Rendering interactive holographic images

Conference Paper · January 1995

DOI: 10.1145/218380.218490 · Source: DBLP

2 authors:

Mark Lucente
Nanohmics, Inc.
47 PUBLICATIONS **1,366** CITATIONS
SEE PROFILE

Tinsley A. Galyean
Massachusetts Institute of Technology
17 PUBLICATIONS **1,287** CITATIONS
SEE PROFILE

Rendering Interactive Holographic Images

Mark Lucente and Tinsley A. Galyean

Massachusetts Institute of Technology

Media Laboratory*

ABSTRACT

We present a method for computing holographic patterns for the generation of three-dimensional (3-D) holographic images at interactive speeds. We used this method to render holograms on a conventional computer graphics workstation. The framebuffer system supplied signals directly to a real-time holographic ("holovideo") display. We developed an efficient algorithm for computing an image-plane stereogram, a type of hologram that allowed for several computational simplifications. The rendering algorithm generated the holographic pattern by compositing a sequence of view images that were rendered using a recentering shear-camera geometry. Computational efficiencies of our rendering method allowed the workstation to calculate a 6-megabyte holographic pattern in under 2 seconds, over 100 times faster than traditional computing methods. Data-transfer time was negligible. Holovideo displays are ideal for numerous 3-D visualization applications, and promise to provide 3-D images with extreme realism. Although the focus of this work was on fast computation for holovideo, the computed holograms can be displayed using other holographic media. We present our method for generating holographic patterns, preceded by a background section containing an introduction to optical and computational holography and holographic displays.

Keywords: Electro-Holography, Holovideo, Computer-Generated Holography, Accumulation Buffer.
CR Categories: I.3.1 [Computer Graphics]: Hardware Architecture - three-dimensional displays, graphics processors; **I.3.7** [Computer Graphics]: Three-Dimensional Graphics and Realism.

1. INTRODUCTION

The practical use of three-dimensional (3-D) displays has long been a goal in computer graphics. Three-dimensional displays are generally electronic devices that provide binocular depth cues, particularly binocular disparity and convergence. (See the Glossary on the next page.) Some 3-D displays provide additional depth cues such as motion parallax and ocular accommodation. The reference by McKenna[1] contains a good discussion of the visual depth cues and a detailed evaluation of 3-D display techniques.

Recently, some 3-D displays have been used interactively. A 3-D display allows the viewer to more efficiently and accurately sense both the 3-D shapes of objects and their relative spatial locations, particularly when monocular depth cues are not prevalent in a scene. When viewing complex or unfamiliar object scenes, the viewer can more quickly and accurately identify the scene contents. Therefore, 3-D displays are important in any application involving the visualization of 3-D data, including telepresence, education, medical imaging, computer-aided design, scientific visualization, and entertainment.

The merit of a 3-D display depends primarily on its ability to provide depth cues and high resolutions. The inclusion of depth cues - particularly binocular disparity, motion parallax, and occlusion - increases the realism of an image. Holography[2] is the only imaging technique that can provide all the depth cues[1]. All other 3-D display devices lack one or more of the visual depth cues. For example, stereoscopic displays do not provide ocular accommodation, and volume displays cannot provide occlusion. Image resolution and parallax resolution are also important considerations when displaying 3-D images. While most 3-D displays fail to provide acceptable image and parallax resolutions, holography can produce images with virtually unlimited resolutions.

In optical holography, a recorded interference pattern reconstructs an image with an extremely high degree of accuracy. A holographic pattern - called *fringes* - can be computed and used to generate a 3-D image, most recently in real time[3]. Both the computing process and the displaying process are significantly more difficult than in other 3-D display systems. Nevertheless, a real--time electro-holographic ("holovideo") display can produce dynamic 3-D images with all of the depth cues and image realism found in optical holography. Therefore, holovideo has the potential to produce the highest quality 3-D images. Also, to view holographic images, the viewer is unencumbered by equipment such as glasses or sensors. Figure 1 illustrates the basic functionality of holovideo.

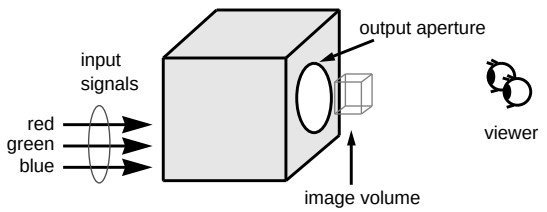

Figure 1: A typical real-time 3-D holographic (holovideo) display. When positioned in the viewing zone, viewers see a 3-D image in the vicinity of the output aperture. The input signals are generated by a computer.

*MIT Media Laboratory, Cambridge, MA 02139. (lucente | tag)@media.mit.edu
Mark Lucente is now at:
IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598

The size and complexity of holographic fringe patterns often precludes their computation at interactive rates. In the field of computational holography[4], a discretized holographic fringe pattern is generated by numerically simulating the propagation and interference of light. Typical sampling sizes are smaller than the wavelength of visible light. Therefore, a computer-generated hologram (CGH) must contain a huge number of samples. Furthermore, the cost of calculating each sample is high if a conventional approach is taken. Even with the power currently available in scientific workstations, researchers in the field of computational holography commonly report computation times in minutes or hours. In recent work[5], Lucente used a massively parallel supercomputer to calculate holographic fringes at interactive rates.

In this paper we describe the first use of a standard graphics workstation to render and display holograms at interactive rates. The graphics workstation provided both a platform for generating image information and the computing power for generating holographic fringes. Our use of a computer graphics workstation also eliminated data transfer bottlenecks that often prohibit interactive computation. We believe that new holographic displays will continue to emerge in the future, necessitating rapid rendering (computation) of holograms. Our work is an important first step toward practical computation systems for holovideo.

