speckleless image, it can also be used as the reference image in image quality metrics such as the structural similarity index in Wany *et al.* [175] or the neural network-based perceptual metric in Zhang *et al.* [176]. The image quality metric can be used in combination with the quantified speckle to find the optimum observed image.

6.4.3 Improving Volume Rendering Sampling

In Chapter 5, we use raycasting direct volume rendering to generate the point light sources for hologram computation and do not consider more complicated lighting. The complexity of generating the point light sources using raycasting is $O(DR_xR_y)$. In practice, processing each compositing plane (with $R_x = R_y = 1024$) takes on average 0.63 ms. The raycasting process takes no more than 1.12% of the total computation time depending on the hologram dimension and the distance between the hologram and the volume. Therefore, lighting models such as diffuse lighting and specular lighting can be easily integrated into our method with very little overhead to the computation time. Other volume visualization and exploration techniques such as in Cheng *et al.* [209, 210]. Also, in the current implementation, the rays are uniformly sampled in both dimensions. This can be improved by sampling the rays more densely in areas where the gradient of the volume is large. In order to cope with more complicated illuminations in the scenes, methods such as the Monte Carlo technique in Veach and Guibas [211], Shirley *et al.* [212], and Meng *et al.* [213] can be used. Instead of computing the light integrals on the image plane, one can compute them at various sampling positions in the volume.

Bibliography

[1] Liang Shi, Fu-Chung Huang, Ward Lopes, Wojciech Matusik, and David Luebke. Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3d computer graphics. *ACM Transactions on Graphics (TOG)*, 36(6):236, 2017.

[2] J Adam Jones, J Edward Swan, and Mark Bolas. Peripheral stimulation and its effect on perceived spatial scale in virtual environments. *IEEE transactions on visualization and computer graphics*, 19(4):701–710, 2013.

[3] Sujal Bista, Jiachen Zhuo, Rao P. Gullapalli, and Amitabh Varshney. Visualization of Brain Microstructure through Spherical Harmonics Illumination of High Fidelity Spatio-Angular Fields. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2516–2525, Dec 2014.

[4] Sujal Bista, Jiachen Zhuo, Rao P. Gullapalli, and Amitabh Varshney. Visual Knowledge Discovery for Diffusion Kurtosis Datasets of the Human Brain. In Ingrid Hotz and Thoma Schultz, editors, *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, pages 213–234. Springer, 2015.

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[6] David George Voelz. *Computational fourier optics: a MATLAB tutorial*. SPIE press Bellingham, WA, 2011.

[7] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. Live image quality assessment database release 2. `http://live.ece.utexas.edu/research/quality`, 2004. Accessed: 2019-04-03.

[8] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, Nov 2006.

[9] T. Smits and W.J. Faber. Chronic (classified historical newspaper images). KB Lab: The Hague. `http://lab.kb.nl/dataset/chronic-classified-historical-newspaper-images`, 2018.

[10] Simon J Thorpe, Karl R Gegenfurtner, Michèle Fabre-Thorpe, and Heinrich H BuÈlthoff. Detection of animals in natural images using far peripheral vision. *European Journal of Neuroscience*, 14(5):869–876, 2001.

[11] Muriel Boucart, Quentin Lenoble, Justine Quettelart, Sebastien Szaffarczyk, Pascal Despretz, and Simon J Thorpe. Finding faces, animals, and vehicles in far peripheral vision. *Journal of Vision*, 16(2):10–10, 2016.

[12] Adam M Larson and Lester C Loschky. The contributions of central versus peripheral vision to scene gist recognition. *Journal of Vision*, 9(10):6–6, 2009.

[13] J Rovamo and V Virsu. An estimation and application of the human cortical magnification factor. *Experimental brain research*, 37(3):495–510, 1979.

[14] Steve Grogorick, Michael Stengel, Elmar Eisemann, and Marcus Magnor. Subtle gaze guidance for immersive environments. In *Proceedings of the ACM Symposium on Applied Perception*, page 4. ACM, 2017.

[15] Xuetong Sun and Amitabh Varshney. Investigating perception time in the far peripheral vision for virtual and augmented reality. In *Proceedings of the 15th ACM Symposium on Applied Perception*, page 13. ACM, 2018.

[16] Xuetong Sun, Sarah B. Murthi, Gary Schwartzbauer, and Amitabh Varshney. High-precision 5dof tracking and visualization of catheter placement in evd of the brain using ar. *ACM Trans. Comput. Healthcare*, 1(2), March 2020.

[17] Manan Raval. Nanophotonic visible light phased arrays. Master's thesis, Massachusetts Institute of Technology, 2016.

[18] Jie Sun, Erman Timurdogan, Ami Yaacobi, Ehsan Shah Hosseini, and Michael R Watts. Large-scale nanophotonic phased array. *Nature*, 493(7431):195, 2013.

[19] Jonathan K Doylend, MJR Heck, Jock T Bovington, Jonathan D Peters, LA Coldren, and JE Bowers. Two-dimensional free-space beam steering with an optical phased array on silicon-on-insulator. *Optics express*, 19(22):21595–21604, 2011.

[20] X. Sun, Y. Zhang, P. C. Huang, N. Acharjee, M. Dagenais, M. Peckerar, and A. Varshney. Correcting the proximity effect in nanophotonic phased arrays. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2020.

[21] Andrew S Glassner. *An introduction to ray tracing.* Elsevier, 1989.

[22] Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)*, 9(3):245–261, 1990.

[23] Dennis Gabor. A new microscopic principle, 1948.

[24] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. Peripheral vision and pattern recognition: A review. *Journal of vision*, 11(5):13–13, 2011.

[25] Michael J Simpson. mini-review: Far peripheral vision. *Vision research*, 140:96–105, 2017.

[26] Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney. Kernel Foveated Rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques (I3D)*, 1(5):1–20, May 2018.

