Rendering and Display for Multi-Viewer Tele-Immersion

Andrew R. Nashel

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

Chapel Hill
2010

Approved by:

Henry Fuchs, Advisor

Ketan Mayer-Patel, Reader

Leonard McMillan, Reader

Leandra Vicci, Reader

Greg Welch, Reader

Abstract

Andrew R. Nashel: Rendering and Display for Multi-Viewer Tele-Immersion.
(Under the direction of Henry Fuchs.)

Video teleconferencing systems are widely deployed for business, education and personal use to enable face-to-face communication between people at distant sites. Unfortunately, the two-dimensional video of conventional systems does not correctly convey several important non-verbal communication cues such as eye contact and gaze awareness. Tele-immersion refers to technologies aimed at providing distant users with a more compelling sense of remote presence than conventional video teleconferencing.

This dissertation is concerned with the particular challenges of interaction between groups of users at remote sites. The problems of video teleconferencing are exacerbated when groups of people communicate. Ideally, a group tele-immersion system would display views of the remote site at the right size and location, from the correct viewpoint for each local user. However, is is not practical to put a camera in every possible eye location, and it is not clear how to provide each viewer with correct and unique imagery.

I introduce rendering techniques and multi-view display designs to support eye contact and gaze awareness between groups of viewers at two distant sites. With a shared 2D display, virtual camera views can improve local spatial cues while preserving scene continuity, by rendering the scene from novel viewpoints that may not correspond to a physical camera. I describe several techniques, including a compact light field, a plane sweeping algorithm, a depth dependent camera model, and video-quality proxies, suitable for producing useful views of a remote scene for a group local viewers.

The first novel display provides simultaneous, unique monoscopic views to several users, with fewer user position restrictions than existing autostereoscopic displays. The second is a random hole barrier autostereoscopic display that eliminates the viewing zones and user position requirements of conventional autostereoscopic displays, and provides unique 3D views for multiple users in arbitrary locations.

To my parents.

Acknowledgments

I would like to first thank my advisor Henry Fuchs for his guidance in the research and writing of this dissertation. Throughout my time in graduate school, he provided me with a wealth of opportunities for cutting edge research and was always a source of inspiration.

I also thank my committee members, Ketan Mayer-Patel, Leonard McMillan, Leandra Vicci, and Greg Welch. Their contributions include many of the topics in this dissertation and their advice led to substantial improvements in the quality of the work. Their comments and suggestions on the dissertation were vital in crafting a suitably rigorous and complete work.

This research would not have been possible without the support of the many staff members of the Department of Computer Science. Herman Towles was invaluable as a project leader and his technical expertise was critical to the success of my work. Andrei State, John Thomas, David Harrison, and Bil Hays were exceptionally helpful. The entire administrative and support staff, especially Janet Jones, took care of everything outside of the technical work.

Much of this research was conducted in collaboration with my fellow UNC students, including Peter Lincoln, Andrei Ilie, and Ruigang Yang. They are exceptional researchers and wonderful colleagues.

This work was supported by funding from Sandia National Laboratories/California, guided by Christine Yang with contributions by Corbin Stewart, and Cisco Systems, led by Mod Marathe and Bill Mauchly. I thank them for recognizing and contributing to this research in its earliest stages.

Many thanks to my friends, including fellow UNC students Ajith Mascarenas, Sharif Razzaque, Yuanxin Liu, Greg Coombe, David Marshburn, Mark Harris, Eli Broadhurst, Dan Samarov, and many more.

Thank you, Megan Orrell! You kept me sane while I finished this dissertation.

And a special thanks to my parents, Cheryl and David, to whom this dissertation is dedicated. Their support is what made this possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With recent increases in available network bandwidth and dropping costs for video equipment, *video teleconferencing* has become widely deployed for business, education and personal use. It enables visual, face-to-face communication between people at distant sites. Unfortunately, conventional systems are limited to capturing and displaying two-dimensional imagery, which does not provide a compelling or convincing sense of presence for multiple viewers at each site. Such systems do not correctly convey eye contact, gaze awareness and certain depth cues, and so are far from replicating the experience of face-to-face conversation. Without these nonverbal cues, it is more difficult for users to fully convey meaning, resulting in reduced trust and turn-taking, and increases in pauses and interruptions.

These problems are exacerbated when groups of people try to communicate with video teleconferencing. With a conventional system, every distant user looking at the camera appears to be making eye contact with all of the viewers. Ideally, we would like to have cameras located at the positions corresponding to every eye of the distant viewers. This imagery would be transmitted to the distant site and presented at the right size and location so that the users each see personalized views. However, it is not practical to put a camera in every possible eye location, and it is not clear how to provide each viewer with correct and unique imagery.

Motivated by the shortcomings in existing video teleconferencing systems, our long-term goal is to create systems in which remote participants can be visualized as if they were sitting across the table, creating the impression of face-to-face conversation. The realization of this interface poses significant scientific and engineering challenges in computer graphics, computer vision, networking and display technology. Measuring the effectiveness of a video teleconferencing system is also a challenge, with many different factors contributing to users' perception of quality. However, there is some general agreement in research and industry on the importance of factors such as resolution, scale, eye contact, and latency.

Video teleconferencing refers to the wide range of systems that support remote interaction via video, using cameras, networking, and displays. The terms *tele-immersion* and *telepresence* usually describe more advanced systems that provide enhanced levels of computer-supported remote presence. Others have used these terms to differentiate

between quality levels of conventional video teleconferencing systems or to set their product apart from other conventional systems. In this dissertation, these terms are used to describe systems that address the camera placement and shared display problems for interaction between groups.

The components of a tele-immersion system include capture cameras, scene reconstruction, networking, rendering, and display. This dissertation deals with each of these components and how they can be improved for groups of viewers. Contributions include prototype systems from capture to display, with algorithms for generating views of the remote scene and two novel multi-user displays: one that presents unique 2D views to several viewers and a 3D display that provides unique stereoscopic views to multiple simultaneous users in arbitrary locations. This work is focused on a two-site scenario, with several viewers at each location.

1.1 Motivation and goals

The goal of tele-immersion is to allow users to feel as if they are present in and can interact with a real, remote location. Tele-immersion systems sample the remote environment and transmit a representation from the distant location to the local site for display. One of the most active areas of tele-immersion is teleconferencing between individuals. Compared to conventional video teleconferencing, tele-immersive systems allow remote users to collaborate with a much higher sense of presence, as if they were in the same shared location. Also, the cost and time savings of teleconferencing versus travel have made expensive tele-immersion systems cost effective for many situations.

Multi-user, or group tele-immersion refers to tele-immersive interaction between groups of users at remote sites. Group tele-immersion presents additional challenges, particularly with respect to the display of the remote environment to a group of users sharing the same physical environment. Our early vision of a group tele-immersion is depicted in Figure 1.1, showing two small groups at distant sites interacting as if they are seated on opposite sides of a single table. The distant users are displayed at the proper scale and in the right position.

In order to support social cues such as eye contact and gaze awareness, a tele-immersion system must be able to provide unique views to each viewer. Unfortunately, in systems with a shared 2D display, as in Figure 1.1, all of the local users see the same imagery of the remote site. When the remote scene is rendered and displayed for a single viewpoint, viewers at any other location sees an incorrect views of the scene. Without a correct view, it is difficult for a user to accurately discern these important social cues. One method for improving the shared view is to render the scene from a *virtual camera* in a location not corresponding to any real camera at the distant site. Such a novel view can be used to improve spatial cues while preserving scene continuity. However, this is extremely difficult to do well, and the quality of such reconstructions for arbitrary views is dramatically lower than provided by common high resolution cameras.

Another option is to use separate displays for each local user, but this may limit sharing in the local environment or make natural interaction difficult. Users could wear head mounted displays and see representation of the remote scene, but camera images

Figure 1.1: Our initial vision of a group tele-immersion scenario, simulated with projected 2D imagery.

of those viewers are compromised by the head gear, preventing eye contact.

Multi-view displays provide different imagery for different viewing angles from the same display area. The most common type of multi-view display is *stereoscopic* (or stereo), which presents different images to the left and right eyes of a viewer to enhance 3D perception. Stereoscopic display is often accomplished using eye wear with passively polarized lenses or rapidly alternating shuttered glasses. Like head mounted displays, these encumbrances that cover the eyes inhibit the eye gaze that such systems could potentially support.

Autostereoscopy is a method of displaying stereo imagery to a viewer without the need for special glasses. There are three basic types of autostereoscopic display: holographic, volumetric, and parallax. Most *autostereoscopic* displays are parallax displays, using either a barrier or lenticular sheet, and emit different 2D images across the viewing field. These multiple views come at the cost of dividing display pixels among the various views. In order to support a wide range of viewer positions, with distinct views over a meter or more, dozens of views are required. This requires a sacrifice of resolution that is an unacceptable trade off. Some autostereoscopic displays can present correct stereo views to multiple users, but there are typically significant restrictions on user location

and movement. For a group tele-immersion system, a display that allows for natural user positions and spacing is desirable.

1.2 Approach

With these challenges in mind, this dissertation describes the research path for developing techniques and displays within the framework of a tele-immersion system. A typical system includes the following components:

1. *Scene acquisition* using video cameras to capture imagery of the environment.
2. *Scene reconstruction* to generate models of the scene.
3. *Encoding* of the scene and *transmission* to the remote site.
4. *Rendering* of the scene for display.
5. *Displays* for presenting the final video output.

Virtual camera techniques are a combination of scene acquisition and rendering to present a novel view that does not correspond to a physical camera. Multi-view displays present unique views to different viewing positions through a combination of rendering and display hardware. With these improvements, we can better support eye contact, gaze awareness and presentation of depth cues.

1.2.1 Virtual camera techniques

To capture the correct view for a particular user, a camera should be placed at the corresponding position for that user's eye at the distant site. While it is prohibitively expensive to directly capture a scene from all possible viewpoints, we have observed that the participants viewpoints usually remain at a constant height (eye level) during most video teleconferencing sessions. Therefore, we can restrict the possible viewpoint to be within a virtual plane without sacrificing much of the realism, and in doing so we significantly reduce the number of required cameras. Based on this observation, we developed a reconstruction technique called the *Line Light Field* (LLF) that uses light field-style rendering [55] to guarantee the quality of the synthesized views, using the imagery from a linear array of cameras.

We developed a complete system for group video teleconferencing using the Line Light Field, which we called called *Group Tele-Immersion* (GTI) [112, 71]. A virtual camera view was synthesized, compressed, and transmitted over a high-speed Internet connection to a remote site, where it was rendered for a life-sized, projected 2D display. The full-duplex prototype system between the Sandia National Labs, California and the University of North Carolina at Chapel Hill has been able to synthesize views at interactive rates, and has been used for regular video conferencing between the sites.

The GTI system was limited to reconstruction of a small group of users, all at the same distance from the camera array. We wanted to support much larger, wall-sized

displays to create the illusion of a single large space divided by a window. We also wanted to handle users at arbitrary locations, particularly at different distances from the display. Because of the wider field-of-view for the larger display, it was not possible to use the LLF algorithm without significant distortion and many additional cameras. Another realization was that reconstructing a model of the remote scene was less important than simply determining the final color of the display pixels.

We developed a reconstruction algorithm based on plane sweeping [111] for use with a 2D video wall, consisting of a modest number of cameras placed around several large flat panel displays. This *Telepresence Wall* system also introduced new depth dependent rendering techniques and allowed for 3D depth estimation of scene objects. However, the reconstruction using the plane sweeping algorithm was of considerably lower quality than conventional high-resolution video imagery. To improve the visual quality of the remote scene while still allowing for arbitrary virtual camera positions, we developed another method for rendering each remote individual at full camera image quality.

1.2.2 Multi-view displays

During the development of these systems, we desired a novel view for each user, but this is not possible with shared 2D displays. If we can improve the display itself, then we can provide these unique views. We have developed two different types of multi-view display to provide perspectively correct views to multiple users.

The first design is for a non-stereo, monoscopic multi-view display for untracked groups of users [57]. Because autostereoscopic display requires localizing viewing zones to within the *interpupillary distance* (IPD) of a user (the spacing between their eyes), untracked users must remain in a fixed or limited number of positions. Most autostereoscopic displays have a limited number of unique viewing zones, typically eight to ten, limiting the total width of the viewing zones to approximately half a meter at the optimal user distance. If we remove the requirement for stereoscopic display, we can provide each user a unique monoscopic display over a wider viewing zone at the required user distance, allowing for less restricted viewing positions and more natural interaction.

We can provide multiple monoscopic views to small groups of users with both barrier and lenticular technology. The optimal display will maximize brightness and resolution for each user, but this requires a non-uniform barrier across the display. Because non-uniform lenticular barriers are difficult to manufacture, we show how to use additional pixels to create guard bands between views with a regular barrier to provide unique and correct imagery to each viewer.

The second display we developed was designed to support autostereoscopic viewing for groups of users. The non-uniform barrier autostereoscopic *Random Hole Display* (RHD) [70] is a parallax barrier autostereoscopic display that uses a barrier pattern of randomly distributed holes instead of a conventional regular pattern. When combined with viewer tracking, the RHD design offers a number of capabilities that are not found in most existing autostereoscopic displays, including stereo display for multiple users in arbitrary viewing positions.

With regular barrier, multi-user autostereoscopic displays, untracked users must remain in a limited number of viewing zones or they will see incorrect imagery. In au-

tostereoscopic display systems with user tracking, multiple viewers are usually not supported because individual display pixels will be seen from multiple views. Because of the regular barrier pattern these visual conflicts are localized and can cover large areas of the display, depending on the viewer positions. We consider the interference between views as aliasing, as the regular barrier interacts with the regular display pixel pattern. By randomizing hole distribution in the barrier, these visual conflicts are distributed across the viewing area as high frequency noise, and can be minimized by changing the parameters of the barrier design.

1.3 Thesis statement

In between conventional 2D teleconferencing and an ideal rendering of remote presence, there exists a continuum of systems that provide groups of viewers with varying degrees of immersive experience.

Through a combination of rendering techniques and display engineering, we can provide more personalized experiences to individuals in a group of viewers via: (1) virtual camera views that improve local spatial cues while preserving scene continuity, and (2) multi-view displays that trade off stereoscopic viewing for a wider range of viewing positions or use randomization to eliminate the spatial viewing conflicts that occur at regular intervals.

1.4 Contributions

This dissertation presents the following innovations in group tele-immersion and multi-view display:

1. Virtual camera techniques that allow for rendering of the remote scene from novel viewpoints that may not correspond to a camera in the remote location, and can render a shared 2D view optimized for a group of viewers.

2. A multiple monoscopic view display that provides unique 2D views for multiple users with fewer user position restrictions than existing autostereoscopic displays. A random hole barrier autostereoscopic display that eliminates the viewing zones and user position requirements of conventional autostereoscopic displays, and provides unique 3D views for multiple users in arbitrary locations.

1. Virtual camera techniques

1.a Line light field The Line Light Field is a low computational cost method for rendering unique views of a scene. It is particularly suited to reconstructing a set of objects at the same distance from the camera array, such as participants in a teleconferencing session. Instead of performing dense, computationally intensive 3D scene acquisition, we exploit the fact that participants motion is usually limited to lateral motions at the

same eye level during a video teleconferencing session. This natural restriction allows us to use a limited number of cameras to capture important views. Based on this observation, we have developed a real-time acquisition-through-rendering algorithm based on light field rendering. The realism of the synthesized view is derived directly from camera images.

1.b Depth-dependent camera The Telepresence Wall is a videoconferencing system with a large, wall-sized display, that supports several simultaneous viewers and captures the remote environment through an array of cameras. We have developed several new techniques for rendering virtual camera views of a remote scene where the users may be at different distances from the display.

For a large scale display, a single perspective view is desirable to provide a continuous display with natural feeling of perspective depth. With a centered perspective view, participants to the side of the display and looking straight ahead, will be appear to be looking to the outside of the display. This prevents local eye contact between users at the side of the display. We want to provide local eye contact between users at the outside of the display, while maintaining the continuous perspective view of the entire scene. We introduce the *depth-dependent camera* to generate views of the scene where the rendering viewpoint changes based on the depth of the object in the scene. The view rays from the center of projection through the image plane are straight, but the center of projection changes for objects at different depths.

1.c Video silhouettes Our final rendering technique addresses the image quality issues inherent in view synthesis. We want to preserve video image quality while generating a single virtual camera view of the remote scene. To accomplish this, we create video silhouettes for each remote user by matching segmented shapes between camera images and eliminating duplicates, and then project the silhouette into the remote scene using virtual camera techniques, including the use of depth-dependent camera techniques.

2. Multi-view displays

2.a Monoscopic multi-view display In a group tele-immersion system without tracking, multiple monoscopic view parallax displays allow for fewer user position restrictions and higher resolution and brightness than existing autostereoscopic displays.

Without user tracking, existing autostereoscopic displays have a fundamental complication in providing unique imagery to a group of users. Each viewing zone must be at IPD scale, but with each user separated at comfortable interpersonal space. The total width covered at the user distance is the number of views multiplied by the pixel zone width. Although more views could be provided, this sacrifices horizontal resolution.

Instead of providing stereoscopic views to each user, we can adapt the parallax barrier or lenticular screen to provide wide angle viewing zones to the users. This will eliminate stereoscopic viewing, but each user will be able move comfortably within a small area, while presenting a unique view of the remote scene to each user.

2.b Random hole display The Random Hole Display is designed to eliminate the zones found in regular barrier and lenticular autostereoscopic displays, allowing individual users to move freely without experiencing jumping between zones. In many conventional autostereoscopic displays, users are restricted to certain positions within viewing zones. Those displays with fixed barriers or lenticular sheets do not allow freedom of movement between zones without a noticeable jump. Furthermore, transitioning between zones can create a reverse stereo image for the viewer. Some autostereoscopic displays use tracking and an adaptive barrier to allow a single user to move freely in a larger viewing area without zones, but these displays do not support groups of users.

Autostereoscopic displays that support multiple users generally require users to remain in one of several fixed viewing zones. Fixed barrier, lenticular, and adaptive barrier displays have no conflict between views in the optimal case of correct viewer positioning, but have significant, localized conflicts without restricted view locations. The RHD accepts the conflicts between views, but the design minimizes the amount of conflict and distributes that interference evenly across the display area.

Furthermore, fixed zone autostereoscopic displays produce the same visual quality per user for any number of total users. This maximizes total visual display quality if there are as many users as zones for these displays, but with fewer viewers some display pixels are not seen by any viewer. With user tracking, the RHD design can provide displays that provide higher quality views per user with fewer users, with a graceful degradation in per user view quality as more users view the display.

1.5 Dissertation outline

The remainder of this dissertation is organized as follows:

Chapter 2 is a review of related and background work. The two primary topics in this review are tele-immersion systems and multi-view displays. The first section covers conventional 2D single user and group teleconferencing systems, including a discussion of the problems with such systems that prompted the research of this dissertation. Next is a review of multi-view displays, with an emphasis on autostereoscopic displays and multiple viewers. Various scene reconstruction techniques, as they relate to human subjects, are briefly reviewed. The final section is a review of existing multiscopic tele-immersion systems.

The body of the dissertation is divided into four chapters covering the tele-immersion systems and multi-view displays we have developed. The group tele-immersion system of Chapter 3 captures a remote group of users using a linear array of cameras and uses a simple method for reconstruction and rendering from a novel viewpoint for 2D display. In Chapter 4, the Telepresence Wall system describes a more sophisticated reconstruction method for a remote group of users, a depth-dependent rendering algorithm, and a high quality video-based algorithm. Chapter 5 covers a multi-user, multi-view display that provides a unique monoscopic view to each of several local users and Chapter 6 describes the multi-user autostereoscopic Random Hole Display.

The dissertation is concluded in Chapter 7 with a review of the work and discussion of possible directions for future work. Readers primarily interested in tele-immersion

rendering algorithms can review Chapter 3 and Chapter 4. Readers interested in multiscopic displays could jump to Chapter 5 for the multi-view monoscopic display and Chapter 6 for the autostereoscopic Random Hole Display.

Chapter 2

Background and related work

In this chapter, I review prior developments in video teleconferencing, view synthesis, and multi-view displays. These topics are the basis of the group tele-immersion rendering techniques and displays described later in this dissertation. Section 2.1 describes the history of video teleconferencing, video conferencing standards, and several modern systems, which typically use 2D display and camera imagery, ranging from small, personal video communication systems to large, multi-display, multi-viewer conferencing systems.

Section 2.2 discusses the two central problems for promoting natural interaction - eye contact and gaze awareness - along with other factors in system design. Section 2.3 explores several solutions to the eye contact problem, including view synthesis and optical path approaches. Section 2.4 describes the use of multi-view displays to support broader gaze awareness in video conferencing systems. Several types of parallax display technologies are reviewed along with existing commercial autostereoscopic displays.

2.1 Conventional video teleconferencing

Video teleconferencing systems are the collection of components necessary to capture imagery of an environment, encode it for transmission, send the data to the remote site, decode the imagery, and then display it. Typically, these components are a video camera connected to a computer with a network connection to another computer with a display. Most video conferencing systems support simultaneous capture, transmission, and playback of audio alongside of the video content. Video teleconferences can be between two sites or multiple locations. In this section I provide a brief history of video teleconferencing, from the earliest conceptions to recent commercial systems.

2.1.1 Early history of video teleconferencing

An early concept of a video teleconferencing system, called the *telephonoscope*, was described shortly after the invention of the telephone in the mid-1870s. George du Maurier sketched the device as a fictional invention of Thomas Edison, and it is considered a prediction of both television and video teleconferencing [24]. Figure 2.1 depicts his sketch,

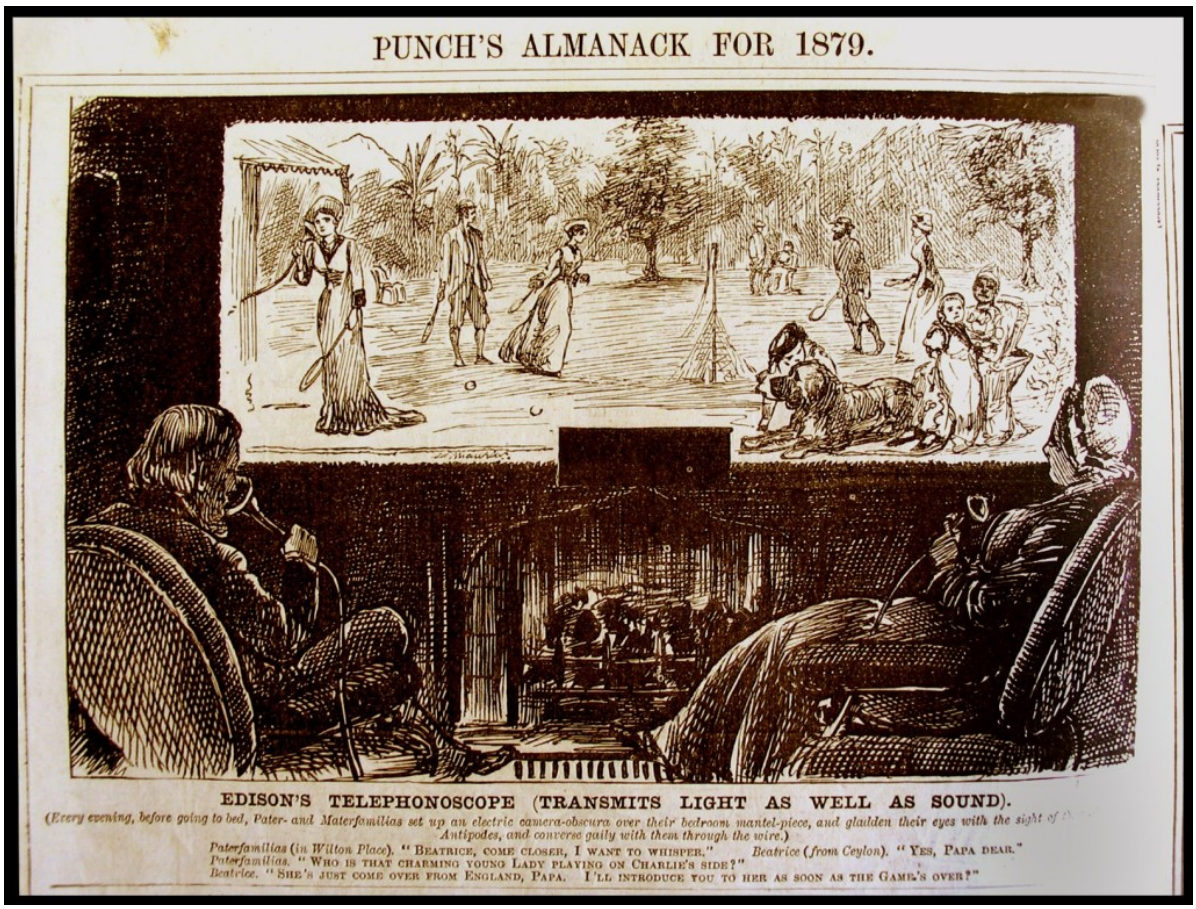


Figure 2.1: The telephonoscope was sketched by George du Maurier as a fictional invention of Thomas Edison in the December 9, 1878 edition of Punch magazine. The text reads:

EDISON'S TELEPHONOSCOPE (TRANSMITS LIGHT AS WELL AS SOUND).
(Every evening, before going to bed, Pater- and Materfamilias set up an electric camera-obscura over their bedroom mantel-piece, and gladden their eyes with the sight of their Children at the Antipodes, and converse gaily with them through the wire.)
Paterfamilias (in Wilton Place). "Beatrice, come closer. I want to whisper." Beatrice (from Ceylon). "Yes, Papa dear."
Paterfamilias. "Who is that charming young Lady playing on Charlie's side?"
Beatrice. "She's just come over from England, Papa. I'll introduce you to her as soon as the Game's over!"

This figure uses a photo from `http://www.flickr.com/photos/seriykotik/208841133/`, available under a Creative Commons Attribution-Noncommercial license.

which not only shows a couple viewing a large scale display, but includes a scenario of a man interacting through sound and imagery with his daughter, who is located on the other side of the world. Other early conceptions of video teleconferencing include Alexander Graham Bell's 1924 pronouncement that "the day would come when the man at the telephone would be able to see the distant person to whom he was speaking" [3].

Of course, the first technology that made steps towards these notions of interactive visual communications was the television. The earliest conceptions of a practical device to capture and transmit imagery one way included various electro-mechanical devices in the mid-1800s that transmitted still imagery via telegraph. The first device to transmit a moving image was built by Scottish inventor John Logie Baird in 1924, and used a spinning mechanical disk to produce an image consisting of 30 vertically scanned lines [7]. He transmitted the first long distance television signal in 1927, between London and Glasgow, and introduced the first color transmission in 1928, using disks with different primary color filters, along with stereoscopic television.

Although several inventors had demonstrated various electronic capture or display components, Philo T. Farnsworth is credited with the invention of the all-electronic television system [25], which he demonstrated in 1928. The electronic television was commercialized by several companies in the 1930s, notably RCA in the United States and Marconi-EMI in Great Britain. Many stations began broadcasting in various formats for electronic receivers during this time. To resolve the conflicts between the many formats, the National Television System Committee (NTSC) issued a technical standard for black-and-white television in 1941, consisting of 525 lines of horizontal resolution at 30 frames per second, with a 4:3 aspect ratio and frequency modulation (FM) for sound. In 1950, NTSC issued a backwards-compatible color standard. The first NTSC-compatible color cameras were demonstrated in 1953.

The first commercial video teleconferencing system was the AT&T Picturephone, introduced in 1964 [23]. It offered a 5.5" $\times$ 5" monochrome CRT display with 250 interlaced lines of vertical resolution updated at 30 frames per second. Designed for a viewing distance of 36 inches, the camera was embedded above the display and had an adjustable field-of-view, allowing for one or two users to be captured. The system used analog encoding for short distance transmission and switched to digital transmission at 6.3 Mb per second for distances over 6 miles. The Picturephone project foresaw many future developments in video conferencing, including high resolution, color transmission, multi-party calls, and inter-frame-based video compression. However, the low quality of the video and the high price of the service hindered widespread adoption.

2.1.2 Video conferencing standards

In the 1980s, commercial products were introduced by numerous companies, among them Mitsubishi, PictureTel, and Compression Labs, but owing to their high cost few were widely deployed [33]. During this time, the International Telecommunication Union (ITU) began setting standards for low bandwidth video transmission. These standards specified methods for encoding the video data and were aimed at real-time applications such as video teleconferencing. The main goal for video encoding is to reduce the size of the video data through compression, and these standards allow for interoperability

between systems from different manufacturers.

The earliest standard for video coding was H.120, but per pixel coding was insufficient to meet quality demands with limited bandwidth. By the end of the decade, the first practical standard for interactive video encoding, called H.261, was developed for use with digital transmission networks such as ISDN[77]. H.261 uses block-based encoding, with different sampling rates for luminance and chrominance, and inter-picture prediction with motion vectors. The standard supports CIF (352x288) and QCIF (76x144) resolution video encoded at rates ranging from 40kbps to 2Mbps. Although now obsolete, H.261 was the foundation for almost all future video encoding standards.

During the same timeframe, the MPEG-1 and 2 standards were developed by the International Organization for Standardization's (ISO) Moving Picture Experts Group to support video and audio coding [11]. These standards are used in devices ranging from video CDs, MP3 players, DVDs and digital cable broadcats. In contrast to the single frame coding delay of H.261, which made it suitable for interactive video, MPEG-1 and 2 are designed for non-interactive broadcast of multimedia data, using bidirectionally predicted frames for motion compensation.

In 1996, the ITU introduced the H.263 standard for low-bitrate video teleconferencing along with the H.323 standard for controlling communication sessions over IP networks [86]. H.263 evolved from H.261 and the MPEG standards, and is widely deployed for content delivery. In 1999, the ISO standardized MPEG-4, which included a wide range of compression methods capable of covering a wide range of bitrates, and supported both broadcast and interactive video applications. The most recent major standard, H.264/AVC, introduced in 2003, is a joint work of the ITU and ISO and provides similar quality at lower bitrates to the earlier standards [107]. It is used for applications including Internet streaming, high definition television and movies, and video teleconferencing.

2.1.3 Modern video conferencing systems

By the 1990s, Internet Protocol-based (IP) videoconferencing became mainstream. Early IP-based video conferencing systems were limited by low resolution and frame rates, but they were inexpensive compared to dedicated hardware systems. One of the first systems was CU-SeeMe, a Macintosh (and later, Windows) based program that used an inexpensive video camera to provide 16-level gray scale video at 320x240 resolution[22]. The system also supported multi-party communication, with reflector software that replicated and distributed streams to many viewers.

The decreasing cost of imaging sensors and the ability to encode and decode video in software has led to the wide deployment of small, inexpensive webcams. These inexpensive video cameras are attached to a computer or directly to a network. The first uses of webcams were as unidirectional monitoring devices, but they were quickly adapted for use for video teleconferencing, and are commonly supported by instant messenger programs. The cameras tend to be small, allowing for attachment to a monitor or integration into a display bezel, and are now found in many laptop computers. Another increasingly common use for small embedded cameras is in mobile devices such as phones. With the advent of high speed wireless networks, bi-directional video communication is

possible between mobile devices.

Figure 2.2: A side view of a Cisco Telepresence 3000 system. Three HD cameras are located in the cluster in the front-top of the central display. The placement of the cameras partially obscures the top of the screen but improves eye contact by reducing the parallax between the camera and the displayed remote users.

Today's mid-range systems typically use dedicated hardware that includes a camera and a display. They are often designed as a dual phone/video conferencing device and typically use standard video compression and transmission protocols [98, 83].

The wide adoption of high definition television (HDTV) has driven down the cost of capture and display hardware. HDTV resolution cameras and large displays are now relatively inexpensive consumer electronics commodities. Another major technological advance is the availability of high bandwidth networking between remote sites.

These advances have led to a growing market for large-scale, high-end video teleconferencing systems. Such systems are typically refered to as "telepresence" to differentiate themselves from lower-end video conferencing systems. Manufacturers include HP [36], Polycom [84], TANDBERG [99], and Cisco [13]. They typically provide one or more high resolution cameras each with a matched, large display to present each of the distant users at life-size and in the appropriate locations.

For example, Cisco Telepresence [13] provides high definition 1920x1080 resolution, 60 frames per second video on one or several large flat panel displays. HD cameras and spatial audio microphones capture the local environment and are encoded with H.264 compression for transmission over high speed, dedicated lines. The system also supports multi-site conferences with view switching, and includes the capture and display hardware, and room treatments such as a conference table and lighting. The three panel Cisco Telepresence 3000 system is shown in Figure 2.2.

2.2 Human factors in video teleconferencing

Decades of video teleconferencing research and development provides some consensus on the importance of certain factors to the quality and usefulness of such systems. Many studies have been performed to test and measure one or several of these factors, often in the context of a technology developed to address that particular issue [88]. The fundamental goal of all teleconferencing systems is to allow people at distant sites to interact effectively.

Two of the most important factors in effective communication are *eye contact* and *gaze awareness*. These are closely related elements, but they are not identical even though they are often used interchangeably. We use the definitions of Monk and Gale of these two terms: gaze awareness is the "ability to gauge the current object of someone elses visual attention," and eye contact is "knowing whether someone is looking at you" [66]. They also define *partial gaze awareness* as knowing the general direction that someone is looking, such as up, down, left, or right.

Eye contact is really a special case of gaze awareness, often referred to as mutual gaze awareness, where two people are simultaneously aware that the other person is looking at their eyes. Gaze awareness is the more general understanding of where the other person is looking. In fact, it is possible for video conferencing systems to provide neither, one or the other, or both. For example, a system using synthetic avatars may convey gaze awareness, but not support eye contact [75].

The first factor, eye contact, is an immediate problem for most video teleconferencing systems because the camera is at least slightly offset from the eye level displayed, making direct eye contact impossible. This occurs in both one-to-one conferences and also with groups of users. Humans are extremely sensitive to small visual parallax in eye contact and are able to accurately judge when others are looking at their eyes [28]. A video camera must be located within 5 degrees vertically and 1 degree horizontally of the image of the distant user's eyes to avoid the appearance of looking away [9].

The second factor, gaze awareness, is particularly a problem for groups because a shared 2D display makes it difficult to discern where a remote person may be looking. This is commonly known as the Mona Lisa effect: if the distant person is looking directly at the camera, they appear to be making eye contact with all of the local viewers. Similarly, if they are looking to one side, then they appear to be looking to that side for every viewer. This is an advantage in the asymmetric situation of a speech to an audience, where the speaker would like to convey the illusion of making eye contact with all viewers. However, this is a disadvantage when a user wants to address a specific person. An ideal system for interactive conferencing would provide correct eye contact between distant viewers and also correctly convey gaze direction.

There is a large body of research on these two interrelated factors, and how their presence or absence has an effect on the effectiveness of communication using video teleconferencing. Studies suggest that eye contact is a critical factor in determining who speaks next in a group conversation [39, 103]. If a video teleconferencing system does not correctly convey gaze awareness, then the turn-taking process is inhibited [104]. Proper eye gaze and spatial awareness are also important for developing trust between groups. Video teleconferencing systems that do not support these cues hinder

trust formation, but spatially faithful conferencing systems improve cooperation [74]. Eye contact also serves to express emotion, monitors feedback, and communicates the nature of interpersonal relationships [88]. Full gaze awareness has not been studied as much as eye contact, but it has been shown to improve task performance versus eye contact alone [66].

Other technical factors for the design of a video teleconferencing system include resolution and latency. While high resolution is obviously useful for reading small text or observing distant objects, sufficient resolution is also important for evaluating gaze direction. The original Multiview user study [73], found that users had difficulty in determining eye contact even when the remote user was looking directly into the camera. The low resolution (800×600) of the displays made it difficult to clearly see the pupils of the remote users and gaze directions were sometime incorrectly evaluated. The next iteration of the Multiview system [74], using higher resolution (1024×768) cameras and displays, provided enough detail for users to correctly evaluate gaze direction. This suggests that meeting a minimum resolution is sufficient for conveying this cue.

Latency is another important factor for the usability of video conferencing systems. In this context, it refers to the time delay between capture at the remote site and display at the distant location. Several components contribute to overall system latency, include camera, processing, network, and display latencies [5]. Synchronization between cameras for view synthesis or other components can also increase overall latency. Immersive systems are very sensitive to latency requirements, and will likely require <100ms end-to-end for effective experiences [52]. Higher delays may lead to users talking over each other, which in turn leads to abrupt halts in conversation.

With higher resolution cameras and displays and faster networks, these factors are mitigated to some degree. The fundamental problems of eye contact and gaze awareness remain. Systems that have been developed to support improved eye contact in teleconferencing can be divided into a two general classes: shared optical path and novel view synthesis. There is a distinct set of solutions for gaze awareness, often involving multi-view displays. An overview of existing solutions to each of these problems follows.

2.3 Eye contact

2.3.1 Optical path approaches

Besides minimizing the camera-to-display offset, another possible approach to improving eye contact is to engineer a system to capture and display from the same viewing direction. In some cases, this is accomplished by embedding cameras into the display surface. A front projected display screen might use louvers to prevent projected light from hitting the camera while allowing light from the scene to be captured [51]. Another approach is to embed sensors directly in a display panel by replacing some of the display elements with image sensors, and using synthetic camera techniques to create a unified camera image [102].

A common method for aligning the capture and display is the use of half-silvered mirrors or glass at a 45 degree angle to provide partial transmission of light from behind,

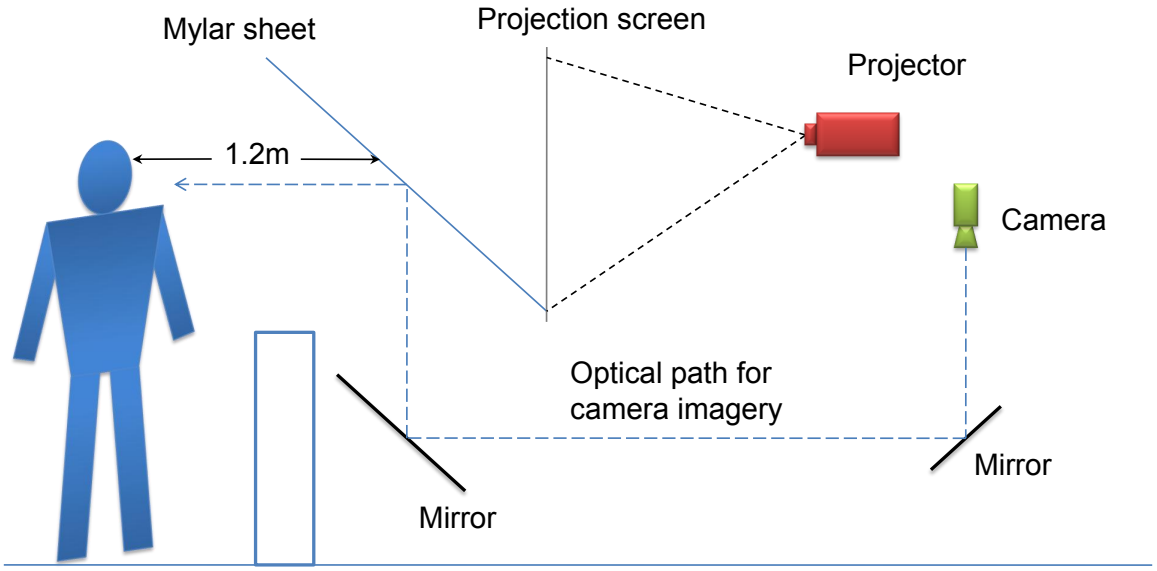


Figure 2.3: A profile view of the geometry of France Telecom's 'magic wall' telepresence system, showing the semi-reflective mylar sheet, through which the viewer sees the projected imagery of the remote site, and which reflects the telephoto camera image.

while light is also partially reflected from the front. This technique, called Pepper's ghost, was originally used in theaters to provide the illusion of a floating figure. Modern systems that produce this illusion have been used to support performances and speeches, with a projection on a large half-silver mirror in place of the remote participant [69]. This technique is also commonly used in teleprompter devices that allow a person making a speech to maintain eye contact with a camera while reading overlaid text.

Video conferencing systems have used these mirrors to provide eye contact by placing the camera on one side of the mirror and the display on the other to align the capture and display gaze axis [18]. For example, the France Telecom 'magic wall' uses a mylar sheet and also several regular mirrors to provide display and telephoto image capture [100]. Figure 2.3 depicts the paths of display imagery and the capture axis. The main problem with this technology is that only half the light in any one direction is transmitted or reflected, which reduces the apparent brightness and requires a more sensitive camera. Another problem is that the angled half-silvered mirror must block the entire display surface, which increases the dimensions required for the display itself.

The blue-c project is an immersive projection and 3D capture environment that is intended for collaborative design between distant individuals [30]. Although users are required to wear shutter glasses for stereoscopic viewing, the system provides several interesting solutions that may be applicable to the eye gaze problem. The blue-c system consists of a surround projection environment with screens made of switchable liquid crystal flat panels. This allows the screens to be used as a rear-projection surface, and then be switched to a clear state to allow a camera to image the user "through" the walls. A 3D model of the user is computed from multiple video streams, and transmitted to the remote site for stereoscopic display.

2.3.2 View synthesis

The ability to synthesize views from novel positions is an alternative to the bulkiness and lower light levels of optical path approaches. Instead of directly rendering a 3D model, view synthesis uses existing 2D imagery to generate new views of a scene. This allows views to be generated from positions where it is either impractical or physically impossible to place a camera. In addition to supporting eye contact by addressing the camera and display offset problem, novel synthesized views can be used to generate imagery that offers a wider field-of-view, higher resolution, or different perspectives. I describe view synthesis techniques used in several video conferencing systems to improve eye contact.

Image warping

A recent approach to eye gaze correction has been warping camera views in software to simulate accurate gaze direction. Although the viewer is not actually looking directly at the camera, they appear to be making eye contact with their distant collaborator. Several systems track the user's eyes and generate new imagery to correct the gaze direction. Gemmell et al. [27] synthesize new eyes and texture map the rest of the face onto a 3D head model. Jerald and Daily [43] warp the imagery of the eyes within the existing video frame so that the eyes appear to be looking at the camera even though the head is turned slightly away. Yang and Zhang [114] present a system that tracks the face using two video cameras, matches features between the two images to generate a 3D model, and warps the model to the correct orientation.

Stereo algorithms

A classic method for generating novel views is through dense stereo reconstruction to build a 3D model of a scene. Stereovision is one of the oldest and most active research topics in computer vision (see [90] for a more complete survey). Stereo algorithms take in two or more images and match feature points between them. Given the disparity of a feature in the two images and knowledge of the camera locations, the depth of that feature in space can be determined, as shown in Figure 2.4.

While many stereo algorithms obtain high-quality results by performing global optimizations, today only correlation-based stereo algorithms are able to provide a dense (per pixel) depth map in real-time. Correlation-based algorithms can be accelerated using either special hardware [26, 46, 108, 49, 17] or assembly level instruction optimization (such as Intel's MMX and SSE extension sets) [68, 35, 34, 82]. Overall these algorithms are quite fragile in practice. The calculated depth can contain substantial outliers due to scene lighting, occlusions, and specular highlights. Increasing the fidelity of scene acquisition leads to higher reconstruction latency and lower frame rates [67].

Several tele-immersion systems use stereo algorithms to generate novel views. The Virtual Team User Environment (VIRTUE) project developed several types of tele-collaboration systems, include a tele-cubicle that performed dense stereo reconstruction to generate virtual camera views corresponding to the remote participants [47]. This complex system required both custom hardware and software, with a multi-stage video

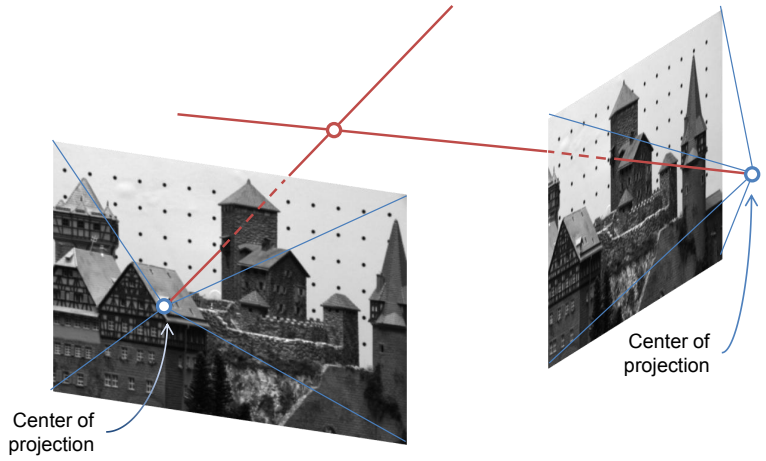


Figure 2.4: Depth from stereo correspondences in a pair of images. Rays from two cameras through similar image regions can be triangulated to find object depths.

processing pipeline that included foreground/background segmentation, image rectification, stereo disparity calculation, head tracking, 3D warping, and composition into a virtual scene. Later refinements included multiple stereo camera pairs that generated several depth maps of the scene and then merged them into a single surface based on the desired viewpoint [16].

Light field rendering

Recently, Image-Based Modeling and Rendering (IBMR) methods have become a popular alternative for synthesizing novel views. The basis for IBMR is reconstructing the *plenoptic function* that describes the flow of light in all positions and in all directions [64]. With a complete plenoptic function, novel views can be easily synthesized by plugging the location and directions for the novel views into the plenoptic function. A class of IBMR methods, called *Light Field Rendering* (LFR), uses many images to pre-record the plenoptic function [55, 29, 95]. LFR methods often achieve a stunning level of realism without using any geometric information. However, applying LFR directly to 3D teleconferencing would require hundreds of cameras, making real-time acquisition and transmission impractical. Schirmacher et al. extended LFR with per-pixel depth information computed with a classic stereo algorithm [91]. Their approach allows real-time online view synthesis, but fidelity is limited by the quality and the speed of the stereo algorithm (1-2 frames/second).

Visual Hulls

An alternative reconstruction technique that is amenable to real-time computation is to use silhouette information to construct an object's *visual hull*. The visual hull can be thought of as a conservative shell that encloses the actual object [53]. Visual hull computation does not require exhaustive matching; therefore, it is quite efficient and robust. Matusik et al. designed an efficient method to compute and shade visual hulls

from silhouette images, allowing real-time rendering of a dynamic scene from a large viewing volume [62]. Lok presented a novel technique to accelerate visual hull computation using commodity graphics hardware [59]. However, these approaches cannot handle concave objects, which results in less than satisfactory close-up views of such objects.

2.4 Gaze awareness

Recall that gaze awareness is more complex than eye contact. To support gaze awareness, a system must correctly depict the direction that each viewer is looking to the other participants. We first discuss single viewer per site gaze awareness, where one viewer is captured from cameras in locations corresponding to each of the remote participants. It is also possible to synthesize views corresponding to those locations from different camera arrangements.

2.4.1 Single viewer per site gaze awareness

Many systems have been developed to improve support for gaze awareness for a single viewer per site. Most of these also support eye contact with a remote viewer, and several support more than two sites interacting simultaneously. If there are several users depicted on screen, multiple cameras must be used to correctly convey gaze awareness to all parties. However, this does not solve the eye contact problem, and generally requires that users stay in a fixed location corresponding to a camera position.

The Hydra system simulates a four-way meeting with a display-camera pair for each of the remote users [92]. The camera above each display allows the system to convey gaze awareness, but the offset between camera and display inhibits eye contact. The MAJIC system captures imagery from a set of several cameras placed behind a semi-transparent screen, allowing for eye contact and gaze awareness [76]. Gaze awareness, besides signaling attention to other people, is also important for identifying attention to objects in the scene. The Clearboard project used gaze awareness to improve shared drawing experiences for two users [41].