The following Background section gives a brief summary of the principles of holography. The computer generation of holograms and their display are reviewed, including a brief description of the real-time holographic display that we used in this research. In the section Method for Rendering Hologram we describe the hologram rendering algorithm, including the initial processing of object scene data to provide for realistic lighting, shading, occlusion, and other pictorial depth cues. Finally, we present results, future work, and a conclusion.

2. BACKGROUND

This section contains an introduction to holography, computational holography, holographic displays, and stereograms. The concepts discussed here were essential to the development of a faster method for hologram rendering.

2.1 Optical Holography

Optical holography[2] uses the physical phenomena of interference and diffraction to record and reconstruct a 3-D image. Holographic imaging became practical in the 1960's with the advent of coherent monochromatic laser light. To produce a hologram, light is scattered from the object to be recorded. A photosensitive medium records the intensity (irradiance) pattern that results when the light scattered from an object interferes with a spatially clean mutually coherent reference beam. The reference beam allows the medium to record both the magnitude and phase of the incident object wavefront, in essence recording variations in both the intensity and the direction of the light. The recording medium must have sufficient resolution to record spatial frequencies that are typically 1500 linepairs/millimeter or more.

To reconstruct an image, the recorded interference pattern modulates an illuminating beam of light. The modulated light diffracts (bends and focuses) and reconstructs a 3-D replica of the wavefront that was scattered from the object scene. Optical wavefront reconstruction makes the image appear to be physically present and tangible. The image possesses all of the depth cues exhibited by the original object, including continuous parallax (vertical and horizontal) and ocular accommodation. Both the image resolution and parallax resolution of an optical holographic image are virtually unlimited.

GLOSSARY

Visual Depth Cues

binocular disparity: the binocular depth cue effected by the slight differences between the two retinal images captured by each eye. The depth sensation caused by binocular disparity is called **stereopsis**.

convergence: a binocular depth cue effected when the eyes rotate to align the retinal images captured by each eye.

occlusion, **overlap**: monocular depth cue effected when one part of image is obstructed by another overlapping part.

ocular accommodation: a monocular depth cue in which the eye senses depth by focusing at different distances.

parallax, **motion parallax**: a (monocular) depth cue sensed from the apparent change in the lateral displacements among objects in a scene as the viewer moves. A display which provides parallax allows the viewer to move around the object scene.

pictorial depth cues: the monocular depth cues found in 2-D images, including occlusion, linear perspective, texture gradient, aerial perspective, shading, and relative sizes.

3-D Display Types

stereoscopic: a 3-D display type that presents a left view of the imaged scene to the left eye and a right view to the right eye. Examples include boom-mounted, head-mounted, and displays using polarizing glasses.

autostereoscopic: a 3-D display type that presents left and right views of the imaged scene without special viewing aids. Examples include lenticular, parallax barrier, slice stacking, and holography. Some provide motion parallax by presenting more than two views.

holovideo: a real-time 3-D electro-holographic display.

Additional Terms

basis fringe: an elemental fringe pattern computed to diffract light in a specific manner. The name "basis fringe" is an analogy to mathematical basis functions. Linear summations of basis fringes were used as holographic patterns.

computational holography, **computer-generated holography**: the numerical synthesis of holograms.

hololine: a horizontal line of samples of a holographic fringe pattern.

horizontal-parallax-only or **HPO**: possessing horizontal parallax but not vertical parallax. The viewer sees the same vertical perspective of the imaged scene regardless of the vertical location of the eyes.

image volume: the volume occupied by a 3-D image.

image resolution: the number of resolvable image features in the lateral dimensions of an image.

parallax resolution: the number of different perspective views available to the viewer.

2.2 Computational Holography

To produce dynamic holographic images, researchers can compute and display holographic fringe patterns. Computational holography[4] generally begins with a 3-D numerical description of the object or scene to be reproduced. Light is numerically scattered from the object scene and propagated to the plane of the hologram. The object wavefront is calculated, and a reference beam wavefront added, imitating optical interference. The resulting total intensity - the fringe pattern - is used by a holographic display to produce the 3-D image. Such a display spatially modulates a beam of light with the fringes, mimicking the reconstruction step in optical holography.

Computational holography produces 3-D imaging with a high degree of realism. Only recently have researchers achieved computational holography at interactive rates, impeded mainly by the enormous data content and by the computational complexity associated with holographic fringes[5]. A typical full-parallax hologram 100 mm $\times$ 100 mm in size has a space-bandwidth product (SBWP) of over 100 gigasamples of information. A larger image requires a proportionally larger number of samples. Several techniques have been used to reduce information content to a manageable size. The elimination of vertical parallax provides great savings in display complexity[3] and computational requirements[5] without greatly compromising the overall display performance. Other less desirable sacrifices include reducing the size of the object scene or the size of the viewing zone.

There are several methods of computing holographic fringes. Most imitate the interference that occurs when a hologram is produced optically using coherent light. Early methods made use of the Fourier transform to calculate the phase and amplitude of the object wavefront at the plane of the hologram[4]. At each sample point in the hologram plane, this object light wavefront was added to a reference beam wavefront, and the magnitude squared became the desired fringe pattern. This approach was used to create planar images. Multiple image planes were separately processed and combined to produce 3-D images. Consequently, the Fourier-transform approach was slow and inefficient for computing 3-D images.

A more straightforward approach to the computation of holographic fringes resembled 3-D computer graphics ray-tracing. Light from a given point or element of an object contributed a complex wavefront at the hologram plane[2]. Each of these complex wavefronts was summed to calculate the total object wavefront which was subsequently added to a reference wavefront. Black (non-scattering) regions of the image volume were ignored, enabling very rapid computations of simple object scenes. Again, for more complex images, computation was prohibitively slow.