[27] X. Meng, R. Du, and A. Varshney. Eye-dominance-guided foveated rendering. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1972–1980, 2020.

[28] X. Meng, R. Du, J. F. JaJa, and A. Varshney. 3d-kernel foveated rendering for light fields. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2020.

[29] PM Daniel and D Whitteridge. The representation of the visual field on the cerebral cortex in monkeys. *The Journal of physiology*, 159(2):203–221, 1961.

[30] A Cowey and ET Rolls. Human cortical magnification factor and its relation to visual acuity. *Experimental Brain Research*, 21(5):447–454, 1974.

[31] Giles S Brindley and WS Lewin. The sensations produced by electrical stimulation of the visual cortex. *The Journal of physiology*, 196(2):479, 1968.

[32] Jonathan C Horton and William F Hoyt. The representation of the visual field in human striate cortex: a revision of the classic holmes map. *Archives of ophthalmology*, 109(6):816–824, 1991.

[33] Jonas Larsson and David J Heeger. Two retinotopic visual areas in human lateral occipital cortex. *The Journal of Neuroscience*, 26(51):13128–13142, 2006.

[34] DJ Tolhurst and L Ling. Magnification factors and the organization of the human striate cortex. *Human neurobiology*, 6(4):247–254, 1987.

[35] Eric L Schwartz. Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding. *Vision research*, 20(8):645–669, 1980.

[36] Hans Strasburger and Maka Malania. Source confusion is a major cause of crowding. *Journal of Vision*, 13(1):24–24, 2013.

[37] David J Yates and Tom Stafford. Insights into the function and mechanism of saccadic decision making from targets scaled by an estimate of the cortical magnification factor. *Cognitive Computation*, 3(1):89–93, 2011.

[38] Christopher W Tyler. Human symmetry detection exhibits reverse eccentricity scaling. *Visual Neuroscience*, 16(05):919–922, 1999.

[39] Mark M Schira, Alex R Wade, and Christopher W Tyler. Two-dimensional mapping of the central and parafoveal visual field to human visual cortex. *Journal of Neurophysiology*, 97(6):4284–4295, 2007.

[40] Hans Strasburger. Invariance of the psychometric function for character recognition across the visual field. *Attention, Perception, & Psychophysics*, 63(8):1356–1376, 2001.

[41] Lester Loschky, Muriel Boucart, Sebastien Szaffarczyk, Clement Beugnet, Alicia Johnson, and Jia Li Tang. The contributions of central and peripheral vision to scene gist recognition with a 180° visual field. *Journal of vision*, 15(12):570–570, 2015.

[42] Dimitri J Bayle, Benjamin Schoendorff, Marie-Anne Hénaff, and Pierre Krolak-Salmon. Emotional facial expression detection in the peripheral visual field. *PloS one*, 6(6):e21584, 2011.

[43] Rogier Landman, Jitendra Sharma, Mriganka Sur, and Robert Desimone. Effect of distracting faces on visual selective attention in the monkey. *Proceedings of the National Academy of Sciences*, 111(50):18037–18042, 2014.

[44] Donghao Ren, Tibor Goldschwendt, YunSuk Chang, and Tobias Höllerer. Evaluating wide-field-of-view augmented reality with mixed reality simulation. In *Virtual Reality (VR), 2016 IEEE*, pages 93–102. IEEE, 2016.

[45] Paul B Kline and Bob G Witmer. Distance perception in virtual environments: Effects of field of view and surface texture at near distances. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 40, pages 1112–1116. SAGE Publications Sage CA: Los Angeles, CA, 1996.

[46] J Adam Jones, David M Krum, and Mark T Bolas. Vertical field-of-view extension and walking characteristics in head-worn virtual environments. *ACM Transactions on Applied Perception (TAP)*, 14(2):9, 2017.

[47] Toshikazu Ohshima, Hiroyuki Yamamoto, and Hideyuki Tamura. Gaze-directed adaptive rendering for interacting with virtual space. In *Virtual reality annual international symposium, 1996., Proceedings of the IEEE 1996*, pages 103–110. IEEE, 1996.

[48] Yoshio Ishiguro and Jun Rekimoto. Peripheral vision annotation: Noninterference information presentation method for mobile augmented reality. In *Proceedings of the 2nd Augmented Human International Conference*, page 8. ACM, 2011.

[49] Brett R Jones, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. Illumiroom: peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 869–878. ACM, 2013.

[50] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.

[51] Thomas A DeFanti, Gregory Dawe, Daniel J Sandin, Jurgen P Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr, and Ramesh Rao. The starcave, a third-generation cave and virtual reality optiportal. *Future Generation Computer Systems*, 25(2):169–178, 2009.

[52] George Robertson, Mary Czerwinski, and Maarten Van Dantzich. Immersion in desktop virtual reality. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 11–19. ACM, 1997.

[53] Hajime Nagahara, Yasushi Yagi, and Masahiko Yachida. Super wide viewer using catadioptrical optics. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 169–175. ACM, 2003.

[54] Kiyoshi Kiyokawa. A wide field-of-view head mounted projective display using hyperbolic half-silvered mirrors. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 207–210. IEEE, 2007.

[55] David Dunn, Cary Tippets, Kent Torell, Petr Kellnhofer, Kaan Akşit, Piotr Didyk, Karol Myszkowski, David Luebke, and Henry Fuchs. 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, 2017.

[56] Ismo Rakkolainen, Matthew Turk, and Tobias Höllerer. A compact, wide-fov optical design for head-mounted displays. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 293–294. ACM, 2016.

[57] I Rakkolainen, M Turk, and T Höllerer. A superwide-fov optical design for head-mounted displays. 2016.