A problem for supporting eye gaze awareness for single users at multiple sites is the rapidly increasing number of simultaneous video streams required as more users participate. In a one-to-one session, only one stream per user is required. With three sites, two streams are required for each pair of users for a total of 6. With 4, 12 streams are required, and so on. The GAZE-2 system supports a hybrid approach with several cameras but only a single video stream, by selecting a view based upon eye tracking [105]. The video is mapped onto a plane in a 3D environment, and the plane is turned to face in the direction that the user is looking. This conveys eye contact to any of the users and informs the other viewers about which way people are looking.

2.4.2 Multiple viewers per site gaze awareness

To correctly convey gaze awareness for multiple local viewers, a system must be able to provide unique views to all of the participants. This requires the ability to display distinct imagery to each local viewer. *Multi-view* displays provide different imagery to viewers in different locations.

Natural gaze awareness limits the use of wearable equipment that might cover the eye, ruling out light polarization [10] or time-division multiplexing techniques that require users to wear polarizing lenses or shutter glasses. The three broad classes of autonomous multi-view technologies that do not require such encumbrances include holographic, volumetric, and parallax.

Holographic displays are typically small and require large amounts of data [31], and volumetric displays are small and typically cannot produce opacity for multiple viewers [44], so these two classes are not widely used for video teleconferencing systems. Parallax displays, based on barriers, lenticular lens sheets, or integral lens sheets placed in front of a display surface, can feasibly be made to support life size imagery of video conferencing participants. They produce a stereoscopic effect by displaying a different image to the left and right eyes of a viewer. Barrier and lenticular displays are common and widely used in existing display systems, and are available commercially. Another practical option for multi-view displays is the use of retroreflective material.

Parallax barrier displays

The most basic kind of autostereoscopic display is a two view parallax barrier. These displays use either a fixed or switchable barrier to present left and right eye views to a single, centrally located viewer. Another common type of multi-view display supports up to 8-16 different views in a set of viewing zones. This type of display allows the viewer to move and experience correct 3D views from various positions or for multiple simultaneous viewers to see different 3D imagery. Commercial systems include a barrier based display from NewSight, a lenticular screen display from Philips, and an optical element projection system from Holografika [72, 81, 2]. Because these displays use a regular pattern barrier or lenticular lens sheet, the same stereo views repeat across the viewing area at regular intervals, leading to potential viewing zone conflicts between multiple users.

Most parallax barrier displays use a regular display, either flat panel LCD or plasma or a CRT in conjunction with a barrier. The barrier consists of an optically opaque film with holes or slits that allows light through. We show a close-up view in Figure 2.5. The barrier sheet is placed a small distance in front of the display panel, usually on the order of a few millimeters. There is a single barrier hole for each a group of subpixels, typically 8-10, across a line of the display. The barrier hole distributes the light from these pixels across the viewing area in front of the display, so that at a certain distance, the left and right eyes of a viewer see two different images. This produces a stereoscopic view which provides an additional sense of depth to the viewer.

A second class of autostereoscopic displays actively tracks the viewer in order to provide correct imagery from a wider range of viewing positions. One of the more extensive

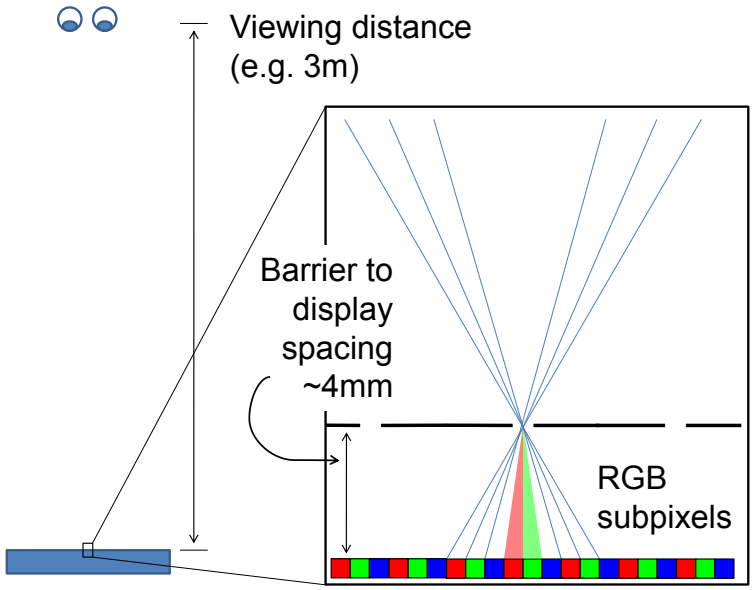


Figure 2.5: Parallax barriers spatially multiplex a display and limit light from the display to passage through particular holes, thus limiting the possible viewing directions.

examples is the Varrier display [89], consisting of 35 LCD panels with conventional film barriers. The visibility of the display pixels is determined by tracking the user's eye positions and illuminating the correct pixels for each eye.

Another method for autostereoscopic display is a variable barrier pattern. Several prototype systems have been developed, including Perlins NYU autostereoscopic display [78]. This display uses an active light blocking shutter that changes in response to the tracked user head position. However, brightness and quality are limited by the active barrier. Similarly, the Dynallax display [79] uses an active LCD barrier for rendering up to 4 distinct views. The combination of tracking and dynamic barrier allows the system to maximize the use of the backing display pixels, improving the resolution and brightness for a single user. The main drawback to a dynamic barrier is that the barrier LCD panel significantly reduces the brightness of the display. Each of these displays are not suitable for a group tele-immersion scenario because each is limited to one or two viewers.

Lenticular displays

Lenticular barrier displays operate in a similar fashion to parallax barrier displays, but with a series of long lenses called lenticules instead of barrier holes. These lenticules are narrow convex-planar lenses and are packed to form a sheet which can be fixed to the front of a display. Each lenticule directs the light from a particular subpixel in a certain direction, as shown in Figure 2.6. Although the directional multiplexing is similar to that of a barrier film, a lenticular sheet transmits significantly more light, leading to a brighter display. The main tradeoff is that the views are not as sharp as those in a

barrier-based display because the simple lens shape cannot precisely focus the light.

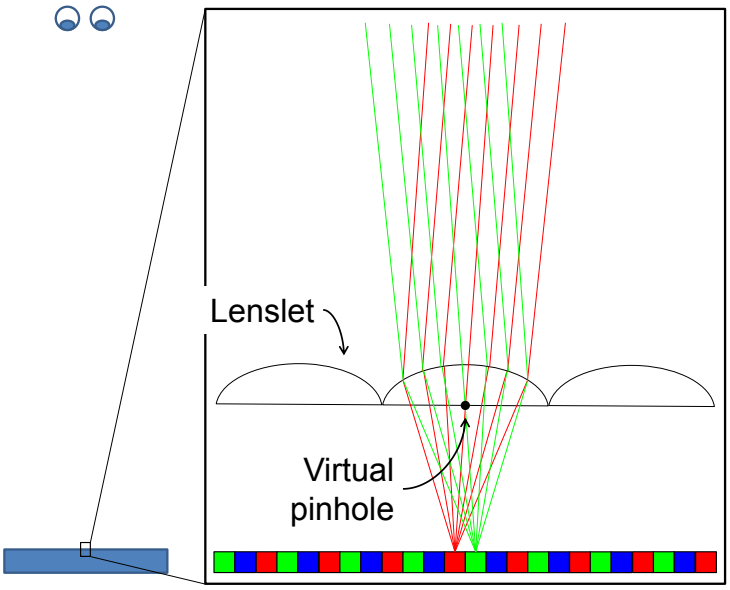


Figure 2.6: Lenticules spatially multiplex a display by directing the light from different pixels in different directions with a series of lenslets.

A notable 3D capture and display system based on lenticular screens is the MERL 3D TV system [63]. Using an array of projectors and cameras, the system captures and displays 16 views for stereoscopic high resolution (1024×768) imagery. The system operates in two modes, either front or rear projection. The rear projection configuration uses back-to-back matched lenticular sheets with a diffuser to multiplex the projected imagery, while the front projection configuration uses a single lenticular sheet in front of a retroreflective screen. Although the system is not bi-directional, the low latency of the system is suitable for teleconferencing.

Retroreflective displays

A retroreflective screen reflects light back in the direction from which it arrived using micro-optical elements, such as mirrors or glass beads, as shown in Figure 2.7. When combined with conventional projectors, a retroreflective screen can be used as a multi-view display. A user sitting near to one projector shining on a retroreflective screen will primarily see the imagery from that projector alone. Several users can each have their own projector to generate several spatially multiplexed views in front of the screen. This kind of multi-view display has been used to implement teleconferencing systems that support gaze awareness [73, 74]. The screens do diffuse light to a limited degree leading to some crosstalk between views. For this reason, it is difficult to support stereoscopic views with two closely spaced projectors.

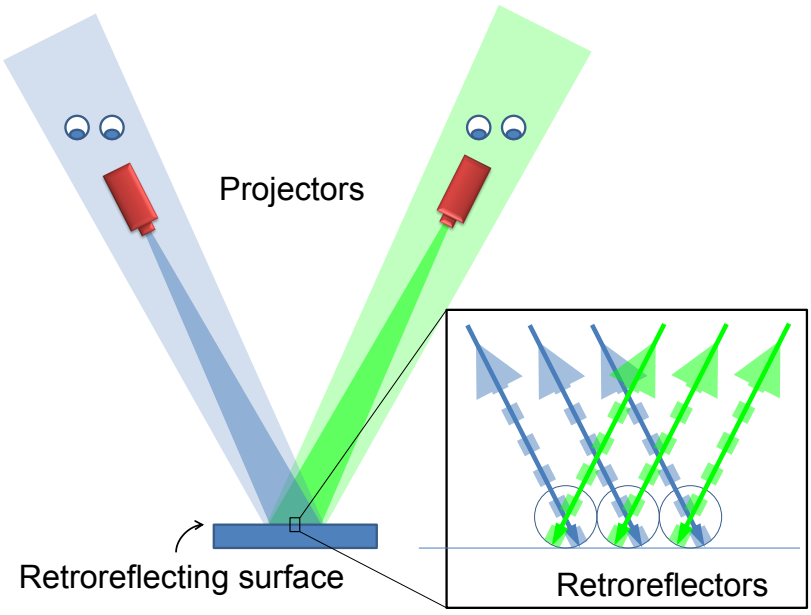


Figure 2.7: Retroreflective surfaces reflect light rays along their incident path. The light from a projector will return in the direction of the projector.

2.4.3 Comparison of multi-view displays

I present a basic taxonomy of parallax multi-view displays in Figure 2.8 and a list of characteristics for selected systems in Table 2.1. The two primary classifications are tracked versus untracked and single versus multi-user. Tracked, single user displays such as the NYU Autostereo, Varrier and Dynallax displays [78, 89, 79] are not suited to group tele-conferencing, nor are untracked steroscopic displays, such Sharp, iZ3D, TWISTER, and Synthagram displays [94, 42, 50, 97]. The Dynallax display does support a limited two viewer mode, at the cost of significantly reduced resolution and brightness.

Some of the displays can support a small number of users, including Newsight, MERL 3D TV, Philips, and Holografika displays [72, 63, 81, 2]. This makes them possible candidates for deployment in a group tele-immersion system. Only a few displays have been designed to ensure that multiple simultaneous viewers see the correct imagery on a shared display.

The Lumisight display [45] supports up to four users around a table using projectors and a light diffusing film. The IllusionHole display [48] supports 3 or more views, with an optional frame sequential stereoscopic mode, with unique views through an oculus at a large display underneath. Both of these displays are designed for tabletop operation, and would be limited to only two or three views if used as a vertically-oriented display.

2.5 Summary

The goal of this dissertation is the development of practical techniques and displays that support eye contact and gaze awareness in video teleconferencing systems for groups of

System	Technology	Viewers	Views	Tracking
Newsight [72]	film barrier, LCD	many	8	no
Philips WOWvx [81]	lenticular, LCD	many	9	no
SynthaGram [97]	lenticular	many	9	no
Holografika [2]	lens element, projectors	many	64	no
Actuality [1]	rotating volumetric	many	many	no
IllusionHole [48]	oculus, LCD	4	4 or 8	no
Lumisight [45]	diffuser, projectors	4	4	no
MERL 3D TV [63]	lenticular, projectors	many	16	no
MultiView [73]	retroreflector, projectors	3-4	3-4	no
Varrier [89]	film barrier, LCD	1	2	yes
Dynallax [79]	LCD barrier, LCD	1 or 2	2-4	yes
NYU Autostereo [78]	pi-cell barrier, projector	1	2	yes
NTII Telepresence [10]	polarized, projectors	1	2	yes

Table 2.1: Selected existing multi-view display systems and their capabilities.

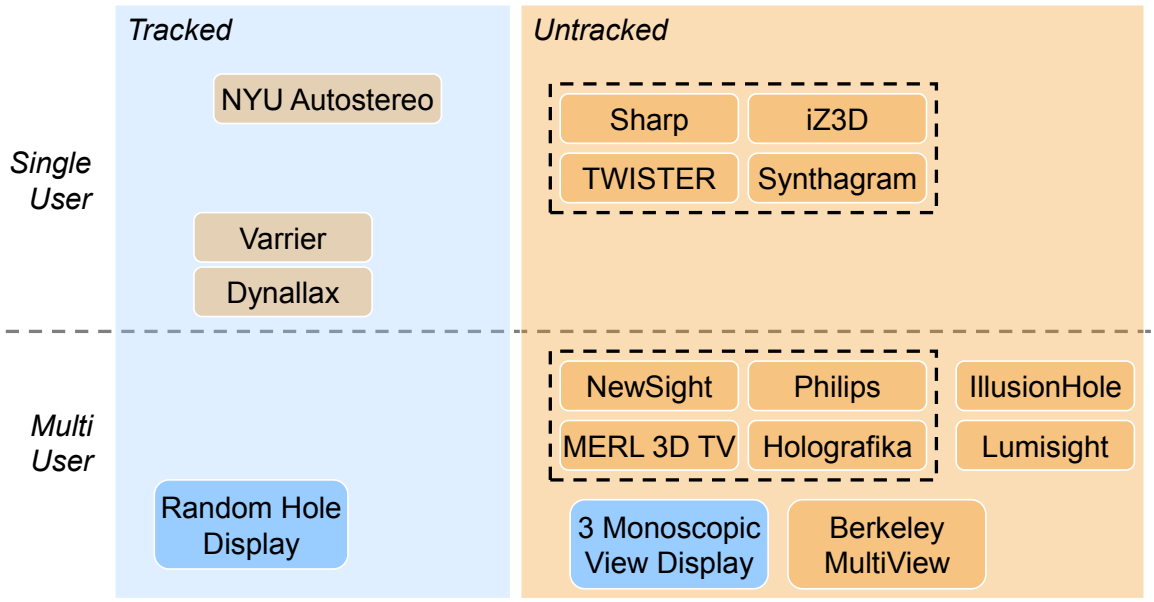


Figure 2.8: Taxonomy of selected parallax multi-view displays: the main division is between tracked and untracked viewing, with a second division between supporting single and multiple viewers.

viewers. Many existing systems address these problems to a limited extent, but none support all of the requirements for fully personalized views. Few existing system provide solutions that support groups of simultaneous viewers in arbitrary positions, at different distances from the display or moving across the display.

My work is characterized by two main themes: virtual camera techniques and display technology. Both are required to fully support group eye contact and gaze awareness. Chapters 3 and 4 describe several virtual camera techniques and systems for large scale display to groups. Chapters 5 and 6 present new multi-view displays that support gaze awareness for multiple users.

Chapter 3

Line light field rendering: Tele-immersion view synthesis for small groups

The first tele-immersion system described in this dissertation was developed to support life size imagery for small groups of remote users sitting at a fixed distance from a display. We have observed that during a video teleconferencing session, participants tend only to move small amounts in the horizontal direction, and hardly at all vertically. This limited view point allows us to consider simplified camera configurations and rendering algorithms compared to a general scene reconstruction system. We also wish to support real-time capture, encoding, and transmission of these views for interactive video teleconferencing.

Reconstructing a 3D model of a scene from 2D images may be computationally expensive and fragile in practice [55]. We can circumvent this problem by synthesizing novel views of the scene using *light field rendering* (LFR) techniques [55] that use many cameras to record the flow of light in all directions. The task of view synthesis then becomes a simple lookup of values in a database of camera rays. The main drawback to light field rendering is the requirement for many dozens or hundreds of cameras to cover a scene, and they would obscure much of the display surface.

Because the motion of video conferencing participants is limited to the horizontal axis, we can reduce the number of cameras necessary to capture a light field of the scene to a 1D linear array of cameras. We refer to this as the *Line Light Field* (LLF). This compact representation makes real-time capture, transmission, and rendering possible.

We also provide a transmission mechanism that aggregates the camera images and performs the rendering locally to the capture system before transmitting the final image across the network. This eliminates synchronization issues with multi-stream video transmission over long distances. We display the synthesized views on a multi-projector abutted display to provide life size rendering of the participants. We show the remote group of users and the rendering of that remote group for the local viewers in Figure 3.1.

The contributions of the Line Light Field project are:

The research in this chapter was conducted by the author with Ruigang Yang from 2001-2006.

(a)

(b)

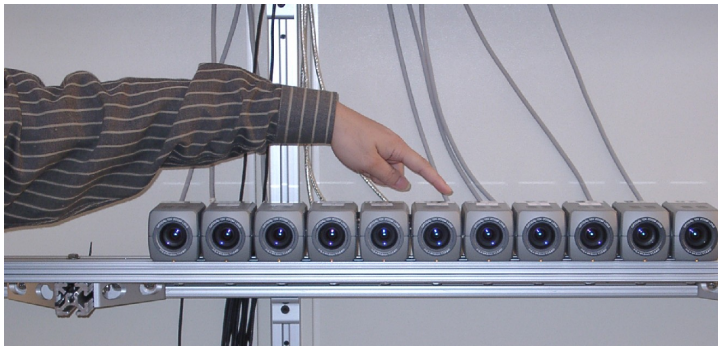
Figure 3.1: (a) shows the remote participants in a typical group teleconferencing configuration. (b) shows the local participants and a life-size, static seamless image synthesized using the LLF method.

- A novel technique that uses a linear array of cameras, as shown in Figure 3.2(a), that permits real-time view synthesis. We also provide an analysis of the sampling requirements to determine how many cameras are necessary for a given scene and viewing volume.
- The software and hardware architecture to collect, process, and transmit many simultaneous, frame synchronized video streams within a local network and across a wide area network.
- A prototype system setup with nodes at the University of North Carolina at Chapel Hill, Sandia National Laboratories, California, and the University of Kentucky, that demonstrate full duplex communication between two sites.

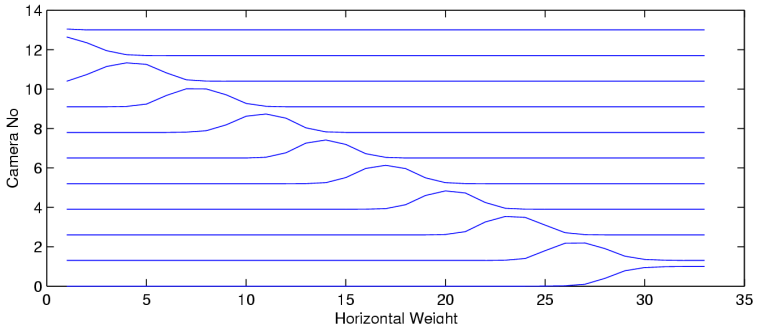
The remainder of this chapter is organized as follows: The technical details of our *Line Light Field* approach are discussed in Section 3.1. I present our prototype system architecture in Section 3.2, our results in Section 3.3, and conclude in Section 3.4 with a discussion of the pros and cons of the technique.

3.1 Line light field rendering

While applying unconstrained Light Field Rendering (LFR) is not practical in a tele-immersion system due to the difficulty of integrating many cameras with a display, we can take advantage of the notion to eliminate scene reconstruction. With a linear array of cameras, novel views at eye level can be synthesized at interactive rates, allowing the participants to view the remote scene from arbitrary viewpoints. The optical axis of the synthesized view is approximately constrained to the plane that passes through the camera array since there are no cameras to capture information from above or below.



(a)



(b)

Figure 3.2: (a) The camera array used to capture the participants. (b) The dynamically updated camera weighting masks (stacked).

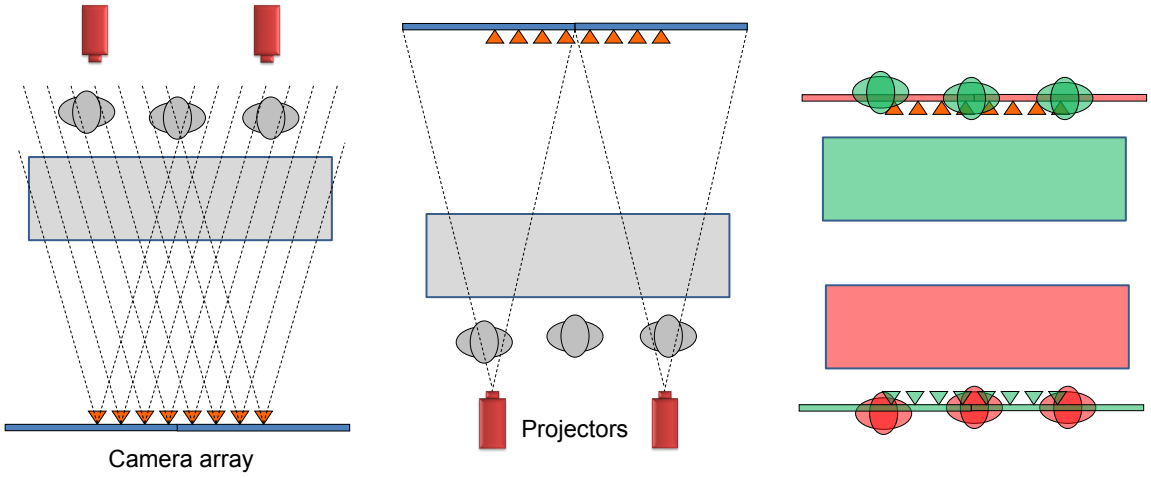
3.1.1 Rendering

Line Light Field Rendering blends the appropriate pixels from the nearest cameras in order to compose the correct scene for the desired viewpoint. The blending process can be accelerated using texture mapping hardware that is commonly found in commodity graphics cards [40, 6, 110]. Our method is a modified version of unstructured lumigraph rendering [6], and consists of the following steps:

- Set the viewpoint and depth of the plane of focus.
- Tessellate the image plane of the virtual camera into a set of narrow rectangles.
- Compute blending weights and texture coordinates between camera images.
- Apply textures to rectangles to generate the desired output.

Setting the viewpoint and plane of focus

The location of the viewpoint is controlled by the viewer. This virtual camera can be translated horizontally and zoomed in or out by the user during a conference. We refer



(a) (b) (c)

Figure 3.3: Relative geometry of local and remote sites: (a) The linear array of capture cameras with substantially overlapping fields of view is mounted just below the projection surface. (b) The display is provided by multiple projectors mounted above the users. (c) The superposition of the relative geometries of the capture and display components.

to this point as C_v in the remainder of this section. In addition, the depth of a *plane of focus* is specified to represents the average depth of the participants. The focal plane's depth is also interactively adjustable. Figure 3.3 shows a top down view of the geometric relationship of local and remote spaces, including the desired viewpoint and focal plane.

Allocate vertices in image plane

The image plane of the virtual camera is tessellated into a series of narrow rectangles. The vertices of these rectangles are back-projected onto the focal plane to guarantee uniform tessellation on the image plane, and leads to better blending.

Blending weights and texture coordinates

A set of blending weights is computed for each vertex on the focal plane. The weights are specified in the coordinate space of the multiple textures. Figure 3.2(b) shows the relative blending weights for each camera. We compute the blending weights as follows:

For a given vertex V, we need to find all of the cameras whose images contain the focal plane point V. We compute the angle θ_i, formed by the desired viewpoint C_v, the focal plane vertex V, and the center of projection of camera C_i, as shown in figure 3.4. A small angle means that the desired view is near that particular texture camera. To compute the blending weights w_i, a set of cameras κ with the smallest angles is selected. The number of cameras is determined by the maximum number of texture sources supported by the video card. The camera with the smallest angle has the largest

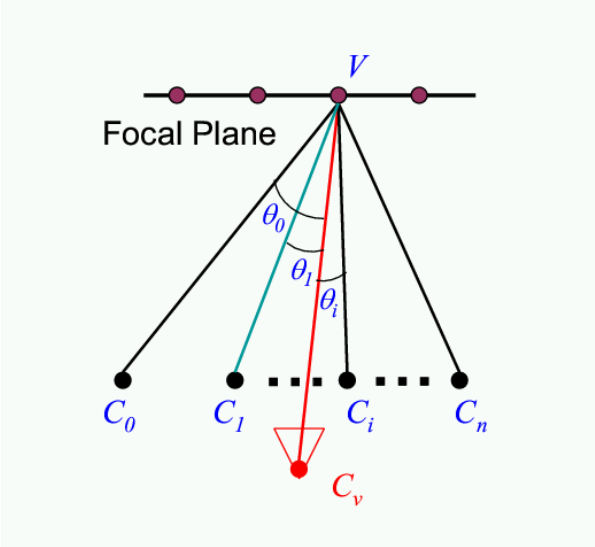


Figure 3.4: Angles between desired view camera C_v and texture cameras $C_{0..\kappa}$, at a vertex V on the focal plane.

blending weight, and the weights for the smaller angles are exponentially enhanced. The computation of the blending weight is given by the following equations:

$$\hat{w}_i = exp(\frac{-\theta_i^2}{\sigma^2})$$

$$w_i = \frac{\hat{w}_i}{\sum_j \hat{w}_j}$$

for all cameras i and j that are included in κ.

The blending factor σ is a user-controllable parameter to control the rate of blending between camera images. For a smaller value of σ, the less the inter-texture blending. We typically use a constant value of 2.5^o. The blending weights for each vertex are normalized to guarantee a constant brightness of the entire image. For the invalid cameras, a blending weight of zero is assigned. At the end of this process, we have a list of weights and texture coordinates for all cameras.

Multiple texture blend

The multiple camera textures are applied to the rectangles in the image plane using the respective blending weights and the texture coordinates. Each rectangle is rendered as many times as there are contributing textures, with the frame buffer used as an accumulation buffer. If multi-texture hardware is available, textures from multiple cameras can be rendered at once, reducing the total number of passes requires.

3.1.2 Orthogonal views

If an object's depth differs substantially from the focal plane, we find that the synthesized view from the Line Light Field is relatively blurry. This effect is caused by under-sampling in our camera system. Our goal is to improve the picture quality without increasing the number of cameras in use.

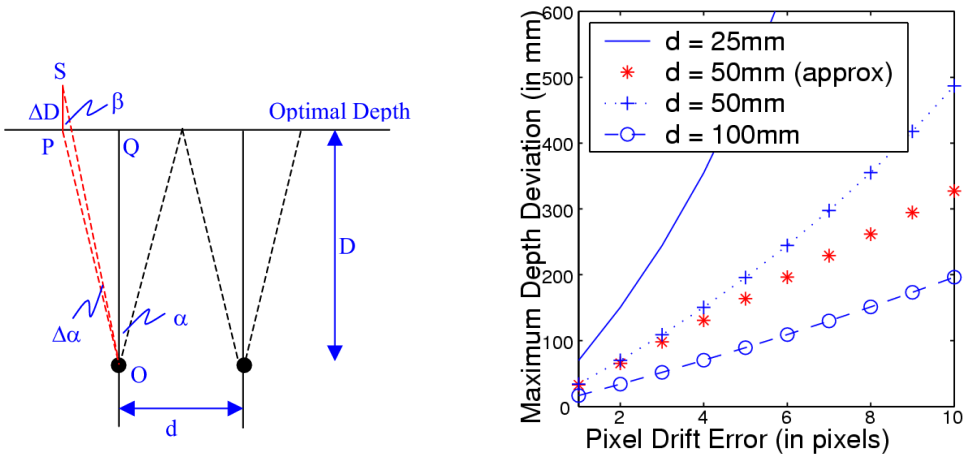
Another objective is to create a continuous, high-resolution, wide field of view image from a perspective further behind the screen, a good compromise for a group of people. As the number of participants grows, we would like to push the center of projection further away, so that every one is more or less the same distance to the center of projection. In this case, we would create an orthogonal view of the remote scene. Unfortunately, normal cameras are designed to take perspective images. However, we can approximate an orthogonal view from an array of cameras. For our 1D linear camera array, if we take the vertical scan line going through the image center for each camera, and piece them side by side, we can get horizontally orthogonal images. In practice, we can always use a small vertical strip of each camera due to the limited resolution of the display device, as well as the human visual system.

This thinking results in an extremely simple view synthesis method. For each camera image, we take out a narrow band in the middle, and juxtapose these bands. We also introduce a small amount of overlap between adjacent bands to accommodate for small registration errors and avoid the harsh boundaries for color mismatches. Unlike the line LFR method in the previous section, there is little inter-camera dependency, since the final color of each pixel in the synthesized view depends on at most two cameras' images. Thus it is possible to distribute (not replicate) the input image data to a number of computers to create *wide FOV* high resolution imagery.

3.1.3 Sampling analysis

We provide a sampling analysis for our linear camera array to determine the number of cameras required for an aliasing-free output. We first assume that all of the cameras are mounted on a horizontal rail and regularly spaced. The optical axes of the cameras are parallel on a horizontal plane. We then define an error tolerance measure (e) in terms of pixel drift, i.e., the distance from a pixel's ideal location in the synthesized view. For a given configuration, we would like to find out how much error there will be, or conversely, given an error tolerance measure, how many cameras are needed. The error tolerance is a view-dependent factor. If the synthesized view corresponds exactly to one of the input camera views, then e is zero.

In the ideal case, only a single vertical scan line through the image center from each input image is used to composite a horizontally orthogonal image. No matter how far away the object is, its projection on the synthesized view would remain the same. This means we can generate correct imagery without knowing the locations of the scene objects, thus avoiding the difficult scene reconstruction problem. But this is not practical since it would require thousands of cameras to create a single image. We use a narrow column of pixels from each camera to approximate the orthogonal view. If we back project the narrow columns into space, they will intersect at a certain distance,



(a) Geometric setup

(b) The maximum depth deviation with respect to pixel drift error.

Figure 3.5: Error analysis for generating orthogonal views.

which we call the optimal depth D. Only the objects at the optimal depth will have the correct imagery on the synthesized views. Objects that are closer will be lost and objects that are further will have duplicates.

Inspired by the sampling analysis for LFR in [8, 56], we evaluate the error tolerance e using a geometric approach. We define the following parameters:

- Camera's field of view FOV
- Camera's horizontal resolution (in number of pixels) W
- Inter-camera distance d

Given a set of camera configuration parameters, and a desired error tolerance e, we want to determine the maximum depth deviation ΔD from the optimal depth D.

From Figure 3.5(a), we see that $\alpha = tan^{-1}(\frac{d}{2}/D)$, $\beta = \angle OPS = 90 + (90 - \alpha) = 180 - \alpha$. In triangle SPO, we have:

$$\frac{\Delta D}{sin(\Delta\alpha)} = \frac{|OP|}{sin(\angle PSO)}$$

Substituting $\angle PSO = 180 - \beta - \Delta\alpha = \alpha - \Delta\alpha$ and $|OP| = \sqrt{(d/2)^2 + D^2}$:

$$\Delta D = \frac{sin(\Delta\alpha)\sqrt{(d/2)^2 + D^2}}{sin(\alpha - \Delta\alpha)}$$

We can then approximate the angular deviation $\Delta\alpha$ in term of pixel drift e, where $\Delta\alpha = (e/W)FOV$. That leads to:

$$\Delta D = \frac{sin(e/W * FOV)\sqrt{(d/2)^2 + D^2}}{sin(\alpha - e/W * FOV)}, \tag{3.1}$$

where FOV is expressed in radians. Furthermore, since $sin(\alpha) = (d/2)/\sqrt{(d/2)^2 + D^2}$, $(e/W)FOV$ is usually a very small number and $(e/W)FOV \ll \alpha, d \ll D$, we can approximate Equation 3.1 as:

$$\Delta D = \frac{e}{W} FOV \frac{D^2}{d/2} \tag{3.2}$$

We can derive a similar equation in case S is closer to the camera instead of farther away.

Let us assume $FOV = 30^o, W = 640,$ and $D = 1000$mm. Figure 3.5(b) shows the maximum depth deviation with respect to pixel drift error under different camera placements $d = 25, 50, 100$ mm. The line of red stars shows the results computed using the rough approximation (Equation 3.2), while the rest are computed using Equation 3.1. Note these are "one-sided" numbers, i.e., they only represent how much *further* away the real depth can be. The total distance variation is roughly twice as long. From the results we can see that it is indeed possible and practical to create crisp orthogonal images for depth variation under 400 millimeters, a reasonable value to accommodate normal human motions during a conference.

3.2 System architecture

A prototype system of this design was implemented at three sites: the University of Kentucky (UKy), the University of North Carolina at Chapel Hill (UNC-CH), and Sandia National Labs, California (SNL/CA). Each site has a total of eight or ten Sony digital Firewire cameras arranged in a linear array, as shown in Figure 3.2(a). These cameras are regularly placed with their centers every 65 millimeters, very close to the minimum distance allowed by the form factor of the camera body. All cameras are synchronized by a wire controlled from a computer and fully calibrated using the method from Zhang [115].

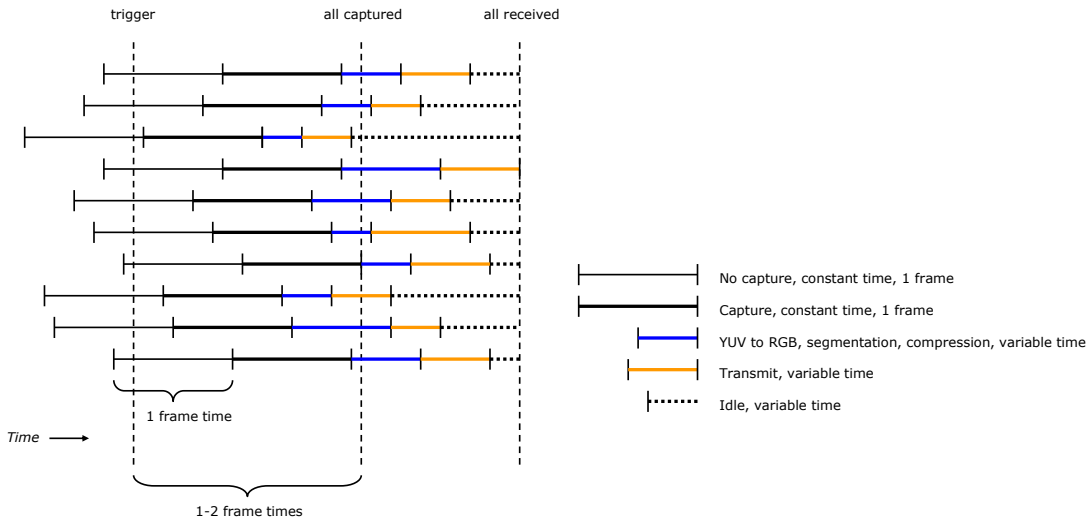


Figure 3.6: Triggering and synchronization timeline for a group of cameras.

Cameras and Synchronization

The video acquisition system (one at each site) includes four to five server computers interconnected through 100Mbit Ethernet. Each server is connected to two Sony cameras and is used to capture and JPEG encode the raw image data at full VGA resolution. The server may optionally segment the foreground participants from the background and encode the mask of the foreground objects in the alpha channel of the image. The JPEG streams are then sent through the network to be decoded on the rendering system.

Camera synchronization ensures that all cameras capture a video frame at the same time. Without synchronization, adjacent video images may show discontinuities in moving objects, such as a participant's body. The cameras used in our system support an external trigger via a special interface, separate from the Firewire used for data transfer. When the camera receives a signal pulse on the trigger interface, the camera captures a frame during the next frame interval of the camera's internal clock, which runs at a preset frame rate. The video data is then read out over the next clock interval. Due to this two frame process, the maximum triggered capture speed is slightly less than half of the set camera frame rate. Also, the data may arrive at the host machine anywhere from one to two frame times after the trigger signal.

When multiple cameras are triggered by the same signal, they are guaranteed to be synchronized to the same video frame and will follow the previously described read out process. However, the camera clocks are not synchronized and may be offset by any amount, as shown in Figure 3.6. Additionally, video processing (color conversion, segmentation, and compression) and network transmission can take a different amount of time for each video stream.

The triggering signal is controlled by trigger server software via a parallel port device. Ideally, triggering will occur as fast as possible (half the set frame rate of the camera).

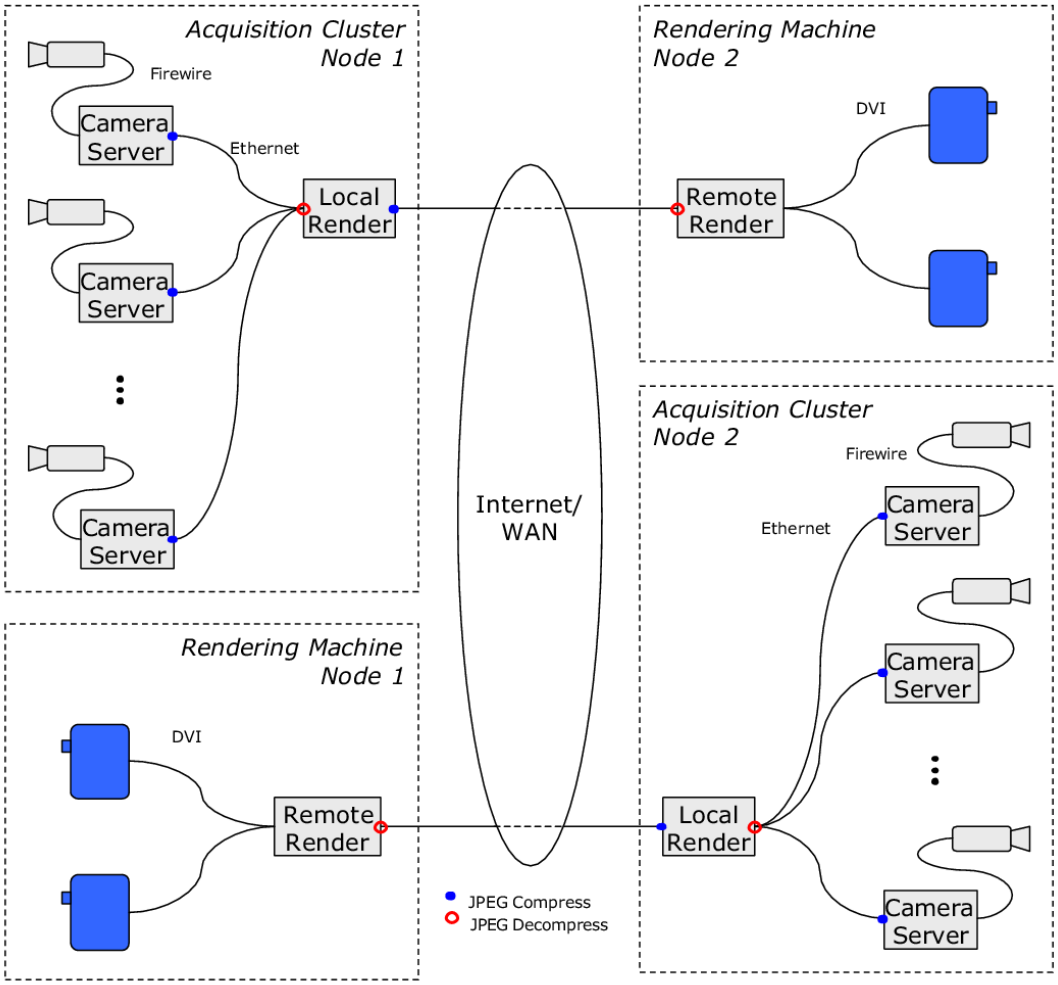


Figure 3.7: The system architecture of our prototype.

However, if the trigger signal is received early by a camera that is still capturing or transmitting video data, the frame will be delayed an entire frame period or not sent at all. To ensure that all of the camera servers are ready to capture a new frame, the trigger server waits until a confirmation message is received from each of the camera servers. Because the time delay in the camera is relatively longer than the other stages of the pipeline (shown in the threaded data processing diagram of Figure 3.8), it is safe for the camera server to send the confirmation signal (trigger out) as soon as the data is received in host memory. In practice, the trigger server is run on the local rendering machine.

Streaming architecture

There are two configurations for running the view synthesis program. One is to send all the video streams over the Internet and synthesize novel views at the remote site (remote rendering). Alternatively we can synthesize views locally and only send the final result

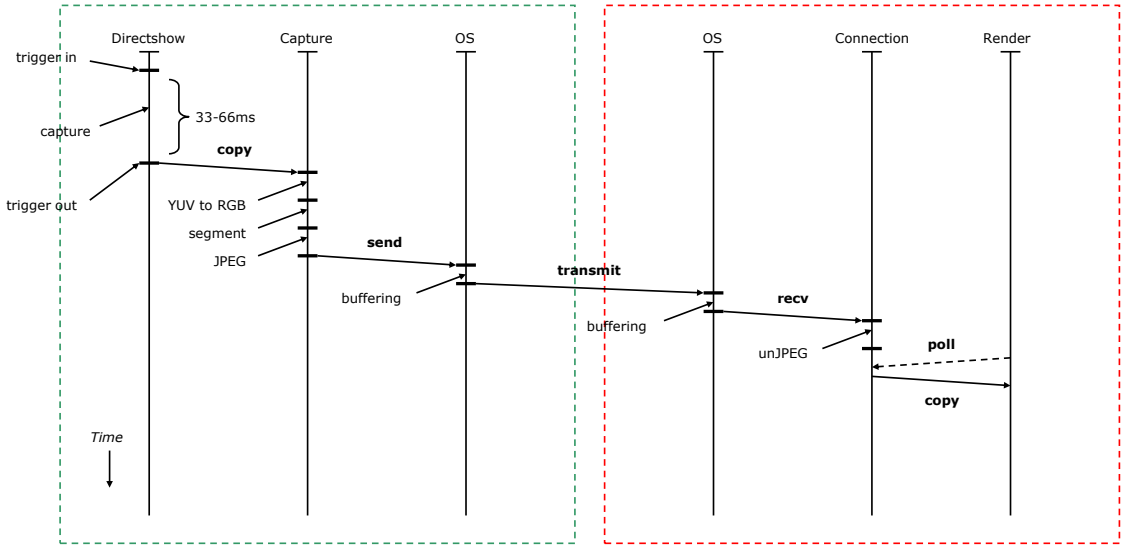


Figure 3.8: Data path for a single video stream, from capture to display. The processes in the left box run on a camera server, and the processes on the right run on the local renderer. The transmission link occurs over a local area network.

to the remote site (local rendering). The first approach has a potentially lower latency for a changing viewpoint, but requires extra stream synchronization mechanisms. The second configuration, shown in Figure 3.7, is easier to manage from a network standpoint, because only a single data stream is sent each way over the Internet.

Remote rendering requires more bandwidth than local rendering, because the Line Light Field blending occurs after transmission over the Internet. In terms of scalability, the bandwidth requirement for remote rendering is $k*R$ where k is the number of cameras and R is the average compressed camera image size. The bandwidth requirement for local rendering, on the other hand, has a resolution fixed by the rendering output, independent of the number of cameras.

3.3 Interactive results

We first show the results from our line LFR method in Figure 3.1(b). To create life-size images, we use a two projector, abutted display at both the UNC and SNL/CA sites, and a single projector at the UKy site. The background is blurry due to the limited number of cameras in use. This can be alleviated using the foreground/background segmentation.

In a teleconferencing session with few participants, we use the view dependent line LFR method to synthesized desired views, shown in Figure 3.9. We put a stationary folder to illustrate the view dependent effect when the local conferees move to different spots. In Figure 3.10, we show the setup at UKy in which a full duplex live session is in progress. The video images were synthesized using the line LFR method.

In terms of performance, we achieve an update rate of 5-10 frames per second (fps)

(a) (b)

Figure 3.9: View dependent effects when the local conferees move to different spots. Notice that we have placed a folder in the scene. When the local conferees are at right (a) the viewpoint of the synthesized view is from right, revealing the front cover (on the right) of the folder. When the conferee moves to left (b) the view changes accordingly, revealing the back cover of the folder.

for VGA input images locally. The bottleneck is in image capture. We can only capture synchronized VGA resolution images at 12-13 fps with one camera per Firewire bus and 7-8 fps with two cameras on the same Firewire bus, as described in Section 3.2. When multiple video streams are sent to the rendering machine, network collisions at the rendering machine reduce the frame rate to 5-10 fps.

The synthesized view, typically rendered at 1024×512 or 2048×768, is read back from the rendering program's framebuffer and sent to the remote site through TCP/IP with JPEG encoding. The remote rendering program is capable of decoding and rendering 1024×512 images at over 30 fps. However, the network bandwidth between UKy and UNC-CH is quite limited. The overall frame rate between these two sites varies from 5 fps to 10 fps, depending on network traffic. We estimated a sustained transfer rate between 3 Mbits and 6 Mbits. Optimization of networking code or the use of a more sophisticated compression scheme is expected to substantially increase the frame rate. Similar performance has been observed between UNC-CH and SNL/CA.

3.4 Discussion

Conventional video teleconferencing solutions are insufficient for replicating the face-to-face experience of group interactions. Resolution limitations, a lack of depth cues, and smaller than life-size imagery are drawbacks of a conventional single camera and display system. We have presented techniques within a system for synthesizing novel, high-resolution views rendered in life-size for group tele-immersion.

The *Line Light Field* method is a practical attempt to bypass the difficult geometry reconstruction problem by using many cameras. Instead of performing dense, computationally intensive 3D scene acquisition, we exploit the fact the participants' motion during a video teleconferencing session is rather limited, usually to lateral motions with their eyes remaining at a fixed level. This natural restriction allows us to use a lim-

Figure 3.10: A live teleconferencing session between the University of Kentucky and the University of North Carolina at Chapel Hill.

ited number of cameras to capture important views. Based on this observation, we have developed a real-time acquisition-through-rendering algorithm based on Light Field Rendering.

The realism of the synthesized view is derived directly from camera images. With smaller, inexpensive cameras becoming available, we believe this method provides a useful solution in the near term. The bottleneck for this method is bandwidth, both network bandwidth and the rendering system's internal bus bandwidth. While these can be improved through technical advancement, a more fundamental drawback of this pure image-based method is that the linear arrangement of the cameras limits the possible range of the viewpoints, requiring the the camera array to be placed at eye level.

Future work will consider these issues in particular:

- New camera arrangements, for example, multiple rows of cameras, above and below the display, with new blending methods to simulate eye level cameras.
- Camera/display integration, such as cameras embedded in the display area.
- Display wall integration, for displaying the rendered images on multi-projector display systems, such as PixelFlex [109].
- Higher speed video capture and transmission, such as automatically synchronized camera systems [82], and improved inter- and intra-stream compression methods.
- Active tradeoff between local and remote rendering to adapt to changing network conditions.
- Combining Light Field-style rendering with improved geometry proxy acquisition for higher fidelity rendering.

- Integrating (multi-)user tracking to set rendering viewpoints and to adapt acquisition algorithms.