In recent work[5], Lucente described a computational method that greatly simplified the ray-tracing approach. By computing fringes that diffracted light only to specific regions of the image volume, this approach had a number of advantages: (1) it enabled the real-valued linear superposition of fringe patterns; (2) it increased the speed of computation by a factor of 2.0; (3) it eliminated the need for an explicit reference beam wavefront; and (4) it eliminated certain imaging artifacts and noise that generally degrade the quality of holographic images. Real-valued summation enabled the efficient use of precomputed elemental fringe patterns (called *basis fringes*), an approach that achieved CGH computation at interactive rates when implemented on a supercomputer[5].

We chose to implement an approach similar to the Lucente approach - using an array of precomputed basis fringes - because it promised to provide the fastest holographic computation. We focused on further reducing computational complexity and on the implementation of holographic computation on hardware designed for computer graphics rendering.

2.3 Holographic Displays

A holographic fringe pattern is used to modulate a beam of monochromatic light to produce an image. In the earliest work in computational holography, the computed fringe pattern was recorded (permanently) in a piece of film[4]. The film provided the SBWP sufficient to represent the fringes. In some cases, the film also provided grayscale. Light passing through this film created a static holographic image.

To create dynamic holographic images, a dynamic spatial light modulator (SLM) must be used. The SBWP of a holographic SLM must be as high as that of holographic film. Ideally, a holographic SLM must display over 100 gigasamples. Current SLMs, however, can provide a maximum of 10 megasamples. Examples of SLMs include the flat-panel liquid-crystal display (LCD) and the magneto-optic SLM. These SLMs are capable of displaying a very small CGH pattern in real time. Early researchers employed a liquid--crystal SLM with an SBWP of 10,000 elements to produce a tiny planar image[6]. Most researchers reported images that were still quite small and essentially two-dimensional[7].

An ideal holographic SLM does not yet exist, but time-multiplexing of a very fast SLM provides a suitable substitute. The display system that we used in this research was similar to previously reported holovideo displays[3]. Our display used the combination of an acousto-optic modulator (AOM) and a series of lenses and scanning mirrors to assemble a small full-color 3-D holographic image at video frame rates. A partial schematic is shown in Figure 2. A general description follows, and a detailed description can be found in the reference by St. Hilaire et al.[3] It is important to note that the holographic rendering method described in this paper is not specific to one display. By incorporating the proper physical parameters, wavelengths and sample size, a hologram generated using this method can be viewed using other holographic displays.

In our holovideo display, as each line (*hololine*) of the fringe pattern was read out of a high-speed framebuffer, it passed through a radio-frequency (RF) signal processing circuit and into the AOM. At any instant, as one line of the holographic pattern traversed the aperture of the AOM in the form of an acoustic wave (at a speed of 617 m/s), a portion equal to roughly 2000 samples modulated the phase of the wavefront of laser light that passed through the AOM. Two lenses reconstructed the diffracted light at a plane in front of the viewer. By reflecting the light off of a synchronized horizontally scanning mirror, the apparent motion of the holographic pattern was cancelled. The scanning motion of the mirror also acted to angularly multiplex the image of the acoustic wave. It extended the apparent width of the imaged holographic pattern to 32768 samples, each sample representing a physical spacing of 1.0 micron.

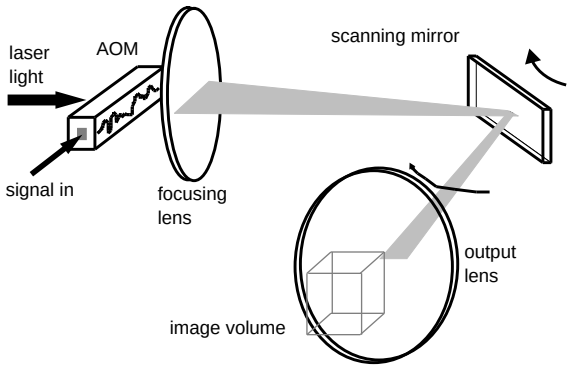

Figure 2: Partial schematic diagram of our holovideo display, shown from roughly the viewer's perspective. The scanning mirror angularly multiplexed the image of the modulated light. A vertical scanning mirror (not shown) positioned each hololine vertically. Electronic control circuits synchronized the scanners to the incoming holo-

The viewer saw a real 3-D image located just in front of the output lens of the system. The image occupied a volume that was 40 mm wide, 35 mm high and 50 mm deep. The size of the viewing zone - i.e., the range of eye locations from which the viewer can see the image - was 16 degrees horizontal. (Both viewing zone and image dimensions were minimized in order to reduce the required bandwidth of the display system.) The viewer experienced the depth cue of horizontal motion parallax. This was a horizontal-parallax-only (HPO) image. Vertical parallax was sacrificed to simplify the display. This restriction does not limit the display's usefulness in most applications. Because the holographic image possessed no vertical parallax, there was no need for diffraction in the vertical dimension. The vertical resolution of 64 lines over 35 mm was equivalent to that of a common 2-D display.

Because our display was a full-color system, the hologram was displayed as a vertical stack of 64 three-color hololines. The display modulated three parallel channels, one for each of the red, green and blue primary colors. The AOM had 3 separate but parallel channels, each modulating a separate beam of incident laser light (633 nm red, 514 nm green, 476 nm blue). Three separate fringe patterns were simultaneously read out of the 3-channel framebuffer.

To be displayed on our holovideo system, a fringe pattern must adhere to three specifications. First, for each of three color separations, the size of the fringe pattern must be 64 hololines of 32768 samples. Second, the sample size is 1.0 micron. Third, a separate fringe pattern must be computed at each of three wavelengths: 633 nm for red, 514 nm for green, and 476 nm for blue.