[58] Robert W Massof, Lawrence G Brown, Marc D Shapiro, G David Barnett, Frank H Baker, and Fuminobu Kurosawa. 37.1: Invited paper: Full-field high-resolution binocular hmd. In *SID Symposium Digest of Technical Papers*, volume 34, pages 1145–1147. Wiley Online Library, 2003.

[59] Kaan Akşit, Jan Kautz, and David Luebke. Slim near-eye display using pinhole aperture arrays. *Applied optics*, 54(11):3422–3427, 2015.

[60] Robert Xiao and Hrvoje Benko. Augmenting the field-of-view of head-mounted displays with sparse peripheral displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1221–1232. ACM, 2016.

[61] Fu-Chung Huang, Kevin Chen, and Gordon Wetzstein. The light field stereoscope: immersive computer graphics via factored near-eye light field displays with focus cues. *ACM Transactions on Graphics (TOG)*, 34(4):60, 2015.

[62] Andrew Maimone, Andreas Georgiou, and Joel S Kollin. Holographic near-eye displays for virtual and augmented reality. *ACM Transactions on Graphics (TOG)*, 36(4):85, 2017.

[63] Andrew Maimone, Douglas Lanman, Kishore Rathinavel, Kurtis Keller, David Luebke, and Henry Fuchs. Pinlight displays: wide field of view augmented reality eyeglasses using defocused point light sources. In *ACM SIGGRAPH 2014 Emerging Technologies*, page 20. ACM, 2014.

[64] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–73, 1992.

[65] David R Huyette, Benjamin J Turnbow, Christian Kaufman, Dale F Vaslow, Benjamin B Whiting, and Michael Y Oh. Accuracy of the freehand pass technique for ventriculostomy catheter placement: retrospective assessment using computed tomography scans. *Journal of Neurosurgery*, 108(1):88–91, 2008.

[66] Udaya K Kakarla, Louis J Kim, Steven W Chang, Nicholas Theodore, and Robert F Spetzler. Safety and accuracy of bedside external ventricular drain placement. *Operative Neurosurgery*, 63(suppl_1):ONS162–ONS167, 2008.

[67] Peter YM Woo, Ben CF Ng, Jacob X Xiao, Daniel Wong, Andrew Seto, Sandy Lam, Carmen Yim, Hong-Yip Lo, Yin-Chung Po, Larry YW Wong, et al. The importance of aspirin, catheterization accuracy, and catheter design in external ventricular drainage-related hemorrhage: a multicenter study of 1002 procedures. *Acta Neurochirurgica*, pages 1–10, 2019.

[68] Faith C Robertson, Muhammad M Abd-El-Barr, Srinivasan Mukundan, and William B Gormley. Ventriculostomy-associated hemorrhage: a risk assessment by radiographic simulation. *Journal of neurosurgery*, 127(3):532–536, 2017.

[69] Andrew Benjamin Cutler, Shervin Rahimpour, Yameng Liu, Nandan Lad, Regis Kopper, and Patrick Codd. Development of augmented reality-based neuro-navigation system for use in external ventricular drain placement. In *Journal of Neurosurgery*, volume 126, pages A1438–A1438, 2017.

[70] David Low, Cheng Kiang Lee, Lee Lian Tay Dip, Wai Hoe Ng, Beng Ti Ang, and Ivan Ng. Augmented reality neurosurgical planning and navigation for surgical excision of parasagittal, falcine and convexity meningiomas. *British Journal of Neurosurgery*, 24(1):69–74, 2010.

[71] Mohammad Najafi and Robert Rohling. Single camera closed-form real-time needle trajectory tracking for ultrasound. In *Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling*, volume 7964, page 79641F. International Society for Optics and Photonics, 2011.

[72] Mohammad Najafi, Purang Abolmaesumi, and Robert Rohling. Single-camera closed-form real-time needle tracking for ultrasound-guided needle insertion. *Ultrasound in medicine & biology*, 41(10):2663–2676, 2015.

[73] Philipp J. Stolka, Pezhman Foroughi, Matthew Rendina, Clifford R. Weiss, Gregory D. Hager, and Emad M. Boctor. Needle guidance using handheld stereo vision and projection for ultrasound-based interventions. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 684–691, Cham, 2014. Springer International Publishing.

[74] Zhencheng Fan, Guowen Chen, Junchen Wang, and Hongen Liao. Spatial position measurement system for surgical navigation using 3-d image marker-based tracking tools with compact volume. *IEEE Transactions on Biomedical Engineering*, 65(2):378–389, 2017.

[75] Marta Kersten-Oertel, Pierre Jannin, and D Louis Collins. The state of the art of visualization in mixed reality image guided surgery. *Computerized Medical Imaging and Graphics*, 37(2):98–112, 2013.

[76] Antonio Meola, Fabrizio Cutolo, Marina Carbone, Federico Cagnazzo, Mauro Ferrari, and Vincenzo Ferrari. Augmented reality in neurosurgery: a systematic review. *Neurosurgical review*, pages 1–12, 2016.

[77] Visish M Srinivasan, Brent R O'Neill, Diana Jho, Donald M Whiting, and Michael Y Oh. The history of external ventricular drainage: Historical vignette. *Journal of neurosurgery*, 120(1):228–236, 2014.

[78] Herbert I Fried, Barnett R Nathan, A Shaun Rowe, Joseph M Zabramski, Norberto Andaluz, Adarsh Bhimraj, Mary McKenna Guanci, David B Seder, and Jeffrey M Singh. The insertion and management of external ventricular drains: an evidence-based consensus statement. *Neurocritical care*, 24(1):61–81, 2016.

[79] Shaun T O'Leary, Max K Kole, Devon A Hoover, Steven E Hysell, Ajith Thomas, and Christopher I Shaffrey. Efficacy of the ghajar guide revisited: a prospective study. *Journal of neurosurgery*, 92(5):801–803, 2000.