We have demonstrated the practicality of the Line Light Field technique using our full duplex 3D video teleconferencing prototype between SNL/CA and UNC-CH, with regular video teleconferencing between the sites. We have also conducted successful but limited tests between UNC-CH and UKy.

Chapter 4

Telepresence wall: View synthesis for a wall sized display

This chapter considers the design of a telepresence system with a wall-sized display for multiple, freely moving participants. The goal of the *Telepresence Wall* (TW) project is to address the problems that arise when multi-user telepresence systems are extended to a much larger format display, covering most or all of a wall. Our objective is to create the illusion that a single large room has been cut into two halves with a large window between them, allowing users in each half to communicate and interact naturally in both directions. The most significant difference from the Line Light Field scenario is a much larger working volume, where users are free to move near to or far from the display, and users may be at different distances from the display simultaneously.

We chose a shared break room as the working scenario, approximately 16' deep by 20' wide, involving seated and standing users, with movable objects such as tables and chairs. Practical considerations include use of vertically oriented large flat panel displays to present a bright image in a well lit room and a modest number of cameras per display, such as 4 to 8. A break room-sized display wall might consist of 6 of these display panel-camera modules, as depicted in Figure 4.1. We tested our algorithms using simulated camera views of a 3D model of a virtual break room environment. The simulated break room provided useful flexibility in testing new camera configurations and also allowed us to avoid the difficult camera calibration problem every time cameras were moved.

The first stage of telepresence wall work was the development of a view synthesis algorithm suitable for a large area and a high resolution display, and the determination of suitable camera configurations compatible with abutted flat panel displays. The second stage was a new technique called the *Depth-Dependent Camera* (DDC) for warping display imagery to support local eye contact while maintaining a perception of depth in the scene. Finally, to address imagery quality problems with view synthesis for a telepresence wall, we developed a rendering technique using video silhouettes that addresses problems of overlapping camera fields of view.

In this chapter, I present our large scale display and video capture system for groups of users and describe the three main contributions of the Telepresence Wall project.

The research in this chapter was conducted by the author with Peter Lincoln from 2008-2009.

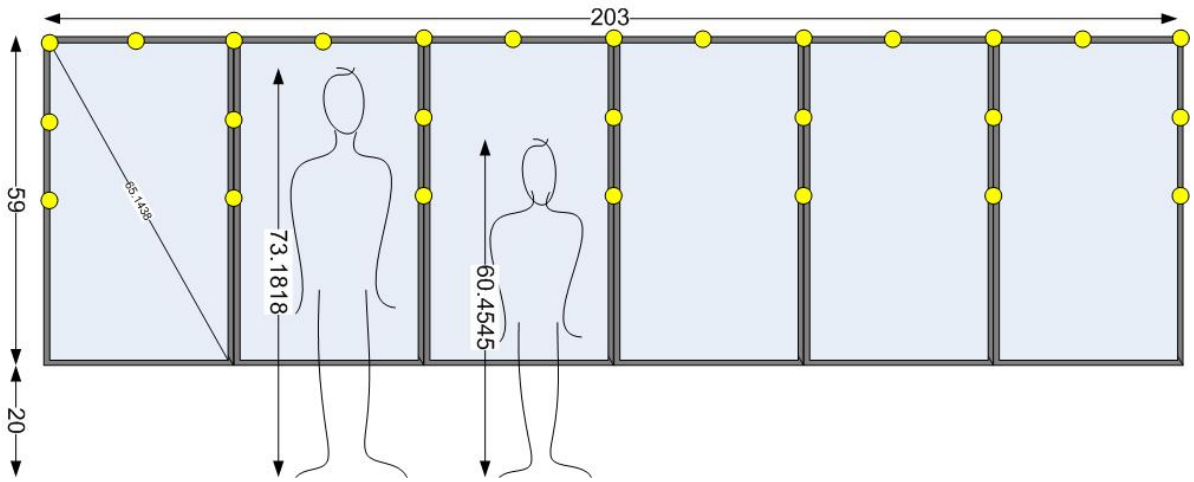


Figure 4.1: A prototype sketch of the display layout for a six panel telepresence wall. The yellow circle represent camera positions around the displays. Used with permission from J. William Mauchly, Cisco Systems.

Section 4.2 explains our view synthesis algorithm using plane sweeping. Section 4.3 describes a new camera model useful in a group teleconferencing scenario. Section 4.4 provides the video silhouette approach for rendering Telepresence Wall imagery. In Section 4.5 I discuss the trade offs between the different techniques and their potential use in the future.

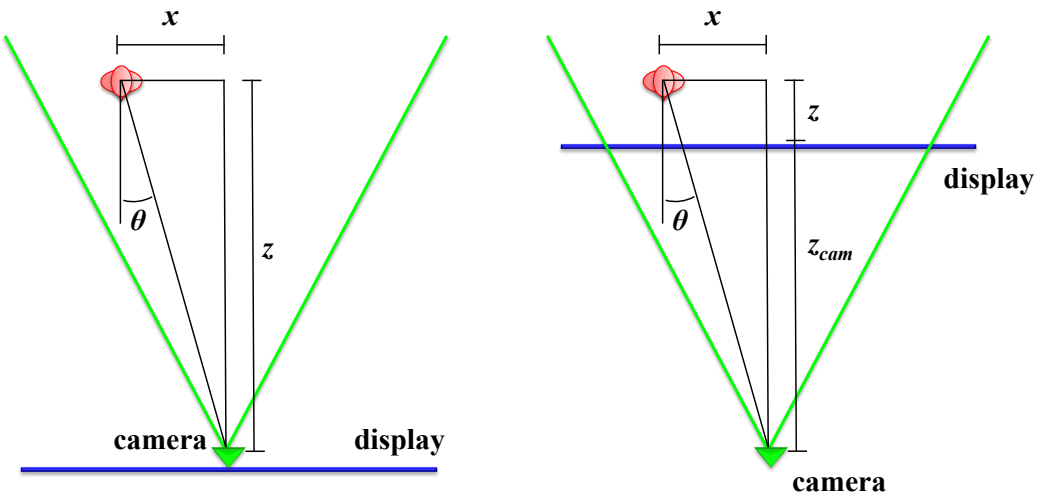
4.1 Introduction

We would like to create the illusion that the break room has been divided in two by a window. However, the window is actually a large display surface. This means it is difficult to place a camera in locations corresponding to the users' views of the display. The real camera or cameras must be placed around or in front of the display wall at the remote site, and this introduces significant perspective and gaze issues.

A solution for capturing the entire breakroom is a panoramic camera, with either a wide angle lens or several cameras with shared centers-of-projection. Such an approach is effective in the Cisco CTS 3000 TelePresence system [12] because the users are all seated at a fixed distance from the camera. However, in a Telepresence Wall scenario, the panoramic camera has two significant problems: severe perspective distortion and lack of eye contact for most viewer positions. Because users can come close to the screen, the close perspective will make them appear unnaturally large.

The main problem is that only those users on the central axis of the camera will be able to make eye contact with viewers. Everyone else will appear to be looking off to one side or the other when they are actually trying to make eye contact with a remote user. We formalize this gaze error as the angular deviation between the apparent viewing direction of the displayed image and the actual gaze direction of the remote user.

As shown in Figure 4.2(a), the gaze error is the angle determined by the offset (x, z)



(a) (b)

Figure 4.2: Top-down view of geometry for calculating gaze error of a remote participant at offset (x, z). The gaze error θ is the angular difference between their actual viewing direction and the apparent viewing direction due to the capture camera offset. (a) shows the gaze error for a camera located at the display, and (b) shows the gaze error for a camera located at z_{cam} behind the display.

between the user and the camera. The angle is calculated by:

$$\theta = |tan^{-1}(\frac{x}{z})| \tag{4.1}$$

We would like to support some amount of eye contact across the width of the display. We define *local eye contact* as the ability for users near to the display wall ($\leq$ 8ft) to make effective eye contact with people shown directly across the display. Because humans are very sensitive to small gaze angles [28, 9], eye contact with remote users is only possible in regions where the gaze error is small. We define the *usable gaze area* as the set of positions with $\leq 10^{\circ}$ gaze angles.

In the case of the panoramic camera located at the display wall, gaze error becomes very large as the horizontal user offset increases, even by small distances. Figure 4.3 shows the gaze error for a remote user at 3 different depths for different horizontal positions across display, with a panoramic camera centered at $(0, 0)$, calculated using Equation 4.1.

For a given depth z, the maximum horizontal offset x for a particular gaze error angle θ is given by:

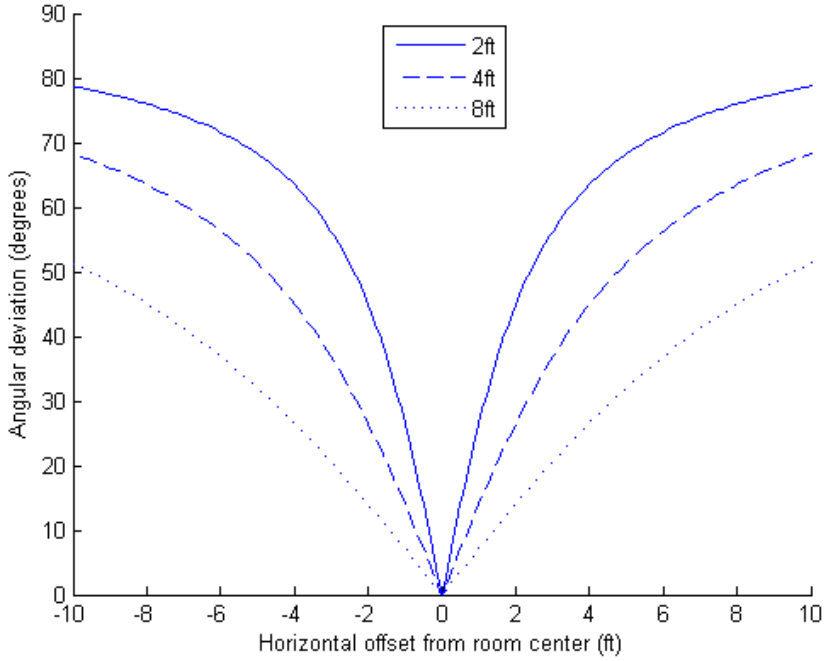


Figure 4.3: The gaze error for a panoramic camera at the display wall, as shown in Figure 4.2(a). The horizontal axis indicates the horizontal (x) offset between the user and camera, and the vertical axis shows the gaze error in degrees. The three lines correspond to 3 different z offsets, with the user at 2, 4, or 8 feet from the display.

$$x = z * tan(\theta) \tag{4.2}$$

For a user at 2' from the display, allowing for a 10° gaze error, the maximum offset is 0.353'. At 4' and 8' from the display, the maximum offsets are 0.710' and 1.411' respectively. The gaze error exceeds 10° at all horizontal positions outside of these horizontal offsets. The total width of the usable gaze area is double the maximum offset, allowing for movement to either side. For the 20' wide wall, this results in only 3.5% , 7.0% , and 14.0% of the viewing width to be classified as within the usable gaze area, at 2', 4', and 8' user distances, respectively. With the majority of the viewing area width outside of the usable gaze area for any user depth less than 8', we can conclude that the panoramic camera model cannot properly support eye contact for a telepresence wall application.

One camera model that would improve local eye contact is an orthographic view of the scene. The parallel projection of light rays from an orthographic camera would render the correct orientation of each user's head with respect to the display surface. When they are looking straight at the display, they would appear to look straight at the display at any horizontal position. However, the orthographic view eliminates perspective, making

it difficult to determine the depth of people or objects in the scene.

We can approximate the local eye contact support of an orthographic view with a distant perspective camera that also preserves perspective depth cues. Most rooms are too small to support a camera at that distance, even if the display were not obstructing its view. Acquiring this distant perspective is important enough for systems to adopt expensive and complex mechanism, such as the folded optical path of the France Telecom 'magic wall' [100]. In order to generate such views, we can synthesize the imagery from a *virtual camera*, positioned behind the display.

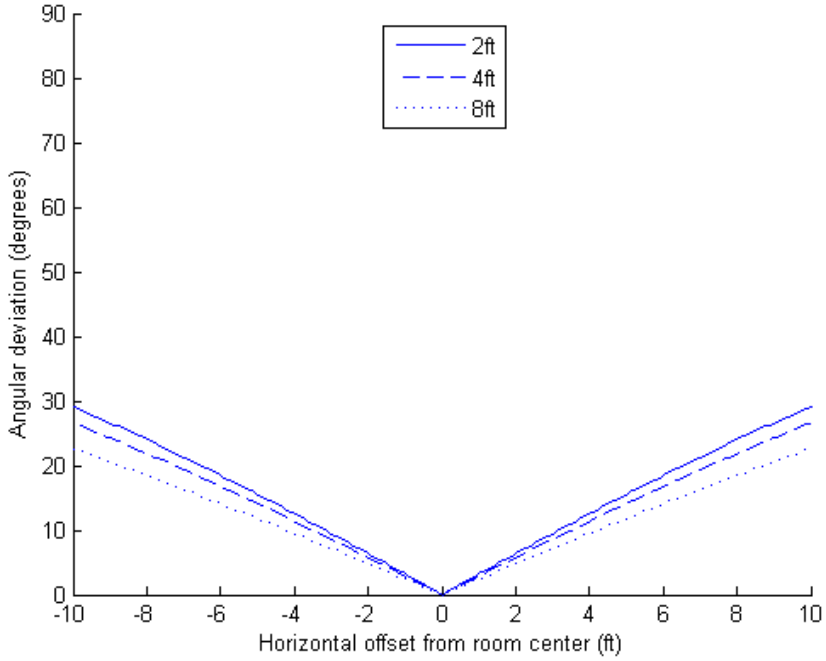
We adapt the gaze error model to include the camera distance behind the display surface, shown in Figure 4.2(b). The gaze error is the angle determined by the offset $(x, z + z_{cam})$ between the user and the camera. The angle is calculated by:

$$\theta = |tan^{-1}(\frac{x}{z + z_{cam}})| \tag{4.3}$$

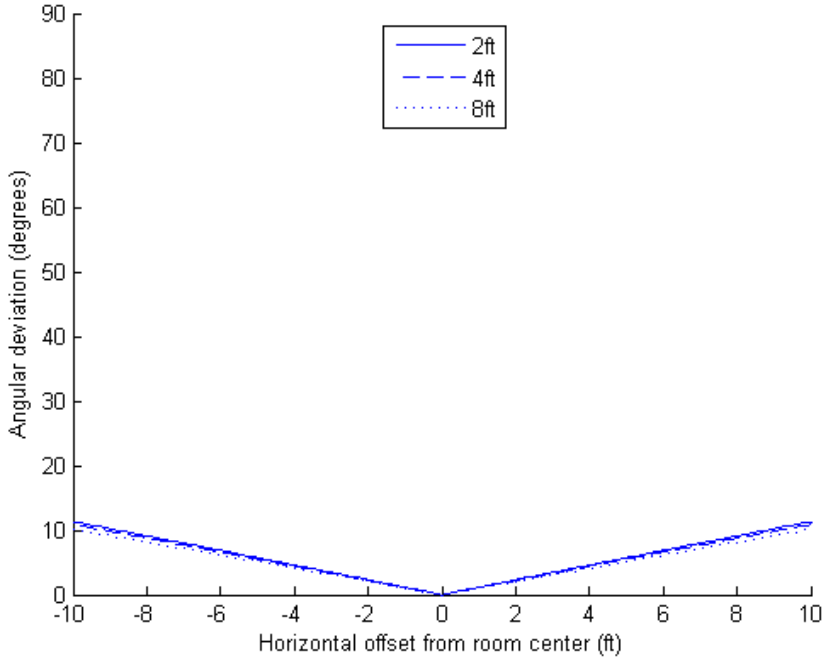
We repeat the gaze error calculations with the virtual camera positioned at different distances behind the display. Figure 4.4 shows the gaze error for virtual cameras at 16' and 48' behind the display. These distant virtual cameras significantly reduce the gaze error across large regions in front of the display. However, the virtual camera at 16' behind the display still only provides 30% of the viewing width at 2' to be classified as within the usable gaze area. By moving the virtual camera to 48' behind the display wall, 88% of the display area at 2' provides less than 10° gaze error.

To generate views from different camera positions, we began with a 3D model of a break room environment, with several participants: men and women, standing and sitting, at various distances from the display wall. The remote scene is generated with 3D modeling software and rendered from various viewpoints corresponding to virtual camera positions centered in the local room at a given distance from the display wall. A top-down view of the first break room model is depicted in Figure 4.5 and an example virtual camera view from the local room is depicted in Figure 4.6. The simulated images include the borders corresponding to the 2" display bezel for each of the five 65" flat panel displays. Total display size is 15' wide by 5' tall, with the bottom 20" from the floor.

For comparison purposes, we directly render views of the break room model from cameras at varying distances from the display, shown in Figure 4.7. The orthographic view (a) shows that users facing the display appear to be facing the display the entire way across the viewing surface. We see that all sense of perspective depth is lost. Virtual cameras that are close to the display, from 8 feet (b) or 16 feet (c), emphasize perspective depth but introduce distortion for objects near to the display, like a panoramic camera. In particular, they make it difficult for users at the sides of the display to make eye contact when looking straight ahead. A camera at 48 feet (d) from the display offers a balance of perspective and a degree of local eye contact across the entire display. We expect that it is necessary to have a fixed viewpoint for multi-user telepresence, because a moving camera may produce a visual swimming effect for the users.



(a)



(b)

Figure 4.4: The gaze error for virtal cameras (a) 16' and (b) 48' behind the display wall, as shown in Figure 4.2(b). The axes and line sets are the same as in Figure 4.3.

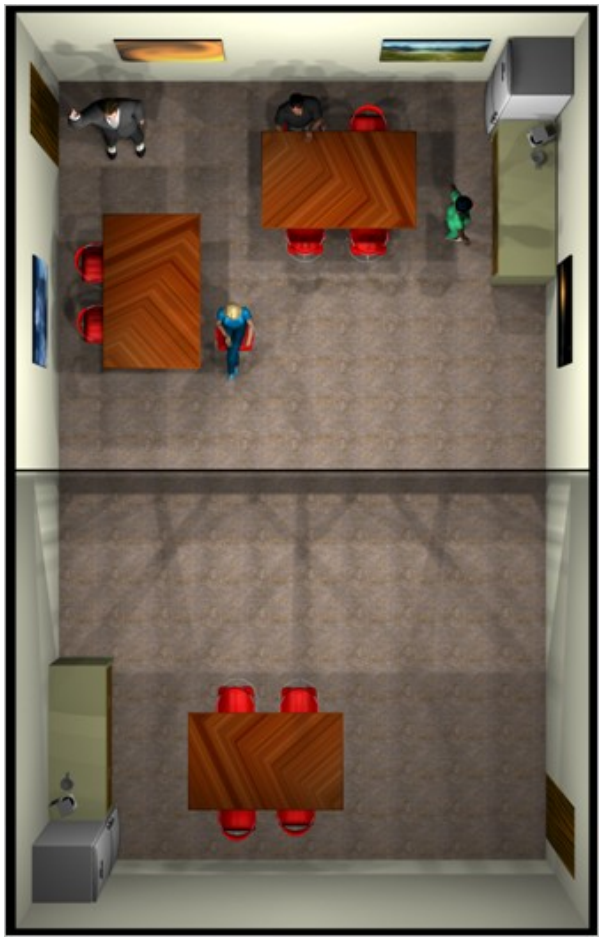


Figure 4.5: Top-down view of the telepresence wall break room scenario 3D model, with the remote and local rooms abutted.

4.2 Plane sweeping view synthesis

We can synthesize a distant perspective view of the scene, as if the camera was 48 feet away, combining multiple camera images into a single coherent view of the scene. The method we have chosen for scene reconstruction is plane sweeping [14], based on recent work that has shown how to implement such an algorithm in graphics hardware [113, 111]. With efficient mapping to commodity hardware, reconstruction algorithms are significantly accelerated over CPU-based methods at low cost. Our variation of the algorithm is based on the plane sweeping view synthesis algorithm, with additional initial setup and per step components. It includes image segmentation to handle silhouette edges with higher resolution and for sweeping segmented regions at higher densities.

4.2.1 Approach

The fixed rendering viewpoint of our simulated break room, allows us to proceed with a major simplifying assumption: that every display pixel corresponds to a fixed ray in the remote room. This allows us to maintain surface probability information on a per

Figure 4.6: A synthesized view from one room into the other in a six panel configuration, as if there was a window in the wall.

pixel basis. This data structure, used by the plane sweep process to improve the quality of surface depth estimation, could be updated by inputs from the capture cameras, model databases, and potentially by other sensors such as depth cameras. The fixed viewpoint restriction also means that each camera-display panel module can operate independently, allowing for scalable display walls.

This output-oriented approach to view synthesis estimates the most likely color for each display pixel. Given a set of calibrated camera images, we synthesize new views as follows: we discretize the 3D space into a set of parallel planes that are parallel to the display surface, and for each plane we project the input images onto it. If there is an object surface located in the plane, each of the images projected from a camera view at that spot should have the same color. We compute the mean color and variance for each pixel in each of the plane images. The final output color for a pixel in the rendered view is the color with minimum variance, or best color consistency among camera views.

This process is depicted in Figure 4.8. For a given candidate plane, we determine the likelihood of surface for each voxel using color consistency measures. In the case (a) where a surface is not at the candidate plane at v, the projections of each camera's imagery onto the candidate plane will have different colors at that point. When the candidate plane is actually at the surface (b), projections of each camera's imagery will have the same color. By measuring the variance between colors, we can determine the probability of a surface at each plane depth for all display pixels. More importantly, we can determine the most likely color for each display pixel.

This rendering method provides a unified framework for handling people, objects, and the room. We can improve the quality of the rendering with knowledge of object

(a)

(b)

(c)

(d)

Figure 4.7: Rendered views of break room scene with a simulated five panel display: (a) orthographic, (b) perspective camera at 8ft, (c) perspective camera at 16ft, and (d) perspective camera at 48ft.

locations obtained from other sources, such as 3D model databases or tracking devices, and use adaptive plane sweeping only in changing regions. We can then update a volume data structure with new 3D position information from the plane sweeping results for more accurate rendering in the future.

Challenges

There are a number of special cases that the plane sweeping algorithm must handle, including shadows and specularity, occlusion, and calibration.

Shadows and specularity These are color related issues which may lead to incorrect color matching, but can be addressed in the software algorithms and by engineering via improved lighting. Shadows are initially considered in the image segmentation component of the algorithm. A new camera image and background image are compared in an hue-saturation-value (HSV) color space to eliminate consideration of shadowed areas by the plane sweeping component. However, eliminating shadows may result in an unnatural appearance so we should maintain the ability to process them in later stages, perhaps as a special case. Similarly, specularity may result in significantly different colors in the camera images of a single surface. We mitigate this by comparing the paths between the object hue and light color rather than just the color values.

Occlusion Surfaces in the scene may be occluded from certain camera views, leading to incorrect color comparisons. Similarly, surfaces at the edges of objects may be visible in some camera views, but not others, also leading to an improper comparison for those surfaces. We can address both of these issues with outlier elimination, by only considering a majority of similar cameras for comparison and blending. However, this is at best a heuristically-driven guess for eliminating certain cameras for consideration. To properly address occlusion of surfaces, we will need to include tags on the source camera rays. If a camera ray is matched with others at a given sweep plane, we should mark that ray in the source camera with a likelihood of occlusion. This will reduce the probability that it is considered for future matches. If silhouette edges are not treated specially, improper comparison and blending may occur. We can tag silhouette edges in the initial image segmentation step to provide uncertainty values to the plane sweeping comparison.

Calibration The plane sweeping algorithm is also dependent upon accurate geometric and color calibration of the source cameras. Even a high quality initial calibration will degrade over time, so future implementations will probably require constant background calibration. To assist in calibration, we may include fiducial markers in the environment or used structured light.

4.2.2 Algorithm

To reconstruct the remote room and render a view from the virtual camera, we run the plane sweeping algorithm on sets of synchronized camera images.

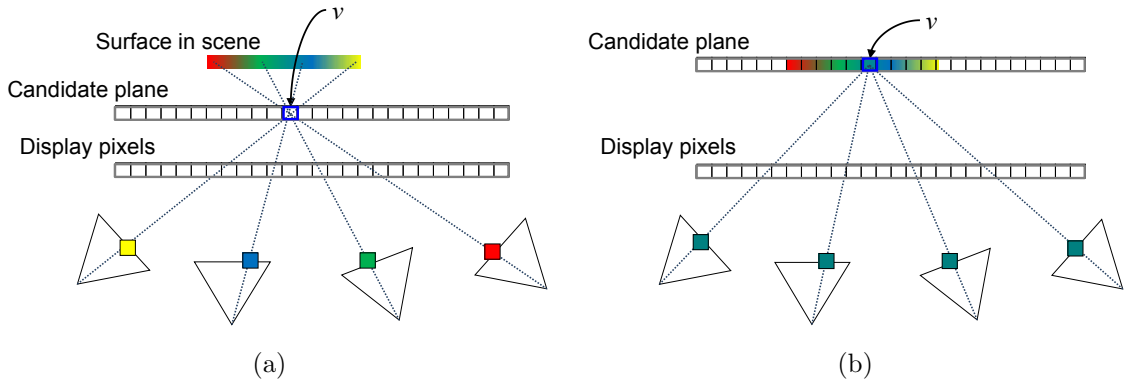


Figure 4.8: Plane sweeping comparisons: For a voxel v in the candidate plane in (a), the four cameras would see different colors, making it unlikely that there is a surface in v. In (b), all cameras see the same color at v, giving a higher likelihood of a surface.

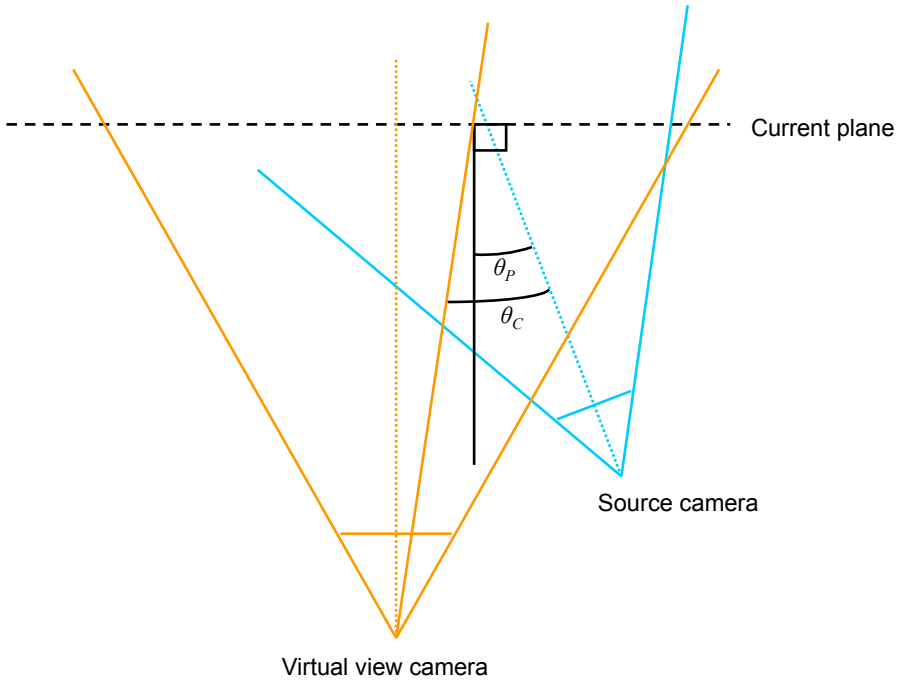


Figure 4.9: Weighting angles between the source camera, virtual view camera, and normal to the current candidate plane: θ_P is the angle between the source camera and plane normal, used for surface probability determination. θ_C is the angle between the source camera and virtual view camera, used for color weighting.

For each display frame to be rendered:

1. *Segment* dynamic object from the static background in each source camera image.
2. *Plane sweep* with segmented images and probability information.
3. *Composite* results into static background.

Segmentation For each new camera image, we segment new or moved objects from the previous frame or a background reference frame. This may include parts of the room that are newly visible, for example, if a table is moved to expose the back wall, or new objects such as a person entering the room. When the system is initialized, the stored background image will be empty and so the entire image will be considered for plane sweeping reconstruction. We call the background and other non-movable objects *static* and we refer to people and moving objects as *dynamic*.

The segmentation process is a multi-stage process for image differencing and shadow detection. The first step is to compile a background reference image from each input camera. To generate the background, a few dozen images of the static scene are averaged to reduce the effect camera noise. Each incoming camera image is then passed to the segmentation algorithm:

1. Convert camera image from RGB to HSV color space
2. Difference hue of HSV camera image with hue of HSV reference image
3. Threshold hue difference image to create the dynamic object mask
4. Filter hue mask to eliminate outliers
5. Difference RGB camera image with RGB reference image
6. Threshold RGB difference image to create the dynamic object/shadow mask
7. Filter RGB mask to eliminate outliers
8. Combine hue mask with partial alpha shadow mask

When successful, this algorithm returns a mask image that identified the dynamic objects in the scene and the shadows that they cast onto static scene objects.

Plane sweeping We perform a per frame front-to-back plane sweep of the segmented dynamic components. This may include multiple window sizes, such as 1x1 and 3x3 pixel windows, or multi-resolution source images (via MIP mapping). We use outlier elimination to exclude the contributions of cameras that do not agree with the majority of the tested pixels. These outliers may not match because of geometric offset, occlusion, shadows, or extreme specularity not accounted for in the color space conversion. For high probability matches, we place occlusion tags on the source camera pixels (as 0.0-1.0 probabilities) to reduce their contribution to future matching.

We formalize the scoring algorithm for a single display pixel at a particular plane depth. This is repeated for all pixels in every plane. Given a set of contributing cameras $C_{1..n}$ with reference camera R and a maximum score difference M, the probability of a surface at a given plane depth d is given by:

$$P_s = \frac{1}{n}\sum_{i=1}^{n} w_{\theta_P} w_R w_{C_i} (1 - \frac{(R - C_i)^2}{M^2}) \tag{4.4}$$

The probability is normalized for n cameras and is the sum of weighted squared differences between the reference camera value and all contributing camera values. The first weighting factor is $w_{\theta_P} = \cos\theta_P$ where θ_P is the angle between C_i and the normal to the candidate plane, as shown in Figure 4.9. The second weight w_R is the alpha weight from projected image of R, and the third weight w_{C_i} is the alpha weight from the projected image of C_i. The color value is determined by a normalized weighted sum of each of the contributing cameras with respect to the angle θ_{C_i} between the contributing camera and the virtual viewpoint instead of the plane normal (see Figure 4.9).

We present the pseudocode for the scoring algorithm in Algorithm 4-1.

Algorithm 4-1: Plane sweep algorithm pseudocode

```
clear output frame buffer Fb

for each depth plane p (from near to far by plane step size) {

  // project each image onto the current depth plane
  for each image Si (from 1 to n)
    Pi = Project Si onto p

  // choose a reference camera
  R = camera closest to output ray

  For each display pixel (x,y) {
    // Wi(x,y) is alpha value from projected Si images
    Wi(x,y) = Pi(x,y).A

    // SSD scoring, comparison of all other
    // camera images to reference camera
    score = 0
    for each window size (1x1,3x3,...)
      score += sum (i=1..n) (Wi(x,y) * (Pi(x,y) - R(x,y))^2)


    prob = normalize(1/score)
```

```
      // check for a better match
      if prob > Sb(x,y) {
        Sb(x,y) = prob  // update best score
        Fb(x,y).RGB = average all Pi(x,y) colors near mean

        // if we have an above-threshold match we
        // can stop using source camera pixels
        if prob > threshold {
          for each image Si (from 1 to n) {
            (s,t) = back project (x,y) to camera plane
            // set occlusion tag on source camera image in alpha
            Si(s,t).A = 0
          }
        }
      }
    } // end pixels
  } // end planes
```

Compositing We composite the output of the plane sweeping step with the existing static background image buffer for display. We use the values determined by the plane sweeping reconstruction to perform a depth-based composition with the background.

4.2.3 Implementation

The algorithm is currently implemented to test images of synthetic models generated using 3ds Max and camera images captured in a lab environment. With the synthetic model, the camera calibration is known from the defined camera parameters. The remote scene, including the room, people, objects, and cameras are modeled using 3ds Max and Poser. The scene is rendered from high resolution virtual capture cameras. Initially, 3ds Max was then used to project each of these camera images separately onto proxy planes and these projections were saved to disk. In later versions, OpenGL was used to accelerate the projection step with graphics hardware.

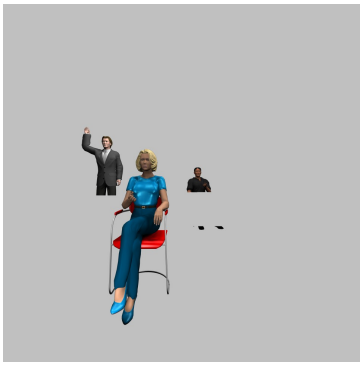
To use live camera imagery, we first calibrate the cameras to generate the correct intrinsic and extrinsic parameters for use in the projection step [38]. For both of these types of input, we use stored video sequences as input for offline processing. We use MATLAB for image segmentation, and custom software for computing the plane sweep scores and compositing.

For our break room scene, we sweep with a depth plane at every 0.5" through the room for a total of 192 different planes. For each depth plane in the sweep, the projected camera images are compared in 1x1 or 3x3 pixel windows to determine scores. Output pixel colors corresponding to a surface are saved as the output image, which is composited with a rendering of the 3D model or a background only sweep.

We have tested many different camera configurations in simulation, ranging from 3 cameras per display up to a total of 9 cameras per display panel. We have tested

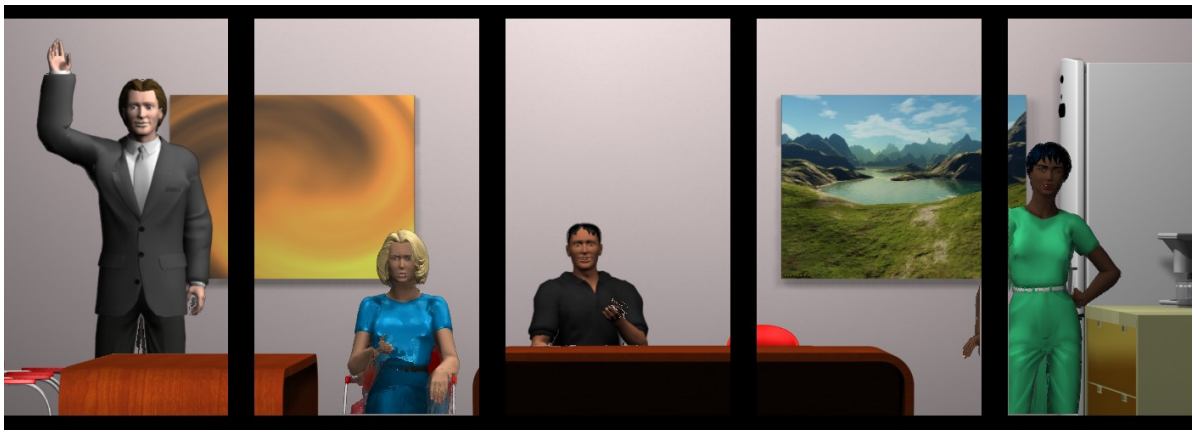
(a)

(b)



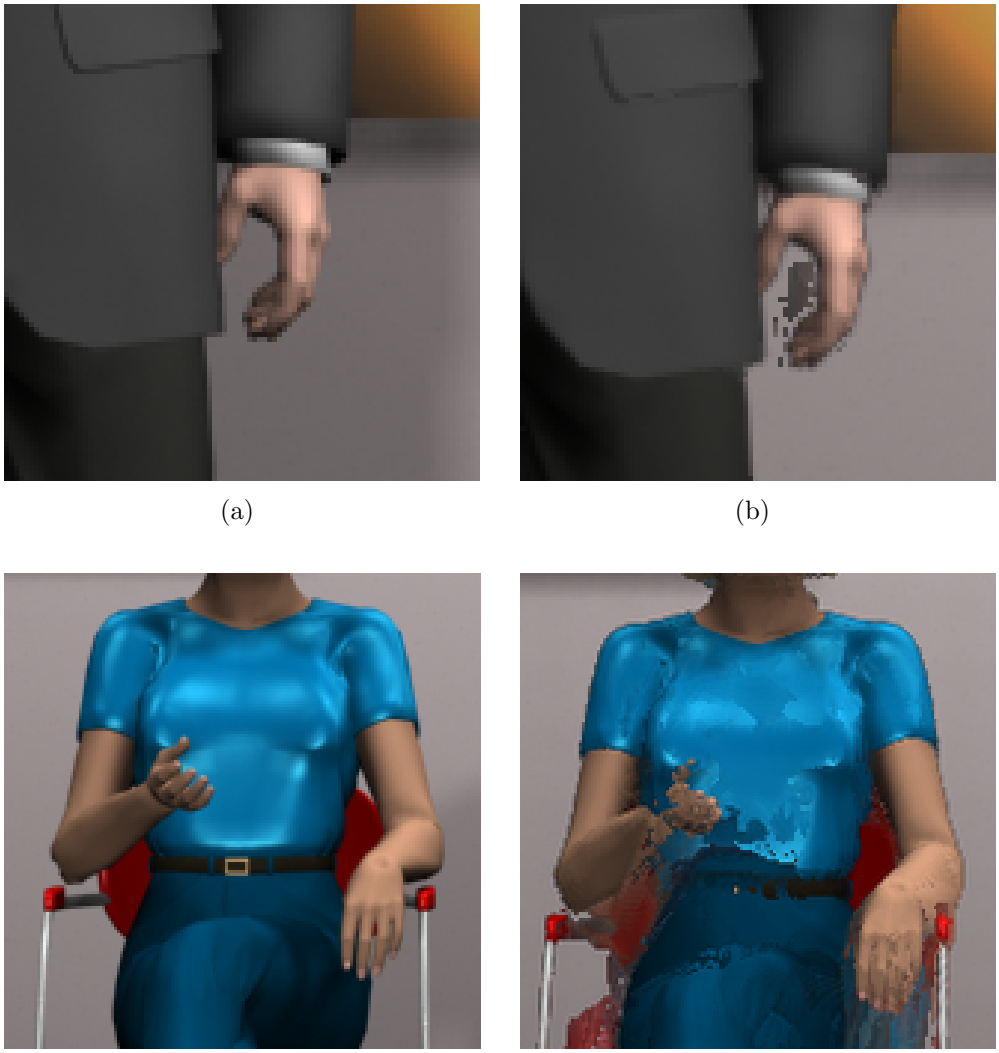
(c)

(d)



(e)

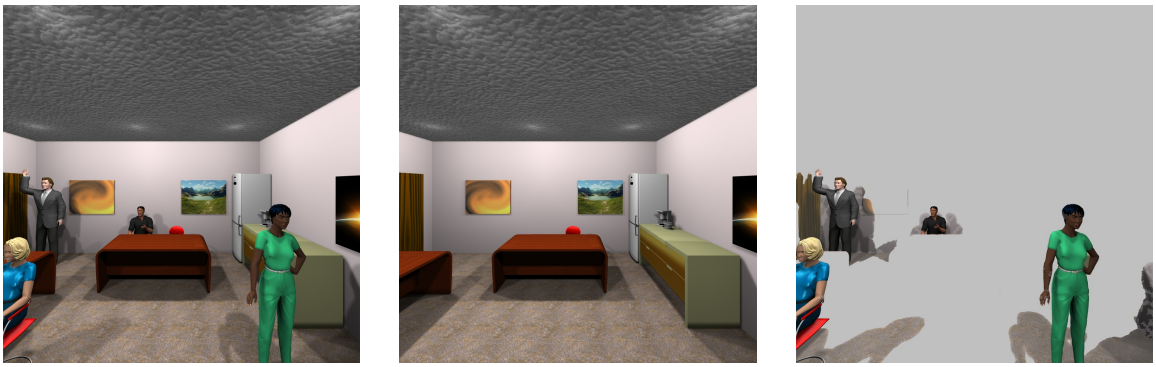
Figure 4.10: Synthetic scene plane sweeping results with perfect segmentation. (a) Selected camera view. (b) Static objects only. (c) Segmented dynamic objects. (d) Rendered view from 48ft. (e) Reconstructed with plane sweeping from 48ft.



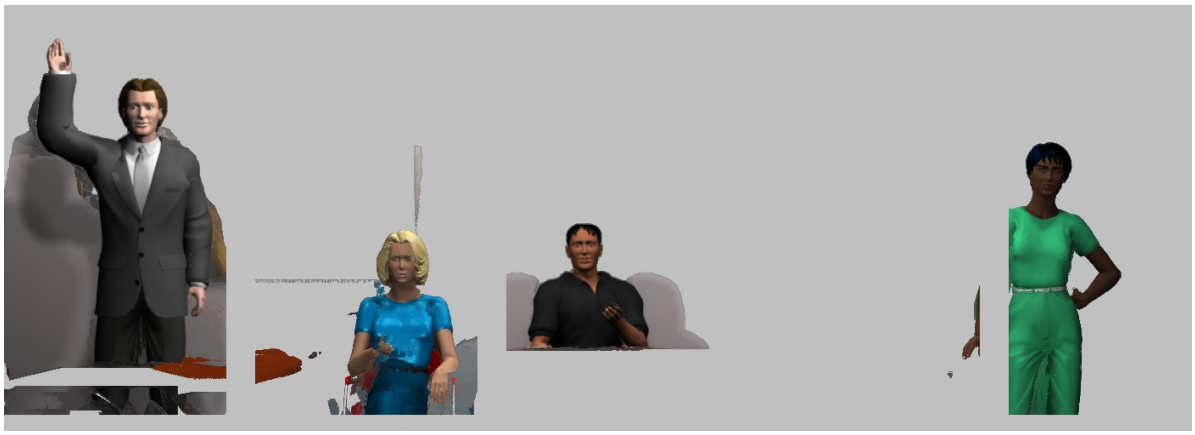
(a) (b)

(c) (d)

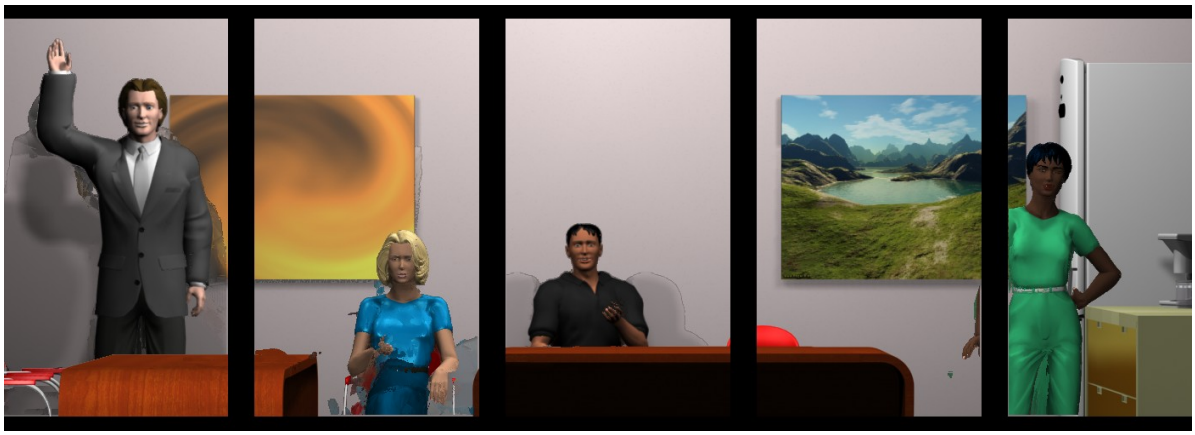
Figure 4.11: Comparison of rendered versus reconstructed views. (a) and (c) are closeups from the rendered view from Figure 4.10(d). (b) and (d) are closeups of the same regions from the reconstructed view shown in 4.10(e). (b) shows incorrect reconstruction due to lack of camera information. (d) shows incorrect reconstruction due to dynamic object self-occlusion.



(a) (b) (c)



(d)



(e)

Figure 4.12: Synthetic scene plane sweeping results with static/dynamic segmentation. (a) Selected camera view. (b) Static objects only. (c) Segmented dynamic objects. (d) Reconstructed with plane sweeping from 48ft before compositing. (e) Reconstructed view composited with static background.

configurations of single and multiple rows of cameras corresponding to different heights on the display wall. The plane sweeping results depicted in this chapter use 9 cameras per display panel in a 3×3 regular 18 inch grid, with 3 cameras on the left side, 3 in the center, and 3 on the right. The cameras between the displays are shared, so there are actually only a total of 33 images required, not 45.

For our live camera capture experiments, we only had six cameras, enough to cover a single display panel. We tested two configurations: two rows of 3, with the central cameras blocking part of the display, and also 3 pairs of two, on the top, left, and right sides of the display.

4.2.4 Results

We present the initial results of plane sweeping reconstruction in Figure 4.10. Parts (a)-(c) shows the idealized segmentation from the 3D modeling software to isolate the effects of the plane sweeping algorithm for study. For comparison, (d) shows the rendered view of the model from a centered camera 48 feet away. (e) shows the rendering of the reconstructed scene using 33 input camera views after it has been composited with the static background model.

Although the reconstruction is similar overall to the rendered model, there are several subtle differences that we show in Figure 4.11. (a) and (b) show a close-up view of the hand of the standing man in the back left corner. In the rendered view (a), the wall is visible between his thumb and fingers. However, in the reconstructed view (b), there are dark pixels in that region. This occurs because there are no clear camera views of the back wall along visible rays through the open hand, and so we are left with an incorrect guess for the depth of the surface along those view rays. Objects with holes or narrow views will often create such a problem for a multi-camera reconstruction system.

In Figure 4.11(c) and (d), we see a different reconstruction problem. The woman's arm is in front of her body and so she is self occluding part of her shirt that is visible from the reconstruction viewpoint from several capture cameras. Because multiple camera rays do not see this part of her shirt, this leads to low probability results and incorrect display pixel colors. More cameras reduce the uncertainty for both of these geometric problems, at the cost of added computation.

We also show results using the complete static/dynamic segmentation step, where the only inputs to the system are the static background image and an image for each camera, in Figure 4.12. Parts (a)-(c) show the dynamic and static image, the static scene, and the results of the segmentation. The shadows from the dynamic objects are preserved in this segmentation with partial alpha values. The reconstructed scene is presented in (d) and composited with the static background in (e). The self-occlusion problem with the woman is still clearly present.

Plane sweeping is limited to determining values at the plane step positions, which may be slightly different from the real depth of the objects in the scene. This is a particular problem for shadows, which may reconstruct slightly behind the back wall. To address this discretization problem, we shift the reconstructed scene forward by half of the plane step distance to ensure that the reconstructed shadows render in front of the static walls.

	Position	$z_{cam} = 0'$	$z_{cam} = 16'$	$z_{cam} = 48'$
Woman in green	(-8, 2)	75.96°	23.96°	9.09°
Woman in blue	(4, 6)	33.69°	10.30°	4.24°
Standing man	(7, 13)	28.30°	13.57°	6.55°
Sitting man	(0, 13)	0°	0°	0°
Cummulative error		137.95°	47.83°	19.88°

Table 4.1: Plane sweeping gaze error for each synthetic participant, from virtual camera positions at 0', 16' and 48'.