2.4 Stereograms

A stereogram is a type of hologram that is composed of a series of discrete 2-D perspective views of the object scene[8][9]. A stereogram has two essential characteristics: it has discretized parallax; and it creates an image at a single plane, the *image plane*. Parallax discretization simplifies fringe computation (and display). Computation is much faster than in the general case of a non-stereogram image. Provided that the number of discrete views is high enough, parallax appears to change smoothly, and the image appears to be 3-D.

An HPO stereogram produces a view-dependent image which presents in each horizontally displaced direction the corresponding perspective view of the object scene. The vertical perspective remains unchanged from any location in the viewing zone. In an image-plane stereogram, the image plane coincides with the physical plane of the hologram. As a result, the contribution to the object wavefront from each view-image pixel does not overlap with contributions from adjacent pixels. This reduces computational complexity. Specifically, discrete samples of the fringe pattern coinciding with a particular pixel are computed using only the view-dependent values of that pixel.

In a stereogram, the viewing zone is treated as an array of discretized subzones. An eye of the viewer, when moving from side to side, sees one discrete view of the scene after another. To compute a stereogram, a sequence of view images is rendered using a different camera geometry for each view image. These images are combined to calculate fringes for display. The only disadvantage of a stereogram is that it may not provide the depth cue of ocular accommodation. Ocular accommodation, however, is generally less important than binocular disparity, motion parallax and occlusion. A stereogram provides all other depth cues, and produces a realistic 3-D image.

The obvious advantage of a stereogram is that conventional computer graphics hardware can be used to render the sequence of view images. This makes pre-existing models and rendering methods applicable to holography.

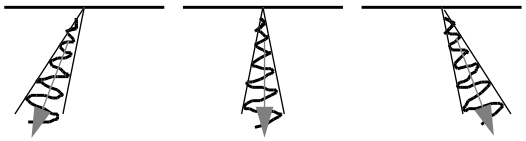

Figure 3a: Three stereogram components. (Top view.) Light is diffracted from each in a specific direction.

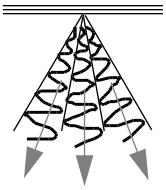

Figure 3b: The composite hologram.

3. METHOD FOR HOLOGRAM RENDERING

The fast computation of a stereogram involved two parts: rendering different view images, and using them to create holographic fringes. Beginning with a finite number of views (in this case eight), our goal was to construct an HPO image-plane stereogram that showed the correctly rendered view image for each viewing angle.

The development of a computational approach began by considering how light should be diffracted (or scattered) by the fringes. Consider an image displayed on a CRT: the light from the 2-D image scatters over the entire (often wide) viewing angle. In contrast, a stereogram must diffract light selectively. A stereogram causes a particular view image to be visible only from a particular part of the viewing zone. Figure 3a shows light diffracting over only a part of the viewing zone for three different views. In Figure 3b, the composite holographic fringe pattern in a single pixel region diffracts specific amounts of light in three different viewing directions. The amount of light corresponds to the pixel brightness values at this particular pixel location for each view image.

This section describes each step of the computational algorithm that we implemented on a graphics workstation. We began by precomputing a set of basis fringes and storing them in memory. As described in the following subsection, each basis fringe controlled the specific directional behavior of diffracted light. The next three subsections describe the three steps used to compute the actual fringe pattern: rendering a set of views (using conventional computer graphics rendering methods), generating a component stereogram fringe pattern from each view using an array of precomputed basis fringes, and compositing these separate stereogram components into the final fringe pattern. The final subsection is a description of the implementation specifics of our research.

3.1 Precomputing the Set of Basis Fringes

The arrays of precomputed basis fringes were designed to diffract light in specific directions. One basis fringe was the holographic pattern that caused light to diffract from a particular view-image pixel location to a particular viewing subzone. An array of basis fringes - one for each view-image pixel - was computed for each viewing subzone. This entire set of basis fringe arrays - one for each viewing subzone - was computed for each of the three (primary color) wavelengths used to create the full-color holographic image.

The set of basis fringes can be computed in several ways. We used an iterative numerical approach to solve the common integral expression that describes the diffraction of optical wavefronts. We began by specifying the direction that light must be diffracted by

each basis fringe. For an image-plane stereogram, each basis fringe was characterized by the diffraction from one view-image pixel region to one viewing subzone, as illustrated in Figure 4. The subzones, each of angular width Δ, divided the viewing zone into non-overlapping pieces across the horizontal range of the viewing zone. The horizontal parallax resolution was equal to the number of subzones. Each pixel region of width w coincided physically with a single rendered view-image pixel location. To make a view image visible from one of the viewing subzones, the basis fringe within each pixel region must be modulated by the pixel value of the rendered view image.

In the analysis that follows, we denoted the modulated light in the plane of the image (and hologram) as $u(x)$ in complex scalar notation. The light at the viewing zone distance z was represented by the complex scalar $v(\theta)$ expressed as a function of viewing angle θ. We defined each basis fringe through a set of constraints on the magnitudes of $u(x)$ and $v(\theta)$. Then, by relating related $u(x)$ and $v(\theta)$ through the law of diffraction, we calculated each basis fringe by numerically solving for the phase of $u(x)$.