[80] Nicholas I Phillips and Nigel W John. Web-based surgical simulation for ventricular catheterization. *Neurosurgery*, 46(4):933–937, 2000.

[81] Olivier Cros, Morten Volden, Jens Jørgen Flaaris, Lars Chebørløv Brix, Christian Fischer Pedersen, Kim Vang Hansen, Ole Vilhelm Larsen, Lasse Riis Østergaard, and Jens Haase. Simulating the puncture of the human ventricle. In *10th Annual Medicine Meets Virtual Reality Conference*, 2002.

[82] G Krombach, A Ganser, Ch Fricke, V Rohde, M Reinges, J Gilsbach, and U Spetzger. Virtual placement of frontal ventricular catheters using frameless neuronavigation: an "unbloody training" for young neurosurgeons. *Minimally Invasive Neurosurgery*, 43(04):171–175, 2000.

[83] Cristian Luciano, Pat Banerjee, G Michael Lemole, and Fady Charbel. Second generation haptic ventriculostomy simulator using the immersivetouch^{TM}system. *Studies in health technology and informatics*, 119:343, 2005.

[84] P Pat Banerjee, Cristian J Luciano, G Michael Lemole Jr, Fady T Charbel, and Michael Y Oh. Accuracy of ventriculostomy catheter placement using a head-and hand-tracked high-resolution virtual reality simulator with haptic feedback. *Journal of Neurosurgery*, 107(3):515–521, 2007.

[85] G Michael Lemole Jr, P Pat Banerjee, Cristian Luciano, Sergey Neckrysh, and Fady T Charbel. Virtual reality in neurosurgical education: part-task ventriculostomy simulation with dynamic visual and haptic feedback. *Neurosurgery*, 61(1):142–149, 2007.

[86] Clemens M Schirmer, J Bradley Elder, Ben Roitberg, and Darlene A Lobel. Virtual reality–based simulation training for ventriculostomy: an evidence-based approach. *Neurosurgery*, 73(suppl_1):S66–S73, 2013.

[87] Rachel Yudkowsky, Cristian Luciano, Pat Banerjee, Alan Schwartz, Ali Alaraj, G Michael Lemole Jr, Fady Charbel, Kelly Smith, Silvio Rizzi, Richard Byrne, et al. Practice on an augmented reality/haptic simulator and library of virtual brains improves residents' ability to perform a ventriculostomy. *Simulation in Healthcare*, 8(1):25–31, 2013.

[88] Bruce L Tai, Deborah Rooney, Francesca Stephenson, Peng-Siang Liao, Oren Sagher, Albert J Shih, and Luis E Savastano. Development of a 3d-printed external ventricular drain placement simulator. *Journal of neurosurgery*, 123(4):1070–1076, 2015.

[89] M. A. Lin, A. F. Siu, J. H. Bae, M. R. Cutkosky, and B. L. Daniel. Holoneedle: Augmented reality guidance system for needle placement investigating the advantages of three-dimensional needle shape reconstruction. *IEEE Robotics and Automation Letters*, 3(4):4156–4162, Oct 2018.

[90] Timur Kuzhagaliyev, Neil T Clancy, Mirek Janatka, Kevin Tchaka, Francisco Vasconcelos, Matthew J Clarkson, Kurinchi Gurusamy, David J Hawkes, Brian Davidson, and Danail Stoyanov. Augmented reality needle ablation guidance tool for irreversible electroporation in the pancreas. In *Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10576, page 1057613. International Society for Optics and Photonics, 2018.

[91] Ehsan Basafa, Pezhman Foroughi, Martin Hossbach, Jasmine Bhanushali, and Philipp Stolka. Visual tracking for multi-modality computer-assisted image guidance. In *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10135, page 101352S. International Society for Optics and Photonics, 2017.

[92] Yoshitaka Masutani, Takeyoshi Dohi, Fumitaka Yamane, Hiroshi Iseki, and Kintomo Takakura. Augmented reality visualization system for intravascular neurosurgery. *Computer aided surgery*, 3(5):239–247, 1998.

[93] Eduardo E Lovo, Juan C Quintana, Manuel C Puebla, Gonzalo Torrealba, José L Santos, Isidro H Lira, and Patricio Tagle. A novel, inexpensive method of image coregistration for applications in image-guided surgery using augmented reality. *Operative Neurosurgery*, 60(suppl_4):ONS–366, 2007.

[94] Philip J Edwards, Derek LG Hill, David J Hawkes, R Spink, Alan CF Colchester, A Strong, and M Gleeson. Neurosurgical guidance using the stereo microscope. In *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 555–564. Springer, 1995.

[95] Philip J Edwards, Andrew P King, Calvin R Maurer, Darryl A De Cunha, David J Hawkes, Derek LG Hill, Ronald P Gaston, Michael R Fenlon, A Jusczyzck, Anthony J Strong, et al. Design and evaluation of a system for microscope-assisted guided interventions (magi). *IEEE Transactions on Medical Imaging*, 19(11):1082–1093, 2000.

[96] Philip J Edwards, Laura G Johnson, David J Hawkes, Michael R Fenlon, Anthony J Strong, and Michael J Gleeson. Clinical experience and perception in stereo augmented reality surgical navigation. In *International Workshop on Medical Imaging and Virtual Reality*, pages 369–376. Springer, 2004.

[97] Ramin Shahidi, Bai Wang, Marc Epitaux, Robert Grzeszczuk, and John Adler. Volumetric image guidance via a stereotactic endoscope. *Medical Image Computing and Computer-Assisted Intervention*, pages 241–252, 1998.

[98] M Scholz, W Konen, S Tombrock, B Fricke, L Adams, M Von Duering, A Hentsch, L Heuser, and AG Harders. Development of an endoscopic navigation system based on digital image processing. *Computer Aided Surgery*, 3(3):134–143, 1998.