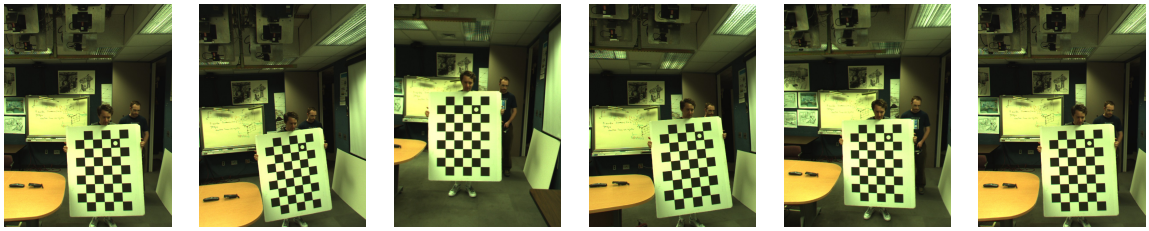
Gaze error analysis

We apply the gaze error measurement to each of the synthetic people in the remote scene, for virtual cameras at 16' and 48' behind the display. Table 4.1 shows the gaze error for each person, compared to the gaze error from a panoramic camera located at the display. The main goal of the virtual camera is to reduce the gaze error of people near to the display, so the two most critical samples are the standing woman in green and the sitting woman in blue. Rendering the woman in green, located 2' from the display, from the panoramic camera results in a gaze error over 75°. As the virtual camera moves behind the display, the error is significantly reduced, to 9.09° with the 48' virtual camera distance. As expected, there is no gaze error for the seated man because he is located at the central camera axis, and gaze error for the standing man is lower than the near participants because he is at the back of the room.

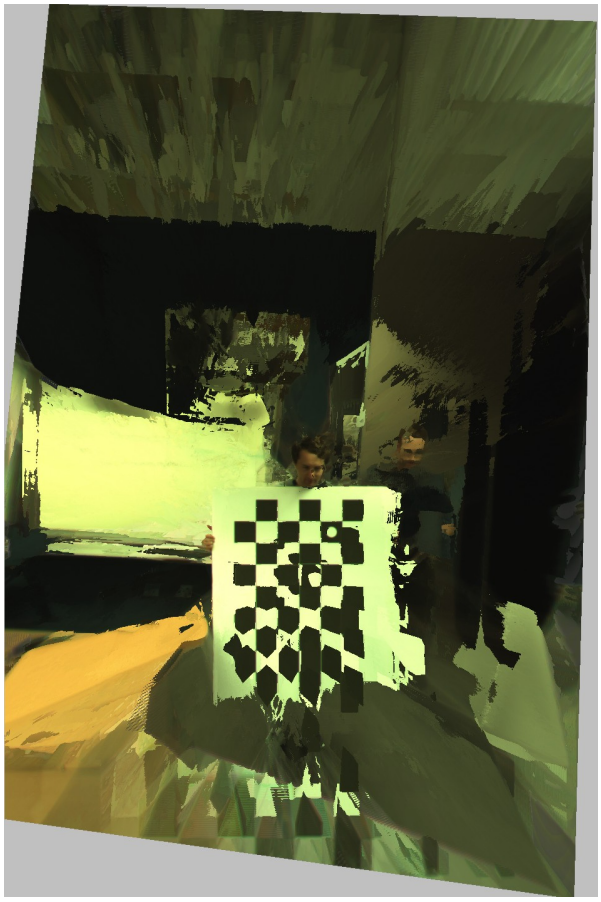
Initial real world tests

Finally, we present a plane sweeping reconstruction using real camera imagery captured in the lab. Figure 4.13 shows six camera images corresponding to two rows of three cameras each, arranged 18 inches apart above and below eye level. This data set was not segmented with a static background and so there are many areas where the surface depth is inaccurate. Many parts of the final image are only seen by two cameras, which makes correct surface calculations difficult. Only the central checkerboard pattern and part of the faces are rendered properly.

The distant virtual camera technique was successful in reducing gaze error angles to



(a) (b) (c) (d) (e) (f)



(g)

Figure 4.13: Initial real world plane sweeping results. (a)-(f) Input camera images. (g) Plane sweeping reconstruction of the scene.

below the 10° threshold for usable gaze angles, even for users near to the display and offset horizontally from the central camera axis. However, based on the noticeable visual artifacts present even in the ideal case of synthesized imagery, with correct calibration and perfect segmentation, we determined that plane sweeping-based reconstruction was not able to provide the image quality that we desired for a Telepresence Wall. Compared to a high definition video image that has no rendering or reconstruction artifacts, like the Cisco TelePresence system [13], even small rendering errors detracted from the tele-immersive experience.

The distant virtual cameras works for reducing gaze error, but further techniques to improve the quality and robustness of the reconstruction must be developed to provide a desired level of visual quality. Prior to plane sweeping, we noticed that the segmented objects had high quality imagery derived directly from the camera frames. These unmodified video silhouettes became the basis of our approach in Section 4.4.

4.3 Depth-dependent camera

For a large scale display, a single perspective view is desirable to provide a continuous display with natural feeling of perspective depth. With a centered, perspective view, remote participants near the center of the display who are looking straight at the display will appear to be looking straight ahead. However, participants to the side of the display looking straight ahead will be appear to be looking off to the outside of the display. This prevents local eye contact between users at the sides of the display with corresponding users at the remote location. We want to provide local eye contact between pairs of users at the outside of the display while maintaining the continuous perspective view of the entire scene.

Conventional rendering uses a single viewpoint from which to calculate the projection of the 3D objects onto the display, just as a camera captures a 2D representation of a real world 3D scene from a single viewpoint. In conventional 3D rendering (and with a pinhole camera model), all rays from the center of projection through the image plane to the remote scene are straight lines, and the center of projection is fixed. Alternative camera models, such as Multiple-Center-of-Projection images [85], have previously decoupled image content from a single camera location.

We introduce the *Depth-Dependent Camera* (DDC) to support local eye contact in large scale telepresence systems while maintaining a unified perspective view across the display. Objects near to the display are reconstructed from a distant or orthographic perspective, allowing for enhanced eye contact for users close to the display wall. More distant objects are rendered from a perspective camera close to the scene to provide appropriate perspective depth.

Compared to other multi-perspective images that change viewing direction across the image, this technique changes perspective per object depth in the scene. This allows objects at differing depths that are displayed at a given pixel to be rendered from different perspectives, when implemented in a telepresence wall system. Users close to the display will have a sense of local eye contact, while preserving perspective depth cues in the remainder of the scene.

4.3.1 Approach

The Depth-Dependent Camera was originally designed as a modification to the plane sweeping algorithm of Section 4.2. Instead of a fixed viewpoint in the projection step of the plane sweep algorithm, we move the projection viewpoint at each plane step. Objects that intersect a given depth plane are reconstructed from this unique viewpoint. We restrict the amount of movement of the viewpoint between plane steps to ensure a continuous view ray through the scene.

For near planes in the plane sweep, we set the projection viewpoint to be extremely far away, approaching an orthographic view. As the depth planes move back through the scene, we bring the projection viewpoint forward to a fixed view position. This results in people and objects near to the screen rendered from a distant perspective or orthographic view. All forward facing participants that are near to the screen will look like they are facing the screen, improving eye contact for the users at the side of the display.

Although the plane sweeping model is well suited to Depth-Dependent Camera rendering, we quickly realized that the DDC model is applicable to any reconstruction method that generates depth, including stereoscopic reconstruction or visual hulls, or direct rendering of 3D models through the graphics pipeline. Because it does depend on knowledge of object depths, it is not suitable for use with purely image-based rendering techniques.

4.3.2 Implementation

We have implemented the DDC as an offline rendering algorithm for 3D models. A conventional rendering stage and a compositing step are all that are necessary to generate DDC images. We have separated these two steps to allow for more flexible testing of various parameters, but they can be combined so that incremental composition happens after each slice is rendered. With incremental composition, it is possible to implement DDC primarily on graphics hardware without the need to save every intermediate slice. We show several slices through a simple scene in Figure 4.14(c)-(g).

Using the 3ds Max MaxScript scripting language [4], we render a set of slices of the 3D model. We choose a start and end position for the DDC path, and a number of slices to be rendered. The near and far clipping planes for the slices are set to the appropriate distance and the renderer draws the scene. We present the pseudocode for rendering the slices in Algorithm 4-2 and for compositing in Algorithm 4-3.

Algorithm 4-2: 3ds Max pseudocode for rendering slices of a scene.

```
Perspective48ft.clipManually = true

slicesize = (roomdepth/numslices)

for i = 1 to numslices do (

        Perspective48ft.pos.y = startpos + i * (endpos/numslices)
```

```
        planedepth = abs(Perspective48ft.pos.y)

        nc = (planedepth + i * (roomdepth/numslices))
        Perspective48ft.nearclip = nc
        Perspective48ft.farclip = nc + slicesize

        py = abs(Perspective48ft.pos.y)

        if py < 100
                then Perspective48ft.fov = 2 * (atan2 36 100)
        else Perspective48ft.fov = 2 * (atan2 36 py)

        bm = bitmap 1920 1080
        bm = render camera:Perspective48ft vfb:false

        save slice bitmap
)
```

Algorithm 4-3: MATLAB pseudocode for compositing rendered slices.

```
first = true;
for i = back:front
    [image, alpha] = read slice i

        a3 = initialize each RGB channel of mask to alpha

    if (first)
        buffer = im;
        first = false;
    else
        ii = immultiply(im, (a3/255));
        bi = immultiply(buffer, (imcomplement(a3))/255);

        buffer = imadd(bi, ii);
    end
end
```

4.3.3 Results

We first present a simple case that we use to test for simultaneous eye contact while maintaining perspective depth. Figure 4.14 shows a scene with two people standing 40ft away from each other. (a) shows the view from a perspective camera 64in away from the woman and slightly to her right. As a result, we see the perspective effect that she appears to be looking to her left when she is in fact looking parallel to the camera's

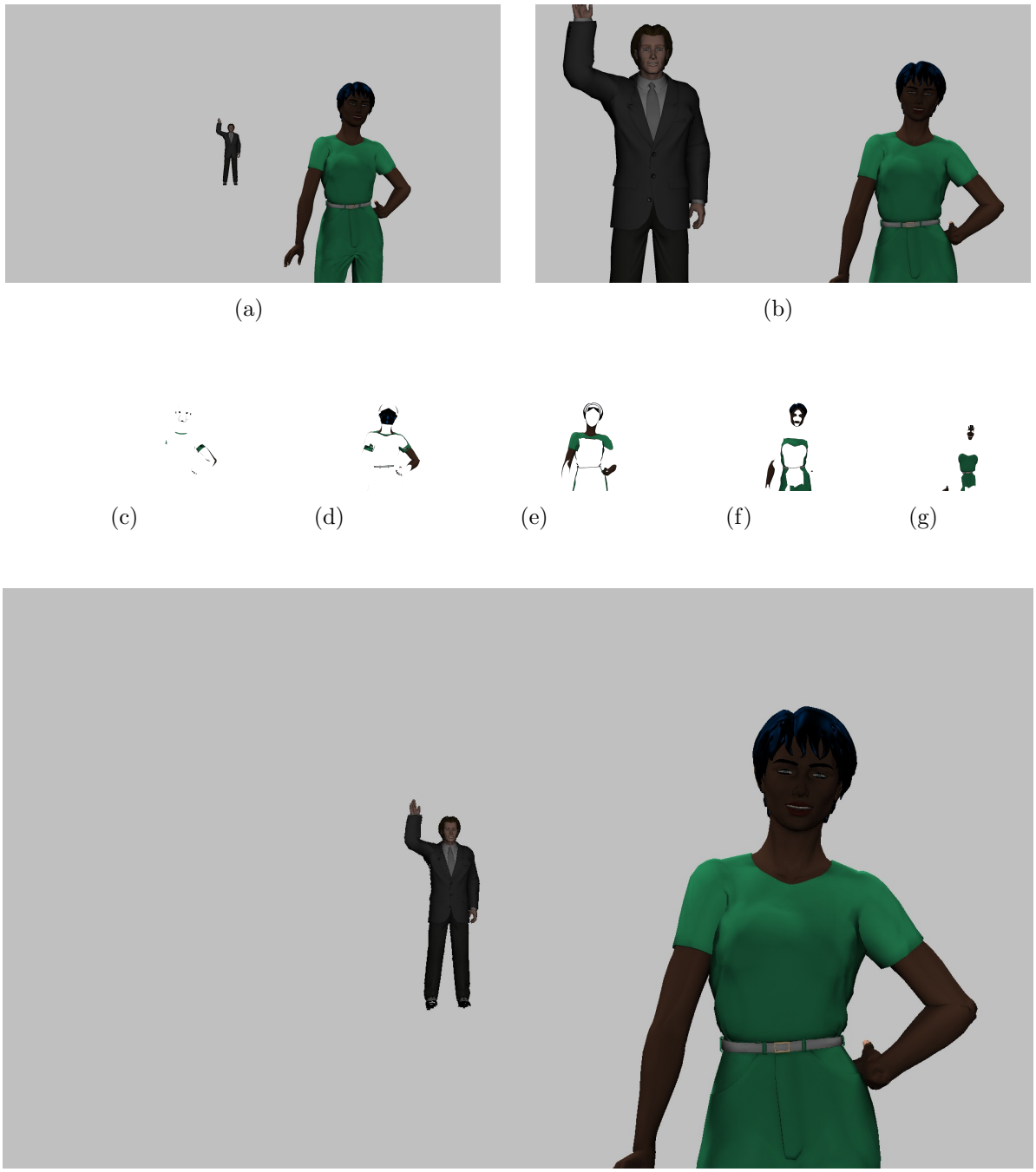
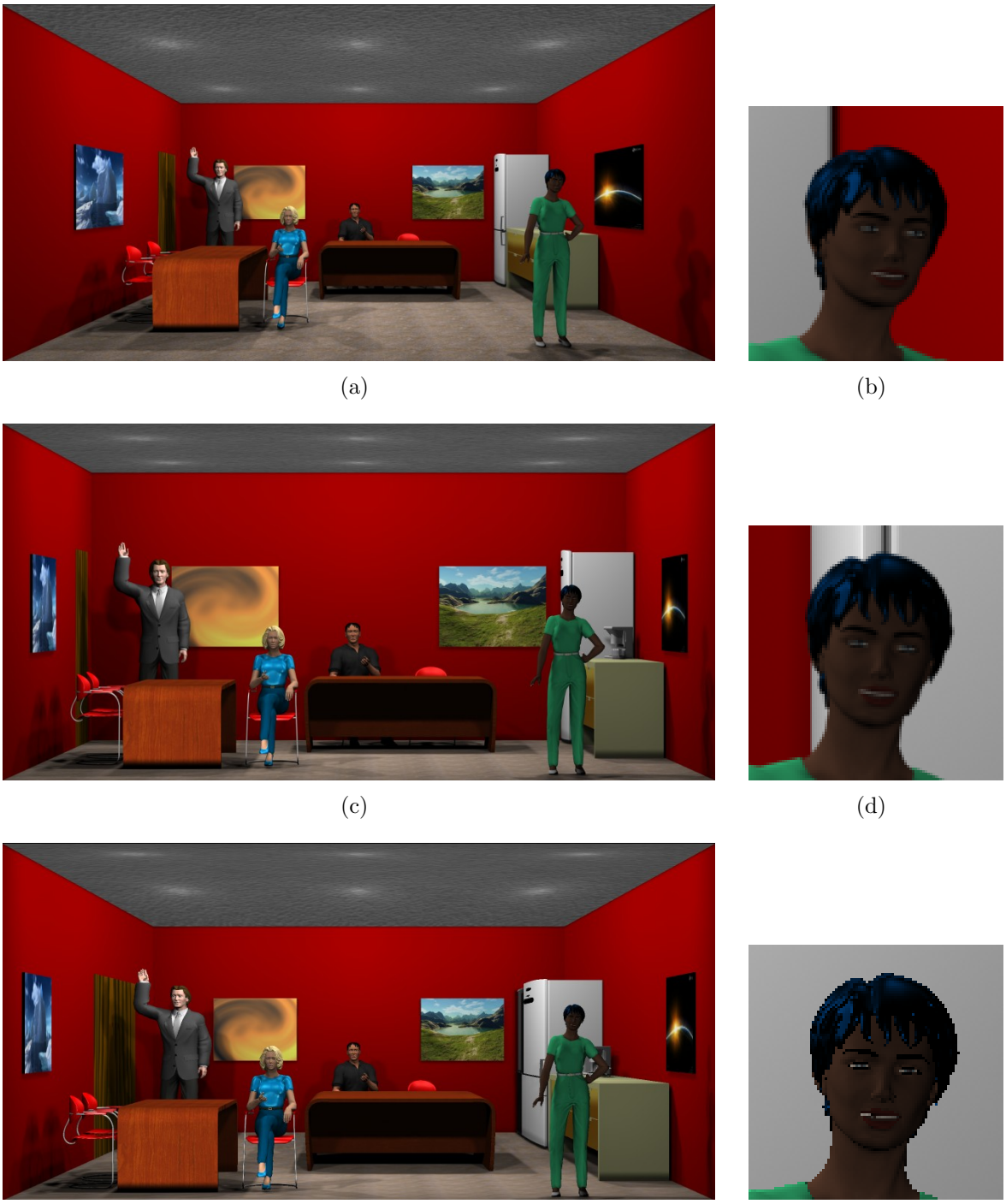


Figure 4.14: Two people standing 40ft apart. (a) Perspective view from 64in. Note that the woman appears to be looking to her left. (b) Perspective view from 1024in. (c)-(g) Selected DDC slices from the scene. (h) The synthesized DDC view preserves close eye contact with the woman and displays the distant man in proper perspective scale.



(a) (b)

(c) (d)

(e) (f)

Figure 4.15: Perspective and DDC views of the break room model: (a),(b) Perspective view from 16ft. (c),(d) Perspective view from 48ft. (e),(f) DDC view using 1024in to 256in camera depths.

viewing axis. (b) shows a distant perspective view from a camera 1024in away from the woman. As expected, this has the visual effect of eliminating the perceived distance between the woman and man, but we see the desired view of the woman looking straight ahead.

We apply the DDC algorithm to the model to generate a view that combines the perspective depth of distant objects and the more orthographic sense of nearby objects. Figure 4.14(c)-(g) show a few of the slices through the woman. We generate a complete set of DDC slices, and we composite them from back to front to generate the view in (h). This provides us with our desired effect, without altering the scene content in any way. If the woman was interacting via a telepresence wall, she would appear to be facing forward even though the camera is to her right. At the same time, the DDC view maintains the appropriate perspective scale for the distant man.

Although this scenario worked with the DDC algorithm, a more challenging environment was needed to test the continuity of DDC images, particularly with objects that run from near to far in the scene. We use a modified version of the telepresence wall scene model, shown in Figure 4.15. (a) shows the perspective view of the scene from a camera 16ft away, and (b) shows a closeup of the woman's head. This view depicts the desired sense of depth for the room, but the woman appears to be looking to her left, when she is actually looking straight at the wall. (c) and (d) show a perspective view from 48 feet away. This produces the desired effect for the woman near the wall, but the sense of depth for the scene is reduced.

We perform a DDC rendering of the scene, with the camera moving from 48' away for the first slice in the room at the display surface, to 16' for slices in the back of the remote scene. The DDC rendering output is shown in (e) and (f), with both local eye contact for the participants near the display and the same sense of depth for distant object as in a perspective view near the display.

In our testing we built a display wall consisting of three 61" flat panel displays with a projected image to simulate additional panels. We show the rendered DDC view display on our multi-panel video wall in Figure 4.16.

One drawback of the DDC system, visible in 4.15(f), is that it requires that anti-aliasing be disabled during the rendering of each slice. Anti-aliased edges at the border of a slice result in halos during compositing. Alternately, we can eliminate many of the aliasing artifacts by filtering of the output images after compositing.

Gaze error analysis

We repeat the gaze error analysis from Section 4.2 to compare the fixed perspective, 48' distant virtual camera ($z_{cam} = 48'$) to the output of the DDC algorithm. Table 4.2 shows the gaze error for each remote person and the difference between the fixed and depth-dependent cameras. The woman in green is rendered from 48' away in the fixed virtual camera. Because she is located 2 feet awy from the display surface, the DDC renders the the slices of the woman from approximately 47' behind the display. This produces a slight increase in gaze error, to 9.27°, but this is within the 10° threshold for the usable gaze area. Similarly, the gaze error for the sitting woman in blue increases by 1.20°, because she is further from the display.

	Position	$z_{cam} = 48'$	DDC	DDC-($z_{cam} = 48'$)
Woman in green	(-8, 2)	9.09°	9.27°	0.18°
Woman in blue	(4, 6)	4.24°	5.44°	1.20°
Standing man	(7, 13)	6.55°	11.31°	4.76°
Sitting man	(0, 13)	0°	0°	0°
Cummulative error		19.88°	26.02°	

Table 4.2: DDC gaze error for each synthetic participant, compared to a virtual camera fixed at 48' behind the display.

4.4 Video silhouettes

The plane sweeping view synthesis of Section 4.2 was able to reconstruct a single distant virtual camera view of the scene and generate imagery similar to a direct rendering of the model. However, there were always circumstances such as self-occlusion or lack of camera information that led to noticeable flaws in the output image, even with synthetic models and perfect calibration and segmentation. The reconstructed imagery was never as compelling to the research group as direct feed high definition video. With the additional difficulty of a high computational load for plane sweeping reconstruction of such a large image, we decided to pursue a rendering technique that preserved video image quality with a virtual camera viewpoint.

The most straightforward way to preserve video quality is to simply display the camera imagery directly on screen, the standard method for most video conferencing systems. For systems with a single camera, this approach produces a single continuous image that is easy to compress, transmit, and display. Single camera systems are necessarily limited in field of view (FOV) and resolution compared to using multiple cameras. However, systems with multiple capture cameras must deal with the potential problem of overlapping fields of view. When the views of two or more cameras overlap, some objects in the scene will be seen multiple times. Directly rendering the cameras onto a display will produce multiple images of those objects.

One solution to the overlapping FOV problem is alignment of the cameras at a common center of projection with aligned fields of view. Such camera clusters have been developed to create seamless high resolution, wide field of view imagery for teleconferencing [60]. Cameras that share approximately the same center of projection, as shown

Figure 4.16: Telepresence wall display with flat panels and projected imagery showing a DDC image of the synthetic scene.

in Figure 4.17(a), have been used in commercial video teleconferencing systems [13]. The clusters correctly map their video imagery onto a surround display. With a wall-sized flat display, a shared center of projection camera cluster exhibits the same eye contact and gaze awareness problems as a single wide field of view camera.

To support gaze awareness across the width of the wall, we must place cameras across that same distance to minimize the angular deviation between cameras and viewers looking straight at the wall. This makes the overlapping fields of view a significant problem. For all regions in the space past the point where the fields of view overlap, there will be double images. Some video conferencing systems limit this effect by restricting the area in which users are located [99], such as in Figure 4.17(b). Such restrictions also severely limit how close users can get to the display, due to gaps between camera fields of view. Figure 4.17(c) shows how there are always regions in between cameras that are unseen, regions that are seen by only one camera, and regions that are seen by several cameras.

A display system that allows users to move both close to and far from the display surface needs cameras with significantly overlapping fields of view. This requires a method to handle objects that appear in multiple camera images. We take an image

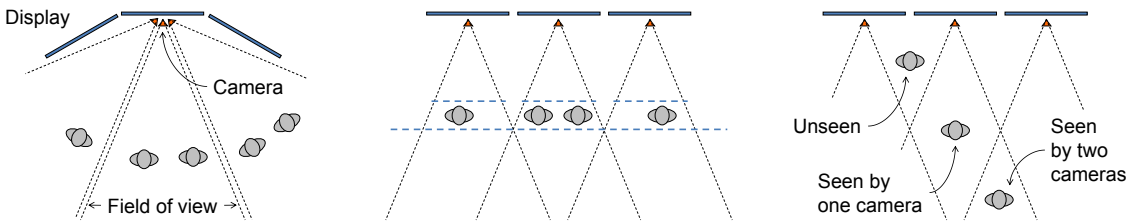


Figure 4.17: Top-down diagram of camera fields of view: (a) A shared center of projection and aligned fields of view prevent overlap between camera images. (b) Front facing camera systems sometimes restrict the usable area to prevent multiple images. (c) There are regions where people or objects will not been seen by any camera, or are visible in one or more cameras.

based approach to eliminating objects, by segmenting objects from the background, comparing them between camera views, and only displaying a single camera view of any given object. We call our technique *video silhouettes* – that is, a silhouette of the person in the scene is used as the surface on which their imagery is rendered.

4.4.1 Approach

To reconstruct and render the remote room from the virtual perspective camera, we run the video silhouette algorithm on sets of synchronized camera images. This algorithm eliminates the multiple images of objects that are seen in several cameras by picking the view of the object from a single preferred camera. The selected image is projected onto a proxy plane from the virtual camera position.

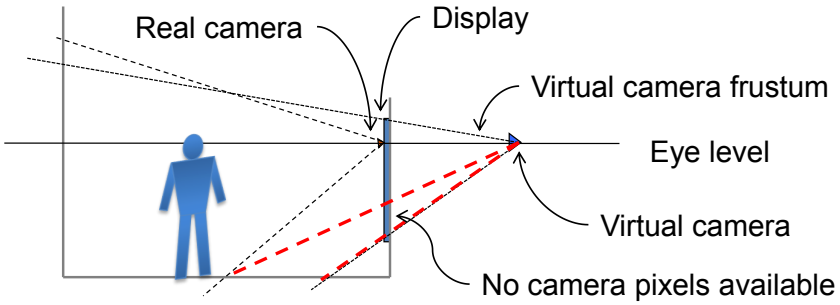
The distant virtual camera addresses two significant problems for a wall-sized telepresence system. It helps to address the eye gaze issue described in Section 4.2, and it handles the offset between the field of view of a user observing the display and the fields of view of the capture cameras. A virtual viewpoint at or near the display, with the large field of view necessary to cover the display, results in a region of the virtual camera image that is not seen by any real cameras, shown in Figure 4.18(a). This leads to an empty band across the bottom of the display. A distant virtual camera eliminates this field of view conflict because the view frustum of the virtual camera does not dip below the frustum of the real camera, as in Figure 4.18(b), and projects on the display panel from top to bottom.

Algorithm

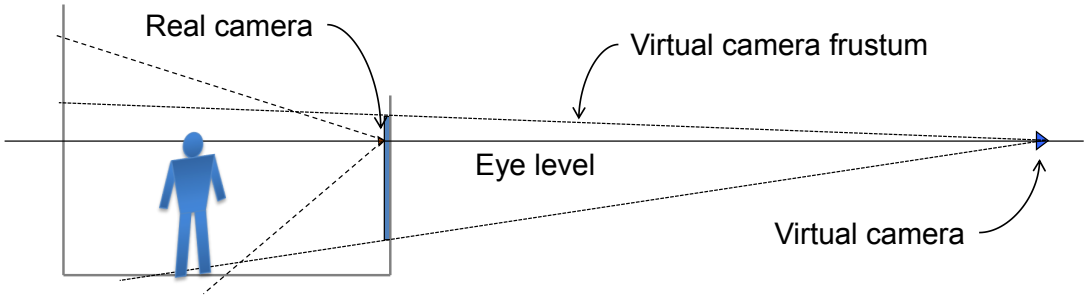
For each display frame to be rendered:

1. *Segment* dynamic components from the static background in each source camera image.

2. *Match* objects between camera views.



(a)



(b)

Figure 4.18: Side profile of telepresence wall geometry with a (a) close or (b) distant virtual camera.

3. *Select* the best object from the matches.

4. *Project* each object onto a proxy plane at the correct depth.

Segmentation The segmentation process is similar to the one used by the plane sweeping view synthesis algorithm but without the computationally expensive RGB to HSV color space transformation described in Section 4.2.1. We perform an efficient hue-based segmentation by differencing the change in each color channel against the mean difference in color change for each pixel. We segment the camera image against a stored reference background derived from a few dozen images of the static scene. The stages of the segmentation process, with intermediate results shown in Figure 4.19, are:

1. Compute RGB difference image between camera image and stored reference background.

2. Calculate mean of the RGB channels of the difference image.

3. Difference each color channel with the mean image and sum.

4. Threshold this image to create the dynamic object mask.

5. Filter hue mask to eliminate outliers by filling holes, and eroding and dilating the components.

6. Apply a Gaussian filter to smooth the mask edges for better compositing.

Matching across cameras We need to determine which objects are visible in which cameras. We match each of the segmented dynamic components in each camera to the components in nearby cameras by computing each component's color histogram. Components that have the most similar histograms are likely to be camera views of the same object in the scene.

The first step of the matching process is to filter out small segmented components. We use a component size heuristic to eliminate those smaller than a fixed level. This is a tunable parameter, but is usually set at half of the pixel count of the smallest view of a person in the scene. As such, it is dependent on the field of view of the camera and the distance to the object. This process removes small components and partial objects seen at camera image edges.

For each component in each image, the color histogram is computed. The number of subpixels of each color is counted. In an 8-bit image, as all of our capture imagery is, this results in three sets of 256 bins each. We append the bins to create a 768 element data structure. To compare one component to another, the Euclidean distance between their color histograms is computed. This matching technique has been used in previous work to track objects in video sequences [61]. Components with similar appearance will have a small distance, so the minimum values represent the best matches. This matching process is computationally inexpensive compared to calculating geometry or matching silhouette edges.

Match selection Once a set of components has been matched, one of those components needs to be selected for rendering. We use a simple selection method that determines the component that is most centered in its camera image. The most centered component is least likely to be at the edge of a camera image, and most likely to be selected in previous and subsequent frames.

Image warping The selected component is projected onto a proxy plane and then composited with the background panorama. We determine the depth of the object and its proxy by triangulation between the camera positions and the center of the segmented component.

4.4.2 Implementation

Our approach for a video quality telepresence wall system is implemented as a real time capture, offline processing, and real time display prototype. We capture synchronized video from a linear array of seven 1024×768 video cameras mounted on a rail, shown in Figure 4.20, corresponding to the width of our three panel display wall. This resulted in a spacing of 1.5' between cameras, for a total array width of 9'. The cameras are

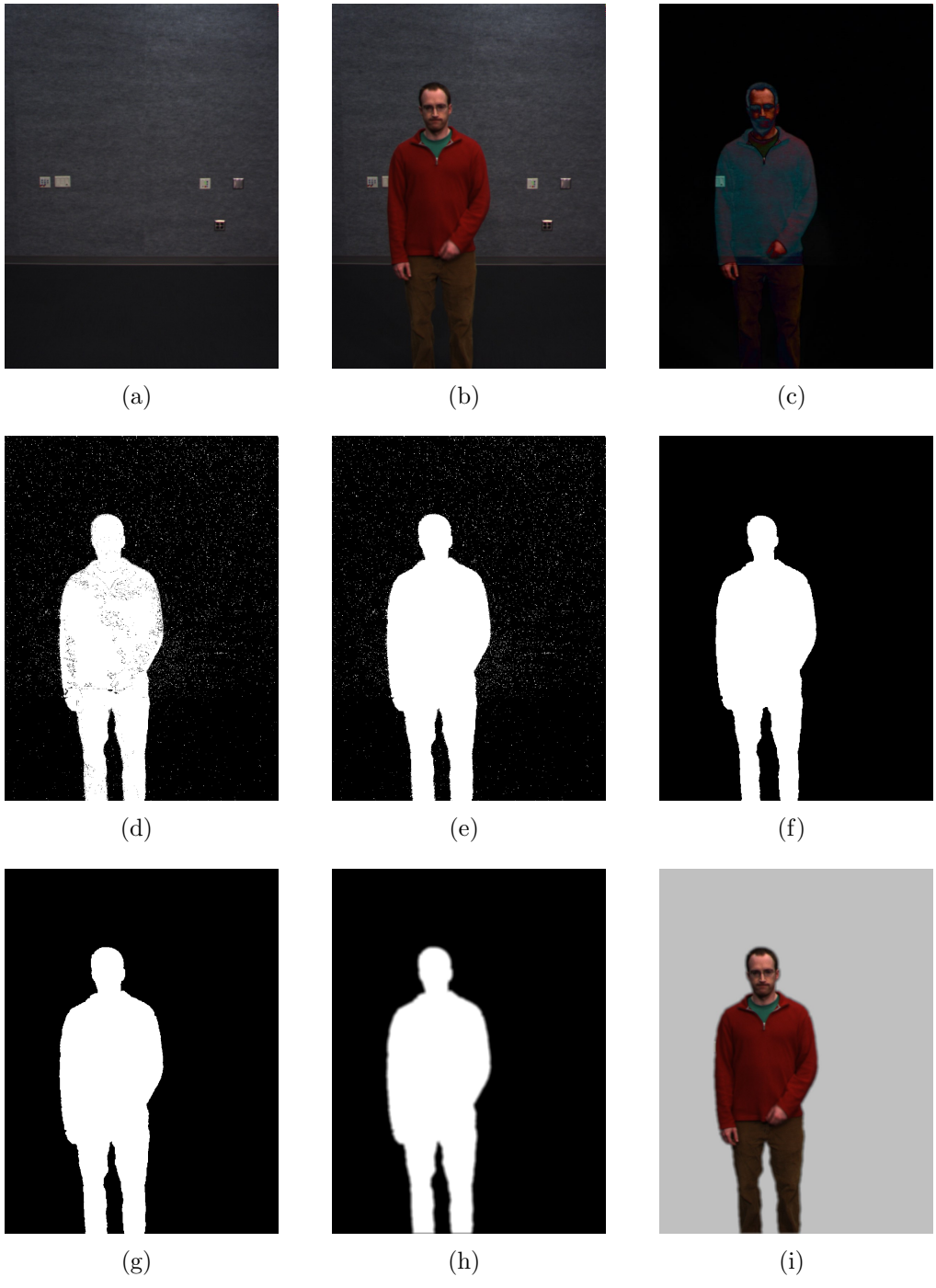


Figure 4.19: Each color channel of background (a) and camera (b) images are differenced separately and then combined to generate (c). This image is thresholded to generate a mask image (d). The mask image is filled (e), eroded (f), dilated (g), and a Gaussian filter is applied (h) to smooth the mask edges. The camera image (b) is combined with the final mask (h) to produce the segmented image (i).

(a) (b)

Figure 4.20: Linear camera array for video silhouettes: (a) Facing the cameras. (b) Looking past the camera array into the capture environment.

rotated 90 degrees so that the long axis is vertical. The camera rail was placed so that the optical centers were at an average eye level, approximately 175cm high. We test two camera orientations: straight ahead and tilted down approximately 15 degrees (as shown in Figure 4.18). The tilted down orientation reduces the size of the unseen region immediately in front of the display panel below the camera.

One advantage of the algorithm design is that it allows for loosely calibrated cameras. There is no requirement for precise calibration, and a manual alignment and measurement of field of view, zoom, position, and orientation are sufficient. Precise calibration will improve the depth calculation and positioning of the proxy, but the video quality of a rendered component is high regardless of the calibration accuracy.

The virtual camera is specified in the same manner as the parameters of the real cameras. The position, orientation, field of view, and resolution are set to generate images of arbitrary size. We typically generate 1920 pixel wide images with a field of view that matches the width of our display wall for the given virtual camera position. Other adjustable system parameters include the object filter size and the histogram matching method.

The stored video streams are processed with our algorithm to generate an output stream of high resolution video. This output is composited with a static background generated from the capture cameras to produce the final display image. A separate player application correctly spaces and scales the imagery for display on our three panel display wall.

4.4.3 Results

We display the output of the video silhouettes algorithm on our three panel display wall in Figure 4.21. This frame is computed from the set of 7 synchronized camera images in Figure 4.22(a). These images are segmented against the static background, producing the imagery shown in 4.22(b). The video silhouette algorithm processes these segmented

Figure 4.21: A photograph of video silhouette imagery shown on the telepresence wall displays.

images to produce a single output image for rendering on the display wall. Two examples are shown in Figure 4.23: (a) shows a scene with a single individual and (b) shows two people interacting with each other. In both cases, the subjects were approximately 8' away from the camera array. In general, all of the frames in a complete video sequence are processed and a movie is created for playback.

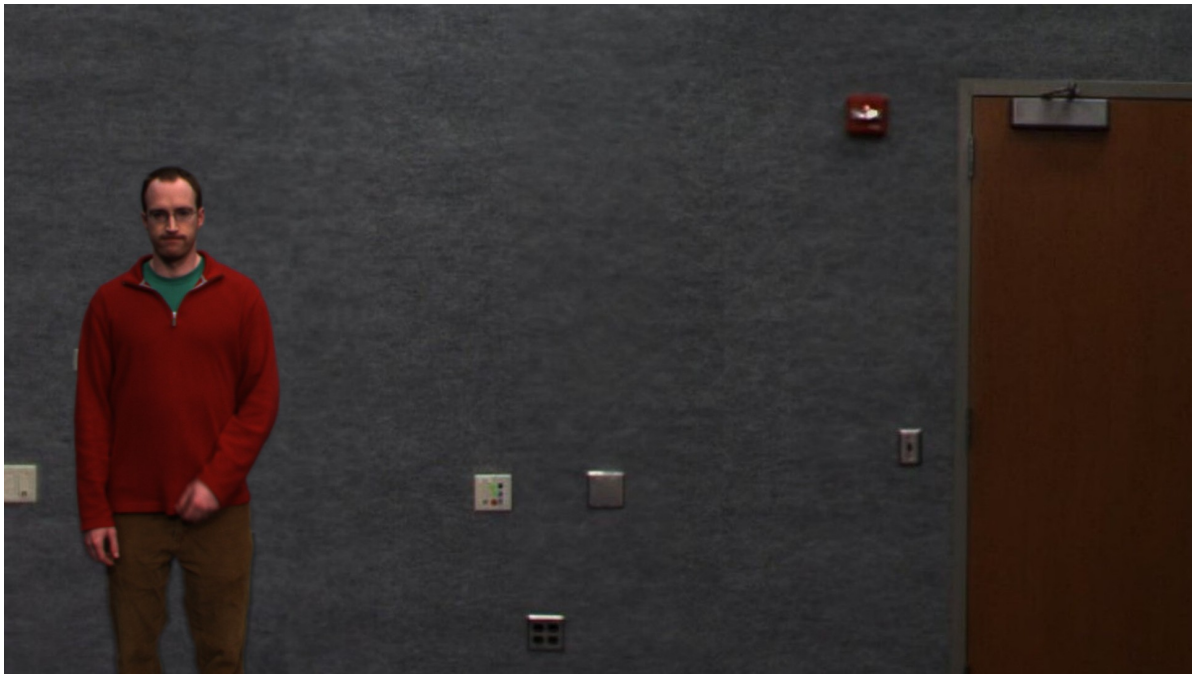
At 15 frames per second, a typical 10 second segment would consist of 150 frames per camera, for a total of 1050 images for the seven camera array. Each frame is stored as an RGB image with lossless compression and averages approximately 1Mbyte per 1024×768 image, for a total of 1GByte of data at an average rate of 100Mbyte per second. The output imagery is approximately 1.8Mbyte per frame with lossless compression. Because these frames are part of continuous video sequences, they lend themselves to significant size reduction through video compression. A 200 frame sequence totaling approximately 375Mbytes of data compresses to 6.5Mbytes using an H.264 codec, equivalent to just under 2Mbit per second. At such data rates, the video imagery may be transmitted over wide area networks at interactive rates.

We have captured and processed several dozen short sequences to test various situations, including users at different distances, multiple users, different motions, and different camera orientations. Figure 4.24 shows 18 consecutive frames from a sequence with two users. As one user moves across the scene to talk to the other, the images from

(a)

(b)

Figure 4.22: (a) Source video imagery from 7 synchronized cameras. (b) Segmented video imagery from 7 synchronized cameras.



(a)

(b)

Figure 4.23: Rendered output of (a) a single user and (b) two participants using 7 cameras.

Figure 4.24: 18 frames from a rendered output sequence.

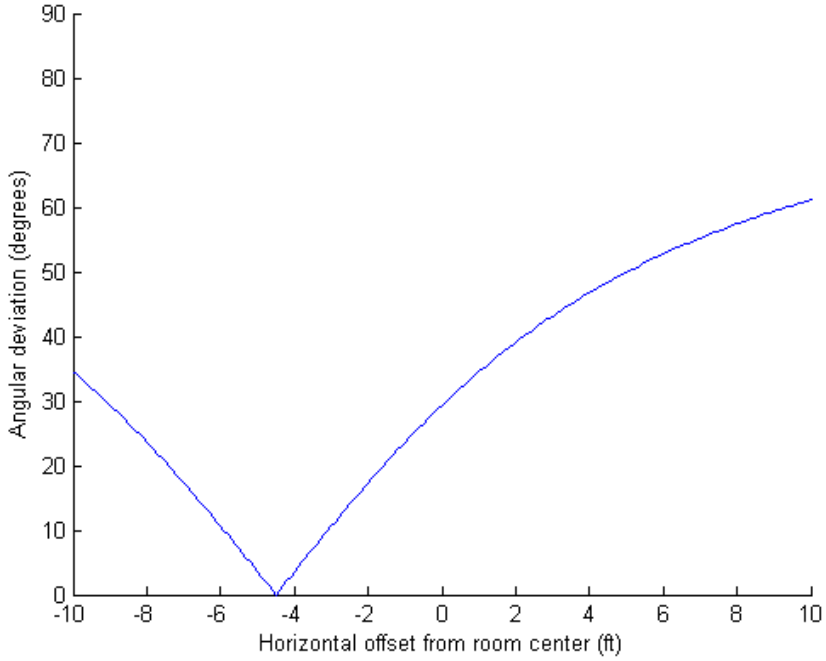


Figure 4.25: The gaze error for a single camera at the display wall, offset from center by 4.5', for a user at 8'. The horizontal axis indicates the horizontal (x) offset between the user and camera, and the vertical axis shows the gaze error in degrees.

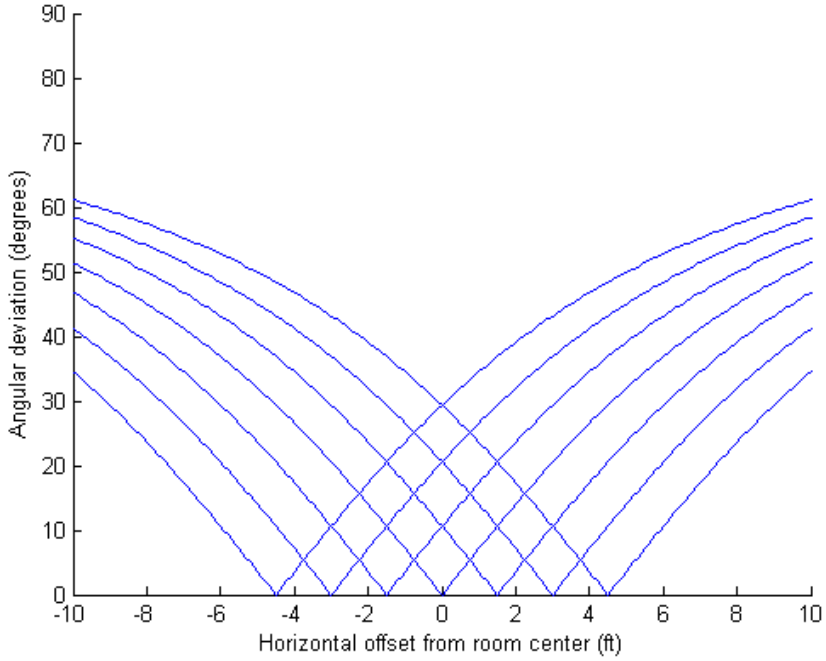
several cameras are segmented, matched, and selected to create a single image of each person. As the user moves, the final video silhouette is chosen from several different cameras. Although difficult to discern in still imagery, the switch between cameras is subtle but occasionally noticeable. In demonstrations, users did not find the switch to be disturbing.

Gaze error analysis

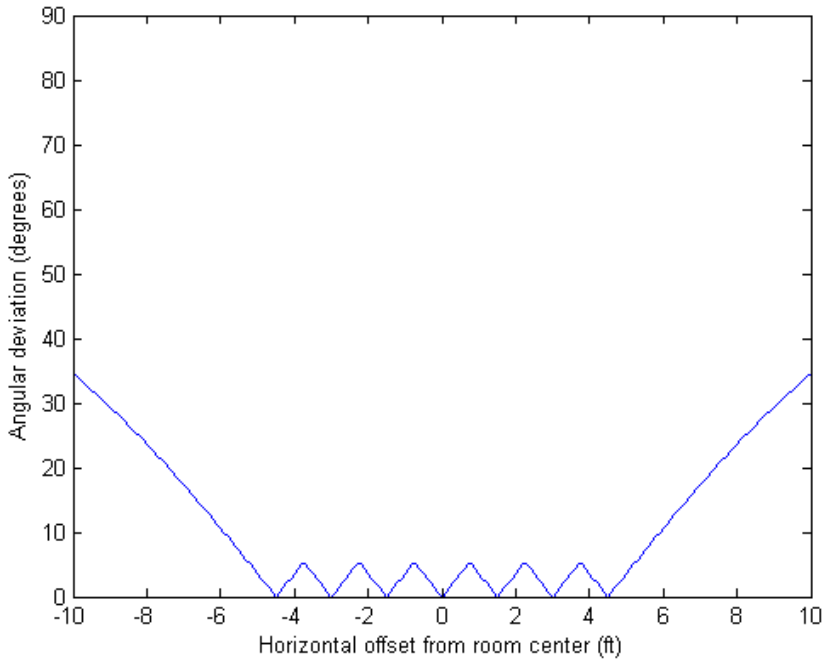
We calculate the gaze error for each camera in the array by the method from Section 4.1. Although the video silhouettes are warped and projected on to proxies from the perspective of the distant virtual camera, the imagery within the silhouette is only from a single camera at the display, corresponding to $z = 0$ and the geometry of Figure 4.2(a).

For example, the gaze error for the outside camera in the array for a user at 8' is shown in Figure 4.25. The gaze error exceeds a 10° threshold for most of the room width. The gaze error for each of the cameras is shown in Figure 4.26(a). However, the camera switching mechanism ensures that as the user moves in front of the camera array, the gaze error angle changes to that from the closest camera position. The gaze error for the camera array at a given horizontal offset is therefore the minimum gaze error from any camera at that position, shown in Figure 4.26(b).

The 1.5' camera spacing for the existing camera array was chosen primarily based on the display size and spacing, to ensure that most cameras were positioned between



(a)



(b)

Figure 4.26: The gaze error for 7 cameras located at 1.5' intervals at the display, with a user at 8'. The gaze error for each camera is shown separately in (a), and the combined minimum gaze error for the system is shown in (b).

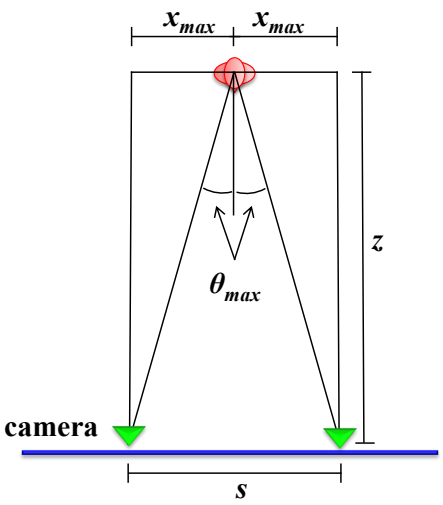


Figure 4.27: Given a maximum allowed gaze angle θ_{max}, the optimal camera spacing s is equivalent to twice the maximum horizontal offset x_{max}.

displays, with only a single camera in front of the active display surface. While the spacing maintained a gaze error of $\leq 10°$ for users at 8' across the width of the array, when users were closer to the display, gaze error exceeded this threshold. While this was sufficient for testing the video silhouettes algorithm and for providing local eye gaze awareness for many user positions, it did not fully satisfy the gaze requirements for users near to the display.