For a single view-image pixel, the constraint on $|u(x)|$ was straightforward: within the pixel region, the diffracted intensity should be constant. Therefore, $|u(x)|$ must be virtually constant. For a single view, the modulated light $u(x)$ should diffract light only to a particular viewing subzone to avoid leakage or ghosting. Within the viewing subzone, $v(\theta)$ should be constant. Therefore, the constraint on $v(\theta)$ was that

$|v(\theta)| = 1$ for $|\theta - \theta_v| < \Delta/2$,

$|v(\theta)| = 0$ elsewhere,

where θ_v is the angle (from normal) connecting the center of a pixel region to the center of the viewing subzone. Finally, the phase of $v(\theta)$ was invisible and therefore was left unconstrained. When computed to satisfy these constraints, a basis fringe diffracts a unity brightness from a given pixel region to a specific viewing subzone.

The values of $u(x)$ and of $v(\theta)$ were related through the laws of optical diffraction[10].We invoked the far-field (Fraunhofer) diffraction approximation (expressed in complex time-harmonic notation):

$$v(\theta) = \frac{e^{ikz(1+\theta^2/2)}}{ikz} \int_{-w/2}^{w/2} u(x)\, e^{-ikx\theta}\, dx \qquad (1)$$

where $k \equiv 2\pi/\lambda$ and λ is the free-space wavelength of the light. The far-field approximation, valid for $z \gg w^2/\lambda$, was appropriate for the dimensions that we used: w = 0.250 mm and z=600 mm.

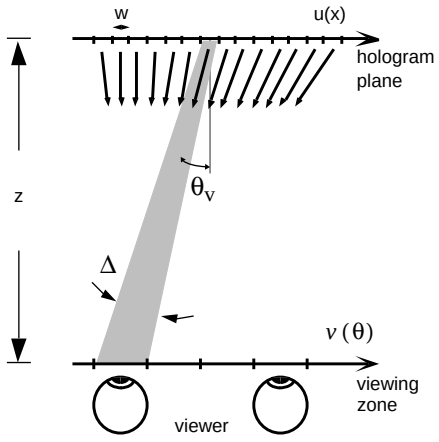

Figure 4: Diffraction caused by an array of basis fringes. Light diffracts from all pixel regions to a particular viewing subzone.

The integral expression in Equation 1 is essentially a Fourier transform integral. The wavefront at the viewing zone, $v(\theta)$, is a Fourier transform of $u(x)$, with additional leading terms to account for phase curvature and for power conservation. Equation 1 reveals that while the constraints on $u(x)$ were spatial, the constraints on $v(\theta)$ determined the spectrum of $u(x)$. An appropriately band--limited $u(x)$ resulted in an angularly limited $v(\theta)$ that satisfies the constraints on $|v(\theta)|$.

For one pixel location and one viewing subzone, the unconstrained phase of $u(x)$ was numerically computed using an iterative constraint approach[11] that uses both the forward and the inverse Fourier transforms:

1. Generate a random fringe pattern of width w.
2. Apply the (spatial) constraints on $u(x)$.
3. Transform into the spatial frequency domain.
4. Apply the (spectral) constraints on $v(\theta)$.
5. Inverse transform back to the spatial domain.
6. Iterate starting at step 2.

The iteration continued until $u(x)$ converged. Satisfactory convergence resulted after typically 10 to 20 iterations. As in work by Lucente[5], the real part of $u(x)$ was used as the basis fringe. A selection of precomputed basis fringes is shown in Figure 5.

We computed the array of basis fringe patterns for each viewing subzone in the same way, with each basis fringe occupying different view-image pixel locations in the image plane. The array for each viewing subzone was computed using a unique viewing direction θ_v. Three different basis sets were computed - one for each color - at the correct red, green or blue wavelengths (633 nm, 514 nm, and 476 nm). These basis sets were stored in memory and used during actual hologram computation.

3.2 Generating the View Images

The first step during the actual computation of the CGH pattern was to render a series of perspective view images from left to right. Each discrete camera location had a unique angle with respect to the scene being rendered. The final hologram diffracted light in the directions corresponding to these camera view-angles. This section describes the recentering shear-camera geometry that we used to render the view images, as well as the bounding and discretization of the image volume.

To render each view image, the camera geometry was changed by moving the camera from left to right at a constant spacing along a straight line. (See Figure 6.) When lateral motion is the only

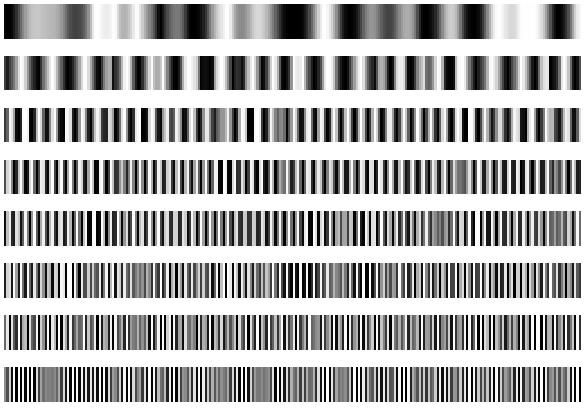

Figure 5: A selection of basis fringes that correspond to the center view-image pixel for each of the eight viewing subzones. Each is 256 samples long. The top fringe represents a mean diffraction angle of 1.8 degrees in red light, corresponding to the left-most viewing subzone. The bottom fringe represents a mean diffraction angle of 14.7 degrees in red light, corresponding to the right-most viewing subzone.

change in camera geometry, the object scene tends to walk out of the frame and becomes invisible in the far left or far right views. Therefore it was necessary to recenter the object scene in the frame of the camera as the camera moved from left to right.

The correct way to recenter the object scene for each camera location was to move the view window, as shown in Figure 6. For the center view (position A in Figure 6), where the camera was directly in front of the scene, the camera geometry was simple. The view normal faced the scene, and the view window was centered with respect to the view normal. For any other view, the view window was off center. As the camera was moved off center (Figure 6, position B), the view window was moved off center in the opposite direction. This is called a shear camera geometry. Note that the view normal does not change. This has the effect of keeping the objects in the center of the frame and in holography is called a recentering camera. If the recentering is done correctly, a point (in the 3-D object scene) that lies on the image plane renders to the same pixel location in each rendered view image.