[99] Ralf A Kockro, Yeo Tseng Tsai, Ivan Ng, Peter Hwang, Chuangui Zhu, Kusuma Agusanto, Liang Xiao Hong, and Luis Serra. Dex-ray: augmented reality neurosurgical navigation with a handheld video probe. *Neurosurgery*, 65(4):795–808, 2009.

[100] D Inoue, B Cho, M Mori, Y Kikkawa, T Amano, A Nakamizo, K Yoshimoto, M Mizoguchi, M Tomikawa, J Hong, et al. Preliminary study on the clinical application of augmented reality neuronavigation. *Journal of Neurological Surgery Part A: Central European Neurosurgery*, 74(02):071–076, 2013.

[101] Frank Sauer, Ali Khamene, Benedicte Bascle, and Gregory J Rubino. A head-mounted display system for augmented reality image guidance: Towards clinical evaluation for imri-guided neurosurgery. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 707–716. Springer-Verlag, 2001.

[102] Calvin Maurer, Frank Sauer, Bo Hu, Benedicte Bascle, Bernhard Geiger, Fabian Wenzel, Filippo Recchi, Torsten Rohlfing, Christopher M Brown, Robert S Bakos, et al. Augmented reality visualization of brain structures with stereo and kinetic depth cues: System description and initial evaluation with head phantom. *Medical Imaging*, 2001:445–456, 2001.

[103] Yuichiro Abe, Shigenobu Sato, Koji Kato, Takahiko Hyakumachi, Yasushi Yanagibashi, Manabu Ito, and Kuniyoshi Abumi. A novel 3d guidance system using augmented reality for percutaneous vertebroplasty. *Journal of Neurosurgery: Spine*, 19(4):492–501, 2013.

[104] Kamyar Abhari, John SH Baxter, Elvis CS Chen, Ali R Khan, Terry M Peters, Sandrine de Ribaupierre, and Roy Eagleson. Training for planning tumour resection: augmented reality and human factors. *IEEE Transactions on Biomedical Engineering*, 62(6):1466–1477, 2014.

[105] Ye Li, Xiaolei Chen, Ning Wang, Wenyao Zhang, Dawei Li, Lei Zhang, Xin Qu, Weitao Cheng, Yueqiao Xu, Wenjin Chen, et al. A wearable mixed-reality holographic computer for guiding external ventricular drain insertion at the bedside. *Journal of Neurosurgery*, 1(aop):1–8, 2018.

[106] Ehsan Azimi, Camilo Molina, Alexander Chang, Judy Huang, Chien-Ming Huang, and Peter Kazanzides. Interactive training and operation ecosystem for

surgical tasks in mixed reality. In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, pages 20–29. Springer, 2018.

[107] Daipayan Guha, Naif M Alotaibi, Nhu Nguyen, Shaurya Gupta, Christopher McFaul, and Victor XD Yang. Augmented reality in neurosurgery: a review of current concepts and emerging applications. *Canadian Journal of Neurological Sciences*, 44(3):235–245, 2017.

[108] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on pattern analysis and machine intelligence*, 21(8):774–780, 1999.

[109] Marc-André Ameller, Bill Triggs, and Long Quan. Camera pose revisited-new linear algorithms. 2000.

[110] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[111] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2016.

[112] Diederick C Niehorster, Li Li, and Markus Lappe. The accuracy and precision of position and orientation tracking in the HTC Vive virtual reality system for scientific research. *i-Perception*, 8(3):177–205, 2017.

[113] Taylor Frantz, Bart Jansen, Johnny Duerinck, and Jef Vandemeulebroucke. Augmenting microsoft's hololens with vuforia tracking for neuronavigation. *Healthcare technology letters*, 5(5):221–225, 2018.

[114] M. A. Khalighi and M. Uysal. Survey on free space optical communication: A communication theory perspective. *IEEE Communications Surveys Tutorials*, 16(4):2231–2258, 2014.

[115] Christopher V. Poulton, Ami Yaacobi, David B. Cole, Matthew J. Byrd, Manan Raval, Diedrik Vermeulen, and Michael R. Watts. Coherent solid-state lidar with silicon photonic optical phased arrays. *Opt. Lett.*, 42(20):4091–4094, Oct 2017.

[116] J. Notaros, M. Raval, M. Notaros, and M. R. Watts. Integrated-phased-array-based visible-light near-eye holographic projector. In *2019 Conference on Lasers and Electro-Optics (CLEO)*, pages 1–2, 2019.

[117] Robert Konrad, Emily A. Cooper, and Gordon Wetzstein. 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, 2016. ACM.

[118] K. Rathinavel, H. Wang, A. Blate, and H. Fuchs. An extended depth-at-field volumetric near-eye augmented reality display. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2857–2866, 2018.

[119] Pierre Gemayel, Bruno Colicchio, Alain Dieterlen, and Pierre Ambs. Cross-talk compensation of a spatial light modulator for iterative phase retrieval applications. *Applied optics*, 55(4):802–810, 2016.

[120] Martin Peckerar, Robert Bass, and Kee Woo Rhee. Sub-0.1 μ electron-beam lithography for nanostructure development. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, 18(6):3143–3149, 2000.

[121] Martin Persson, David Engström, and Mattias Goksör. Reducing the effect of pixel crosstalk in phase only spatial light modulators. *Optics express*, 20(20):22334–22343, 2012.

[122] X Sun, Y Zhang, P-C Huang, Mario Dagenais, M Peckerar, and A Varshney. Proximity effect on nanophotonic phased arrays. In *Frontiers in Optics*, pages JW3A–101. Optical Society of America, 2019. To appear.

[123] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[124] Alan J Fenn, Donald H Temme, William P Delaney, and William E Courtney. The development of phased-array radar technology. *Lincoln Laboratory Journal*, 12(2):321–340, 2000.