In order to design a camera arrangement that meets a desired threshold, we can derive a set of rules based on the equations for gaze error (Equation 4.1) and maximum horizontal offset (Equation 4.2). Figure 4.27 shows the geometry for adjacent cameras, given a maximum gaze error θ_{max}. For users at a given depth z from the cameras, the horizontal spacing s for two cameras in order to ensure a maximum gaze error θ_{max} is given by:

$$s = 2 * z * tan(\theta_{max}) \tag{4.5}$$

Similarly, we can derive the size of the area that guarantees a maximum gaze error. For an array of n cameras, the maximum horizontal space H_{max} at a given depth z with a maximum gaze error θ_{max} is:

$$H_{max} = n * s = n * 2 * z * tan(\theta_{max}) \tag{4.6}$$

However, to design a camera array to support a particular gaze error, we would start

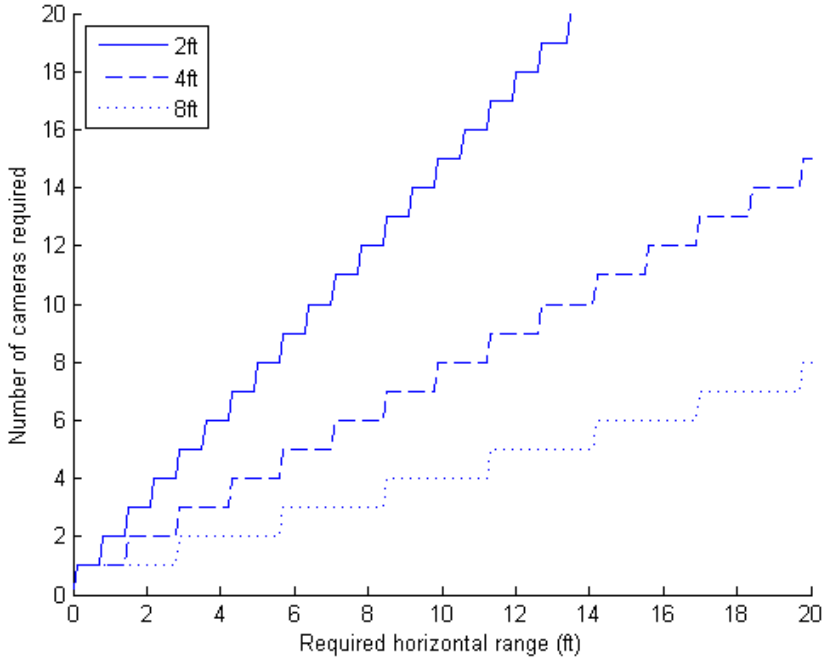


Figure 4.28: The number of cameras required to ensure a 10° maximum gaze error for a given horizontal distance, calculated by Equation 4.7. The three lines correspond to 3 different z offsets, with the user at 2, 4, or 8 feet from the display.

with a desired area and then determine the number of cameras required. The number of cameras required to ensure a maximum gaze error θ_{max} over a horizontal range H is determined by dividing the horizontal space desired by the maximum camera spacing and rounding up:

$$n = \lceil \frac{H}{s} \rceil = \lceil \frac{H}{2 * z * tan(\theta_{max})} \rceil \tag{4.7}$$

For example, to ensure a 10° maximum gaze error over a 15' wide region starting at 4' from the camera array, we calculate $n = 6$ from Equation 4.7. Figure 4.28 shows the number of cameras required to ensure a 10° maximum gaze error for various distances and horizontal ranges. The number of cameras required to support close user distances ($z = 2'$) grows quickly. Increasing the minimum allowable spacing significantly reduces the number of required cameras.

4.5 Discussion

Plane sweeping and video silhouettes are both view synthesis techniques designed for group tele-immersion with a wall size display. Both algorithms generate a synthesized

view of the scene from the perspective of a distant virtual camera. The plane sweeping approach generates novel imagery from a distant perspect camera, which reduces the gaze error compared to a panoramic camera located at the wall. The video silhouettes approach reduces gaze error by switching to the camera nearest to the target participant. Both techniques preserve local eye contact across the width of the display wall

The Depth-Dependent Camera model can be applied to either technique, to any number of other reconstruction algorithms, or used with conventional 3D rendering. It enhances local eye contact by rendering objects close to the wall from a distant camera, and preserves perspective depth by rendering objects further from the display wall with a closer virtual camera.

The visual quality limitations of the plane sweeping algorithm led us to develop the video silhouettes algorithm, but there are still possibilities for using plane sweeping reconstruction for a telepresence wall. As more cameras are added, the potential quality of the plane sweeping reconstruction increased, and faster computers will be able to take advantage of this additional data. Other techniques that may be used to improve scene reconstruction include a complete probability distribution data structure. With such a structure, we could inject models of known objects such as a table into the reconstruction process. Multi-pass plane sweeping could also be used to refine the surface depth estimates. There will almost always be some visual artifacts from rendering of scene reconstructions, but we may use the surface models from plane sweeping as a geometric proxy for rendering video data.

The Depth-Dependent Camera technique provides a method for combining different perspective views of a scene into a single continuous image, based on the depth of the objects. We use this technique to support eye contact across a large display wall, but it may be useful for entirely different applications, such as enhancing nearby details of a scene while maintaining awareness of the scope of a large model. Instead of linking camera distance to the object depth, the DDC model allows for a more general set of transformations. The camera may translate, rotate, or zoom depending on object depth.

Video silhouettes provide a computationally inexpensive method for generating continuous, high quality views of a large remote scene, with objects at varying depths. Because imagery from those cameras near to a user are composited into the final rendering, local eye contact is maintained for those users looking straight at the video wall. Because each silhouette object is treated separately, it is also possible to render each object from entirely different virtual camera viewpoints. When combined with Depth-Dependent Camera techniques, local eye contact and perspective depth could be rendered in a single continuous image.

The gaze error equations of Section 4.1 will allow designers of view synthesis algorithms to determine the correct position of virtual cameras in order to ensure a maximum gaze angle across a given display. Similarly, the rules derived from the gaze error equations for a camera array in Section 4.4 provide for the correct number of camera and their spacing.

Chapter 5

Monoscopic multi-view display: Generating unique views over a wide area

View synthesis techniques can only partially address eye contact and gaze awareness in a multi-user telepresence system. To be able to fully convey gaze awareness to multiple users at a single location, a system must provide a correct view of the remote scene to each viewer. With a shared 2D display, a remote user looking directly at a camera will appear to be making eye contact with all of the local viewers. If the remote user looks to one side, all of the local viewers will perceive that user looking to that side. Conveying correct gaze awareness requires both camera imagery corresponding to the viewpoint of each user and a method to display unique imagery to each as well.

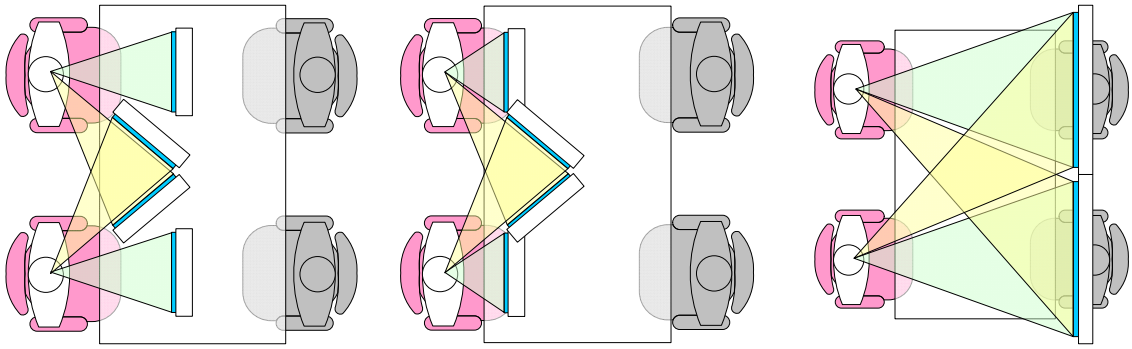
One possible solution is to use separate displays for each local viewer, but it is difficult to design flexible display configurations for different numbers of users. We rule out the use of head-mounted displays because those prevent eye contact. Multi-view displays provide a practical method to resolve this problem. Such displays send different imagery in multiple directions simultaneously from a single surface.

The most common type of multi-view display is stereoscopic, which generates an image for a user's right and left eyes to improve 3D perception. Many systems use special eye wear such as shutter glasses or polarized lenses, but these encumbrances also block the eyes from camera acquisition. Stereoscopic displays that do not require any visual encumbrances are called autostereoscopic. These are typically parallax displays that divide the source display pixels into a number of different views by way of a barrier or lenses. Other autostereoscopic techniques, such as holography and volumetric displays, have particular drawbacks that make them unsuitable for group tele-immersion.

Many different designs for parallax autostereoscopic displays have been prototyped, and several are commercially available. Most of the commercial displays are intended for applications such as entertainment or advertising, where any 3D image is novel and exciting. They are able to provide *a* 3D view from many locations, but they are typically unable to provide the *correct* view of a 3D scene from more than a few selected viewpoints. The display pixels are divided among the several views. Every additional

---

The research in this chapter was conducted by the author with Peter Lincoln from 2007-2008.



(a) (b) (c)

Figure 5.1: Potential multi-viewer display configurations. (a),(b) present unique views to each user with separate monitors. (c) represents the use of multi-view displays.

viewpoint leads to lower resolution and brightness. In order to support a wide range of user positions, such as in a group meeting, a conventional autostereoscopic display would need to provide dozens of views, which leads to unacceptable decreases in resolution and brightness. Displays that provide correct autostereoscopic views to multiple users usually have significant restrictions on user position and movement.

In this chapter, I describe a multi-view display that adapts the design of conventional autostereoscopic parallax displays to provide unique monoscopic images to several viewers. By trading off stereoscopic viewing, we can provide unique views to several users over the range of space required for comfortable group interaction. When combined with multiple video cameras at the correct locations, we can provide the correct gaze awareness for each viewer in a group tele-immersion system.

5.1 Background

The scenario for this work on multi-view conferencing is a meeting between two small groups, as if they were on opposite sides of a conference table, similar to the goals of the Line Light Field of Chapter 3. To correctly convey gaze awareness for the users at each site, a tele-immersion system must provide unique, view appropriate imagery of the remote scene to each viewer. This includes capturing the imagery from the correct location and displaying the remote users at the correct scale.

For the simple case of two viewers, it is possible to use multiple displays to provide distinct views in approximately the correct locations, as shown in Figure 5.1(a) and (b). One significant problem with such configurations is that movement is severely restricted. If a viewer moves from the designated viewing location, they are likely to see imagery meant for the other viewer. The monitors must also be placed closer to the viewer than the corresponding location of the remote participants across the conference table. The viewer will perceive the depth of the display at a different depth than the remote user.

This led to the consideration of multi-view displays that could be placed across the conference table at locations corresponding to the remote users, shown in Figure 5.1(c).

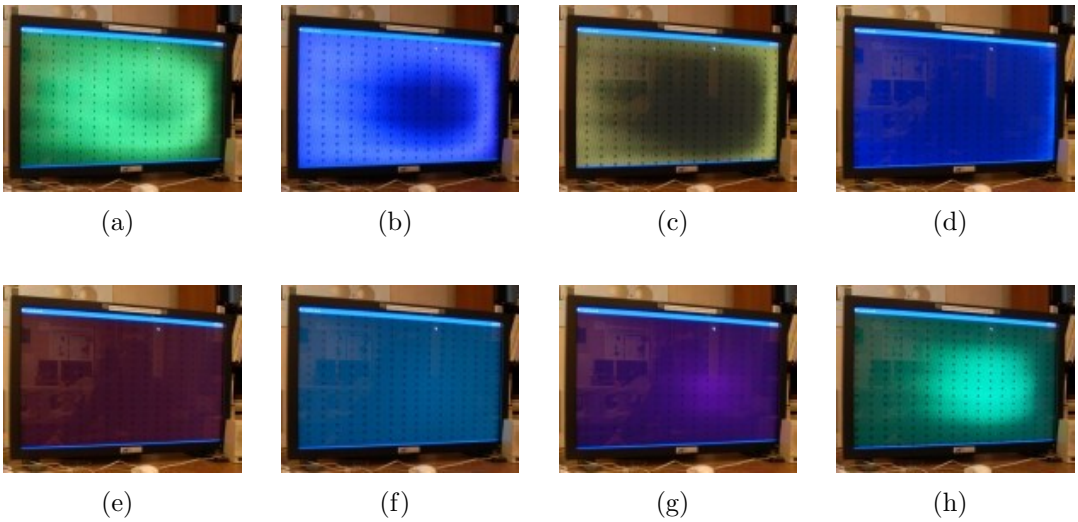


Figure 5.2: The Newsight 40" autostereoscopic display simultaneously provides 8 different views, spaced every 6cm, at 4m from the display.

Multi-view displays are commercially available in sizes large enough to display the upper body of users across the table. Several companies, including Philips and NewSight [81, 72], produce 40+" diagonal autostereoscopic displays, using lenticular lens and parallax barrier technology, respectively. If the multiple viewers of an autostereoscopic display are positioned such that their eyes are in distinct but non-adjacent viewing zones, it may be possible to provide unique views to the users at natural group seating distances.

5.1.1 Using a conventional multi-view display

We evaluated a NewSight 40" display for use as a multi-viewer teleconferencing display. The display provides 8 different views (shown in Figure 5.2) using a barrier to spatially divide the light from the subpixels of the backing display panel, in this case an LCD. The barrier has a hole for every 8 subpixels. These views are separated by the interpupillary distance (IPD) of approximately 6 cm at a default viewing distance of 4m from the display. This creates a total viewing zone width of 48 cm. The viewing distance is adjustable through a calibration process that changes the selection of active display subpixels for each view. The closest configurable distance for autostereoscopic display is 1.8 m, which is a reasonable distance between users across a conference table. This close calibration reduces the total width of the viewing zones to approximately 24 cm. The zones repeat every 24 cm as well, making it impossible to provide unique stereoscopic imagery to multiple viewers with the close calibration, no matter what their spacing

As shown in Figure 5.3(a), at the calibrated viewing distance, each view is separated by the IPD, e, in order to provide a stereoscopic effect. The total width of the viewing zones is this distance times the number of views, in this case $8e$. The number of possible stereoscopic viewing positions is one less than the number of viewing zones. We describe a viewing position as a pair (a, b) where a and b are the views seen by the left and right

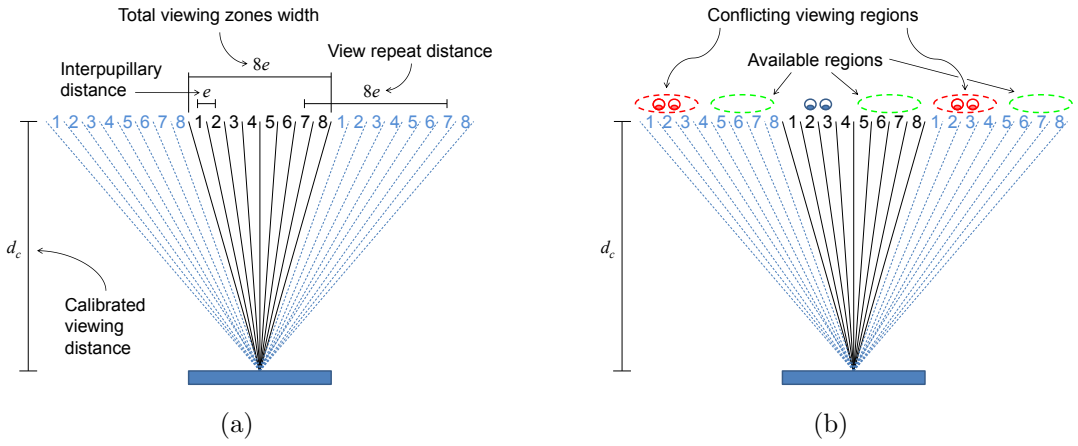


Figure 5.3: Repeating viewing zones in a regular barrier display: (a) shows the spacing of repeating zones; (b) shows the regions of conflict between viewers.

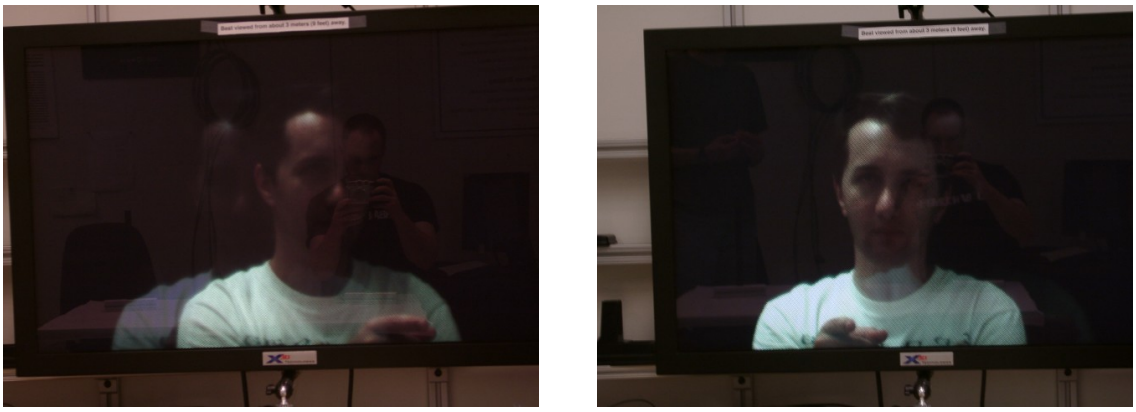
eye of a viewer. Correct stereo images are seen from positions (1,2), (2,3), ... (6,7), or (7,8). One limitation on viewing positions is that a viewer observing the last view and the first view, (8,1), sees incorrect imagery that is reversed and offset by $7e$.

A fundamental property of any regular barrier autostereoscopic display is that views repeat across space, because each pixel is seen through neighboring barrier holes. The image presented at a particular horizontal location x will also be seen at horizontal locations $x + i \times 8e$ where i is $.., -2, -1, 0, 1, 2, ...$ For certain applications, such as an advertising billboard, this is an advantage because stereoscopic views are presented to many locations in front of the display.

However, this is a disadvantage for providing unique views in several locations. A stereoscopic view presented to a user necessarily covers two views, but this has the effect of making up to 4 viewing zones unusable at other locations. Figure 5.3(b) depicts a user seeing views (2, 3). Any other stereo view that sees view 2 or 3 in either eye will make distinct imagery impossible, making viewing positions (1,2), (2,3), and (3,4) unusable in any other repeating locations. This leaves views 5 to 8 available for a second observer. In the actual display, a non-trivial amount of crosstalk between views is intentionally introduced to reduce sharp transitions between adjacent views. This further reduces the possible range of locations for multiple users to see distinct views.

In practice, these restrictions limit the 8 view display to only two possible viewers, for example, one observing views (2,3) and the other seeing (6,7). In testing with live video imagery from two offset cameras corresponding to the spacing between two local viewers, we still noticed substantial crosstalk, or ghosting, between views, as shown in Figure 5.4. Imagery intended for one viewer is partially seen by the other. At the desired viewing distance, a viewer cannot move much without seeing even more of the imagery intended for the other viewers. It is not possible to support three viewers with the 8 view display.

One solution to allow more movement or support additional users would be to increase the number of pixels per barrier hole. This would increase the number of views,



(a) (b)

Figure 5.4: Newsight 40" autostereoscopic 8 view display showing two live views. Notice the ghosting between views; imagery from one view is partially seen in the other.

but at the cost of reduced resolution and brightness per view. Another method is to change the distribution of light from the display. We noticed that as a viewer moves further away from the display than the calibrated distance, they eventually see a single monoscopic image from the display. Because both of their eyes are within a single viewing zone, there are fewer zones of conflict due to the repeating effect. If the width of each view can be increased at a closer viewing distance, it may be possible to provide unique monoscopic views to several users without viewing conflicts, supporting 2 or 3 viewers without sacrificing more resolution and brightness.

5.1.2 Lenticular parameters

Our design goals prevent use of wearable equipment, and so multi-view displays that rely on light polarization or time-division multiplexing are ruled out. The two primary candidates for developing an autostereoscopic display are parallax barriers and lenticular lenses. Both technologies are used in conjunction with a regular display panel, such as an LCD or plasma flat panel. Both techniques reduce the brightness and resolution of the backing display to provide multiple views. We selected lenticular lenses based on their successful use in various multi-view display systems [81, 63, 87]. I briefly describe some of the characteristics of lenticular lens design and their effect on light distribution in an autostereoscopic display

In order to locate the image position of an object in the optical system of a lenticule in front of a display, we first consider the general case of a lens with an index of refraction n, and two spherical surfaces with curvature radii of R_1 and R_2. An object O at a distance s from the lens produces an image I at the image distance s'. Figure 5.5 depicts the relative positions of O and I in the case where the object is located inside the focal point f of the lens, a common configuration in a lenticular multi-view system. The position of the image may be located graphically through ray diagrams, and the figure shows the following rays from the top of the object:

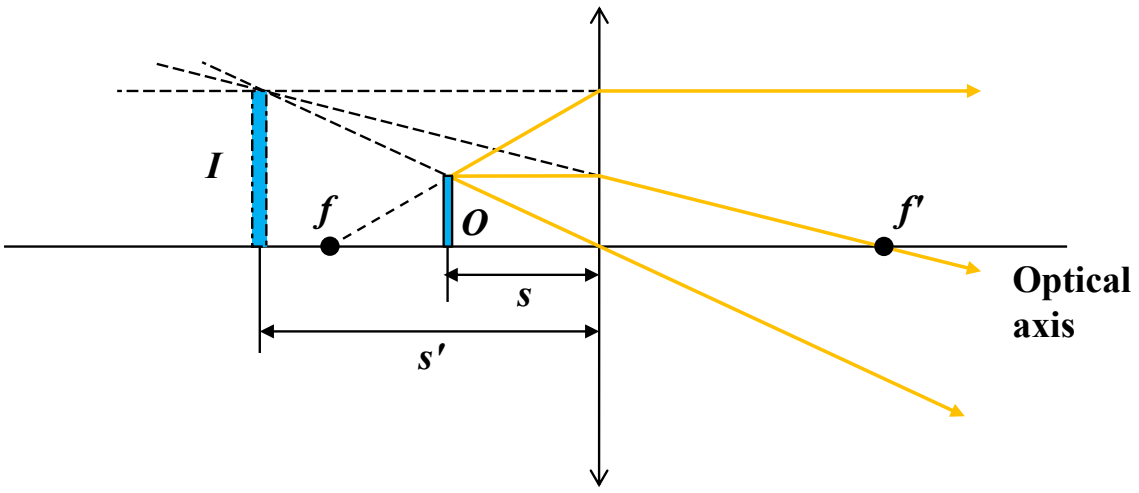


Figure 5.5: Ray diagram for locating the image I of object O when the object is located within the focal point f of a converging lens.

1. A ray from the focal point through O to the lens, where it is refracted parallel to the optical axis.

2. A ray from O parallel to the optical axis. After being refracted by the lens, the ray travels through the other focal point.

3. A ray from O through the center of the lens, which is not refracted.

When these rays are projected back from the lens, they converge at the top of the image, showing the position and scale. Figure 5.6 shows the position of several subpixels by this projection method. When the object is inside the focal point, the image is located behind the object, with the same orientation and some amount of magnification.

The relation between s and s' is given by the thin lens formula [93]:

$$\frac{1}{s} + \frac{1}{s'} = (n-1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \tag{5.1}$$

The focal length f of a thin lens is the image distance corresponding to an object at infinite distance. As $s \to \infty$, $f = s'$ and so Equation 5.1 can be rewritten as:

$$\frac{1}{f} = (n-1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \tag{5.2}$$

Equation 5.2 is known as the lens makers' equation, and allows for the calculation of f or the lens radii, given the alternate parameters. Substituting from 5.2, the thin lens

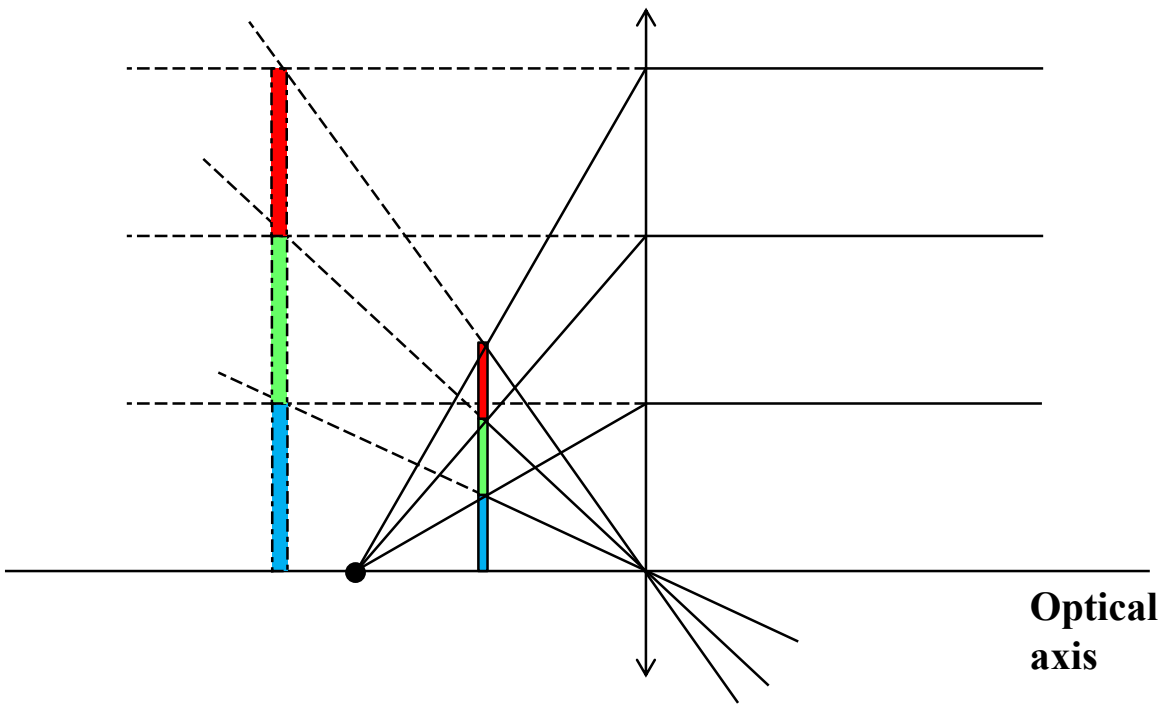


Figure 5.6: Expansion of the ray diagram for an RGB subpixel triplet.

formula (5.1) can be rewritten as:

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f} \tag{5.3}$$

The back surface of the lenticule is flat, and so R_2 is infinite, simplifying Equation 5.2 to:

$$\frac{1}{f} = \frac{n-1}{R_1} \tag{5.4}$$

Substituting this result in Equation 5.3 gives:

$$\frac{1}{s} + \frac{1}{s'} = \frac{n-1}{R_1} \tag{5.5}$$

And solving for s', we have:

$$s' = \frac{1}{\frac{n-1}{R_1} - \frac{1}{s}} \tag{5.6}$$

The magnification M of the object in the lens is defined as:

$$M = -\frac{s'}{s} \tag{5.7}$$

Equations 5.6 and 5.7 determine the position and scale of the object in the optical system of the lens, given the index of refraction n, the lenticular radius R_1, and the distance from the lens to the display pixels s.

From these initial parameters and the lens equations, we can construct an approximate model of the radiance from a given lenticule covering a set of subpixels to determine the light spread of each view from the display. The offset and overlap between the light emitted from each subpixel determines the number of usable views and their position in front of the display. Conversely, we can design a lenticular multi-view display by determining the lenticular parameters needed to produce a certain number of views and their positions.

We can vary the light distribution in an autostereoscopic display by changing these parameters. Figure 5.7 depicts the effects of several variations. Lenticules spread the light from display subpixels in particular directions. Reducing the thickness of the lens from (a) to (b), increases the light spread between each view. Decreasing the lens radius, as in (c), increases the overlap between views. Increasing the width of each lenticule, as in (d) provides more views but decreases resolution and brightness per view. Beyond these basic parameters, there are two signficant considerations in the design and analysis of a lenticular multiview system: color banding and cylindrical aberration.

Color banding

If lenticular sheets are placed vertically with respect to the display, each viewing zone will distribute light from a single color due to the arrangement of RGB subpixels. We rotate the sheets relative to the display normal to handle the horizontal RGB subpixel offset. By angling the lens sheet, a viewer in a zone sees a combination of RGB subpixels, allowing for a full color image. Figure 5.8 shows the subpixel structure of a flat panel display. To produce an equal amount of red, green, and blue light per view, the lens angle θ_l is determined by:

$$\theta_l = \tan^{-1}(\frac{p_w}{3 \cdot p_h}) \tag{5.8}$$

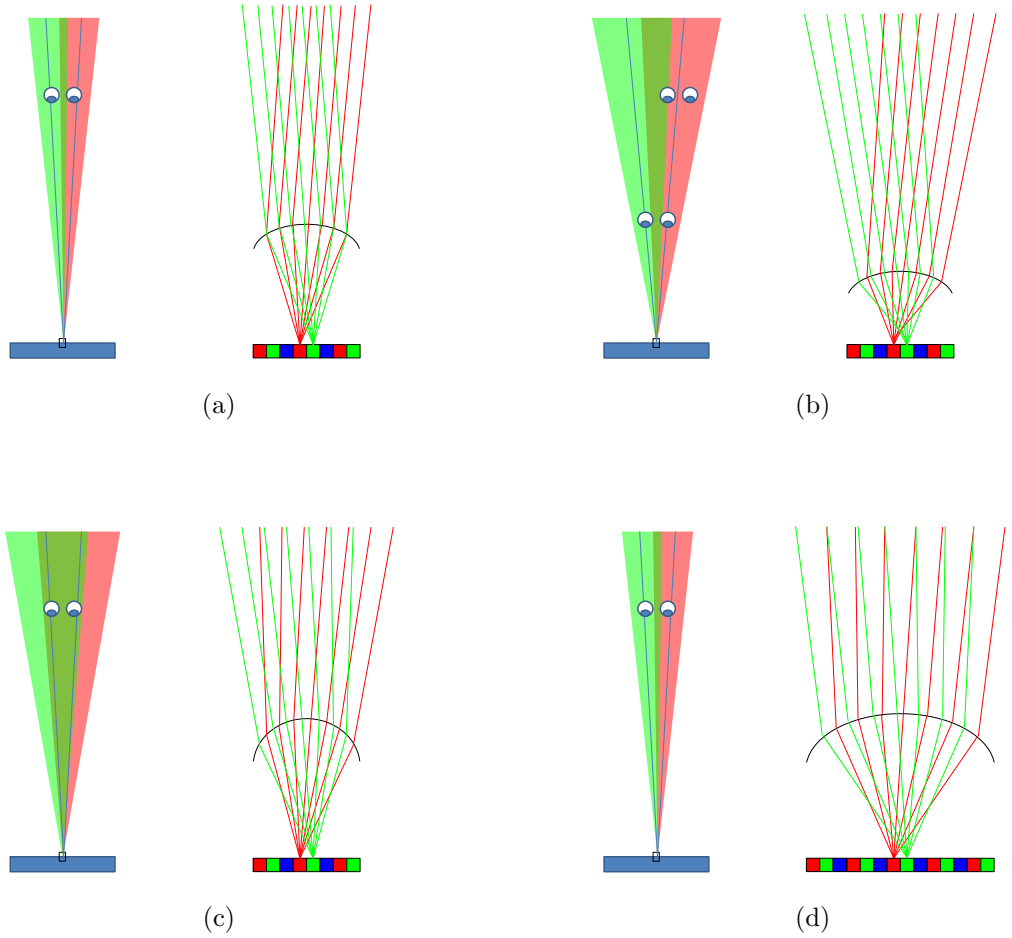


Figure 5.7: Effect of varying lenticular parameters in a multi-view display: (a) For comparison, a diagram of light spread from two subpixels through a lenticule. (b) Reducing the thickness increases the light spread of each view. (c) Decreasing the lens radius increases the overlap between views. (d) Increasing the width of each lenticule provides more views but decreases resolution per view.

where p_w is the pixel width and p_h is the pixel height.

Rotating the lens sheet relative to the display pixel grid also reduces the Moiré effect that results from the interference pattern between the lens pattern and display [80]. For a display with square pixels, θ_l is 18.4349°.

Cylindrical aberration

The lens model discussed above produces an object and image relationship based on paraxial optics, which applies only to small angle approximations for angles near the optical axis. While the paraxial approximation is sufficient for some lens systems, a lenticule covering a set of pixels typically has lens radius that is small relative to its width. This leads to significantly different refraction angles for rays passing through the

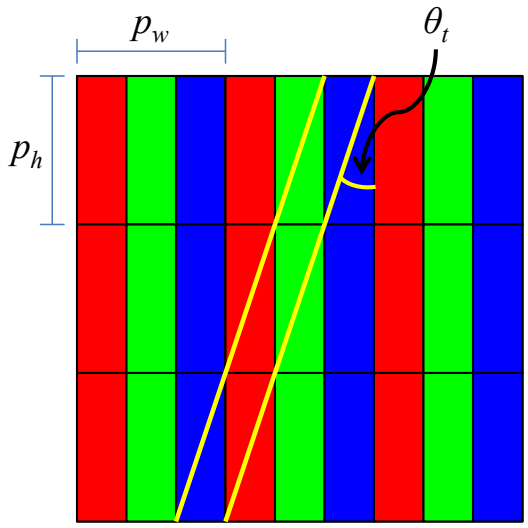


Figure 5.8: Diagram illustrating the subpixel structure of LCD panels. Columns of pixels are composed of three columns of subpixels. The pair of parallel, diagonal lines represents a participant's view of the display.

lens at different distances from the optical axis. In a typical lens, the most significant defect arises from the spherical shape of the lens and is referred to as spherical aberration. In the case of a lenticule, the aberration occurs only along one axis, and so we refer to the defect as cylindrical aberration.

The approximation for refraction in paraxial optics uses only the first term from the Maclaurin series expansion of the sine function [19]:

$$\sin I = I - \frac{I^3}{3!} + \frac{I^5}{5!} - \frac{I^7}{7!}... \tag{5.9}$$

Cylindrical or spherical aberration is a third order monochromatic aberration (occurring in all colors of light) along the optical axis. Third-order aberrations occur when the $\sin I$ in Snell's law is approximated by the first two terms of the sine expansion of Equation 5.9. For rays intersecting the lens outside of the paraxial region, we determine the angle of refraction by substituting the two term approximation in Snell's law [19]:

$$n'(I' - \frac{I'^3}{3!}) = n(I - \frac{I^3}{3!}) \tag{5.10}$$

We use this relation to determine the focal point for rays intersecting the lens at height

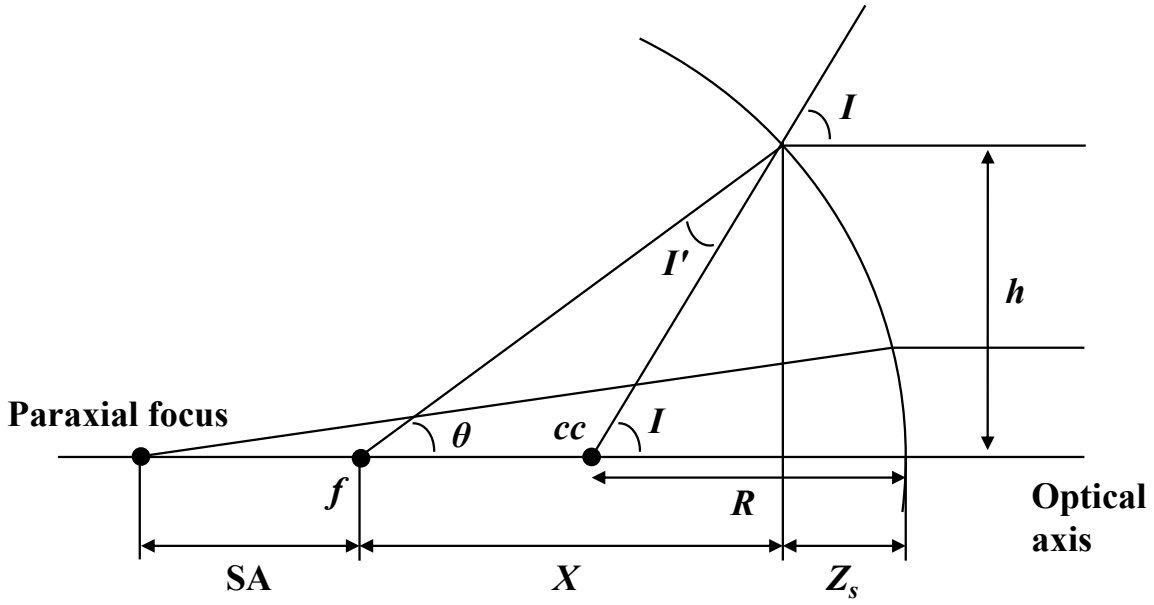


Figure 5.9: The spherical aberration SA is determined by the distance between paraxial focus and the focal point of a parallel ray entering the lens at height h.

h from the optical axis, as shown in Figure 5.9. The angle of incidence on the lens is given by:

$$I = \sin^{-1}(\frac{h}{R}) \tag{5.11}$$

Substituting this value into Equation 5.10 and solving for I' allows us to find θ by:

$$\theta = I - I' \tag{5.12}$$

The longitudinal distance from the point of intersection on the lens to the focal point is given by:

$$x = \frac{h}{\tan\theta} \tag{5.13}$$

In addition to the distance determined from refraction, there is also a shift due to the sag z_s in the surface. Sag is calculated as [19]:

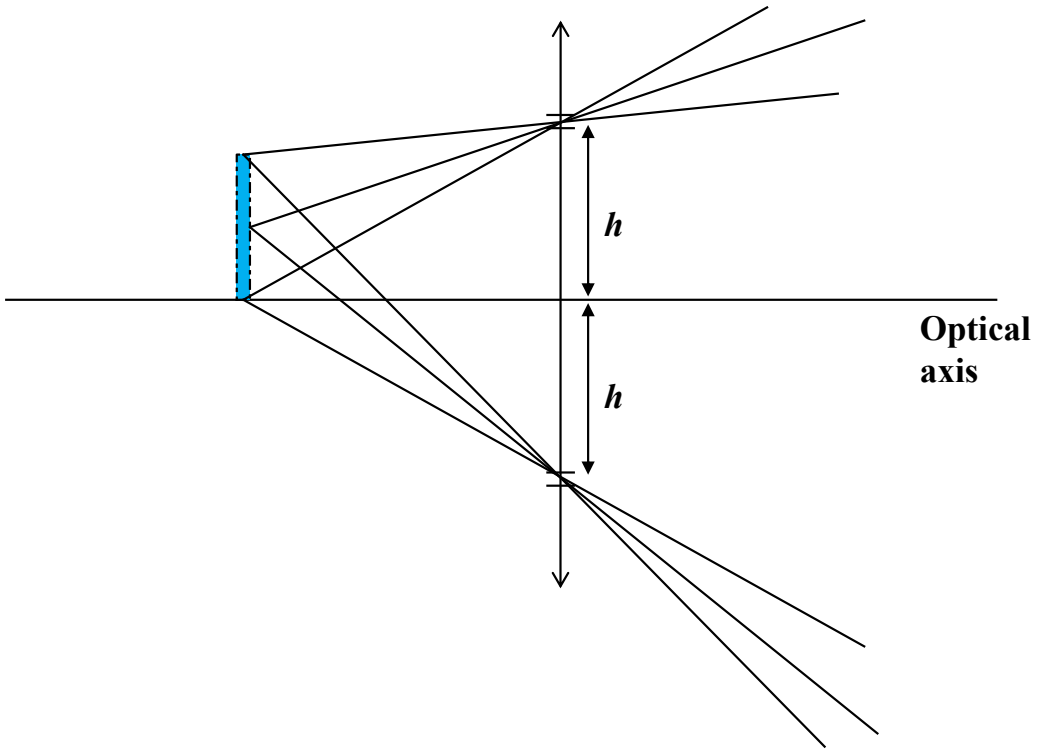


Figure 5.10: The regions of angular energy from the image of a subpixel through the pair of slits corresponding to height h.

$$z_s = \frac{h^2}{2R} \tag{5.14}$$

This results in a focal distance of $x + z_s$. The spherical aberration SA at height h is the difference between the paraxial focus and the calculated focal distance. We can use this solution for focal distance with Equation 5.6 to find the image distance s' for each height h. The distribution of radiance from each subpixel through a slit at height h is determined by the angles from the top and bottom of the image of the subpixel through the slit location, shown in Figure 5.10.

Note that the focal distance f is the same for the slit at height h above and below the optical axis. This allows us to make one calculation for the focal length and calculate the angular energy dispersal through both slits. The total energy from a subpixel through a lenticule is the integration of energies through all of the slit heights, from 0 (at the optical axis) through half of the lenticule pitch p:

$$E = \int_0^{p/2} e(h)\, dh \tag{5.15}$$

We calculate the energy by angle for each subpixel to determine the spread and location of potential viewing zones.

5.2 Display prototype

We have developed a prototype multi-view lenticular display to convey gaze awareness to multiple users in a video teleconferencing scenario. A custom lenticular sheet is placed in front of a flat panel display, directing the light from various pixels in different directions. The parameters for the display and lenticular lens sheet are defined based on the requirements of our multi-user scenario. Ideally, we would like to provide distinct views of four remote users to four viewers sitting 1.8 m apart across a conference table, as shown in Figure 5.11.

Due to the wide aspect ratio of available HDTV display, we expect to display two remote users per monitor. This requires two monitors, each with a lenticular sheet, of a size sufficient to display the upper bodies of two users at life size, side by side. The displays are angled inward so that the central axis of each points to the middle of the group of viewers. This allows a nearly symmetric distribution of views across both displays.

The views themselves should be spread wide enough to cover all four local viewers, while providing unique images to each. Because there is significant overlap in light dispersion of neighboring subpixels, we will need to incorporate *guard bands*, views without imagery that prevent crosstalk between active viewing locations. For four viewers, we expect a minimum of 8 different views, 4 active and 4 guard bands. To provide additional flexibility in the number of viewers and to allow for some natural user movement, we would like to provide two views to each user, and allow for 1-2 guard bands between each view. We chose to provide 15 different views, each wider than the IPD, corresponding to 15 subpixels (5 complete pixels), shown for one display in Figure 5.12. These views should cover at least 3 m of horizontal space in front of the display to allow the users to sit next to one another with a comfortable spacing.

In addition to the display itself, we have created a multi-camera capture and rendering system for displaying unique live video to several viewers from the correct perspectives. By calibrating the display from the viewpoint of each user, we are able to determine which pixels are visible from each position. With this per view calibration, we can combine the appropriate camera images into a single image sent to the display. This image is transmitted through the lenticular sheet, resulting in the unique, correct views displayed to each user, allowing the proper determination of eye gaze between remote and local viewers.

5.2.1 Display calibration

The goal of the display calibration process is to determine which pixels are visible from each of the desired user locations. A complete calibration includes the initial

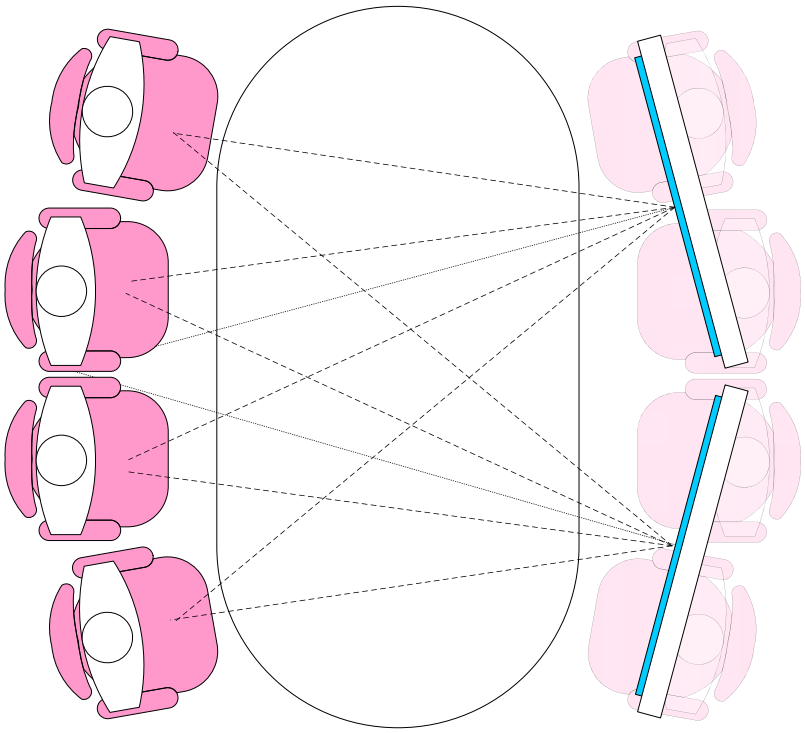


Figure 5.11: The four-on-four teleconferencing scenario goal includes four remote users depicted on two large multi-view displays that provide unique views to each of four local viewers.

positioning and alignment of the lenticular sheet on the display panel, camera capture of a series of images to determine pixel visibility, and generation of the final display masks. The calibration method can calibrate arbitrary multi-view displays from a given set of viewpoints and is not dependent on any knowledge of the display technology.

Lens alignment

The first step in the calibration process is alignment of the lens sheet on the display. A single color diagonal line corresponding the rotation angle θ_l is draw on the screen. The lenticular sheet is placed on the display and rotated until the line is fully visible from a single viewpoint. The lens sheet is then shifted horizontally until the maximum brightness is seen at the desired viewpoint. A second view is determined by illuminating a parallel line half the total number of subpixels away. In the 15 view prototype, this is 7 or 8 subpixels. Because each display panel has two lenticular sheets, the alignment process is performed for both.

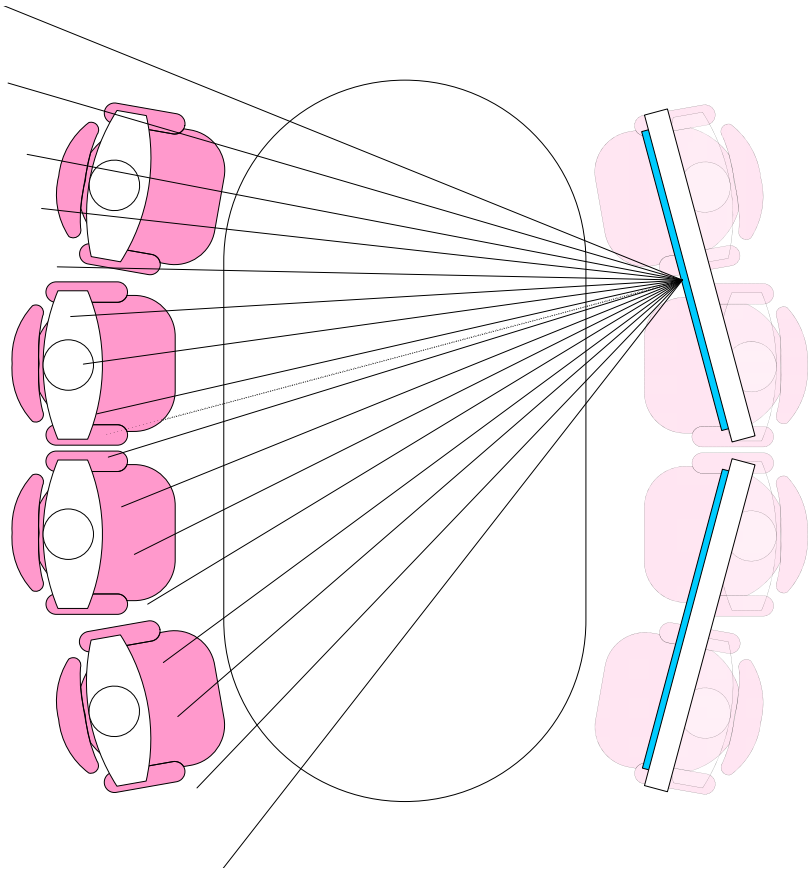


Figure 5.12: 15 views per display.