When rendering the view images, we restricted the scene volume to match the size of the image volume achievable in our holovideo display. If an object crosses the frame, clipping occurs. This clipping (called image vignetting) is a window violation, which often hinders the feeling of depth. Therefore, we chose to confine objects to lie inside the restricted volume so that objects did not cross the frame boundaries.

We chose the resolution of the rendered view images to be 128 by 64. In the horizontal dimension, the CGH was capable of imaging to resolutions that were beyond the typical acuity of the human visual system. We chose a horizontal resolution of 128 to give an image pixel width of 0.25 mm, or roughly the smallest feature size that a human viewer can discern at typical viewing distances. We matched the vertical resolution of the view images to that of our holovideo display (64 lines).

3.3 Computing the Component for Each View

We chose to use the image-plane stereogram geometry because it ensured that a view-image pixel contributed only to the part of the fringe pattern that coincided with that pixel region. A given precomputed basis fringe was multiplied by its respective pixel value at each sample in the pixel region. When sent to the display, the resulting fringe pattern diffracted the correct amount of light to the correct viewing subzone. For one view, the fringe pattern in this pixel region was not a function of any other pixel values. In other words, in a given stereogram view component, the fringe contributions from each pixel did not overlap. Therefore, because of the image-plane stereogram geometry, each of the pixel values for one view image independently modulated each basis fringe in one array.

We designed the holovideo rendering algorithm to perform the multiplication of basis fringes by pixel values using a single texture-mapping instruction, eliminating the need to multiply the basis fringe for each pixel one by one. The array of precomputed basis fringes for a particular view was multiplied by the appropriately resized pixel map of the rendered view image. The result, a single stereogram component, diffracted light to form the image for a given viewing subzone.

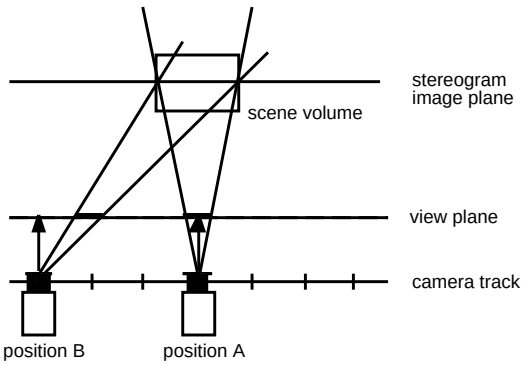

Figure 6: Camera geometry for rendering view images.

3.4 Compositing the View Components

After each stereogram component was generated, it was composited into the final CGH pattern. The approach that we use allowed for the linear superposition of computed fringe patterns[5]. Therefore, the components were simply summed. An array of basis fringes patterns were modulated by an array of weighting values (a view image) and accumulated in the accumulation buffer. The process is illustrated in Figure 7. Below is a piece of pseudocode that describes the process.

```
Set the accumulation buffer to zero.
For each view i {
      Load ith array of basis fringes.
      Generate ith view image.
      Modulate fringes with the view image.
      Add result to accumulation buffer.
}
Display sum contained in accumulation buffer.
```

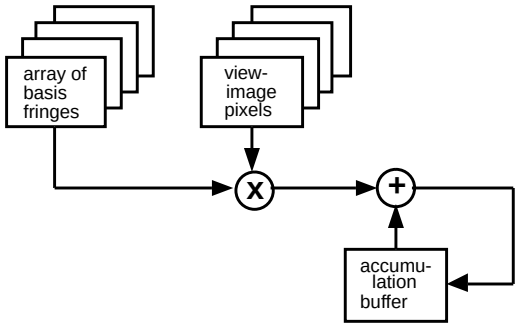

Figure 7: The modulate and sum process of rendering.

3.5 Implementation Specifics

Our algorithm for rendering holographic fringe patterns was implemented initially on an SGI Skywriter with a VGXT graphics subsystem and more recently on an Onyx with a RealityEngine2 (RE2). In both cases, the graphics hardware subsystems had two enabling features: an integrated accumulation buffer and a reconfigurable framebuffer. The accumulation buffer was an essential part of the fast algorithm. The reconfigurability of the framebuffer enabled the graphics subsystem to drive our holovideo display directly, but was not essential to the rendering algorithm.

We reconfigured the graphics framebuffers to provide three analog output signals in a format that was suited to our holovideo display. The active data array was set to 2048×1024 in each of the three 8-bit buffers. (The display system unfolds the 2048×1024 video signal to a 32768×64 pattern.) We adjusted the clock rate of the three 2-MB buffers to generate signals at a rate of 110 MB/s/channel. We also adjusted blanking intervals to minimize the image noise caused by signal blanking that occurred within each hololine. We could not eliminate blanking, but the length of the blanking intervals was reduced to less than 4 percent of the active data line.

The following is a description of the algorithm illustrated in Figure 7 as we implemented it on the SGI graphics subsystems. The algorithm had three steps: render the view images, modulate the arrays of basis fringes using the rendered view images, and sum the results in the accumulation buffer. The view images were rendered

using straightforward GL library calls. The camera geometry was adjusted for each view using the method described earlier in the section 3.2 Generating the View Images. Each of the 128×64-pixel view images was stored in memory. The framebuffer was loaded with the array of basis fringes for a particular view. The 32768×64 fringe pattern was folded into the 2048×1024 framebuffer. The 128×64 view image was mapped over the entire framebuffer and used to modulate the basis fringes. The graphics subsystem used the view image as a texture map on the surface of a polygon that covered the entire framebuffer. The "blending" feature of the graphics subsystem was set to multiply the current value in the framebuffer with the pixel value from the texture map. Effectively, the basis fringes were weighted by the view-image pixels. The resulting stereogram component was added to the accumulation buffer and the process was repeated for each view. The final holographic pattern was moved from the accumulation buffer to the framebuffer for viewing. All of these steps were performed in parallel for all three colors, generating a total of 6 MB of CGH.