[125] Manan Raval, Ami Yaacobi, and Michael R. Watts. Integrated visible light phased array system for autostereoscopic image projection. *Opt. Lett.*, 43(15):3678–3681, Aug 2018.

[126] A. Grinenko, M. P. MacDonald, C. R. P. Courtney, P. D. Wilcox, C. E. M. Demore, S. Cochran, and B. W. Drinkwater. Tunable beam shaping with a phased array acousto-optic modulator. *Opt. Express*, 23(1):26–32, Jan 2015.

[127] Hajime Nagahara, Yasushi Yagi, and Masahiko Yachida. Super wide viewer using catadioptrical optics. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '03, pages 169–175, New York, NY, USA, 2003. ACM.

[128] Kiyoshi Kiyokawa. A wide field-of-view head mounted projective display using hyperbolic half-silvered mirrors. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–4, Washington, DC, USA, 2007. IEEE Computer Society.

[129] Jannick P. Rolland, Myron W. Krueger, and Alexei Goon. Multifocal planes head-mounted displays. *Appl. Opt.*, 39(19):3209–3215, Jul 2000.

[130] Kurt Akeley, Simon J. Watt, Ahna Reza Girshick, and Martin S. Banks. A stereo display prototype with multiple focal distances. *ACM Trans. Graph.*, 23(3):804–813, August 2004.

[131] P. Chakravarthula, D. Dunn, K. Akit, and H. Fuchs. FocusAR: Auto-focus augmented reality eyeglasses for both real world and virtual imagery. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2906–2916, 2018.

[132] X. Xia, Y. Guan, A. State, P. Chakravarthula, K. Rathinavel, T. Cham, and H. Fuchs. 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, 2019.

[133] Kaan Akşit, Ward Lopes, Jonghyun Kim, Peter Shirley, and David Luebke. Near-eye varifocal augmented reality display using see-through screens. *ACM Transactions on Graphics (TOG)*, 36(6):189, 2017.

[134] Jonghyun Kim, Youngmo Jeong, Michael Stengel, Kaan Akşit, Rachel Albert, Ben Boudaoud, Trey Greer, Joohwan Kim, Ward Lopes, Zander Majercik, Peter Shirley, Josef Spjut, Morgan McGuire, and David Luebke. Foveated AR: Dynamically-foveated augmented reality display. *ACM Trans. Graph.*, 38(4):99:1–99:15, July 2019.

[135] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[136] Andrew Maimone, Gordon Wetzstein, Matthew Hirsch, Douglas Lanman, Ramesh Raskar, and Henry Fuchs. Focus 3D: Compressive accommodation display. *ACM Trans. Graph.*, 32(5):153–1, 2013.

[137] Douglas Lanman and David Luebke. Near-eye light field displays. *ACM Transactions on Graphics (TOG)*, 32(6):220, 2013.

[138] Andrew Maimone and Henry Fuchs. Computational augmented reality eyeglasses. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 29–38. IEEE, 2013.

[139] Gordon Wetzstein, Douglas Lanman, Matthew Hirsch, and Ramesh Raskar. Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Trans. Graph.*, 31(4), July 2012.

[140] Fu-Chung Huang, Kevin Chen, and Gordon Wetzstein. 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, July 2015.

[141] Seungjae Lee, Changwon Jang, Seokil Moon, Jaebum Cho, and Byoungho Lee. Additive light field displays: Realization of augmented reality with holographic optical elements. *ACM Trans. Graph.*, 35(4):60:1–60:13, July 2016.

[142] Seungjae Lee, Changwon Jang, Seokil Moon, Byounghyo Lee, Jaebum Cho, and Byoungho Lee. See-through light field displays for augmented reality. In *SIGGRAPH ASIA 2016 Virtual Reality Meets Physical Reality: Modelling and Simulating Virtual Humans and Environments*, SA '16, pages 3:1–3:2, New York, NY, USA, 2016. ACM.

[143] Vitor F. Pamplona, Manuel M. Oliveira, Daniel G. Aliaga, and Ramesh Raskar. Tailored displays to compensate for visual aberrations. *ACM Trans. Graph.*, 31(4):81:1–81:12, July 2012.

[144] Fu-Chung Huang, Gordon Wetzstein, Brian A. Barsky, and Ramesh Raskar. Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays. *ACM Trans. Graph.*, 33(4):59:1–59:12, July 2014.

[145] Rahul Narain, Rachel A. Albert, Abdullah Bulbul, Gregory J. Ward, Martin S. Banks, and James F. O'Brien. Optimal presentation of imagery with focus cues on multi-plane displays. *ACM Trans. Graph.*, 34(4):59:1–59:12, July 2015.

[146] Nitish Padmanaban, Yifan Peng, and Gordon Wetzstein. Holographic near-eye displays based on overlap-add stereograms. *ACM Trans. Graph.*, 38(6), November 2019.

[147] Lantian Mi, Chao Ping Chen, Yifan Lu, Wenbo Zhang, Jie Chen, and Nizamuddin Maitlo. Design of lensless retinal scanning display with diffractive optical element. *Opt. Express*, 27(15):20493–20507, Jul 2019.

[148] Jie Chen, Lantian Mi, Chao Ping Chen, Haowen Liu, Jinghui Jiang, and Wenbo Zhang. Design of foveated contact lens display for augmented reality. *Opt. Express*, 27(26):38204–38219, Dec 2019.

[149] Changwon Jang, Kiseung Bang, Gang Li, and Byoungho Lee. Holographic near-eye display with expanded eye-box. *ACM Trans. Graph.*, 37(6), December 2018.

[150] G. A. Koulieris, K. Akit, M. Stengel, R. K. Mantiuk, K. Mania, and C. Richardt. Near-eye display and tracking technologies for virtual and augmented reality. *Computer Graphics Forum*, 38(2):493–519, 2019.