Image capture

One the lenticular sheet is aligned on the display, the capture stage of the calibration process collects images of the display from the position of each viewer. The images are saved and post-processed to determine the the set of subpixels visible from each location. A camera is placed at the desired viewpoint, aimed at the center of the display. Multiple viewpoints can be calibrated simultaneously if enough cameras are available. We use a high resolution (1600×1200) grayscale Scorpion camera from Point Grey Research and we save each image to disk in a lossless image format. Several techniques exist for the composition of these images into a mask of all pixels visible from each viewpoint.

Methods for pixel visibility determination

The most straightforward method for determining the visibility of each subpixel is to illuminate every subpixel, one at a time, for the entire display. If the subpixel is seen in the camera image for a given viewpoint, then that subpixel is considered visible for that viewpoint. For a $w \times h$ resolution display and v different viewpoints, this would require $3 \times w \times h \times v$ different images. For a 1080p HDTV display, this is approximately 6 million images per viewpoint, which is impractical.

A faster method for determining pixel visibility is the use of Gray codes, an error

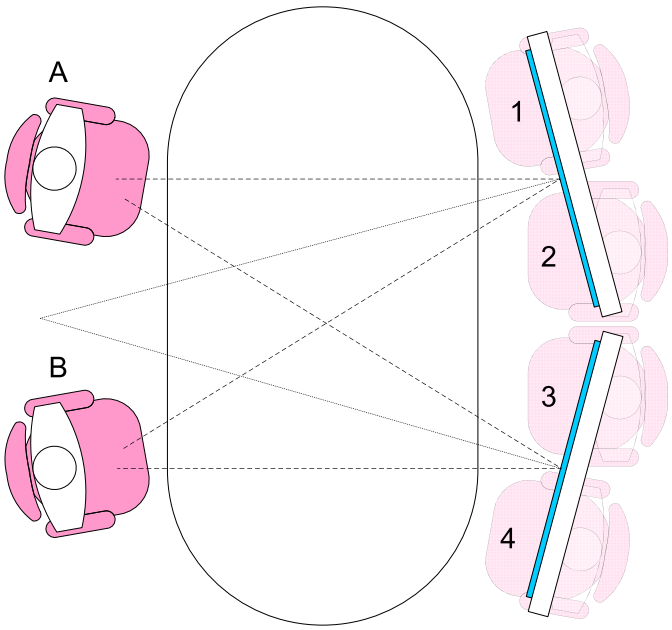


Figure 5.13: The prototype two-on-four teleconferencing scenario. Participants A and B are local. The four remote participants are shown on two large multi-view displays that provide unique views to A and B.

resistant binary encoding for mapping the display to the camera [54]. The displayed patterns are a series of black-and-white stripes that divide the display in successively finer slices. A sequence of such images can be used to uniquely identify the visibility of each pixel. Unlike conventional binary coding imagery, the stripe boundaries in a Gray code sequence do not occur at the same pixel locations, which prevents significant errors in the mapping. To generate a complete mapping of pixels to the viewpoint, a Gray code sequence requires $\log_2 w + \log_2 h$ images for each color channel. For an 1080p HDTV display, only 66 images are required for a complete view calibration.

In theory, such a Grey code mapping is optimal, but in practice the combination of camera sensitivity and the light spread of a multi-view display make it difficult to accurately capture the Gray code images. Large illuminated regions may bleed into black areas, which will make edge detection difficult and degrade the output mask quality.

Line sweeping

We have developed a hybrid technique that preserves the precision of the single pixel method but uses a significantly reduced number of images. Horizontal and vertical lines with single pixel thickness are swept across the display. One image is captured for a vertical line at every X coordinate position for the entire resolution of the display. Similarly, one image is captured for every Y coordinate. For a display with $w \times h$ resolution, a total of $w + h$ images are captured. A 1080p HDTV display will require

(a) (b)

Figure 5.14: Multi-view display calibration: Example camera images for horizontal (a) and vertical (b) line sweep for the lenticular multi-view display.

$1920 + 1080$ images per color channel, for a total of 9000 images per viewpoint.

For displays with physically offset color subpixels, such as an LCD flat panel display, the line sweep is performed separately for the red, green, and blue color channels. For displays where the color components of each pixels are spatially aligned, such as a DLP or LCD projector, only a single line sweep is necessary, and any color channel or a combined white color sweep may be used. Figure 5.14 shows frames from the horizontal (a) and vertical (b) line sweeps across the lenticular display.

If all of the views are captured simultaneously, then the total time required to generate the complete mask is the number of images divided by the camera capture frame rate. The Point Gray Scorpion camera running at 7.5 frames per second, would optimally require $9000/7.5 = 1200$ seconds per display. In practice, the actual speed of the capture process is a few hours for a 1080p HDTV display.

Processing

The camera images are post-processed using MATLAB to generate the final view mask. The algorithm processes the vertical and horizontal images to generate X and Y masters, which are camera space representations for which pixels are visible. The final stage of the algorithm maps these masters from the camera to the display space to create the mask for each viewpoint. The pseudocode for the calibration processing is presented in Algorithm 5.2.1. For each color channel, R, G, and B, the calibration process is as follows:

1. Set camera resolution array, M_x, to zero.
2. For all vertical line images:
   (a) For each pixel (x, y):
      i. If the value is above threshold, set $M_x(x, y)$ to x

3. Set camera resolution array, M_Y, to zero.

4. For all horizontal line images:

   (a) For each pixel (x, y):

      i. If the value is above threshold, set $M_y(x, y)$ to y

5. Set display resolution array, M_d, to zero.

6. For each element, (x, y), in arrays M_x and M_y

   (a) If $M_x(x, y)$ and $M_y(x, y)$ are both non-zero, set $M_d(x, y)$ to 1.

Algorithm 5.2.1: MATLAB psuedocode for line sweep calibration

```
% initialize X and Y masters
X_master = zeros(y_camera_res, x_camera_res);
X_values = uint8(zeros(y_camera_res, x_camera_res));

Y_master = zeros(y_camera_res, x_camera_res);
Y_values = uint8(zeros(y_camera_res, x_camera_res));

% process vertical lines
for pixelValue = x_screen_start:x_screen_end
    I = imread(fname); % read image

    BW = im2bw(I,threshold); % thresholded image/mask
    BW(1,1:64) = 0; % mask out data bits in image

    BWi = ~BW; % inverse mask
    Z = pV * BW; % value mask representation

    X_master = X_master .* BWi; % clear new pixel values
    X_master = X_master + Z;
end

% process horizontal lines
for pixelValue = y_screen_start:y_screen_end
    I = imread(fname); % read image

    BW = im2bw(I,threshold); % thresholded image/mask
    BW(1,1:64) = 0; % mask out data bits in image

    BWi = ~BW; % inverse mask
    Z = pV * BW; % value mask representation
```

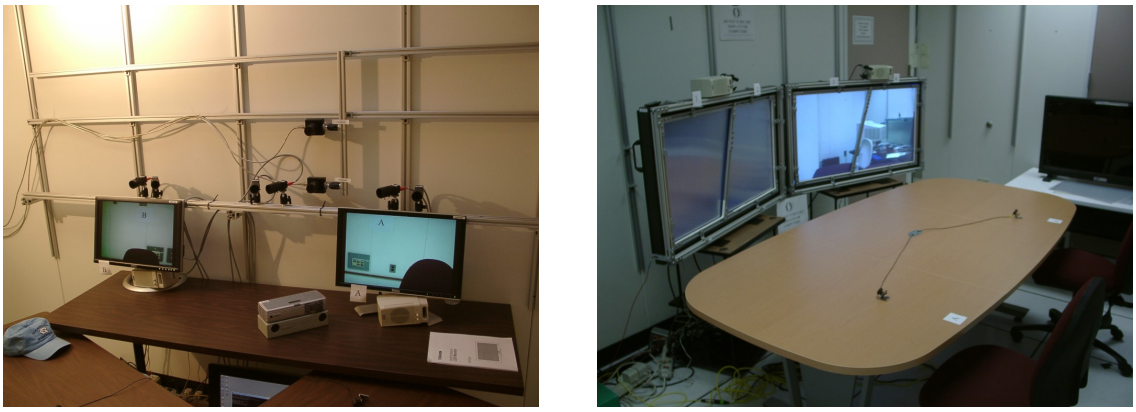
```
    Y_master = Y_master .* BWi; % clear new pixel values
    Y_master = Y_master + Z;
end

% scan camera image to create monitor image of complete pattern
screen_master = zeros(y_screen_res, x_screen_res);

% scan through the x and y masters
for x = 1:x_camera_res
    for y = 1:y_camera_res
        ys = Y_master(y, x);
        xs = X_master(y, x);

        % if there is a pixel in both masters
        if ((ys ~= 0) && (xs ~= 0))
            screen_master(ys, xs) = 1;
        end
    end
end
```

Mask Combination

Once the set of visible subpixels for each viewpoint has been determined, we generate a set of corresponding masks. These masks are binary images that are used in the rendering stage. If a pixel is visible from more than one viewpoint, there are two methods for determining output pixel color. In black blending mode, if two viewers observe the same subpixel, then the subpixel is turned off. Equal weighted blending averages the value for a subpixel seen by multiple views. In theory, equal weighted blending will trade off increased brightness for increased crosstalk between views, but in practice the two calibrated views have almost no overlapping subpixels due to their spacing and the use of guard bands.

5.2.2 Remote camera calibration and capture

In addition to the multi-view display, the system must acquire imagery corresponding the particular viewpoints of each observer. For the two-view display, two cameras are required. The cameras are positioned in the remote environment at locations approximating the position of each viewer. We have implemented a half-duplex system, with multi-viewpoint capture and display only in one direction. Figure 5.15(a) shows the camera pairs corresponding the viewer positions of (b). In order to accurately depict the remote participants at the correct scale, the capture cameras must be calibrated to the display size. We use a checkerboard pattern the size and relative position of the multi-view display to adjust the zoom and alignment of each camera to match the display.



(a) (b)

Figure 5.15: The half-duplex capture and monoscopic multi-view display system. (a) shows the capture configuration, with two camera per viewpoint, corresponding to viewer positions A and B at the display site and an additional central camera pair for comparison purposes. (b) shows the two monoscopic multi-view displays for viewers in positions A and B.

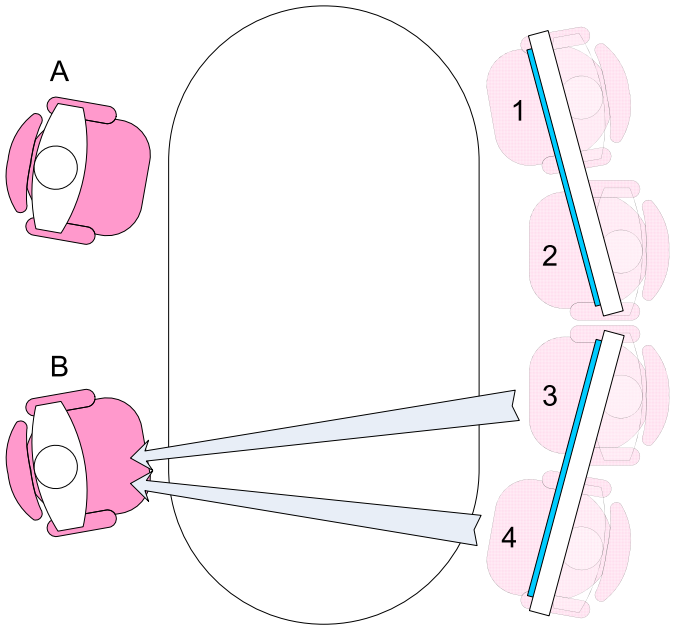
5.3 Results

To support display of two side-by-side users, we use a 47" flat panel 1080p HDTV display which provides a 41" × 23" viewing area. MicroLens Technology, a manufacturer of lenticular sheets, was selected based on their experience developing lens sheets for displays of this scale. Although MicroLens offers large format sheets in several configurations, including different materials and lens densities (given in lenses per inch, LPI), their prototype lenses made with UV casting are limited to a width of 26 inches. As a result, two lenticular sheets were required to cover a single display, with a bar placed across the front to hold the two sheets in place.

Manufacturing restrictions in producing a lens capable of spreading 15 subpixels across a 3m wide area led to a significant amount of overlap between views. In order to achieve the desired view spread, the focal distance of the lens was significantly different than the thickness. As a result, there is significant overlap between the light output from adjacent subpixels at 1.8 m from the display. Accounting for the typical spacing between two individuals seated at a table, this reduced the number of effectively different views to two. Intermediate views were turned off to serve as guard bands between the active viewing areas. The actual view configuration for the prototype is shown in Figure 5.13. Increasing the total light spread and decreasing the amount of overlap between views would require decreasing the radius of each lenticule. However, the manufacturer was unable to construct tooling to reliably cut such radius lenticules. As radius decreases, the angle between lenses requires a sharper cutting tip, which are subject to breakage.

We show the multi-view display with distinct imagery from the two views in Figure 5.16. The display is calibrated from positions corresponding the the users at A and B.

MicroLens Technology, Inc., Indian Trail, NC, USA, `http://www.microlens.com/`



(a)

(b)

(c)

Figure 5.16: The remote participants at 3 and 4 are pointing at the user in position B. The multi-view display presents simultaneous views to different locations: (b) shows the view of the remote users from position A, and (c) shows the view from position B.

(a) (b)

Figure 5.17: The remote participants at 3 and 4 are pointing at the user in position A. (a) shows the view of the remote users from position A, and (b) shows the view from position B.

The camera imagery for each view is masked and rendered. The right display shows the remote users in positions 3 and 4 point to the local viewer at position B. Figure 5.16(b) shows the view from the user at position A, and the users are clearly pointing away. Figure 5.16(c) shows the users pointing directly at the viewing at position B. Similarly, Figure 5.17 shows simultaneous views of the multi-view display where the remote users are point at position A.

The system is able to capture and mask simultaneous multiple video streams per display. Viewers were able to correctly determine the gaze direction of the remote users, particularly with respect to eye contact and whether the remote users were addressing the other viewing position. However, we would like to be able to design and build multi-view displays that effectively support more than two distinct views of the remote scene.

5.3.1 Display analysis

In order to properly characterize the light distribution from the prototype multi-view display, we first measure the light output from the backing Toshiba 47" HD display without any lenticular sheet. We need to determine the amount of energy radiating from a pixel by output angle. Imagery of the display was captured by a camera at a fixed height at many angular positions. A region of each camera image was averaged to determine a relative light intensity for that capture angle. Figure 5.18 shows the measured light intensity by angle in front of the display.

Given the light output from the display, we can then calculate the light distribution when the lenticular sheet is added. The manufacturer provided many of the input parameters for the lenticular sheet: a pitch of 9.68443 lenses per inch, a refractive index of 1.56, and a lens radius of 1.9304 mm. Additionally, we measured the lens thickness to be 2.286 mm. To handle the color banding effect, we rotate the lenticular sheet on the

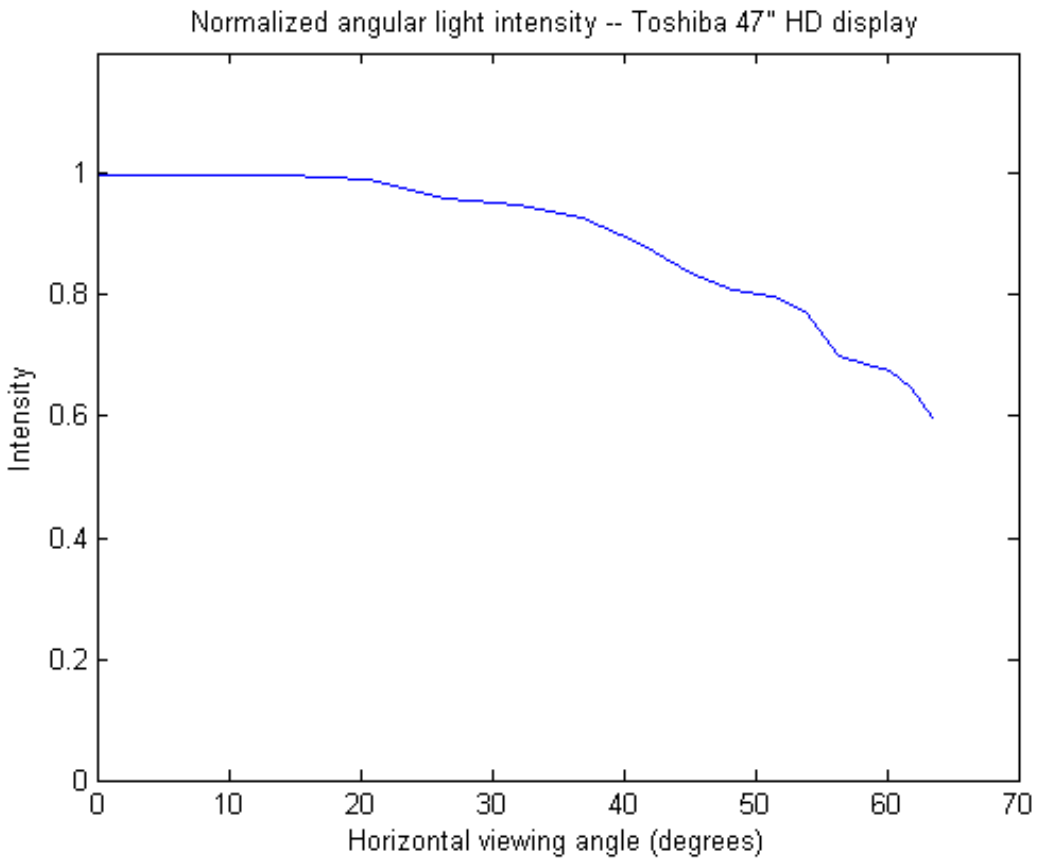


Figure 5.18: The normalized angular light intensity from Toshiba 47" HD display, by horizontal angle from the central axis.

display by 18.43°. This results in an effective horizontal lens width across the display of 2.7668 mm, with each lenticule covering 15 subpixels. At this rotation angle, the effective lens radius increases to 2.0349 mm.

With these parameters, we can then calculate the output angles for the set of slits covering the lenticule. We first present the calculations for a slit in the paraxial region, for a selection of subpixels behind a given lenticule. At a slit height h of 0.05 mm, the angle of incidence I is given by Equation 5.11, with angular variables given in degrees:

$$I = \sin^{-1}(\frac{0.05}{2.0349}) = 1.408$$

From Equation 5.10, the two-term approximation to Snell's law is:

$$1.56(I' - \frac{I'^3}{3!}) = 1(1.408 - \frac{1.408^3}{3!})$$

Solving for I' and θ:

$$I' = 0.9025$$

$$\theta = I - I' = 1.408 - 0.9025 = 0.5055$$

Solving for the longitudinal distance and sag by Equations 5.12 and 5.14:

$$x = \frac{0.05}{\tan(0.5055)} = 5.6675$$

$$z_s = \frac{0.05^2}{2 * 2.0349} = 0.0006$$

The distance to the focal point for the slit at height 0.05 mm is thus $5.6675 + 0.0006 = 5.6681$ mm. The distance and magnification corresponding to the slit are then given by Equations 5.6 and 5.7:

$$s' = \frac{1}{\frac{1.56-1}{2.0349} - \frac{1}{2.286}} = -3.8311$$

$$M = -\frac{-3.8311}{2.286} = 1.6759$$

For a particular subpixel behind the lenticule, we can determine the angles from its image through the slits at height 0.05 mm. With 15 subpixels behind each lenticule, we label the subpixels from -7 to 7, with subpixel 0 centered at the optical axis of the lens. We calculate the extents of the subpixel image based projection of the subpixel. Then, as shown in Figure 5.10, we project the top and bottom extents of the subpixel image through the slit holes to determine the output angles. Table 5.1 presents the angular output for several different slit heights and subpixels.

The total energy from a given subpixel is found by integrating the energy from each slit pair from the optical axis to the outside edge of the lenticule. We approximate this integral by iterating over a finite set of equally spaced slits. After calculating the minimum and maximum output angles for each slit as above, we accumulate energy by angle in proportion to the output angle width. We then account for the angular light distribution of the backing display by multiplying the energy per slit by the normalized

Slit height	Focal distance	Subpixel	Angular projections
0.05 mm	5.67 mm	-7	(-27.77°, -31.20°), (-26.59°, -30.09°)
		0	(1.52°, -3.01°), (3.01°, -1.52°)
		7	(30.09°, 26.59°), (31.20°, 27.77°)
0.5 mm	5.60 mm	-7	(-32.74°, -35.83°), (-21.01°, -24.85°)
		0	(-5.14°, -9.59°), (9.59°, 5.14°)
		7	(24.85°, 21.01°), (35.83°, 32.74°)
1.0 mm	5.35 mm	-7	(-37.38°, -40.13°), (-14.73°, -18.88°)
		0	(-11.92°, -16.18°), (16.18°, 11.92°)
		7	(18.88°, 14.73°), (40.12°, 37.38°)

Table 5.1: Lenticular output angles for selected slit heights and subpixel positions.

intensity shown in Figure 5.18.

We plot the total energy by angle for the central subpixel in Figure 5.19. The energy is spread over 40° from the center of the lenticule. The strong peaks on the sides of the energy curve are caused by the strong cylindrical aberration near the edge of the lenticules, with increased refraction bending more rays to a common region. Figure 5.20 shows similar plots for the outer subpixels, -7 and 7. There are similar peaks near the edges of the light distribution, but they are unequal because the pixel offset from the center results in greater refraction through the near side of the lenticule. These outer pixels exhibit the highest peak energy per angle of any subpixel. As expected, these two pixels produce symmetric energies across the optical axis.

We can calculate the energy when two subpixels are used to render the same view. This increases the brightness of the display for the viewer and slightly increases the effective viewing zone width. Figure 5.21 shows the light energy from subpixels 5 and 6, and the combined energy when those subpixels are assigned to the the same view.

To determine possible unique viewing positions, we must find subpixels without substantial angular overlap to provide distinct views to several users. If we only consider the subpixels behind a single lenticule, we can model several sets of subpixels that meet this criterion. Figure 5.22(a) shows a possible two view configuration, corresponding to

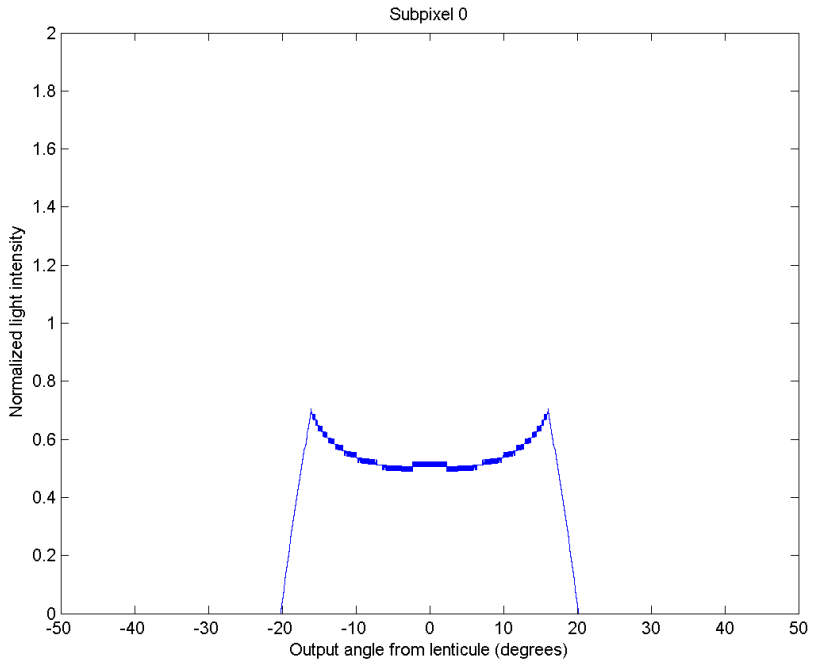
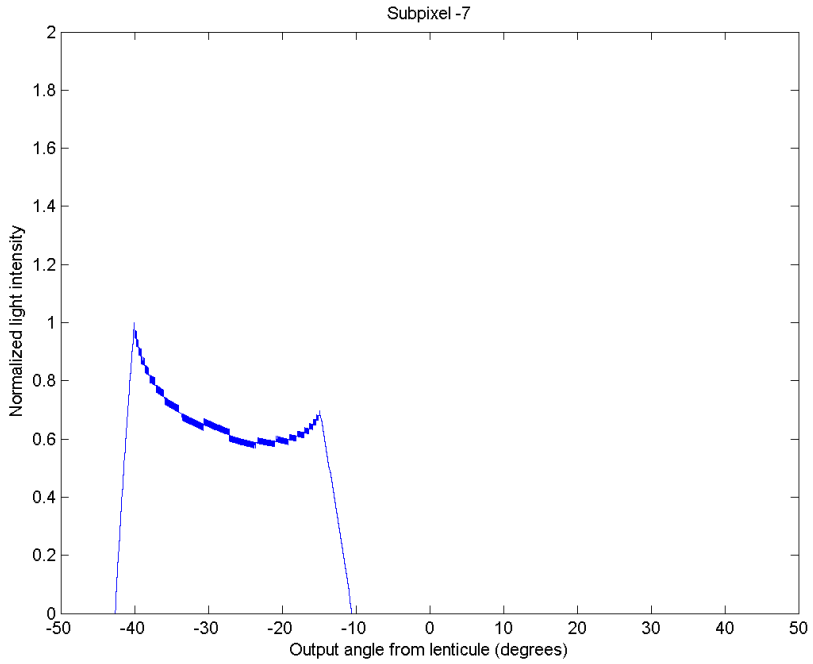


Figure 5.19: Energy from subpixel 0 by angle, normalized by the peak energy of all subpixels.

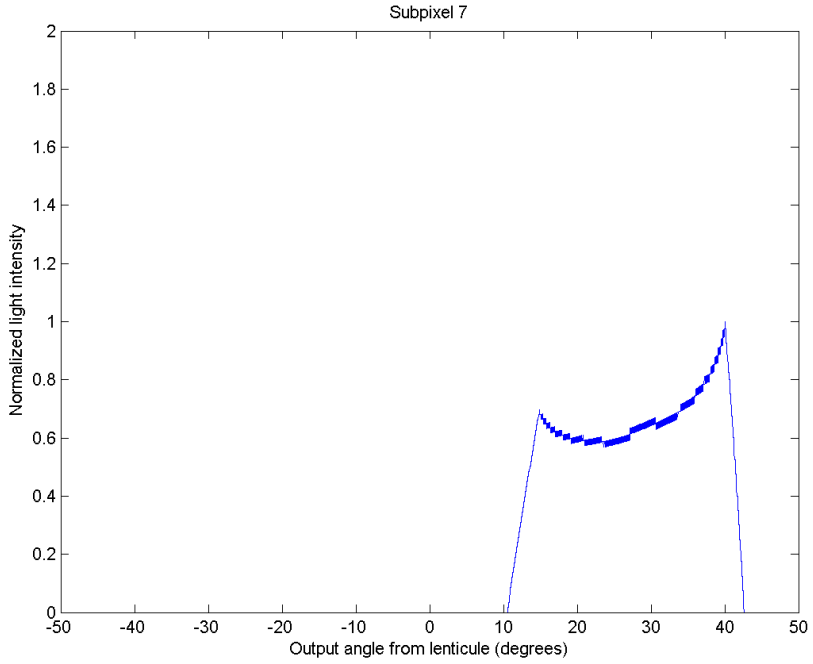
subpixels -5 and 5. This provides a field of view of approximately 30° for views centered at -20° and 20°. Similarly, we can develop a 3 view configuration with subpixels -7, 0, and 7, shown in Figure 5.22(b). Each of these views will have at least 20° field of view, centered at -30°, 0°, and 30°.

However, we must consider the repeating zone effect that occurs when subpixels behind a neighboring lenticule are refracted through the primary subpixel. When a given view is active, all of the corresponding subpixels across the display are on, radiating through their primary lenticule and neighboring lenticules. Figure 5.23(a) shows the distribution of energy when the view corresponding to subpixels labeled -5 is active. As before, the light from subpixel -5 behind the primary lenticule is refracted as expected. In addition, the light from the -5 subpixel in the adjacent lenticule is also refracted through the primary lenticule. Similarly, Figure 5.23(b) shows the distribution for subpixels labeled -7.

With the current lenticular sheet, this repeating zone effect makes it impossible to develop a three view configuration, such as the example shown in Figure 5.22(b). Any three views that are chose will result in almost complete overlap between at least two views, due to the energy from pixels under neighboring lenticules. However, we are able to provide two distinct views with the manufactured sheet, while accounting for the repeating zone effect. Figure 5.24 shows the configuration used in the prototype display system. The energy for a subpixels its corresponding subpixel under a neighboring lenticule are shown for views -4 and 4. When these energies are superimposed, as in Figure 5.24(c), we see that there are distinct 20° fields of view for the two views, even



(a)



(b)

Figure 5.20: Energy from (a) subpixels -7 and (b) 7 by angle, normalized by the peak energy of all subpixels.

including the repeating views.

The calculated angular energy spread ranges from 40.22° for the central subpixel to 31.99° for the outer subpixels -7 and 7. The angular distance between the centers of subpixel -7 or 7 and the corresponding subpixel under a neighboring pixel is 56.5°. Similar anglular distances are found for each viewing zone repeat. So, in order to provide n distinct views within this angular repeat r, each view should subtend approximately r/n degrees. To provide 3 distinct views within the 56.5° zone repeat, the lenticules must be modified to decrease the per view spread to one-third of this angle.

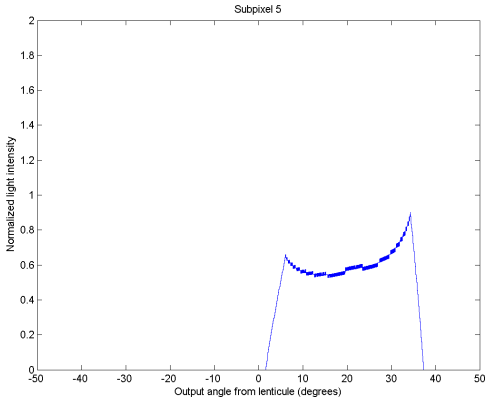
We can decrease the angular spread by decreasing the lens radius (to a point). Figure 5.25 shows the relationship between lens radius and the per view angular distribution of energy, using the other parameters from the existing lenticular sheet. The minimum angle is provided at a radius of approximately 1.3 mm. This minimum corresponds to a focal distance close to the actual distance between the lens and subpixels; that is to say, the subpixels are in focus. At smaller radii, the focal length is closer than the subpixels and so the angular spread increases. At larger radii, the focal length is further than the subpixels and the angular spread increases.

5.4 Discussion

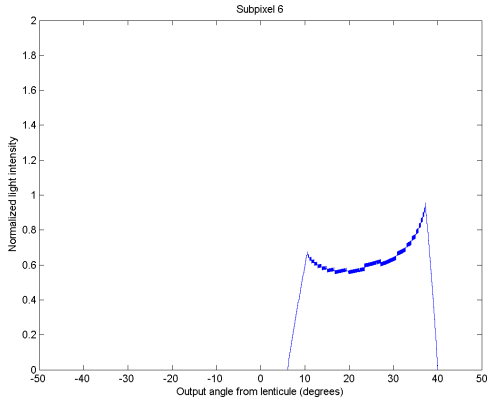
The current prototype system demonstrates an improvement in teleconferencing systems by using multiple cameras and a multi-view display. Each local participant can view a distinct and spatially appropriate view of the remote participants, allowing for correct determination of gaze and attention. However, we are limited to two distinct views because of the characteristics of the lenticular sheet in front of the display.

The limitations of the display are inherent to the lens radius and the distance betwen the sheet and subpixels. To achieve the desired spread of the viewing zones over 3 m, the thickness of the lens is less than the focal length determined by the radius of the lenticule. This results in a significant amount of overlap between adjacent views. As discussed earlier, there were manufacturing limitations in the lenticular sheet construction that limited the lenticule radius to a minimum of just over 2.0 mm. Based on the model presented in Section 5.3.1, we can create three distinct views by decreasing the lens radius to approximately 1.5 mm. This will require improved manufacturing of lenticules, or an alternative technique such as parallax barriers. Barriers could achieve better view separation at the cost of reduced brightness. This would also allow the use of fewer subpixels to achieve a similar number of distinct views, which can increase the display resolution.

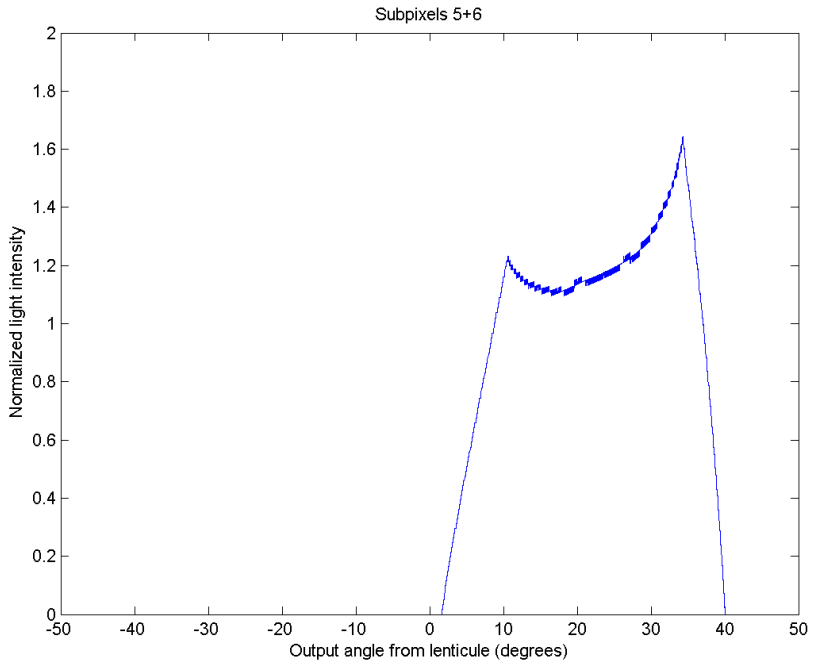
The multi-view display calibration method is not restricted to calibrating lenticular lens displays. Because it does not require a display model, it is able to calibrate any kind of multi-view display from a given set of viewing positions. The flexibility of this calibration method allows it to be used with custom displays and to generate new viewing masks for existing multi-view displays from novel viewpoints unsupported by the existing display software. This calibration method may also allow us to build a model of the display mask, whether lenticular or barrier, and to determine the set of pixels that are visible from locations that are not calibrated. Such a model would allow



(a)

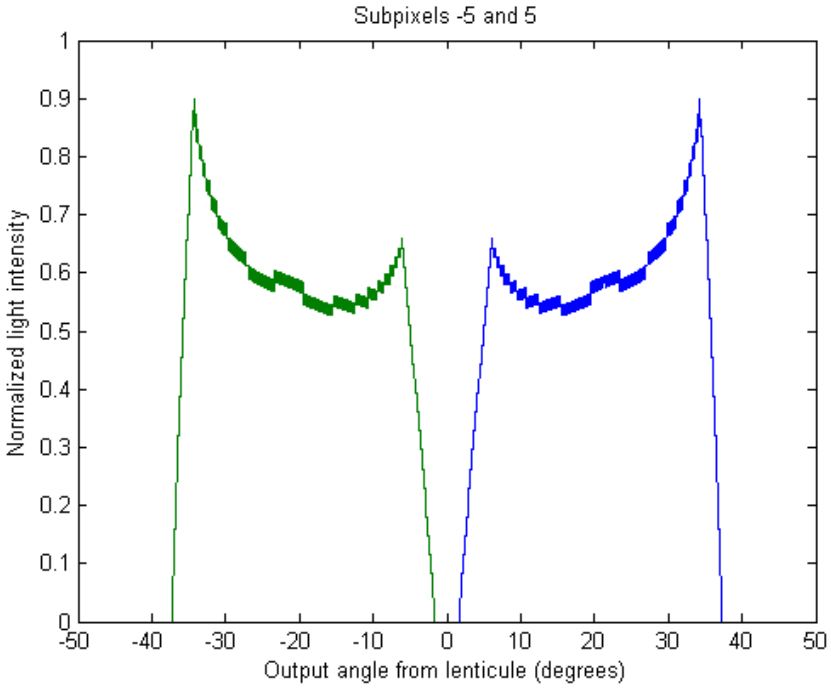


(b)

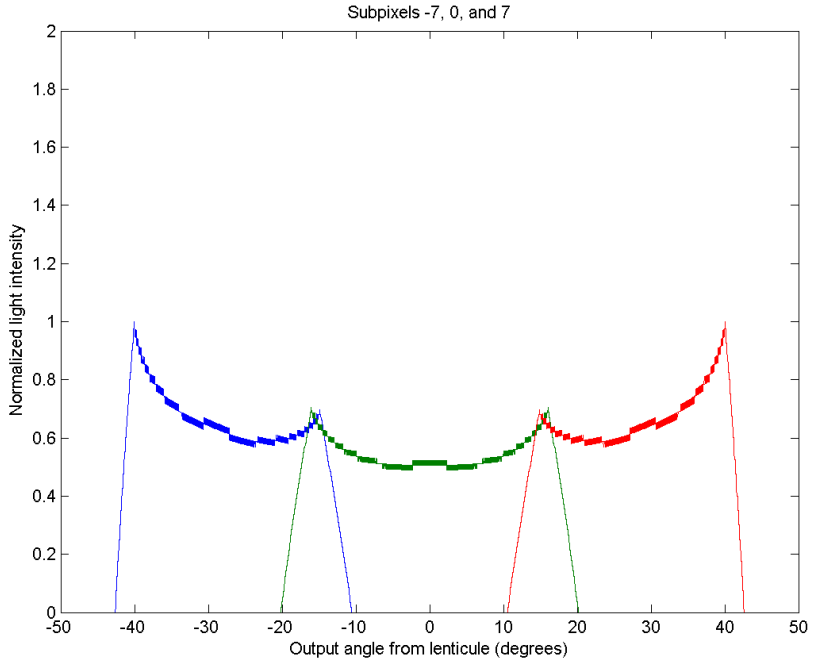


(c)

Figure 5.21: Energy from (a) subpixels 5 and (b) 6 by angle, normalized by the peak energy of all subpixels. (c) Combined light energy for subpixels 5 and 6.

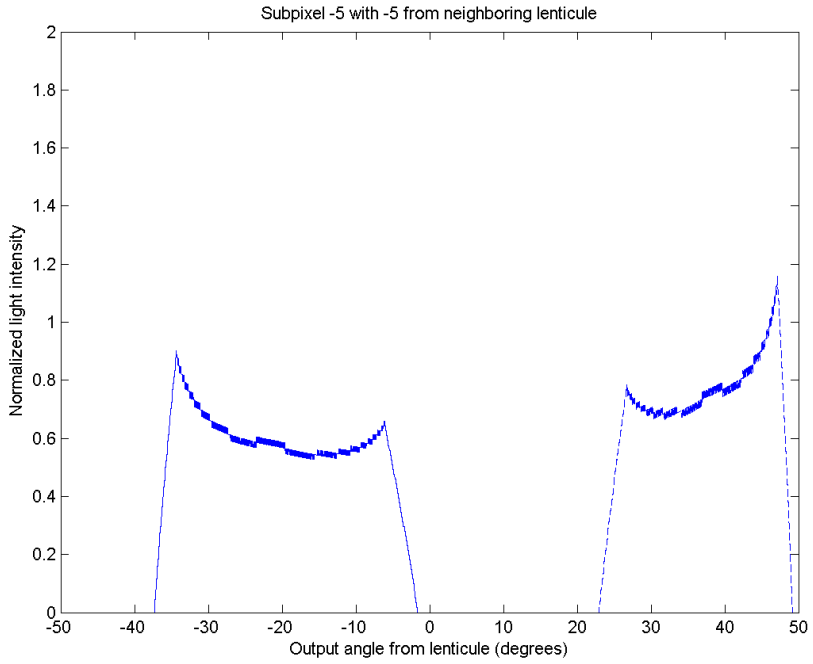


(a)

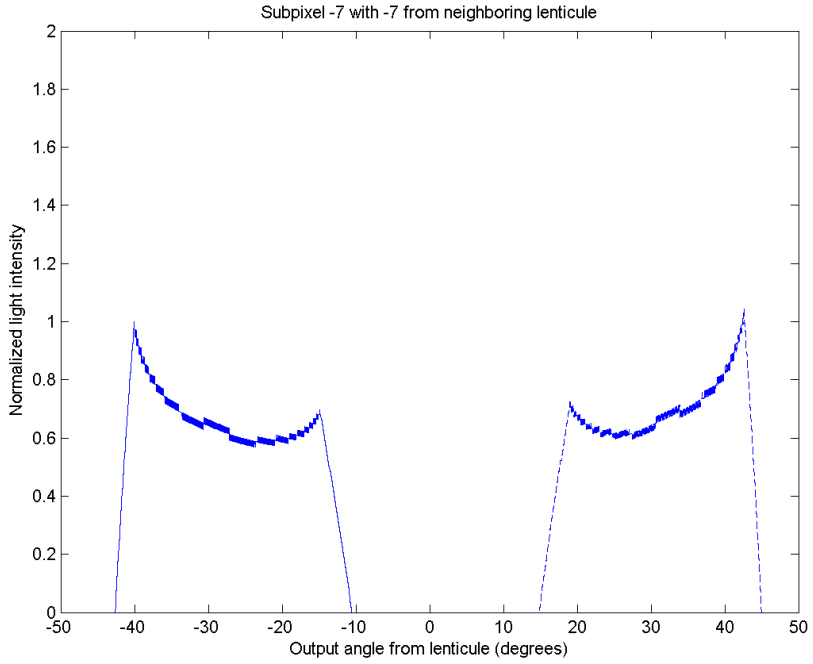


(b)

Figure 5.22: Light energy from (a) subpixels -5 and 5 and (b) -7, 0, and 7 by angle, normalized by the peak energy of all subpixels.

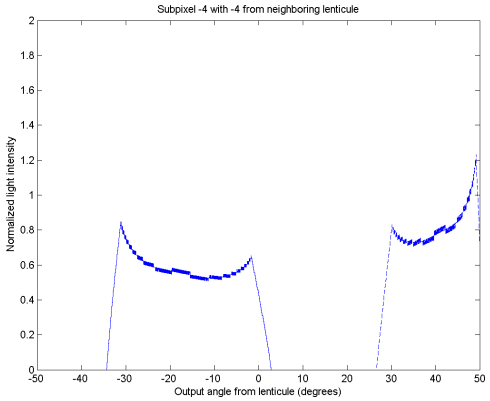


(a)

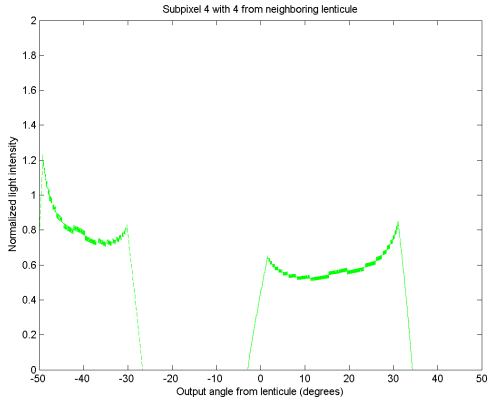


(b)

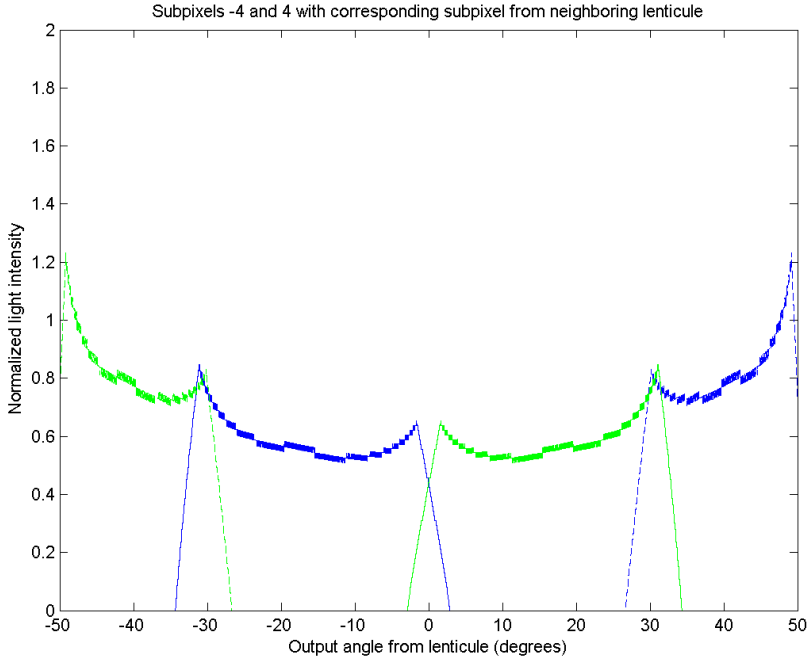
Figure 5.23: Light energy from (a) subpixel -5 and the corresponding -5 subpixel in the neighboring lenticule, and (b) subpixel -7 and the corresponding -7 subpixel, normalized by the peak energy of subpixels under the primary lenticule.