Figure 8: Two photographs of holographic images generated on our real-time holographic display. Left: polygonal model of head with texture-mapped face. Right: polygonal model of a car. View images for both objects were rendered in hardware at a resolution of 128×64.

4. RESULTS

Holographic fringes were computed on either SGI platform and used to generate 3-D images on our real-time holographic display. Figures 8, 9, and 10 are photographs of typical images, each generated using eight full-color view images at a resolution of 128 by 64. Figure 8 shows a human head and a car, both generated from polygonal models. Figure 9 shows a simple image of three cut cubes, photographed from different locations in the viewing zone. Figure 10 shows a time sequence of an object generated using an interactive modeling system.

The horizontal image resolution of 128 was sufficient to produce the appearance of continuous surfaces. Some noticeable image artifact was due to the 4-percent blanking, which produced the faint vertical stripes visible in Figures 8, 9, and 10. The vertical resolution, limited by the display system to 64, is also evident in these photographs. (We remark that photographing holographic images poses a challenge, and the quality of these illustrations suffers as a result.) As discussed in the Future Work section, image quality can be improved using a new approach that is more precise than the stereogram approach.

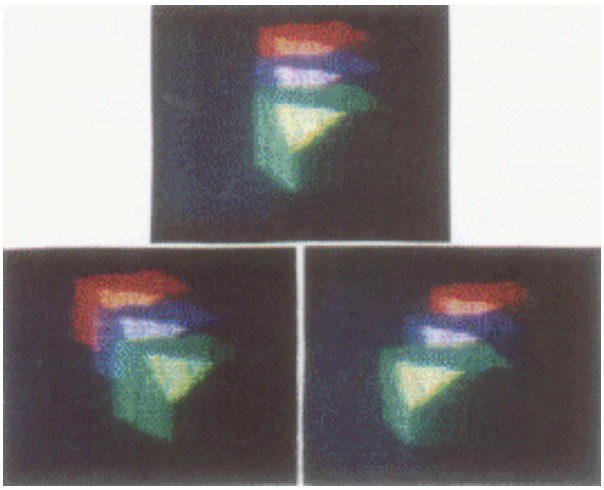

Figure 9: An image, composed of three cut cubes located at different depths, photographed from different locations in the viewing zone. Top: center. Left: left. Right: right.

Image depth and color were very good. The images gave a good sensation of depth, though the limited viewing zone and lateral dimensions weakened the impact. Objects appeared to occupy a depth of over 50 mm. Color was superb. The three wavelengths used as primary colors spanned a very broad color space that was much larger than that of a typical high-quality 2-D display.

Computation time was 2 seconds (for the 6-MB fringe pattern) using the Onyx/RE2 platform (two-processor Onyx, 4-raster-manager RE2) and 5 seconds for the Skywriter/VGXT platform. (For comparison, the same hologram required over 8 minutes when implemented using a standard method on the SGI Skywriter.) Computation time comprised two parts: view-image rendering and fringe computation. The time spent rendering the view images was less than 10 percent of total hologram computation time. Fringe computation was O(v*n), where v was the number of views (parallax resolution) and n was the total number of samples in the fringe pattern. The sample count n was essentially a measure of image volume. Fringe computation time was independent of view-image resolution, which only affected the rendering time. The precomputation of the basis fringes did not contribute to the computation time per CGH frame. The precomputed basis fringes were computed only once and stored in memory.

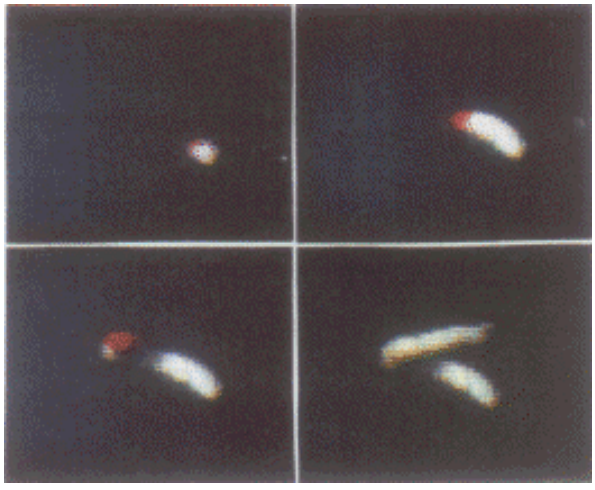

Figure 10: A sequence of images generated using the interactive "sculpt" system, showing the potential of holographic rendering and display for interactive applications.

Transfer of the 6-MB fringe pattern from the accumulation buffer to the framebuffer required negligible time since these two buffers resided in the same hardware. Therefore, our implementation on either graphics subsystem avoided a common data transfer

bottleneck. For comparison, in past work by Lucente[5] using a data-parallel supercomputer, 0.4 s was spent transferring the 6-MB fringe pattern from the 16 K processors to the framebuffer.

The benchmark computation time of 2 s for our rendering method was fast enough to provide interactivity to a patient viewer. We anticipate that computation speeds will continue to increase as computational power increases and further computational efficiencies are obtained. In fact, the work using the Skywriter/VGXT platform was performed two years ago. Our recent upgrade to the Onyx/RE2 platform more than doubled speed, and new graphics hardware should continue to offer faster implementations of our hologram rendering algorithm.