[151] A Lohmann and D Paris. Synthesis of binary holograms. *IEEE Journal of Quantum Electronics*, 2(4):153–153, 1966.

[152] James P Waters. Holographic image synthesis utilizing theoretical methods. *Applied physics letters*, 9(11):405–407, 1966.

[153] Mark E Lucente. Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1):28–35, 1993.

[154] Hiroshi Yoshikawa, Susumu Iwase, and Tadashi Oneda. Fast computation of fresnel holograms employing difference. In *Practical Holography XIV and Holographic Materials VI*, volume 3956, pages 48–56. International Society for Optics and Photonics, 2000.

[155] Mark Lucente and Tinsley A Galyean. Rendering interactive holographic images. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 387–394. ACM, 1995.

[156] Shuhong Xu, Farzam Farbiz, Sanjeev Solanki, Xinan Liang, and Xuewu Xu. Adaptive computation of computer-generated holograms. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '08, pages 35:1–35:2, New York, NY, USA, 2008. ACM.

[157] M. Hossein Eybposh, Nicholas W. Caira, Mathew Atisa, Praneeth Chakravarthula, and Nicolas C. Pégard. DeepCGH: 3D computer-generated holography using deep learning. *Opt. Express*, 28(18):26636–26650, Aug ts , url = http://www.opticsexpress.org/abstract.cfm?URI=oe-28-18-26636, doi = 10.1364/OE.399624,.

[158] Yifan Peng, Suyeon Choi, Nitish Padmanaban, Jonghyun Kim, and Gordon Wetzstein. Neural holography with camera-in-the-loop training. *ACM Transactions on Graphics (TOG)*, 39(6), 2020.

[159] Remo Ziegler, Simon Bucheli, Lukas Ahrenberg, Marcus Magnor, and Markus Gross. A bidirectional light field-hologram transform. In *Computer Graphics Forum*, volume 26, pages 435–446. Wiley Online Library, 2007.

[160] Remo Ziegler, Simone Croci, and Markus Gross. Lighting and occlusion in a wave-based framework. In *Computer Graphics Forum*, volume 27, pages 211–220. Wiley Online Library, 2008.

[161] Kaan Akşit. Patch scanning displays: spatiotemporal enhancement for displays. *Opt. Express*, 28(2):2107–2121, Jan 2020.

[162] Ralph W Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[163] Kazunobu Masuda, Yusuke Saita, Ryusuke Toritani, Peng Xia, Kouichi Nitta, and Osamu Matoba. Improvement of image quality of 3d display by using optimized binary phase modulation and intensity accumulation. *Journal of Display Technology*, 12(5):472–477, 2015.

[164] Praneeth Chakravarthula, Yifan Peng, Joel Kollin, Henry Fuchs, and Felix Heide. Wirtinger holography for near-eye displays. *ACM Trans. Graph.*, 38(6), November 2019.

[165] Mihir Parikh. Corrections to proximity effects in electron beam lithography. i. theory. *Journal of Applied Physics*, 50(6):4371–4377, 1979.

[166] Allen M Carroll. Proximity-effect correction with linear programming. *Journal of Applied Physics*, 52(1):434–437, 1981.

[167] DGL Chow, JF McDonald, DC King, W Smith, K Molnar, and AJ Steckl. An image processing approach to fast, efficient proximity correction for electron beam lithography. *Journal of Vacuum Science & Technology B: Microelectronics Processing and Phenomena*, 1(4):1383–1390, 1983.

[168] MC Peckerar, S Chang, and CRK Marrian. Proximity correction algorithms and a co-processor based on regularized optimization. i. description of the algorithm. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, 13(6):2518–2525, 1995.

[169] Christie R Marrian, Steven Chang, and Martin C Peckerar. Proximity correction for electron beam lithography. *Optical Engineering*, 35(9):2685–2693, 1996.

[170] Martin Peckerar, David Sander, Ankur Srivastava, Adakou Foli, and Uzi Vishkin. Electron beam and optical proximity effect reduction for nanolithography: New results. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, 25(6):2288–2294, 2007.

[171] Pengcheng Li. A review of proximity effect correction in electron-beam lithography. *arXiv preprint arXiv:1509.05169*, 2015.

[172] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[173] Felix Heide, Steven Diamond, Matthias Nießner, Jonathan Ragan-Kelley, Wolfgang Heidrich, and Gordon Wetzstein. Proximal: Efficient image optimization using proximal algorithms. *ACM Trans. Graph.*, 35(4), July 2016.

[174] Amir Beck and Marc Teboulle. Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications*, pages 42–88, 2009.

[175] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[176] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[177] K. Rathinavel, G. Wetzstein, and H. Fuchs. 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, 2019.

[178] Michal Makowski, Maciej Sypek, Izabela Ducin, Agnieszka Fajst, Andrzej Siemion, Jaroslaw Suszek, and Andrzej Kolodziejczyk. Experimental evaluation of a full-color compact lensless holographic display. *Optics express*, 17(23):20840–20846, 2009.

[179] Zixiang Lu and Yuji Sakamoto. Holographic display method for volume data by volume rendering. *Optics express*, 27(2):543–556, 2019.

[180] T. Todd Elvins. A survey of algorithms for volume visualization. *SIGGRAPH Comput. Graph.*, 26(3):194–201, August 1992.

[181] E. Keppel. Approximating complex surfaces by triangulation of contour lines. *IBM Journal of Research and Development*, 19(1):2–11, Jan 1975.

[182] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal surface reconstruction from planar contours. *Commun. ACM*, 20(10):693–702, October 1977.

[183] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, August 1987.

[184] Peter Shirley and Allan Tuchman. A polygonal approximation to direct scalar volume rendering. In *Proceedings of the 1990 Workshop on Volume Visualization*, VVS '90, pages 63–70, New York, NY, USA, 1990. Association for Computing Machinery.