(a)

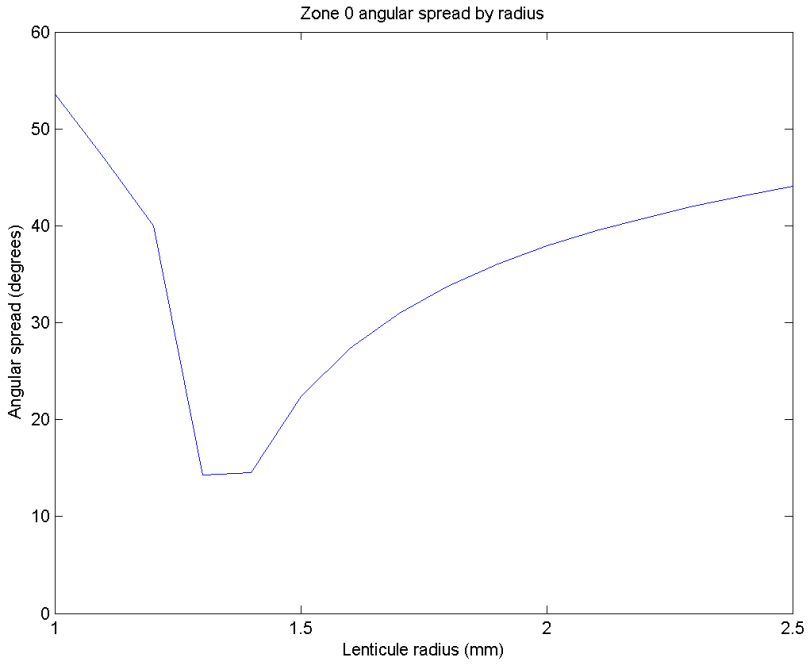


(b)

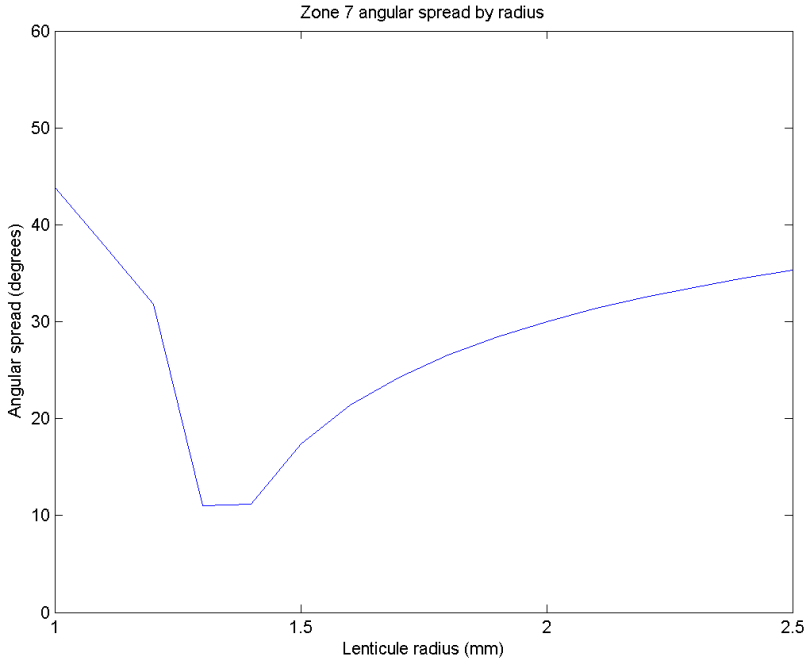


(c)

Figure 5.24: Light energy from (a) subpixel -4 and the corresponding -4 subpixel in the neighboring lenticule, and (b) subpixel 4 and the corresponding 4 subpixel, normalized by the peak energy of subpixels under the primary lenticule. (c) Combined light energy for subpixels 4 and -4 and their corresponding neighbors.



(a)



(b)

Figure 5.25: Angular viewing zone spread by radius for (a) subpixel 0 and (b) subpixel 7. The minimum values corresponds to a focal length close to the distance to the subpixels.

for rendering of novel views for users moving in front of the display.

Chapter 6

Random hole display: A non-uniform barrier autostereoscopic display

In this chapter, I present a novel design for an autostereoscopic display using a randomized hole distribution parallax barrier. The Random Hole Display (RHD) design eliminates the repeating zones found in regular barrier and lenticular autostereoscopic displays, enabling multiple simultaneous viewers in arbitrary locations. The primary task of a multi-user autostereoscopic display is to deliver the correct and unique view to each eye of each observer. If multiple viewers see the same pixels behind the barrier, then a conflict occurs. Regular barrier displays have no conflicts between views for many viewer positions, but have significant, localized conflicts at regular intervals across the viewing area when users are at certain positions.

By randomizing the barrier pattern, the RHD exhibits a small amount of conflict between viewers, distributed across the display, in all situations. Yet it never exhibits the overwhelming conflicts between multiple views that are inherent in conventional autostereoscopic displays. With knowledge of user locations, the RHD presents the proper stereoscopic view to one or more viewers. It further mitigates viewing conflicts by allowing display pixels that are seen by more than one viewer to remain active by optionally blending the similar colors of desired views. Interference between views for random hole barriers and for a conventional regular barrier pattern are simulated. Results from a proof-of-concept Random Hole Display are presented.

The display is a parallax barrier consisting of a fixed planar barrier in front of the native display surface. In a conventional autostereo display, the barrier consists of alternating clear and opaque stripes. The barrier in the new display contains clear holes in a uniformly-distributed pseudo-random pattern. Only a small fraction of the surface area of the barrier consists of holes, so that from any single viewing position, only a small fraction of the native display surface is visible.

The prototype implementation of the RHD uses a barrier pattern with a randomized distribution of barrier holes in front of a flat panel display. The collection of these tiny holes restricts the view from any 3D position in front of the display to a subset of tiny circular regions on the projection surface. The arrangement of the holes in the screen, their size and density, as well as the distance of the screen in front of the display surface is constructed so as to minimize the overlap between visible regions of multiple

eyes. As the number of observers increases, and thus the number of views increases, the fraction of overlap regions will increase, degrading the image quality for all views. This degradation can be countered by decreasing the density of holes. In turn, the resulting decrease in the image brightness can be countered by increasing the brightness of the backlight in the display.

6.1 Approach

Parallax autostereoscopic displays, based on barriers or lenticular sheets, operate by occluding certain parts of an image from a particular viewing direction while making other parts visible. They provide different imagery to the left and right eyes of a viewer, allowing for 3D perception of a scene. This is commonly achieved by dividing the horizontal resolution of a display surface behind the parallax barrier among several views.

To support multiple viewers, some autostereoscopic displays provide many views to allow for several possible viewing positions. This allows a single viewer to experience correct 3D views from various positions. Examples include the MERL 3D TV system which uses projection display with lenticular elements [63], and commercial systems such as Philips 3D displays [81]. Several systems are described in more detail in Section 2.4.2.

Recall the discussion of conventional autostereoscopic display in Section 5.1.1: to preserve horizontal resolution, multi-view autostereoscopic displays have a limited number of distinct views, typically eight to ten. Autostereoscopic display requires sizing individual views to the scale of the interpupillary distance of a user, approximately 6cm. At the optimal distance where this spacing occurs, the maximum width of the display's views is approximately half a meter. This leads to two fundamental problems for groups of users viewing such an autostereoscopic display.

Figure 5.3 depicts a regular barrier autostereoscopic display with 8 viewing zones. The stereo viewer in zones 2 and 3, at the optimal distance from the display, will see all of the pixels corresponding to zones 2 and 3. Due to the regular pattern of the barrier, this view repeats in front of the display at the regular interval of the view repeat distance. Any other viewer must be restricted from entering any of these repeat areas or they will see the same output as the viewer of 2 and 3. This severely limits the lateral movement and potential viewing positions for additional viewers.

The second problem occurs when two viewers are at different distances from the display Viewers at different distances from a regular barrier display see backing pixels at different frequencies. If a viewer is at the calibrated distance, they see the pixels at a particular frequency which corresponds to the correct spacing for the zones. Viewers at a different distance from display will observe the display at a frequency that does not correspond to the regular spacing of viewing zone pixels. This leads to interference between views, where the second viewer at 2 will they other viewer's imagery in certain regions of the display. The superposition of these pixel sets leads to a beating pattern of pixels seen by both users simultaneously, no matter what their lateral position. This restricts multiple users to approximately the same distance from the display.

6.1.1 View interference as aliasing

With regular barrier, multi-user autostereoscopic displays, untracked users must remain in certain viewing areas or they will see incorrect imagery or the same imagery as other viewers. In autostereoscopic display systems with user tracking, multiple viewers are usually not supported because individual display pixels will be seen from multiple views. These visual conflicts are localized and can cover large areas of the display, depending on the viewer positions, because of the regular barrier pattern. This interference between views is a form of aliasing.

Aliasing is a long recognized problem in computer graphics, generating numerous artifacts such as jagged edges and Moiré patterns. Solutions include pre- and post-filtering images and supersampling [65]. Although filtering methods for antialiasing in autostereoscopic displays have been proposed [116], these operate on image quality and depth-of-field rather than between views. Supersampling is not possible because the barrier pattern fixes the sampling rate of the underlying display.

A different solution to the aliasing problem is stochastic sampling, which replaces aliasing with high frequency noise that is less objectionable to the human visual system [15]. There are many classes of stochastic sampling, but an immediately useful form is the Poisson disk distribution, which enforces a minimum distance between randomly placed sample points. This ensures uniform distribution over the larger pattern and trades off perceptually difficult low and mid frequency noise for less troublesome high frequency noise [21].

We apply stochastic sampling to the construction of multi-view displays by constructing a parallax barrier autostereoscopic display that uses a barrier with a Poisson disk pattern of holes. This RHD design offers a number of capabilities that are not found in most existing autostereoscopic displays, including display for multiple users in arbitrary viewing positions. By randomizing hole distribution in the barrier, visual conflicts between views are distributed across the viewing area as high frequency noise, and can be minimized by changing the parameters of the barrier design.

The primary limitations for most parallax barrier autostereoscopic displays are limited brightness and resolution. A typical 8 view regular parallax barrier display will block 7/8 of the light from the backing panel or projection, and have 1/8 of the horizontal resolution. The brightness and resolution of a random hole pattern barrier will be similarly limited by the density of the barrier holes.

By allowing for a small number of pixels to be seen in multiple views, a random hole display can be brighter than a conventional regular barrier with an equivalent number of views. Regular barrier displays cannot allow individual pixels to be seen by multiple users because the regularity of the barrier pattern would mean many pixel conflicts in a localized area. With a random distribution, the conflicts will be randomly distributed across the entire viewing area and the conflicting pixels may be turned off or their color may be blended. When multiple viewers are looking at similar scenes, as in the different perspectives of a single remote scene in group tele-immersion, it is likely that many conflicting pixels will have similar colors, allowing those pixels to stay on. Additionally, different random hole barriers that are optimized for a particular number of viewers could be used to maximize the number of visible display pixels.

6.1.2 Interference analysis

The critical difference between a conventional regular barrier autostereoscopic display and one with random hole patterns is the distribution of view interference caused by the barrier pattern. We define view interference as display pixels that are seen by more than one eye. The total interference is the fraction of backing display pixels that are seen by two or more views.

In general, the amount of interference for n views in a barrier display is the sum of pairwise intersections of all views. The number of interfered pixels, where i and j are viewing positions and I_i and I_j are the sets of pixels visible from each position, is given by:

$$I_{conflict} = \sum_{i,j=1,i\neq j}^{n} count(I_i \cap I_j) \tag{6.1}$$

The average amount of interference between two randomized samples is the product of their sampling frequency. Consider a 3x3 grid of pixels with a single randomly chosen sample. The chance that any particular pixel is selected is 1/9. A second random sample has the same 1/9 chance to select a particular pixel. The chance that these samples end up selecting the same box is the product of the sampling rate, in this case $1/9 \times 1/9 = 1/81$. When a third random sample is introduced, there is a 1/81 chance of intersection with the first sample and a 1/81 chance of intersection with the second, and the overall interference is cumulative, for a 3/81 change of interference. There is a $(1/9)^3$ chance that the same pixel is selected in all three random samples.

When extended to multiple random samples over a larger area, this relation still applies. The amount of interference between any two views is the square of the barrier duty cycle c, the ratio of holes to opaque regions. With each additional view, all existing views must be considered for interference. The number of comparisons is 1 for 2 views, 3 for 3 views, 6 for 4 views, 10 for 5 views, etc. The n-th term of this sequence is given by $(n^2 + n)/2$. In a random barrier display, the amount of interference for n views is given by:

$$I = \frac{n^2 + n}{2} \cdot c^2 \tag{6.2}$$

For example, a barrier with a 1/9 duty cycle and two stereoscopic viewers (for a total of four), the average amount of interference is $(4^2 + 4)/2 \times (1/9)^2 = 10 \times (1/9)^2$, or 12.35% of the total visible pixels. Figure 6.1 shows the interference as calculated by Equation 6.2 for 1 to 10 views and four different duty cycles. As expected, the amount of interference grows more quickly with each additional view.

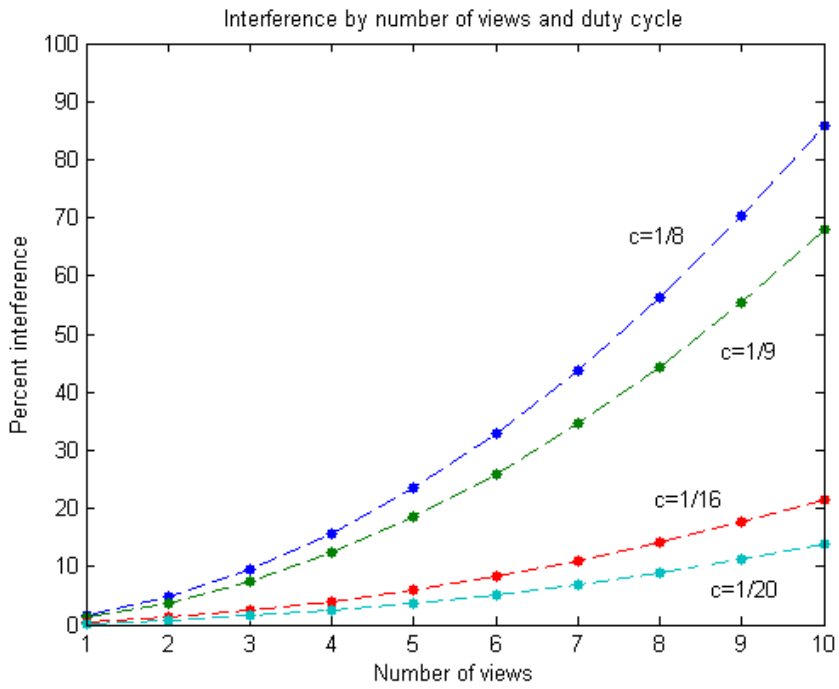


Figure 6.1: Interference by views and duty cycle, calculated by Equation 6.2.

The main problem with a truly random pattern for a barrier is that samples bunch in places and leave large gaps in others. Instead, we use a Poisson disk distribution, in which samples are randomly places, but with a minimum distance constraint ensuring that no two samples are too close. Such a distribution trades off aliasing for noise, like a random sampling, but ensures more even coverage.

Figure 6.2 shows the Fourier transforms of regular barrier, random and Poisson disk distribution patterns. The Fourier transform of the regular barrier pattern shows strong spikes corresponding to the fixed sampling frequency, while the random pattern shows no structure in the Fourier transform. The Fourier transform of the Poisson disk shows a DC spike at the origin and noise beyond the Nyquist limit, resembling the random sampling. We see that the Poisson disk pattern in Figure 6.2(e) is more evenly distributed than the random pattern in (c). This will result in a more uniform light distribution from any given region of the display.

6.1.3 Barrier simulation

We can measure the interference between views for conventional and random hole barrier autostereoscopic displays in simulation. The following simulation results are based on parameters of a desktop-scale autostereoscopic display, including pixel count, display size, barrier hole size, and number of holes. The uniform barrier of a conventional display is compared to a barrier with jittered hole positions (an approximation of Poisson disk distribution) for a display scan line. The display is fixed in virtual space at (1m, 0m), with one stereo viewer centered one meter from the display, at (1m, 1m). Figure 6.3

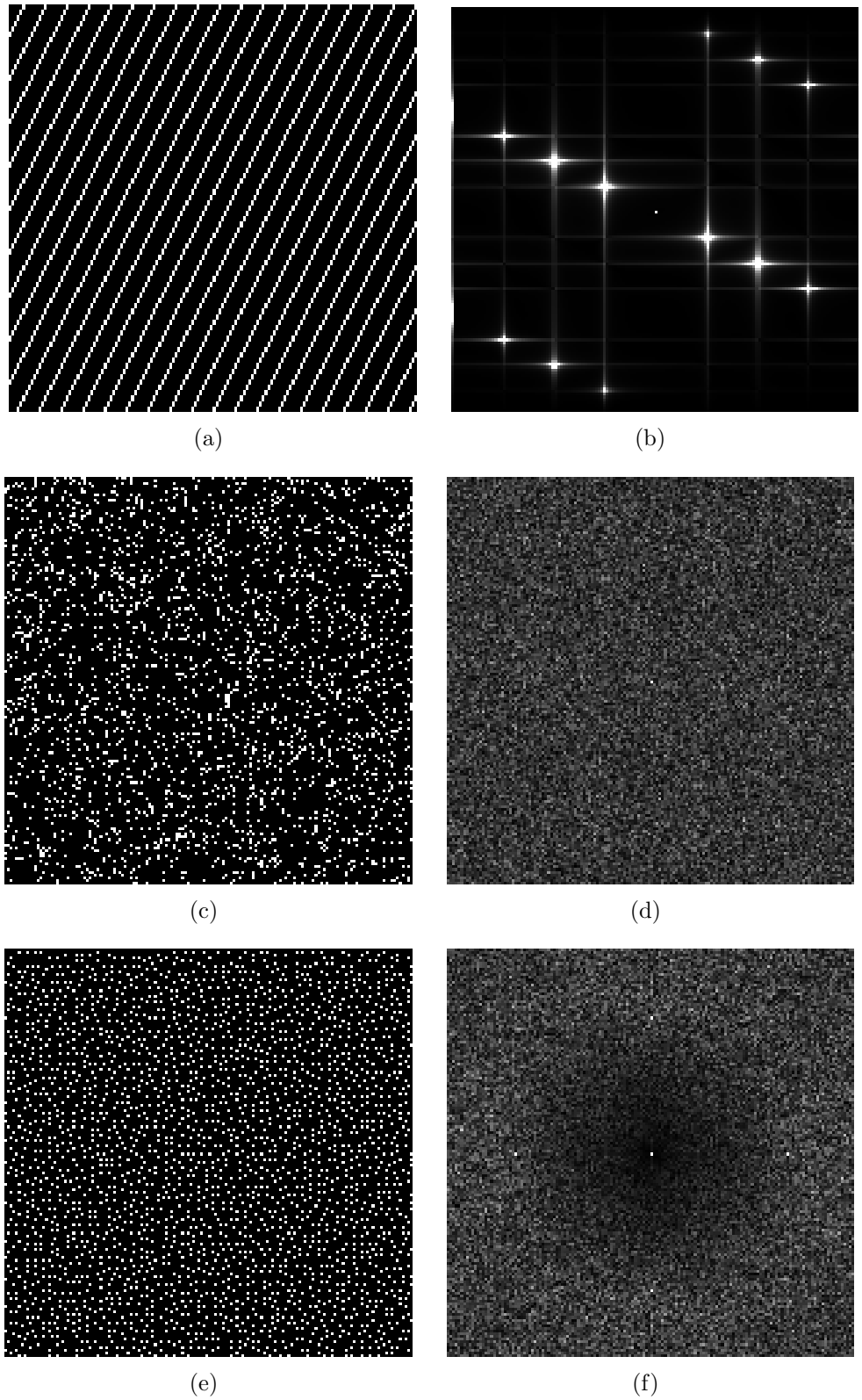
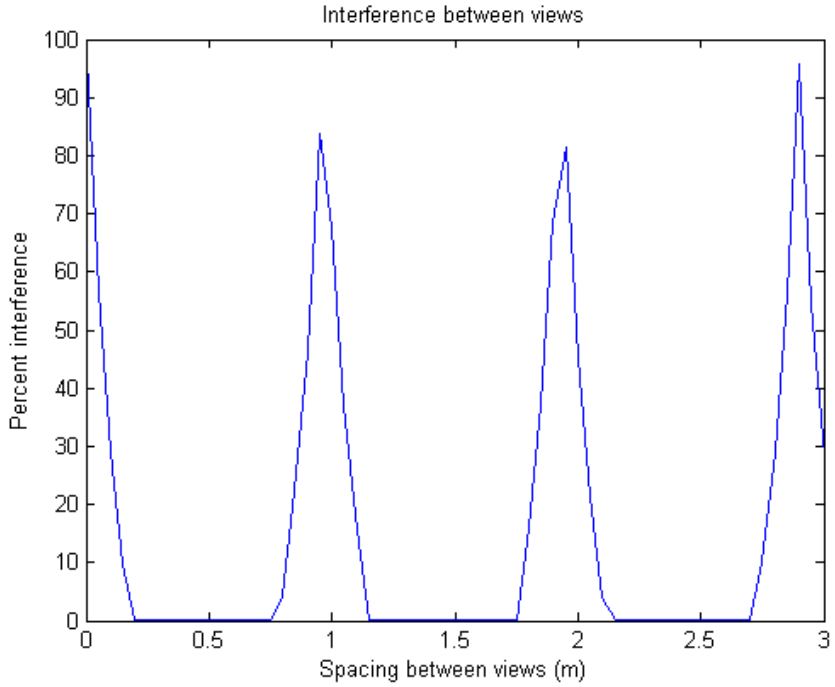
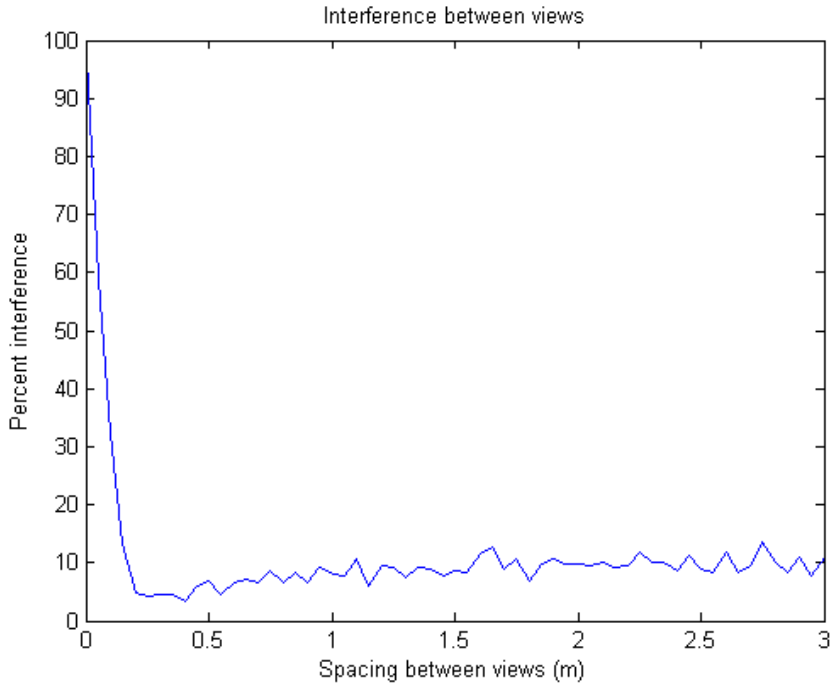


Figure 6.2: Fourier transforms of barrier patterns: (a) Regular barrier pattern; (b) FFT of (a). (c) Random pattern; (d) FFT of (c). (e) Poisson disk distribution pattern; (f) FFT of (e).

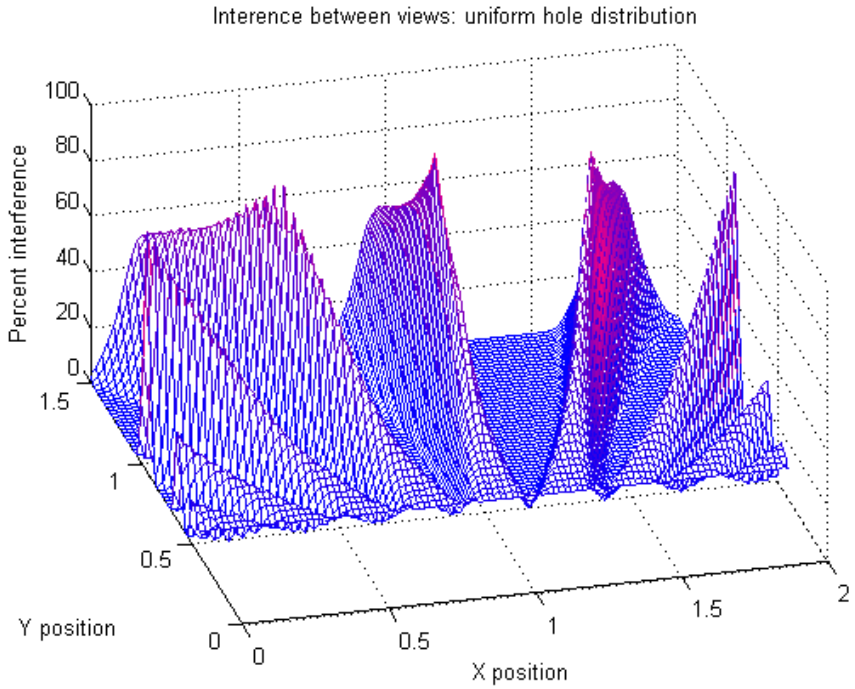


(a)

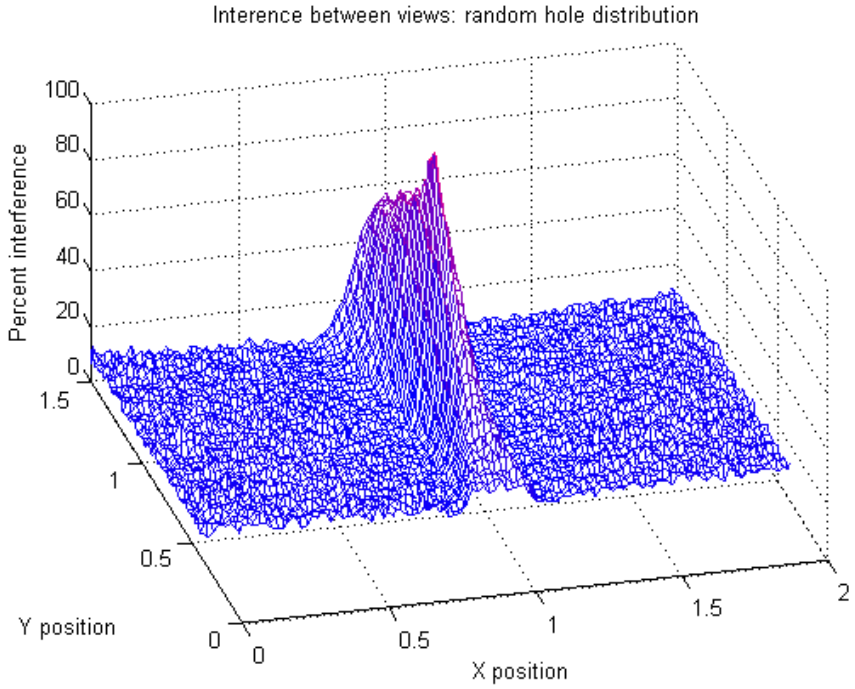


(b)

Figure 6.3: Interference between two viewers, one fixed at (1,1) as the second viewpoint moves horizontally from 0 to 3 m away at the same distance: (a) regular barrier of a conventional autostereoscopic display, and (b) randomized barrier.



(a)



(b)

Figure 6.4: Interference between viewers, one fixed at (1,1) and the other at the plotted (x,y) position: (a) regular barrier of a conventional autostereoscopic display, and (b) randomized barrier.

(a) (b)

Figure 6.5: Film barrier test shows significant ghosting between views.

shows the interference, measured as a percent of visible pixels seen by multiple views, when a second user at the same distance from the display moves horizontally away from the first viewer for (a) the uniform barrier display and (b) the jittered hole barrier.

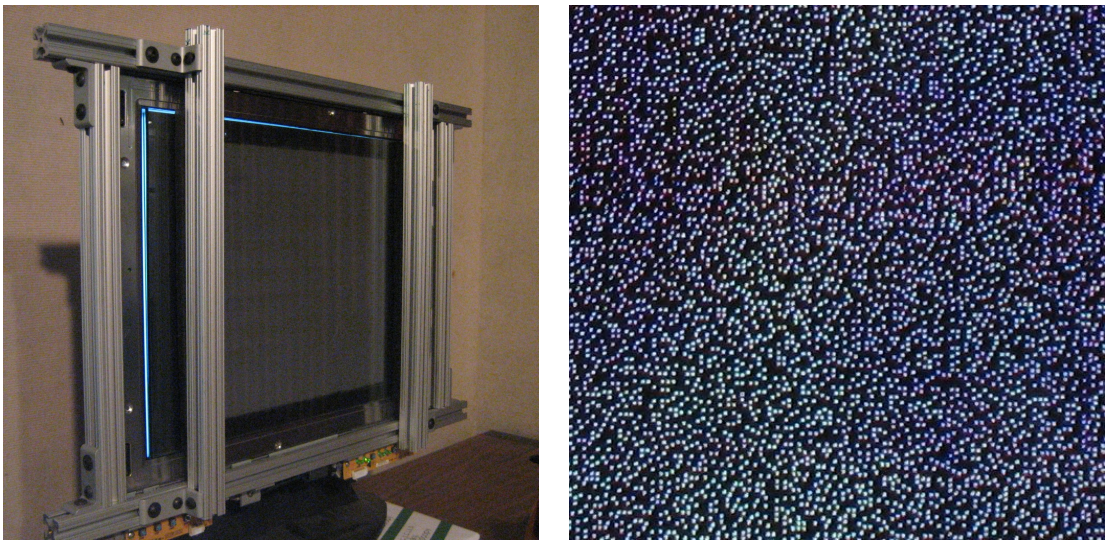
We repeat this simulation over a wide range of postions for the second viewer in the space in front of the display. The interference between views of the fixed user and another user at uniformly distributed positions every 0.02m over a 2m wide by 1.5m deep area in front of the display is computed and shown in Figure 6.4.

Near the display, interference rises equally as the minimum viewing distance is approached. The same pixel is seen by both eyes of a single viewer through neighboring holes and the interference is caused by this near viewer alone. Spines representing areas of high interference are spaced at regular angles from the display. This is the zoning effect fundamental to regular barrier autostereoscopic displays. When two stereo viewers are located in the same zone, interference is very high. In between these spines, view interference is very low, as the second stereo viewer is in a different viewing zone.

The experiment is repeated with a random hole barrier, and all other parameters are kept the same. The expected spine of interference when the two stereo viewers are on the same viewing axis remains. Elsewhere, the random distribution of barrier holes eliminates the other spines of high interference and distributes the interference as noise across the viewing area. There are no areas where the interference is zero, but it is at a low level across the viewing volume.

6.2 Implementation

Our initial random hole display system used a custom film barrier attached to a plastic spacer and placed in front of a flat panel display. The random hole barrier pattern has a Poisson-disk distribution of holes, generated subject to minimum and maximum hole spacing constraints. The barrier field is divided into pixels from which a single subpixel is chosen, based on the specified fractional fill factor (e.g. 1/4, 1/8, 1/9, etc.). This pattern is exposed on a film using a CT scanner film printer.



(a) (b)

Figure 6.6: Views of the RHD prototype: (a) the barrier fixed in front of the LCD panel, and (b) a photograph of a small part of the actual barrier in front of the fully lit display panel.

Problems with this physical barrier include the high reflectivity of the film surface and diffusion of display light in the plexiglass spacer. These two factors lead to noticable reflections between the display and barrier, causing a halo effect around lit areas, as shown in Figure 6.5. We can reduce this problem by lowering display brightness and limiting the brightness of displayed imagery, but it is not possible to eliminate the crosstalk between views. Also, the limited resolution of the CT scanner film printer limits the number of barrier holes per 16 x 10 sheet and the minimum size of each hole is much larger than individual display pixels. These issue led to the consideration of a barrier with physical holes to allow light to pass without significant reflections between the barrier and display.

Our proof of concept Random Hole Display, shown in Figure 6.6, uses a plastic barrier separated from a 100dpi 20" flat panel LCD display by a 1/4" glass spacer. The barrier pattern was laser cut with a Poisson disk distribution of holes, each 1/100" square, with 1/9th hole fill factor and a 2/100" minimum spacing constraint. The pattern covers a 10" x 10" area with 1000 x 1000 backing pixels. To simulate tracking user positions, we calibrate the prototype using the line sweeping method described in Section 5.2.1.

The masks produced by this calibration are passed to the renderer along with the desired imagery for each view. By comparing masks for each view, the visibility of each display pixel is determined. Some pixels are seen by only one view, and so the corresponding imagery is displayed as usual. Other pixels are not seen by any view and remain black. Pixels that are seen by multiple views make up the view interference. A pixel with similar colors in all of the masked imagery remains active, but one with different contributing color values is set to black.

6.3 Results

The proof of concept Random Hole Display is able to present several simultaneous views, each directed to arbitrary locations in the viewing area. Source images are filtered by the mask corresponding to a particular view, and then the source images are blended to form a single output image. Figure 6.7 shows four source images and their masked values. For example, he images of '1' and '2' are masked and then blended together by adding the images and turning any intefering pixels black to create the output image shown in Figure 6.8(a). When the display is viewed from the positions from which the masks were calibrated, the user sees the '1' and '2' in their left and right eyes, respectively, as shown in Figure 6.8(b) and (c).

Figure 6.9 shows photographs from four viewing positions, corresponding to the two stereo views of the users in (e). The expected interference between views is noticeable, but the unique view content is easily distinguished. In typical usage, two stereo views are shown, but the RHD is capable of presenting four monoscopic views to any location as well. Stereo views have been calibrated at various distances from the display, as close as 50cm and as far as 4m. Simultaneous stereo views in many different viewing positions have been tested, with views at the same distance from the display, and varying separations, both laterally and away from the display.

Limited user testing has shown that viewers are able to judge the perceived depth of simple geometric primitives relative to the display surface, both in front and behind. They are also able to fuse stereo imagery of more complex scenes, such as the 3D model in Figure 6.10. These images also exhibit a limitation of the current prototype: dark bands across the display. There are areas of the plastic barrier that were not cut as well as others, leading to smaller apertures in these bands bands. The manufacturing artifacts decrease the visual quality, but multiple simultaneous views from arbitrary positions remain distinct.

6.3.1 Blending

In the RHD design, we expect interference between various views, even for a single stereo viewer. As additional viewers observe the display, the interference will increase as each new view interferes with all existing views. The blending method used to combine masked images from several views can have a noticable effect on the final output.

The first blending method that was tested simply involved turning off any conflicting pixels. This *blank* method eliminates active interference between views, but reduces the overall brightness and resolution. The second method modifed this by blending only interfering pixels of *similar* color, within 12.5% of the maximum value, and turning off the others. We also investigated a *random* selection of which source image color to use for an interfering pixel and using an *average* value of the source image pixels. We compared these four different blending techniques using calibrated masks from two stereo viewing positions, depicted in Figure 6.9(e).

We use the peak signal-to-noise ratio (PSNR) to measure the quality of the blended image[37]. The source signal is the original unblended view, consisting of all pixels visible from a single view. The noise is the error introduced by blending that view with

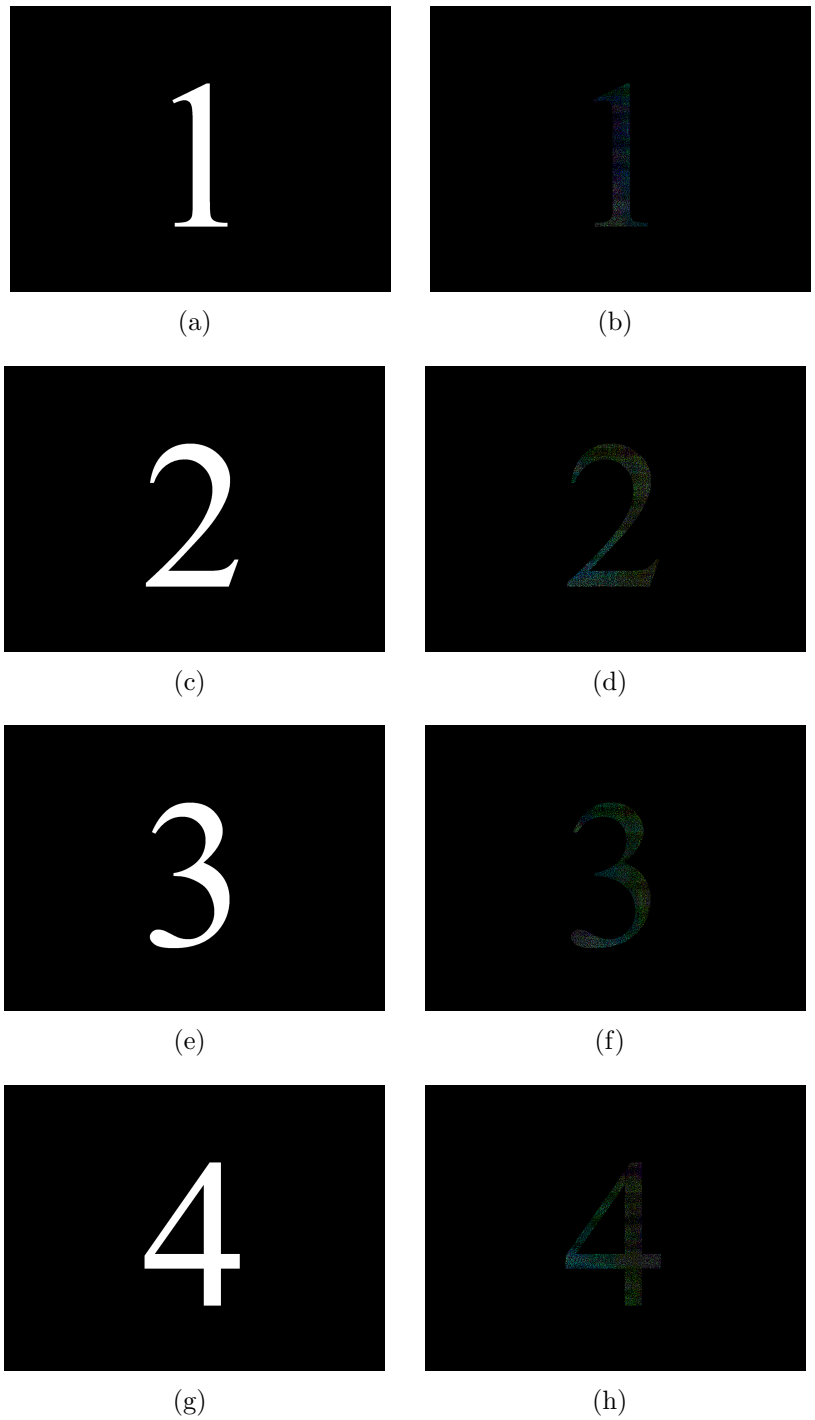
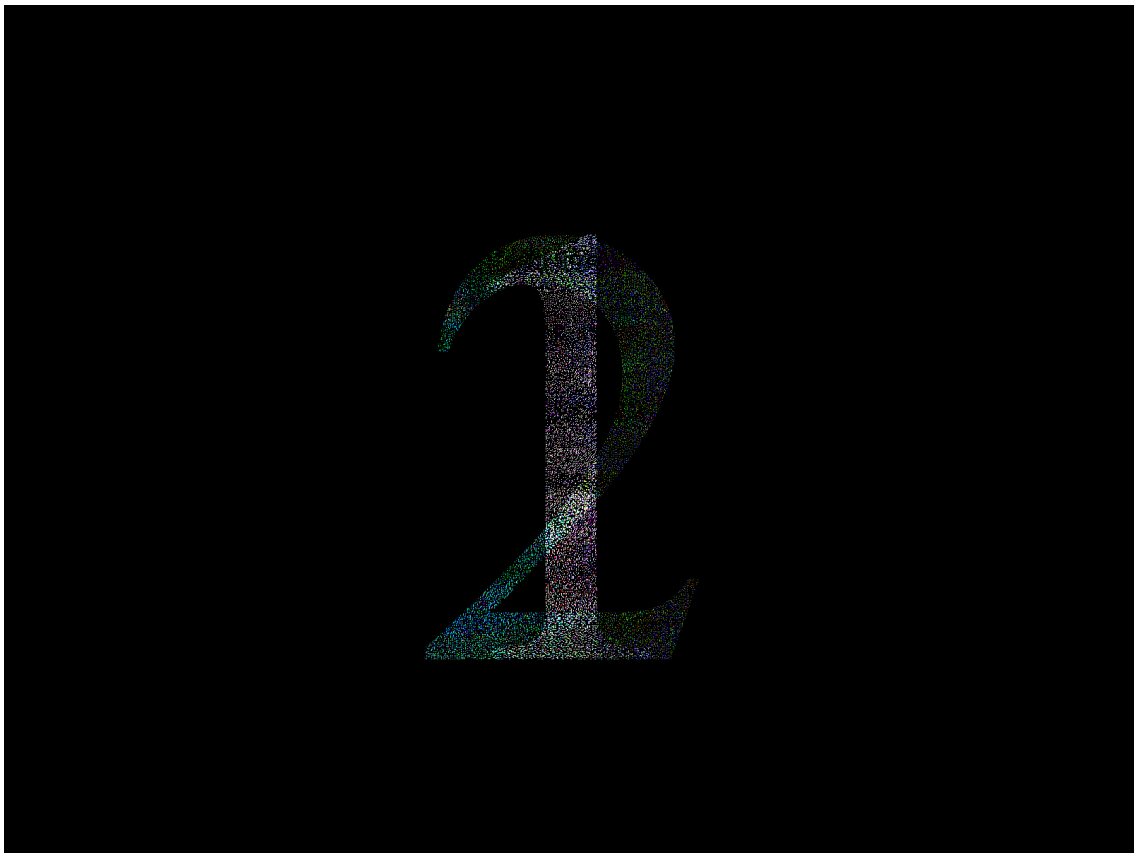


Figure 6.7: The "Numbers" data set. The left column shows the source image and the right shows the masked image for the associated viewing position.



(a)

(b)

(c)

Figure 6.8: Two masked images are blended to form the (a) output image sent to the display. Photographs from the (b) left and (c right eye positions of a viewer centered 1 m from the display.

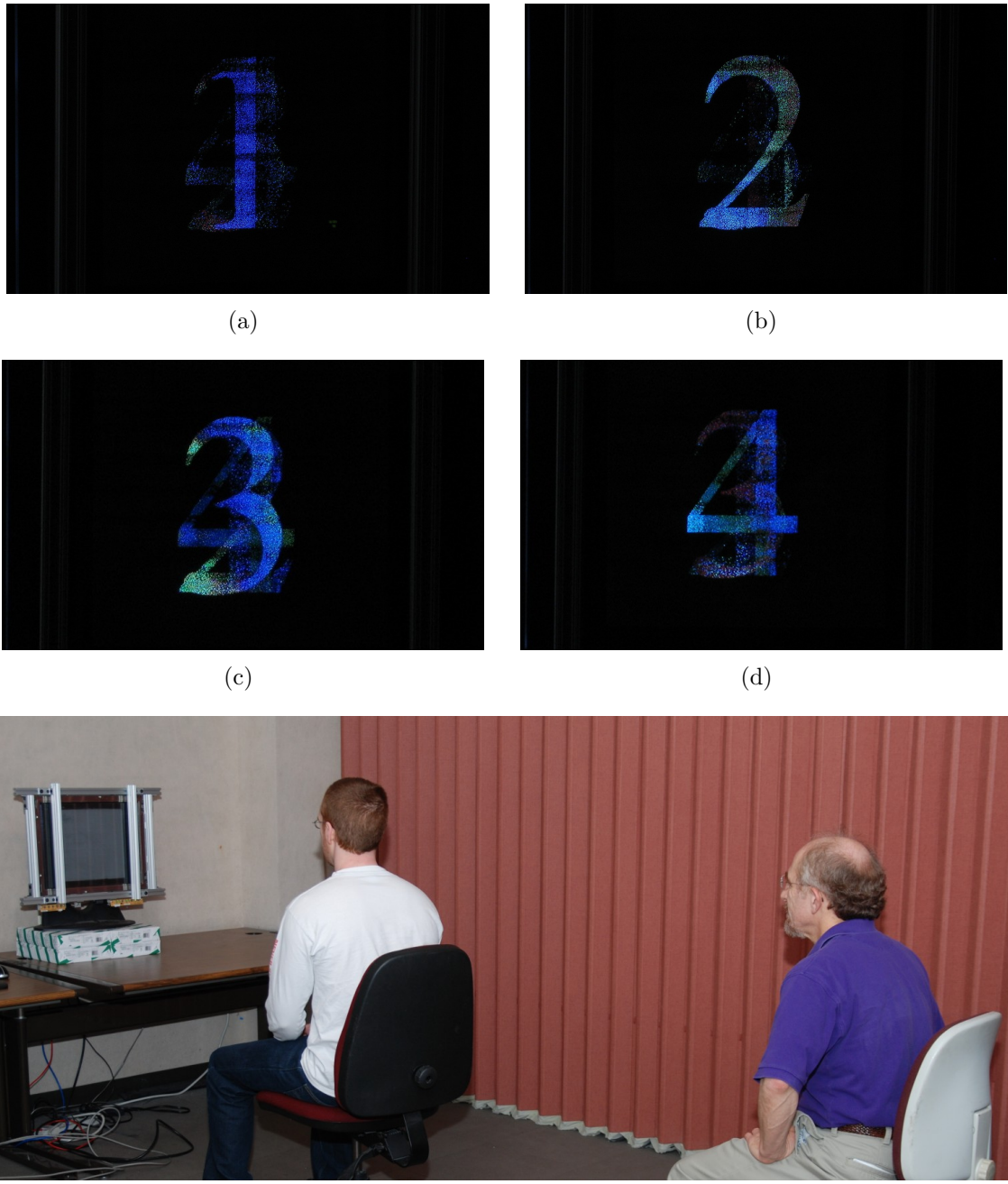
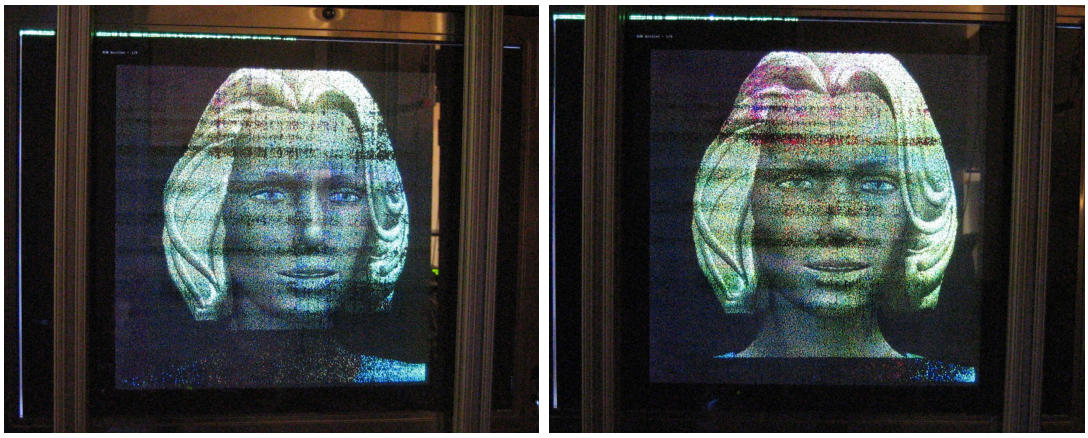


Figure 6.9: Photographs of four simultaneous views of the Random Hole Display, at (a, b) 1.5m and (c, d) 3m from the display, the two stereo viewing positions shown in (e).



(a) (b)

Figure 6.10: Photographs of simultaneous (a) left and (b) right eye views of a color 3D model.

one or more additional views, by one of the various methods describe above. A higher PSNR reflects a blended view that is closer to the original unblended view. PSNR values are given in decibels (dB), with values above 20 dB typically considered acceptable for compression codecs [101].

PSNR is defined using the mean squared error (MSE) for two $m \times n$ color images I and K, which is defined as:

$$MSE = \frac{1}{3mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2 \tag{6.3}$$

And PSNR is defined as:

$$PSNR = 20 \cdot log_{10}(\frac{MAX_I}{\sqrt{MSE}}) \tag{6.4}$$

MAX_I is the maximum possible pixel value of the image. Since all of the images are 8 bits per pixel, MAX_I is 255. Identical images will have zero MSE and an infinite PSNR.

We apply PSNR measurements to several combinations of views and for several different source image sets. The first set is shown in Figure 6.7, with four views of white numbers. The second set, shown in Figure 6.11, is of a rendering of a woman and the third, shown in Figure 6.12 is the same woman rendered in front of a colored background. We present the results in Tables 6.1 and 6.2.

We see that in all cases, the *blank* blending method produces the worst results. When blending two simple views, with the "Numbers" data set, turning off conflicting

pixels has relatively little effect. However, when the source imagery contain larger areas in each view, conflicts increase and are turned off, leading to large areas of black. This is a significant difference from the original content, and the calculated PSNR values indicate that this would not be acceptable for a viewer.