5. FUTURE WORK

The holovideo rendering algorithm presented here is applicable not only to image-plane stereograms but to all types of synthetic holograms. Research is underway to adapt this rendering method to the computation of 3-D volume (non-stereogram) images using the newly developed method of "diffraction-specific" fringe computation that also allows for holographic bandwidth compression[12]. Such a 3-D volume image possesses more depth and tangibility because it produces an actual 3-D real image and provides the depth cue of ocular accommodation. Finally, the computation of full-parallax holographic images is possible, though no full-parallax holographic 3-D display yet exists.

We have generated holograms from live scenes. The simplest approach is to use an array of horizontally displaced video cameras to provide view images in our hologram rendering algorithm. Faster frame rates will enable the production of real-time holographic live video of actual scenes.

6. CONCLUSION

Three-dimensional displays are superior to 2-D displays in many applications of computer graphics. Holographic displays produce images with the greatest number of depth cues and therefore the greatest sensation of 3-D. With their ability to produce images in real time with high resolutions and full color, holographic displays promise the highest degree of image realism.

Our research has demonstrated that existing computer graphics workstation technology can be used to generate holographic fringe patterns at interactive rates for the real-time display of 3-D images. A workstation such as the SGI Onyx coupled with a specialized rendering engine is capable of driving a holovideo display directly. As holographic displays become larger and more practical, the computational method presented here can be scaled up to produce holographic fringes for images that occupy a larger volume and provide a larger viewing zone.

We have presented only a single case in a wide range of hologram types, computation techniques, display architectures, applications, and hardware and software support. Nevertheless, the importance of this work is that it breaks the deadlock that has impeded the development of methods for both the computing and the displaying of 3-D holographic images. We have reported the use of an existing computer graphics system to generate a high-bandwidth holographic signal for a holovideo display. Our research shows that a computer graphics workstation provides sufficient computational support for the development of real-time electronic holography. Furthermore, our work heralds the marriage of the two fields of computer graphics and electronic holography.

ACKNOWLEDGMENTS

We thank many: Stephen A. Benton for guidance in the holovideo project; Michael Halle for contributions to implementation issues; David Zeltzer; Pierre St. Hilaire and John D. Sutter for help with the holovideo display optics; Scott Pritchett, John Hallesy, Gregory Eitzmann, and Jim Foran at Silicon Graphics, Inc. for help with framebuffer reconfiguration; and Katherine Wrean for help in preparing this document.

Components of this research have been sponsored by Honda Research and Development Co., Ltd.; NEC Corporation; International Business Machines Corp.; the Advanced Research Projects Agency (ARPA) through the Naval Ordnance Station, Indian Head, Maryland (under contract No. N00174-91-C0117); the Television of Tomorrow research consortium of the Media Laboratory, MIT; Apple Computer Inc.; and NHK (Japanese Broadcasting Corp.).

Silicon Graphics, Inc. (Mountain View, CA, USA) are the creators of "Skywriter" and "Onyx" computer graphics workstations, the "VGXT" and "RealityEngine" graphics subsystems, and GL graphics software.

REFERENCES

[1] Michael McKenna and David Zeltzer. Three Dimensional Visual Display Systems For Virtual Environments. *Presence: Teleoperators and Virtual Environments.* Vol. 1 #4, 1992, pp. 421-458.

[2] P. Hariharan. *Optical Holography.* Cambridge: Cambridge University Press. 1984.

[3] Pierre St. Hilaire, Stephen A. Benton and Mark Lucente. Synthetic Aperture Holography: A Novel Approach To Three Dimensional Displays. *Journal of the Optical Society of America A*, Vol. 9, #11, Nov. 1992, pp. 1969 - 1977.

[4] William J. Dallas. Computer-Generated Holograms. Chapter 6 in *The Computer in Optical Research, Topics in Applied Physics*, Vol. 41, ed. B.R. Frieden. Berlin: Springer-Verlag. 1980, pp. 291-366.

[5] Mark Lucente. Interactive Computation Of Holograms Using A Look-up Table. *Journal of Electronic Imaging*, Vol. 2, #1, Jan 1993, pp. 28-34. Also: Mark Lucente. Optimization of hologram computation for real-time display. in *SPIE Proc. #1667 Practical Holography VI*, 1667-04 (SPIE, Bellingham, WA, 1992), pp. 32-43.

[6] F. Mok, J. Diep, H.-K. Liu and D. Psaltis. Real-time Computer-generated Hologram By Means Of Liquid-Crystal Television Spatial Light Modulator. *Optics Letters*, Vol. 11 #11, Nov. 1986, pp. 748-750.

[7] S. Fukushima, T. Kurokawa and M. Ohno. Real-time Hologram Construction And Reconstruction Using A High-resolution Spatial Light Modulator. *Applied Physics Letters*, Vol. 58 #8, Aug. 1991, pp. 787-789.

[8] Stephen A. Benton. Survey of Holographic Stereograms. In *Processing and Display of Three-Dimensional Data. Proceedings of the SPIE* 367, 1983, pp. 15-19.

[9] Michael Halle. *The Generalized Holographic Stereogram*. Master's thesis, Massachusetts Institute of Technology, 1991.

[10] Joseph W. Goodman. *Introduction to Fourier Optics*. New York: McGraw-Hill. 1968.

[11] J. R. Fienup. Iterative Method Applied to Image Reconstruction And to Computer-Generated Holograms. *Optical Engineering*, Vol. 19 #3, May/June 1980, pp.297-305.

[12] Mark Lucente. *Diffraction-Specific Fringe Computation for Electro-Holography.* Ph.D. Thesis, Massachusetts Institute of Technology, 1994.