[185] Gregory M. Nielson and Bernd Hamann. The asymptotic decider: Resolving the ambiguity in marching cubes. In *Proceedings of the 2nd Conference on Visualization '91*, VIS '91, pages 83–91, Washington, DC, USA, 1991. IEEE Computer Society Press.

[186] H. K. Tuy and L. T. Tuy. Direct 2-d display of 3-d objects. *IEEE Computer Graphics and Applications*, 4(10):29–34, Oct 1984.

[187] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[188] Marc Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261, July 1990.

[189] Lee Westover. Footprint evaluation for volume rendering. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, pages 367–376, New York, NY, USA, 1990. Association for Computing Machinery.

[190] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS '01.*, pages 29–538, Oct 2001.

[191] M. Berger, J. Li, and J. A. Levine. A generative model for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 25(4):1636–1650, April 2019.

[192] Jan Novák, Iliyan Georgiev, Johannes Hanika, Jaroslav Křivánek, and Wojciech Jarosz. Monte carlo methods for physically based volume rendering. In *ACM SIGGRAPH 2018 Courses*, SIGGRAPH '18, New York, NY, USA, 2018. Association for Computing Machinery.

[193] Michal Makowski, Maciej Sypek, Andrzej Kolodziejczyk, and Grzegorz Mikula. Three-plane phase-only computer hologram generated with iterative fresnel algorithm. *Optical Engineering*, 44(12):125805, 2005.

[194] Nobuyuki Masuda, Tomoyoshi Ito, Takashi Tanaka, Atsushi Shiraki, and Takashige Sugie. Computer generated holography using a graphics processing unit. *Optics Express*, 14(2):603–608, 2006.

[195] Kyoji Matsushima and Masahiro Takai. Recurrence formulas for fast creation of synthetic three-dimensional holograms. *Applied Optics*, 39(35):6587–6594, 2000.

[196] Kyoji Matsushima, Hirohito Nishi, and Sumio Nakahara. Simple wave-field rendering for photorealistic reconstruction in polygon-based high-definition computer holography. *Journal of Electronic Imaging*, 21(2):023002, 2012.

[197] Yuji Sakamoto and Yoshinao Aoki. Autostereoscopic visualization of volume data using computer-generated holograms. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 90(11):31–39, 2007.

[198] Zixiang Lu and Yuji Sakamoto. Holographic display methods for volume data: polygon-based and mip-based methods. *Applied optics*, 57(1):A142–A149, 2018.

[199] Takuo Yoneyama, Chanyoung Yang, Yuji Sakamoto, and Fumio Okuyama. Eyepiece-type full-color electro-holographic binocular display with see-through vision. In *Digital Holography and Three-Dimensional Imaging*, page DW2A.11. Optical Society of America, 2013.

[200] Eugene Hecht and A. R. Ganesan. *Optics.* Pearson, Chennai, 4th edition, 2008.

[201] Ruofei Du and Amitabh Varshney. Social Street View: Blending Immersive Street Views with Geo-Tagged Social Media. In *Proceedings of the 21st International Conference on Web3D Technology*, Web3D, pages 77–85. ACM, July 2016.

[202] Ruofei Du, Sujal Bista, and Amitabh Varshney. Video Fields: Fusing Multiple Surveillance Videos into a Dynamic Virtual Environment. In *Proceedings of the 21st International Conference on Web3D Technology*, Web3D, pages 165–172. ACM, July 2016.

[203] Ruofei Du, David Li, and Amitabh Varshney. Experiencing a Mirrored World With Geotagged Social Media in Geollery. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI, pages 1–4. ACM, May 2019.

[204] Ruofei Du, David Li, and Amitabh Varshney. Geollery: A Mixed Reality Social Media Platform. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI)*, CHI, pages 1–13. ACM, May 2019.

[205] Ruofei Du, David Li, and Amitabh Varshney. Project Geollery.com: Reconstructing a Live Mirrored World With Geotagged Social Media. In *Proceedings of the 24th International Conference on Web3D Technology*, Web3D, pages 1–9. ACM, July 2019.

[206] Eric Krokos, Catherine Plaisant, and Amitabh Varshney. Virtual Memory Palaces: Immersion Aids Recall. *Springer VR 2018*, 1(15):1–20, May 2018.

[207] Hani Nejadriahi and Volker J. Sorger. On-chip integrated all-optical fast fourier transform: Design and analysis. In *Frontiers in Optics 2017*, page JW4A.46. Optical Society of America, 2017.

[208] Desmond C. Ong, Sanjeev Solanki, Xinan Liang, and Xuewu Xu. Analysis of laser speckle severity, granularity, and anisotropy using the power spectral density in polar-coordinate representation. *Optical Engineering*, 51(5):1 – 8, 2012.

[209] Hsueh-Chien Cheng, Antonio Cardone, and Amitabh Varshney. Volume Segmentation Using Convolutional Neural Networks With Limited Training Data. In *Proceedings of 2017 IEEE International Conference on Image Processing*, ICIP. IEEE, September 2017.

[210] Hsueh-Chien Cheng, Antonio Cardone, Somay Jain, Eric Krokos, Kedar Narayan, Sriram Subramaniam, and Amitabh Varshney. Deep-learning-assisted Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–14, January 2018.

[211] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on*

Computer Graphics and Interactive Techniques, SIGGRAPH 95, page 419428, New York, NY, USA, 1995. Association for Computing Machinery.

[212] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM Trans. Graph.*, 15(1):136, January 1996.

[213] Xiaoxu Meng, Quan Zheng, Amitabh Varshney, Gurprit Singh, and Matthias Zwicker. Real-time Monte Carlo Denoising with the Neural Bilateral Grid. In Carsten Dachsbacher and Matt Pharr, editors, *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association, 2020.