The *similar* blending method produces better PSNR than the *blank* method, primarily for a stereo pair. The similar metric is able to blend two views in the "Background" data set with 16 to 18 dB PSNR because large areas of the backgrounds are similar color. However, when used to blend four views, the quality of the output imagery ranges from approximately 12 to 15 dB, which is below the useful threshold. Even though large parts of the background is similar between views, there is more conflict in the other areas, especially around object edges. These results imply that this is not an acceptable blending method for most cases.

The *random* and *average* blending methods produce significantly better PSNR measurements, in all conditions, than the previous two. In several cases, they exceed 30 dB, with the *average* method peaking at 39.48 dB for view 3 when blending between views 3 and 4 using the "Numbers" data set. The *average* method generally produces the highest PSNR values when four views are blended and *random* produces better values when only two views are blended. Because these two methods are perceptually superior to the methods that force conflicting pixels to black, we recommend that blending between views combine between or choose one of the active values.

6.4 Discussion

The Random Hole Display allows for multiple stereo viewers in arbitrary locations, without the restrictions of conventional autostereoscopic displays on viewing positions. By randomizing the barrier hole pattern, the aliasing interference between views is changed to high frequency noise, which is less visually objectionable than regions of conflict or repeating patterns. This interference is further mitigated by comparing the image pixels and optionally displaying pixels seen by multiple views.

The current prototype system uses view masks from static calibration positions. Future versions of the RHD will track users and generate masks for every viewing position in each frame, using a real-time masking technique similar to the Varrier approach [89]. Higher pixel density displays, such as QuadHD resolution monitors, and camera-based user eye tracking will allow for encumbrance free autostereoscopic viewing with high resolution for multiple viewers. The RHD concept may also be combined with an active barrier, allowing optimal hole density for various numbers of viewers. Random hole patterns may be generated to favor multiple eyes and viewers along the horizontal axis, with higher distribution density along the vertical axis. This will provide brightness and resolution similar to the regular barrier displays.

When a view of any barrier display is very close, the display will appear as a collection of points of light on a black background. For conventional barrier displays, users at close distances cannot be supported due to the barrier pattern, so this is not a concern. However, the RHD design allows for a greater range of viewing distances, so visual acuity may become a more serious problem. Barrier holes could be replaced with a pseudo-

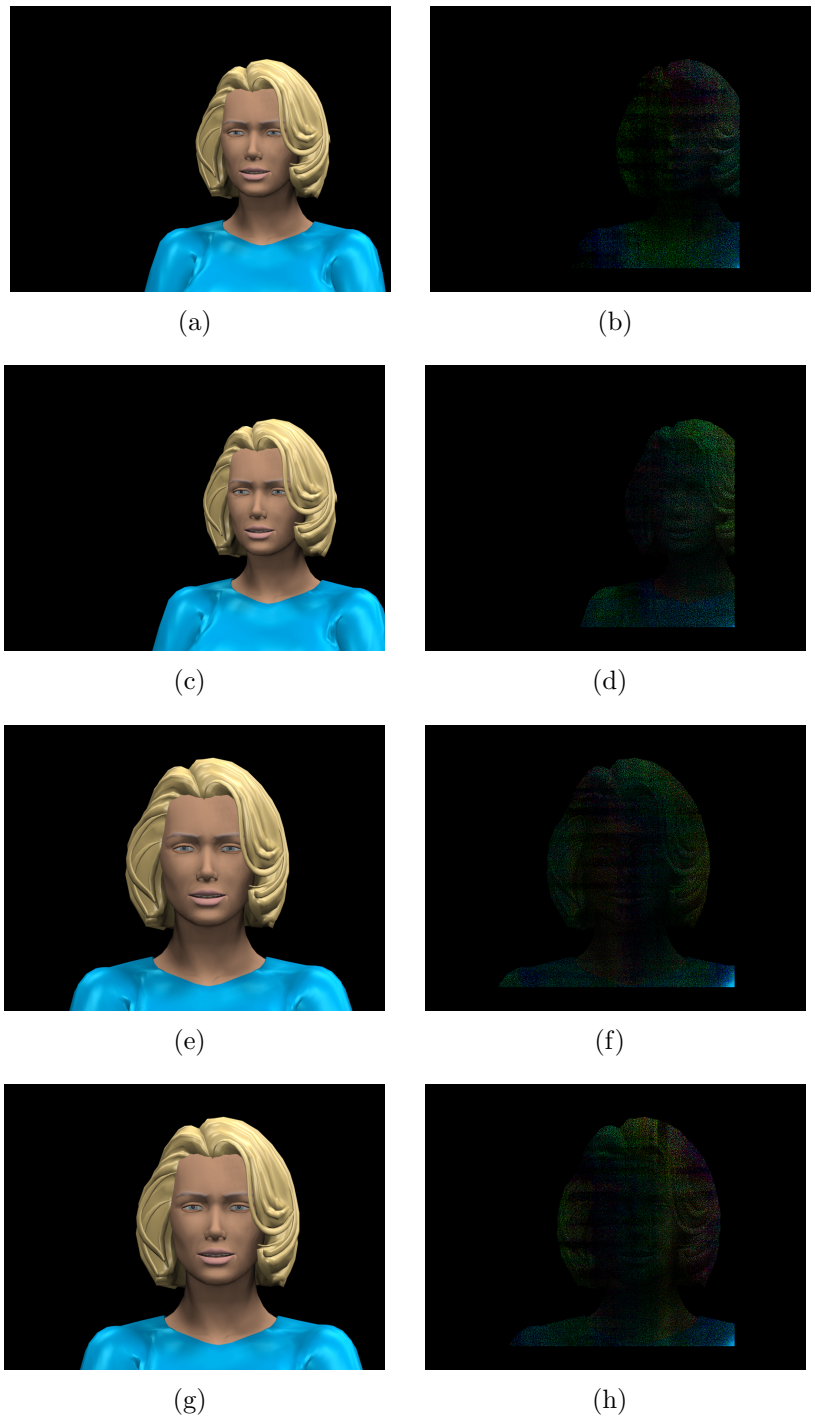


Figure 6.11: The "Woman" data set used for blending analysis. The left column shows the source image and the right shows the masked image for the associated viewing position.

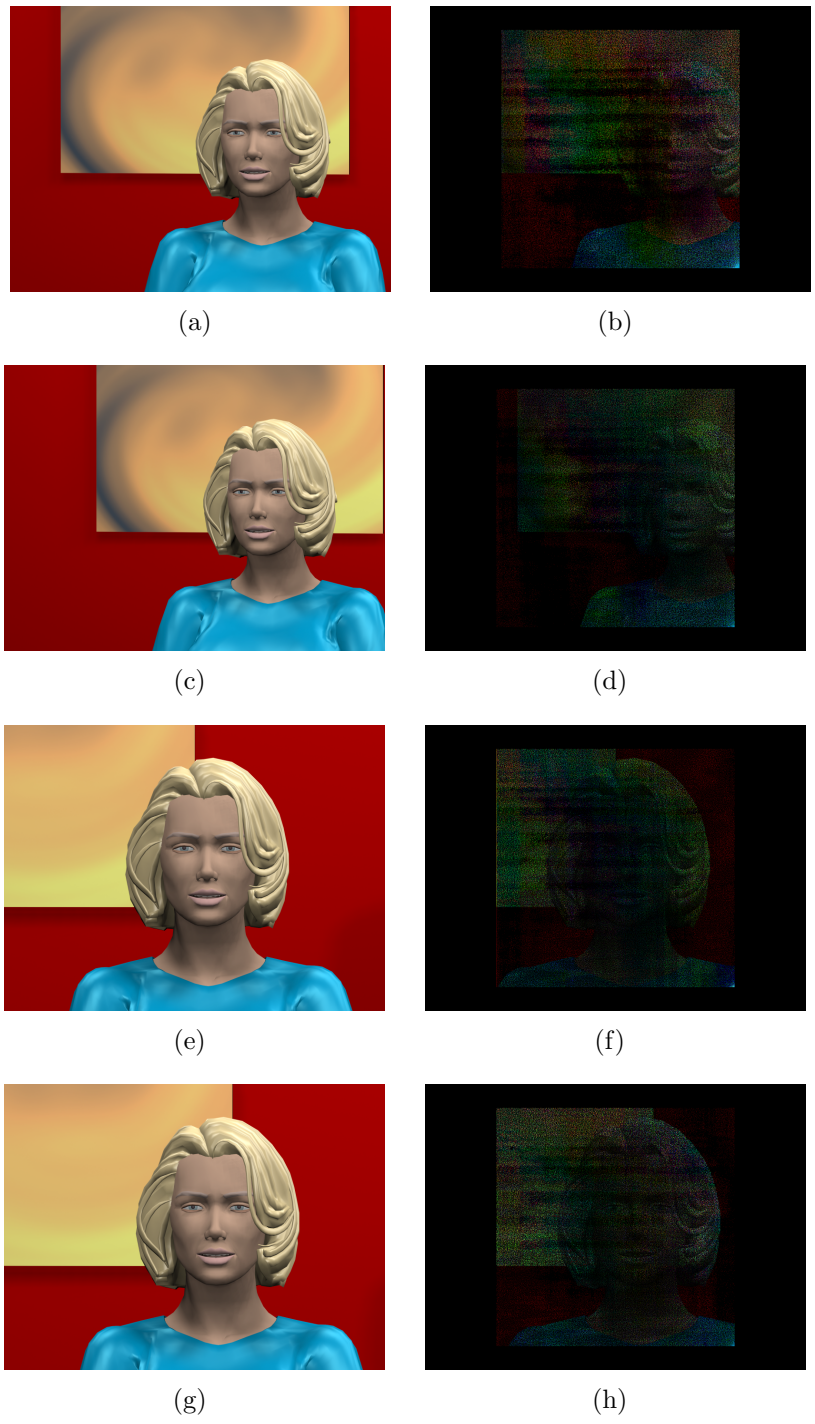


Figure 6.12: The "Background" data set used for blending analysis. The left column shows the source image and the right shows the masked image for the associated viewing position.

View	Views blended	Blending method	Data set PSNR (dB) Numbers	Woman	Background
1	1, 2	*blank*	20.05	12.21	9.56
		similar	21.93	16.34	16.31
		random	25.85	23.69	26.74
		average	26.13	21.78	20.23
2	1, 2	*blank*	18.29	13.32	10.41
		similar	24.33	20.08	18.70
		random	22.49	24.22	25.16
		average	24.36	22.97	20.70
3	3, 4	*blank*	33.42	13.32	10.72
		similar	33.73	17.75	15.21
		random	37.02	25.73	24.58
		average	39.48	22.67	21.41
4	3, 4	*blank*	32.63	12.22	10.07
		similar	38.69	19.08	16.89
		random	35.45	25.43	29.36
		average	38.69	21.77	21.73

Table 6.1: Calculated PSNR for different blending techniques between stereo view combinations.

View	Views blended	Blending method	Data set PSNR (dB) Numbers	Woman	Background
1	1, 2, 3, 4	*blank*	18.46	11.03	8.11
		similar	19.65	14.56	12.97
		random	22.48	19.23	20.47
		average	25.28	22.15	23.58
2	1, 2, 3, 4	*blank*	16.32	11.27	8.22
		similar	16.89	14.74	12.74
		random	19.68	19.31	19.88
		average	22.43	22.32	22.95
3	1, 2, 3, 4	*blank*	29.34	11.29	8.40
		similar	31.01	15.18	12.96
		random	34.31	20.20	19.46
		average	37.38	23.23	22.77
4	1, 2 3, 4	*blank*	30.82	11.10	8.83
		similar	31.37	15.39	13.65
		random	34.20	20.44	20.37
		average	36.96	23.51	23.94

Table 6.2: Calculated PSNR for different blending techniques between 4 views (two stereo pairs).

randomly distributed lenslet array to increase light utilization for a brighter display and to eliminate the black gaps.

6.4.1 Sampling

Image quality may be further improved through a combination of pre-filtering display imagery and applying a reconstruction filter to the input data. In conventional autostereoscopic displays, the sampling rate of source imagery may lead to aliasing both within and between views, as the captured image data may not map directly to display pixels on a one-to-one basis. These aliasing and sampling issues in regular autostereoscopic displays may be mitigated using a resampling algorithm derived from a ray space analysis of the input data and output display characteristics [116]. The sampling grid of the barrier in such a display imposes a limit on the bandwidth that can be represented, which results in a display depth of field proportional to this frequency limit. Given a continuous input signal, antialiasing for such a display simply requires a low pass pre-filter matched to this limit. However, given light field data with discrete samples, the signal must first by resampled for display, requiring reconstruction followed by reparameterization. The display pre-filter may then be applied to eliminate aliasing within each resampled view.

In the case of a multi-view video system, pre-filtering light field imagery prior to transport also leads to reduced bandwidth requirements for transmission, because high frequency content is removed [106]. In a symmetrical video conferencing system, the target display bandwidth is known prior to the compression step, based on the display size and barrier density. In a multi-point, multi-view video conferencing system, a single source transmitted to a number of sites must be correctly pre-filtered for each different output display, whether or not they use a random or regular barrier. With broadcast multi-view video, the characteristics of every target display may not be known, and so it is not possible to pre-filter the light field data. If multi-view video standards specify display bandwidth requirements, it would be possible to develop target classes of autostereoscopic displays for which pre-filtering may be performed with suitable frequency cutoffs.

Considered within such a framework, the RHD concept does not have the same aliasing issues between views as a conventional autostereoscopic display or require all of the same resampling methods to eliminate aliasing within a single view. In the case of an input light field, a continuous signal must still be reconstructed from the samples in order to be mapped to the display. However, the reparameterization step to find a common mapping between input and output is no longer possible nor necessary due to the antialiasing mechanism inherent to the randomized sampling of the barrier.

The limited resolution for any single RHD view still imposes a limited depth of field on the display. Even though the sampling is non-uniform, the average sampling rate across the display is that of a regular barrier with the same duty cycle. This results in an average display depth of field across the RHD proportional to the duty cycle. Similarly, the maximum spatial frequency that can be displayed is also limited, which allows for a similar pre-filtering mechanism for reducing aliasing as in a regular barrier display.

A unified resampling filter for the RHD combines reconstruction of the input signal

and display pre-filtering, allowing reparameterization to occur in the barrier itself. Such a filter will eliminate aliasing artifacts, but will also exhibit a display depth of field proportional to the average barrier frequency. In the RHD prototype, which has a duty cycle similar to existing regular autostereoscopic displays, this will result in a shallow depth of field, possibly leading to blurred views. Further input resampling techniques, such as scaling the spacing between capture cameras or depth compression, may be applied when the depth of field of the acquired scene exceeds that of the display. This combination of display resampling and input processing allows for a final displayed view that eliminates both aliasing and blurring.

Chapter 7

Conclusions and future work

This dissertation investigates the issues that arise when a group of people at one site communicate with a remote group using video teleconferencing. It is not possible to fully convey the important cues of eye contact and gaze awareness between groups with a conventional video teleconferencing system. This can lead to reduced trust and turn-taking between users, and increases in pauses and interruptions. In order to replicate the experience of face-to-face conversation as closely as possible, we must develop techniques to support these cues. In this chapter, I summarize how the virtual camera algorithms and multi-view displays presented in this dissertation address these issues.

7.1 Contributions

The two main areas of research for this dissertation are virtual camera techniques and multi-view displays designed for a group of local viewers using a tele-immersion system. Virtual camera techniques allow for rendering of the remote scene from novel viewpoints that may not correspond to a camera in the remote location. This allows us to address local gaze awareness issues with a shared 2D display by rendering the scene from a distant perspective camera.

The first multi-view display is a multiple monoscopic view display that provides unique 2D views for multiple users by adapting the spacing and design of a lenticular barrier. This results in fewer user position restrictions than existing autostereoscopic displays. The second display is an autostereoscopic display that eliminates the viewing zones and user position requirements of conventional autostereoscopic displays by randomizing the pattern of barrier holes. This allows for unique 3D views to be presented to multiple viewers in arbitrary locations.

The contributions of each chapter include:

- **Line light field rendering.** A compact light field representation of a remote scene allows for real-time capture, synthesis, and rendering of a novel view of a group of users (Section 3.1). This system also provides application level camera stream aggregation to eliminate multi-stream synchronization issues (3.2).

- **Telepresence wall.** Three different rendering methods for preserving local gaze awareness across the width of a very large scale display, including plane sweeping view synthesis (Section 4.2), the Depth-Dependent Camera (4.3), and video silhouettes (4.4) are presented. I introduce a gaze-based error metric that can be used to guide placement of virtual cameras and the number and spacing of real cameras.

- **Monoscopic multi-view display** Using a wide angle lenticular lens sheet, it is possible to uniquely distribute display pixels to a wider range of locations than with an autostereoscopic display (Section 5.2). I also introduce an efficient multi-view display calibration technique that can simultaneously calibrate multiple view points (5.2.1). I present guidelines for the lens parameters need to achieve useful light distributions based on optical analysis of lenticular displays.

- **Random hole display** By randomizing the distribution of holes in a parallax barrier autostereoscopic display, it is possible to provide unique stereoscopic views from arbitrary positions. I present the notion of view interference in autostereoscopic displays as a type of aliasing (Section 6.1) and a random hole display proof of concept that can provide stereoscopic views to multiple viewers (6.2). Perceptual quality metrics show that randomized or averaged blending between views is superior to turning off conflicting pixels.

It is important to note that these techniques do not completely solve the problems of eye contact and gaze awareness for groups of viewers. However, they provide the ability to make design trade offs in group tele-immersion systems that were not previously possible. This allows for a more personalized experience for each user, either through local gaze awareness or novel views of the remote scene.

The virtual camera algorithms allow for views of a scene that do not correspond to any physical camera. This is an advantage in and of itself, because it is difficult to capture a large environment with a single physical camera without seeing a large amount of perspective distortion. When the virtual camera is positioned at a distant location behind the display, we lose some sense of perspective depth but gain a more orthographic view of the scene. This near-orthographic view is what provides local gaze awareness to users across the display. The depth-dependent camera can provide local gaze awareness for objects close to the screen, while also presenting better perspective depth cues for objects further away. This comes at the cost of requiring a view synthesis algorithm, like plane sweeping, or it requires multiple rendering passes for each depth slice of a rendered model. As graphics hardware continues to rapidly improve, both of these options are feasible.

The multi-view displays described in this dissertation address particular limitations caused by the regular barrier of conventional autostereoscopic displays, especially with regard to possible user positions. The monoscopic multi-view display trades off stereoscopic views for more unique viewing positions over a wider area. This allows users to move more freely in front of the display instead of being restricted to very particular locations. The random hole display provides autostereoscopic display from arbitrary user positions, unlike conventional autostereoscopic displays. To support multi-user gaze

awareness, both types of display must receive real or synthesized imagery appropriate for the viewer positions.

7.2 Lessons learned

We conclude with a discussion of some additional lessons learned over the course of the research for this dissertation. Each of these reflect the experiences of the author with tele-immersion techniques and systems.

Good segmentation is critical

Many view synthesis and scene reconstruction techniques rely on segmentation of objects from the background. In this dissertation, both the plane sweeping and video silhouettes methods rely on segmentation in early stages. If the segmentation does not completely include the dynamic objects, it is likely that holes will appear in the output. If the segmentation is too inclusive, shadows may be included as part of the foreground objects.

One of the main ways to improve the quality of segmentation is to enhance the lighting of the scene. This includes positioning lights to eliminate shadows in the camera images, using diffused light to reduce specular effects, and making sure that the color temperature of all lights are similar. Another factor is the flicker rate of the light source. Fluorescent lights, even with high frequency ballasts, produce different colors over very short time windows. Interaction with short camera shutter times can lead to varying colors between frames and between cameras. This can lead to incorrect segmentation, not to mention imagery that is very different than the human eye would perceive in the same scene.

Consistent lighting is not only important for segmentation, but also for color matching for scene reconstruction algorithms and visual quality. High quality commercial telepresence systems have carefully designed lighting for these reasons. LED-based lighting is a promising development that should help address these issues by offering fine control over the amount, direction, and timing of lighting.

Camera calibration is tedious

Many dozens of hours have been spent calibrating cameras for use in the tele-immersion systems described in this dissertation. Determining camera parameters is necessary, but extremely tedious, typically requiring the manual movement of calibration patterns to many different positions. Also, for several view synthesis techniques, cameras must be calibrated so that their color gamuts match, requiring additional patterns and time. Furthermore, camera calibrations degrade over time – cameras seem particularly sensitive to vibration. This can lead to the need for recalibration on a regular basis; for best results, it is a given that new calibrations should be made each time data is captured for processing.

Systems that require regular camera calibrations are thereby necessarily limited in their possible deployment. The overhead of repeated manual calibrations increases the cost and complexity just to preserve the functionality of the system. In my experience,

companies are reluctant to develop products that require this particular kind of maintenance. Ideally, cameras could simply be plugged in and would calibrate themselves based on scene content, and then constantly update their calibration as the system operates. Wide commercial deployment of multi-camera systems requiring calibration is unlikely until this occurs. Luckily, research in automatic and continuous camera calibration has begun to address this very issue. Once such methods are more robust and reliable enough to make manual calibration unnecessary, multi-camera systems will become more appealing.

Demo on the target display

In developing multi-user tele-immersion systems, we always try to show demonstrations of our techniques on the intended display, or to simulate that display at the correct location and scale. This is obviously important for conveying eye contact and gaze awareness, but it also helps viewers better understand camera placement issues and the trade offs of various virtual camera positions than viewing rendered imagery on a regular computer monitor. If a user can move to one side of a large scale display, then they can actually experience the local gaze awareness that our algorithms support.

It is also important to convey the size and scale of the display even if the viewers are not in the actual display space. We can do this by taking photos and video of the display itself, while it is showing the rendering output. For example, the imagery from the telepresence wall may be presented as the rendering output that is sent to the display, as shown in Figure 4.23, or it may be shown as an image of that rendering on the display itself, as in Figure 4.21. I have found that the latter is more effective in conveying the effect of a large scale display to distant users.

7.3 Future work

There are a number of directions for possible future work based on this research, along with related systems that address similar design issues. I describe several of these possibilities, including extensions to the rendering techniques and improvements to the multi-view displays.

Hybrid video silhouettes with extra information

To improve the identification and segmentation of users of a system like the telepresence wall, we want to use additional sensors and information, in combination with the existing video capture methods. A significant problem is the differentiation between people that are in front of one another with respect to the display wall. In some camera views they may be segmented as a single object, and in others they will each be seen separately. Using additional cameras, placed on the side or top of the room, will allow us to identify each person separately. We can use visual hull techniques to generate models used to improve segmentation, accurately determine the depth of each person, and to generate better geometric proxies for rendering.

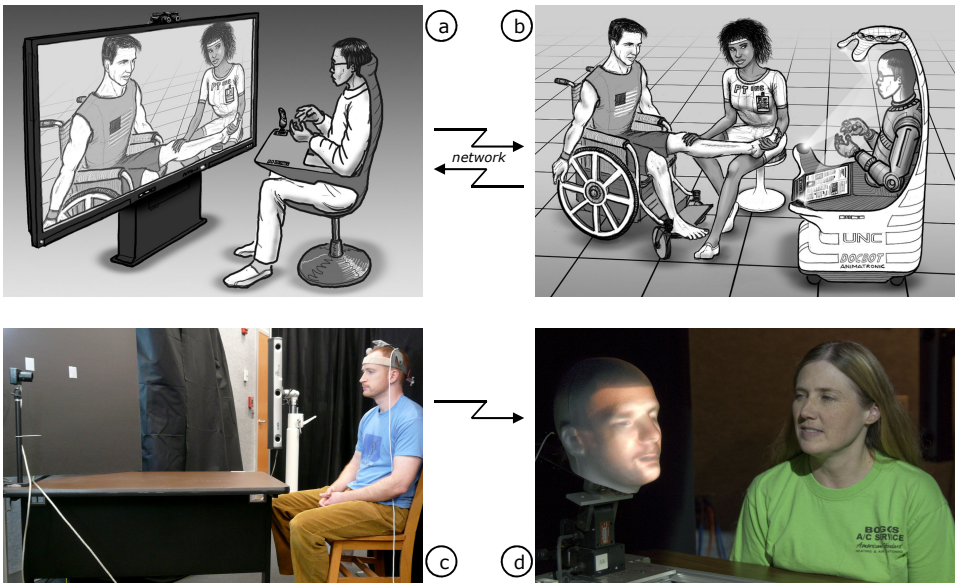


Figure 7.1: Shader Lamps Avatars (SLA): (a) and (b) show a full-duplex telepresence scenario for remote medical consultation. The physician in (a) interacts with a remote patient and therapist in (b) by means of a camera-equipped SLA. In our prototype system the user in (c) wears a tracking system and is captured by a video camera. In (d) we show the animatronic styrofoam head illuminated by a projector.

Tabletop autostereoscopic display

Previous work in multi-view tabletop displays has sought to provide distinct imagery to multiple users across the same display surface to enable more practical uses, such as a shared modeling application [45]. A limited number of multi-view tabletop displays can provide stereoscopic imagery to multiple viewers [96]. However, none of these systems was compelling enough to encourage wide adoption. More recently, multi-touch tabletop displays, where one or more users can interact with digital content by touching the display [32, 20], have become popular. Such systems are now commercially available, although to date they are typically used for entertainment-oriented applications. None of these systems currently support stereoscopic viewing for multiple users.

Conventional autostereoscopic displays only multiplex different views horizontally, and so users viewing the displays from different orientations are not able to see a stereoscopic view. This prevents their effective use as a tabletop display. The Random Hole Display design can provide stereoscopic views in any orientation, making it particularly suited to a tabletop orientation. The challenge will be integrating the touch sensing surface with a multi-view barrier.

Animatronic avatars

Instead of presenting a view of a remote participant on a conventional or even multi-view display, we have developed a system called Shader Lamps Avatars (SLA) to support the projection of the imagery of a person onto a humanoid animatronic model. This

offers advantages even beyond correct multi-view display of a remote user. By also capturing their motion, we can move the model to mimic the orientation and pose of the user, to correctly convey attention to the local viewers. Such a physical avatar provides an infinite number of possible viewing positions for the local users. We show a concept application and the prototype system in Figure 7.1. Our initial results have been published and demonstrated [58].

7.4 Conclusion

While video conferencing is widely deployed, the conventional systems currently in use do not provide the sense of presence that is the hallmark of normal face-to-face conversation. The research described above is but one step on the path to improving this deficiency by addressing eye contact and gaze awareness issues. The proofs-of-concept described in this dissertation suggest that such scientific and engineering challenges are resolved, natural interaction via video teleconferencing will be a mainstream realization.

Bibliography

[1] Actuality Systems Inc. (2001). Volumetric 3-D Display. http://www.actuality-systems.com. 25

[2] Agocs, T., Balogh, T., Forgacs, T., Bettio, F., Gobbetti, E., Zanetti, G., and Bouvier, E. (2006). A large scale interactive holographic display. In *Virtual Reality Conference, 2006*, pages 311–311. 21, 24, 25

[3] Anon. (1924). Pictures by wire sent with success for the first time. *New York Times*. 20 May 1924. 12

[4] Autodesk (2009). Autodesk 3ds max, `http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659302`, 23 jun 2009. 62

[5] Baker, H. H., Bhatti, N., Tanguay, D., Sobel, I., Gelb, D., Goss, M. E., MacCormick, J., Yuasa, K., Culbertson, W. B., and Malzbender, T. (2003). Computation and performance issues in coliseum: an immersive videoconferencing system. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 470–479, New York, NY, USA. ACM. 16

[6] Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M. (2001). Unstructured Lumigraph Rendering. In *Proceedings of SIGGRAPH 2001*, pages 43–54, Los Angels. 29

[7] Burns, R. W. (1998). *Television: An International History of the Formative Years*. The Institution of Electrical Engineers. 12

[8] Chai, J.-X., Tong, X., Chan, S.-C., and Shum, H.-Y. (2000). Plenoptic Sampling. In *Proceedings of SIGGRAPH 2000*, page 307318, New Orleans. 33

[9] Chen, M. (2002). Leveraging the asymmetric sensitivity of eye contact for videoconference. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 49–56, New York, NY, USA. ACM. 15, 43

[10] Chen, W.-C., Towles, H., Nyland, L., Welch, G., and Fuchs, H. (2000). Toward a compelling sensation of telepresence: demonstrating a portal to a distant (static) office. In *VIS '00: Proceedings of the conference on Visualization '00*, pages 327–333, Los Alamitos, CA, USA. IEEE Computer Society Press. 21, 25

[11] Chiariglione, L. (2009). Mpeg-1 `http://www.chiariglione.org/mpeg/`. 13

[12] Cisco (2009a). Cisco telepresence system 3000, `http://www.cisco.com/en/US/products/ps8333/`, 22 jun 2009. 42

[13] Cisco (2009b). Cisco telepresence, `http://www.cisco.com/en/US/netsol/ns669/networking_solutions_solution_segment_home.html`, 20 jun 2009. 14, 61, 68

[14] Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 358, Washington, DC, USA. IEEE Computer Society. 47

[15] Cook, R. L. (1986). Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72. 119

[16] Cooke, E., Feldmann, I., Kauff, P., and Schreer, O. (2003). A modular approach to virtual view creation for a scalable immersive teleconferencing configuration. volume 3, pages III–41–4 vol.2. 19

[17] Darabiha, A., Rose, J., and MacLean, W. J. (2003). Video-Rate Stereo Depth Measurement on Programmable Hardware. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–210. 18

[18] De Silva, L., Tahara, M., Aizawa, K., and Hatori, M. (1995). A teleconferencing system capable of multiple person eye contact (mpec) using half mirrors and cameras placed at common points of extended lines of gaze. *Circuits and Systems for Video Technology, IEEE Transactions on*, 5(4):268–277. 17

[19] Dereniak, E. L. and Dereniak, T. D. (2008). *Geometrical and Trigonometrical Optics*. Cambridge University Press. 92, 93

[20] Dietz, P. and Leigh, D. (2001). Diamondtouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, New York, NY, USA. ACM. 143

[21] Dippé, M. A. Z. and Wold, E. H. (1985). Antialiasing through stochastic sampling. *SIGGRAPH Comput. Graph.*, 19(3):69–78. 119

[22] Dorcey, T. (1995). Cu-seeme desktop videoconferencing software. *Connexions*, 9(3). 13

[23] Dorros, I. (1969). Picturephone. *Bell Laboratories Record*. 12

[24] du Maurier, G. (1878). Edison's telephonoscope (transmits light as well as sound). *Punch magazine*. December 9th, 1878. 10

[25] Farnsworth, P. T. (1930). *Television System*, U.S. Patent 1,773,980, 26 Aug 1930. 12

[26] Faugeras, O., Hotz, B., Mathieu, H., Viville, T., Zhang, Z., Fua, P., Thron, E., Moll, L., Berry, G., Vuillemin, J., Bertin, P., and Proy, C. (1993). Real time sorrelation-based stereo: Algorithm, amplementations and application. Technical Report 2013, INRIA. 18

[27] Gemmell, J., Toyama, K., Zitnick, C., Kang, T., and Seitz, S. (2000). Gaze awareness for video-conferencing: a software approach. *Multimedia, IEEE*, 7(4):26–35. 18

[28] Gibson, J. J. and Pick, A. D. (1963). Perception of another person's looking behavior. *American Journal of Psychology*, 76:386–394. 15, 43

[29] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The Lumigraph. In *Proceedings of SIGGRAPH 1996*, pages 43–54, New Orleans. 19

[30] Gross, M., Wrmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-meier, E., Svoboda, T., Gool, L. V., Lang, S., Strehlke, K., Moere, A. V., Oliver, Zrich, E., and Staadt, O. (2003). blue-c: A spatially immersive display and 3d video portal for telepresence. In *ACM Transactions on Graphics*, pages 819–827. 17

[31] Halle, M. W., Benton, S. A., Klug, M. A., and Underkoffler, J. S. (1991). The ultragram: A generalized holographic stereogram. In *Masters thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology*, pages 142–155. 21

[32] Han, J. Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA. ACM. 143

[33] Hanzo, L., Cherriman, P., and Streit, J. (2007). *Video Compression and Communications: From Basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers*. Wiley-IEEE Press. 12

[34] Hirschmuler, H. (2001). Improvements in Real-Time Correlation-Based Stereo Vision. In *Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 141–148, Kauai, Hawaii. 18

[35] Hirschmuller, H., Innocent, P., and Garibaldi, J. (2002). Real-Time Correlation-Based Stereo Vision with Reduced Border Errors. *International Journal of Computer Vision*, 47(1-3). 18

[36] HP (2009). Hp halo, `http://h71028.www7.hp.com/enterprise/us/en/halo/index.html`. 14

[37] Huynh-Thu, Q. and Ghanbari, M. (2008). Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801. 127

[38] Ilie, A. and Welch, G. (2005). Ensuring color consistency across multiple cameras. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1268–1275, Washington, DC, USA. IEEE Computer Society. 54

[39] Isaacs, E. A. and Tang, J. C. (1993). What video can and can't do for collaboration: a case study. In *MULTIMEDIA '93: Proceedings of the first ACM international conference on Multimedia*, pages 199–206, New York, NY, USA. ACM. 15

[40] Isaksen, A., McMillan, L., and Gortler, S. J. (2000). Dynamically Reparameterized Light Fields. In *Proceedings of SIGGRAPH 2000*, pages 297–306. 29

[41] Ishii, H. and Kobayashi, M. (1992). Clearboard: a seamless medium for shared drawing and conversation with eye contact. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 525–532, New York, NY, USA. ACM. 20

[42] iZ3D (2009). iz3d. iZ3D, `http://www.iz3d.com/`, 20 Jun 2009. 24

[43] Jerald, J. and Daily, M. (2002). Eye gaze correction for videoconferencing. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 77–81, New York, NY, USA. ACM. 18

[44] Jones, A., McDowall, I., Yamada, H., Bolas, M., and Debevec, P. (2007). Rendering for an interactive 360 light field display. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 40, New York, NY, USA. ACM. 21

[45] Kakehi, Y., Iida, M., Naemura, T., Shirai, Y., Matsushita, M., and Ohguro, T. (2005). Lumisight table: An interactive view-dependent tabletop display. *IEEE Comput. Graph. Appl.*, 25(1):48–53. 24, 25, 143

[46] Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. (1996). A Stereo Engine for Video-rate Dense Depth Mapping and Its New Applications. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 196–202. 18

[47] Kauff, P. and Schreer, O. (2002). An immersive 3d video-conferencing system using shared virtual team user environments. In *CVE '02: Proceedings of the 4th international conference on Collaborative virtual environments*, pages 105–112, New York, NY, USA. ACM. 18

[48] Kitamura, Y., Konishi, T., Yamamoto, S., and Kishino, F. (2001). Interactive stereoscopic display for three or more users. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 231–240, New York, NY, USA. ACM. 24, 25

[49] Konolige, K. (1997). Small Vision Systems: Hardware and Implementation. In *Proceedings of the 8th International Symposium in Robotic Research*, pages 203–212. Springer-Verlag. 18

[50] Kunita, Y., Ogawa, N., Sakuma, A., Inami, M., Maeda, T., and Tachi, S. (2001). Immersive autostereoscopic display for mutual telexistence: Twister i (telexistence wide-angle immersive stereoscope model i). In *Virtual Reality, 2001. Proceedings. IEEE*, pages 31–36. 24

[51] Kuo, H. P., Hubby, L. M., Naberhuis, S. L., and Birecki, H. (2006). *See-through display*, U.S. Patent Application 11/491,360 (US 2008/0018555 A1), 21 Jul 2006. 16

[52] Lanier, J. (1998). Tele-immersion: The ultimate qos-critical application. In *in First Internet2 Joint Applications/ Engineering QoS Workshop*. 16

[53] Laurentini, A. (1994). The Visual Hull Concept for Silhouette Based Image Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162. 19

[54] Lee, J. C., Dietz, P. H., Maynes-Aminzade, D., Raskar, R., and Hudson, S. E. (2004). Automatic projector calibration with embedded light sensors. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 123–126, New York, NY, USA. ACM. 98

[55] Levoy, M. and Hanrahan, P. (1996). Light Field Rendering. In *Proceedings of SIGGRAPH 1996*, pages 31–42, New Orleans. 4, 19, 27

[56] Lin, Z.-C. and Shum, H.-Y. (2000). On the numbers of samples needed in light field-rendering with constant-depth assumption. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*. 33

[57] Lincoln, P., Nashel, A., Ilie, A., Towles, H., Welch, G., and Fuchs, H. (2009a). Multi-view lenticular display for group teleconferencing. In *IMMERSCOM 2009: 2nd International Conference on Immersive Telecommunications*. ICST. 5

[58] Lincoln, P., Welch, G., Nashel, A., Ilie, A., State, A., and Fuchs, H. (2009b). Animatronic shader lamps avatars. In *ISMAR Symposium and Expo 2009*. 144

[59] Lok, B. (2001). Online Model Reconstruction for Interactive Virtual Environments. In *Proceedings 2001 Symposium on Interactive 3D Graphics*, pages 69–72, Chapel Hill, North Carolina. 20

[60] Majumder, A., Meenakshisundaram, G., Seales, W. B., and Fuchs, H. (1999). Immersive teleconferencing: A new algorithm to generate seamless panoramic video imagery. 67

[61] Mason, M. and Duric, Z. (2001). Using histograms to detect and track objects in color video. *Applied Image Pattern Recognition Workshop,*, 0:0154. 71

[62] Matusik, W., Buehler, C., Raskar, R., Gortler, S., and McMillan, L. (2000). Image-Based Visual Hulls. In *Proceedings of SIGGRAPH 2000*, pages 369–374, New Orleans. 20

[63] Matusik, W. and Pfister, H. (2004). 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 814–824, New York, NY, USA. ACM. 23, 24, 25, 87, 118

[64] McMillan, L. and Bishop, G. (1995). Plenoptic Modeling: An Image-Based Rendering System. In *Proceedings of SIGGRAPH 1995*, pages 39–46. 19

[65] Mitchell, D. P. and Netravali, A. N. (1988). Reconstruction filters in computer-graphics. *SIGGRAPH Comput. Graph.*, 22(4):221–228. 119

[66] Monk, A. F. and Gale, C. (2002). A look is worth a thousand words: Full gaze awareness in video-mediated conversation. *Discourse Processes*, 33(3):257–278. 15, 16

[67] Mulligan, J. and Daniilidis, K. (2000). View-independent Scene Acquisition for Tele-Presence. Technical Report MS-CIS-00-16, Computer and Information Science Dept., U. ofPennsylvania. 18

[68] Mulligan, J., Isler, V., and Daniilidis, K. (2002). Trinocular Stereo: A New Algorithm and its Evaluation. *International Journal of Computer Vision (IJCV), Special Issue on Stereo and Multi-baseline Vision*, 47:51–61. 18

[69] Musion (2009). Musion eyeliner, `http://www.eyeliner3d.com/`, 22 jun 2009. 17

[70] Nashel, A. and Fuchs, H. (2009). Random hole display: A non-uniform barrier autostereoscopic display. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, pages 1–4. 5

[71] Nashel, A., Yang, C. L., and Stewart, C. (2005). Group tele-immersion:enabling natural interactions between groups at distant sites. Technical Report SAND2005-5206, Sandia National Laboratories. 4

[72] Newsight (2009). Newsight multiview displays, `http://www.newsight.com/index.php?id=61`, 20 jun 2009. 21, 24, 25, 85

[73] Nguyen, D. and Canny, J. (2005). Multiview: spatially faithful group video conferencing. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 799–808, New York, NY, USA. ACM. 16, 23, 25

[74] Nguyen, D. T. and Canny, J. (2007). Multiview: improving trust in group video conferencing through spatial faithfulness. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1465–1474, New York, NY, USA. ACM. 16, 23

[75] Ohya, J., Kitamura, Y., Takemura, H., Kishino, F., and Terashima, N. (1993). Real-Time Reproduction of 3D Human Images in Virtual Space Teleconferencing. In *Proceedings of Virtual Reality Annual International Symposium*, pages 408–414. 15

[76] Okada, K.-I., Maeda, F., Ichikawaa, Y., and Matsushita, Y. (1994). Multiparty videoconferencing at virtual social distance: Majic design. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 385–393, New York, NY, USA. ACM. 20

[77] Okubo, S. (1995). Reference model methodology-a tool for the collaborative creation of video coding standards. *Proceedings of the IEEE*, 83(2):139–150. 13

[78] Perlin, K., Paxia, S., and Kollin, J. S. (2000). An autostereoscopic display. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 319–326, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. 22, 24, 25

[79] Peterka, T., Kooima, R. L., Girado, J. I., Ge, J., Sandin, D. J., Johnson, A., Leigh, J., Schulze, J., and DeFanti, T. A. (2007). Dynallax: Solid state dynamic parallax barrier autostereoscopic vr display. *Virtual Reality Conference, IEEE*, 0:155–162. 22, 24, 25

[80] Peterka, T., Sandin, D. J., Ge, J., Girado, J., Kooima, R., Leigh, J., Johnson, A., Thiebaux, M., and DeFanti, T. A. (2006). Personal varrier: autostereoscopic virtual reality display for distributed scientific visualization. *Future Gener. Comput. Syst.*, 22(8):976–983. 91

[81] Philips (2009). Philips 3d solutions. `http://www.philips.com/3Dsolutions`. 21, 24, 25, 85, 87, 118

[82] Point Grey Research (2009). `http://www.ptgrey.com`. 20 Jun 2009. 18, 39

[83] Polycom (2009a). Polycom hdx 400, `http://www.polycom.com/products/telepresence_video/telepresence_solutions/personal_telepresence/hdx4000.html`. 14

[84] Polycom (2009b). Polycom rpx hd 400, `http://www.polycom.com/products/telepresence_video/telepresence_solutions/immersive_telepresence/rpx_hd400.html`. 14

[85] Rademacher, P. and Bishop, G. (1998). Multiple-center-of-projection images. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 199–206, New York, NY, USA. ACM. 61

[86] Rijkse, K. (1996). H.263: video coding for low-bit-rate communication. *Communications Magazine, IEEE*, 34(12):42–45. 13

[87] Roberts, D. E. (2003). History of lenticular and related autostereoscopic methods. White paper, Leap Technologies, LLC. 87

[88] Rose, D. and Clarke, P. (1995). A review of eye-to-eye videoconferencing techniques. *BT Technology Journal*, 13:127–131. 15, 16

[89] Sandin, D. J., Margolis, T., Ge, J., Girado, J., Peterka, T., and DeFanti, T. A. (2005). The varrier autostereoscopic virtual reality display. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 894–903, New York, NY, USA. ACM. 22, 24, 25, 132

[90] Scharstein, D. and Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42. 18

[91] Schirmacher, H., Ming, L., and Seidel, H.-P. (2001). On-the-Fly Processing of Generalized Lumigraphs. *EUROGRAPHICS 2001*, 20(3). 19

[92] Sellen, A., Buxton, B., and Arnott, J. (1992). Using spatial cues to improve videoconferencing. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 651–652, New York, NY, USA. ACM. 20

[93] Serway, R. A. (1986). *Physics for Scientists & Engineers*, volume 2. Saunders College Publishing. 88

[94] Sharp (2003). Actius rd3d. Actius RD3D, `http://www.sharp3d.com/`, 15 Dec 2003. 24

[95] Shum, H. Y. and He, L. W. (1997). Rendering with Concentric Mosaics. In *Proceedings of SIGGRAPH 1997*, pages 299–306. 19

[96] Smith, R. T. and Piekarski, W. (2008). Public and private workspaces on tabletop displays. In *AUIC '08: Proceedings of the ninth conference on Australasian user interface*, pages 51–54, Darlinghurst, Australia, Australia. Australian Computer Society, Inc. 143

[97] StereoGraphics (2009). Synthagram. Synthagram monitor Series, `http://www.vrlogic.com/html/stereographics/synthagram.html`, 20 Jun 2009. 24, 25

[98] TANDBERG (2009a). Tandberg 150 mxp, `http://www.tandberg.com/products/video_systems/tandberg_150_mxp.jsp/`. 14

[99] TANDBERG (2009b). Tandberg telepresence, `http://www.tandberg.com/products/telepresence/index.jsp`. 14, 68

[100] Telecom, F. (2003). The telepresence wall which eradicates frontiers and distance, `http://www.francetelecom.com/sirius/rd/en/galerie/mur_telepresence/pdf/doc.pdf`. 17, 45

[101] Thomos, N., Boulgouris, N., and Strintzis, M. (2006). Optimized transmission of jpeg2000 streams over wireless channels. *Image Processing, IEEE Transactions on*, 15(1):54–67. 131

[102] Uy, M. (2006). *Integrated sensing display*, U.S. Patent 20,060,007,222, 12 Jan 2006. 16

[103] Vertegaal, R. (1999). The gaze groupware system: mediating joint attention in multiparty communication and collaboration. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 294–301, New York, NY, USA. ACM. 15

[104] Vertegaal, R., van der Veer, G., and Vons, H. (2000). Effects of gaze on multiparty mediated communication. In *In Proceedings of Graphics Interface*, pages 95–102. Morgan Kaufmann. 15

[105] Vertegaal, R., Weevers, I., Sohn, C., and Cheung, C. (2003). Gaze-2: conveying eye contact in group video conferencing using eye-controlled camera direction. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 521–528, New York, NY, USA. ACM. 20

[106] Vetro, A., Yea, S., Zwicker, M., Matusik, W., and Pfister, H. (2007). Overview of multiview video coding and anti-aliasing for 3d displays. volume 1, pages I –17 –I –20. 137

[107] Wiegand, T., Sullivan, G., Bjontegaard, G., and Luthra, A. (2003). Overview of the h.264/avc video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576. 13

[108] Woodfill, J. and Herzen, B. V. (1997). Real-Time Stereo Vision on the PARTS Reconfigurable Computer. In Pocek, K. L. and Arnold, J., editors, *IEEE Symposium on FPGAs for Custom Computing Machines*, pages 201–210, Los Alamitos, CA. IEEE Computer Society Press. 18

[109] Yamaashi, K., Cooperstock, J., Narine, T., , and Buxton, W. (1996). Beating the limitations of camera-monitor mediated telepresence with extra eyes. In *SIGCHI 96 Conference Proceedings on Human Factors in Computer Systems*. 39

[110] Yang, J., Everett, M., Buehler, C., and McMillan, L. (2002). A Real-Time Distributed Light Field Camera. In *Proceedings of Eurographics Workshop on Rendering*, pages 77–86. 29

[111] Yang, R. (2003). *View-dependent pixel coloring: a physically-based approach for two-dimensional view synthesis*. PhD thesis, The University of North Carolina at Chapel Hill. Director-Welch, Greg. 5, 47

[112] Yang, R., Nashel, A., and Towles, H. (2004). Interactive 3d teleconferencing with user-adaptive views. In *ETP '04: Proceedings of the 2004 ACM SIGMM workshop on Effective telepresence*, pages 50–51, New York, NY, USA. ACM. 4

[113] Yang, R., Welch, G., and Bishop, G. (2003). Real-time consensus-based scene reconstruction using commodity graphics hardware. *Computer Graphics Forum*, 22(2):207–216. 47

[114] Yang, R. and Zhang, Z. (2002). Eye Gaze Correction with Stereovision for Video-Teleconferencing. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 479–494. 18

[115] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334. 34

[116] Zwicker, M., Matusik, W., Durand, F., and Pfister, H. (2006). Antialiasing for automultiscopic 3d displays. In *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering*, pages 73–82. 119, 137