Reconfigurable image projection holograms

Wendy Plesniak
Michael Halle
Surgical Planning Lab
Department of Radiology
Brigham and Women's Hospital
75 Francis Street, L1-050
Boston, Massachusetts 02115
E-mail: wjp@bwh.harvard.edu

V. Michael Bove Jr., FELLOW SPIE
James Barabas
MIT Media Laboratory
20 Ames Street
Cambridge, Massachusetts 02139

Ravikanth Pappu
ThingMagic, Inc.
One Broadway, 5th Floor
Cambridge, Massachusetts 02142

Abstract. We introduce reconfigurable image projection (RIP) holograms and a method for computing RIP holograms of three-dimensional (3-D) scenes. RIP holograms project one or more series of parallax views of a 3-D scene through one or more holographically reconstructed projection surfaces. Projection surfaces are defined at locations at which the hologram reconstructs a variable number of real or virtual images, called holographic primitives, which collectively compose the surface and constitute exit pupils for the view pixel information. RIP holograms are efficiently assembled by combining a sweep of 2-D parallax views of a scene with instances of one or more precomputed diffractive elements, which are permitted to overlap on the hologram, and which reconstruct the holographic primitives. The technique improves on the image quality of conventional stereograms while affording similar efficient computation: it incorporates realistic computer graphic rendering or high-quality optical capture of a scene, it eliminates some artifacts often present in conventional computed stereograms, and its basic multiply-and-accumulate operations are suitable for hardware implementation. The RIP approach offers flexible tuning of capture and projection together, according to the sampling requirements of the scene and the constraints of a given display architecture.
[DOI: 10.1117/1.2390678]

Subject terms: electroholography; computer-generated hologram; holographic stereogram.

Paper 050817R received Oct. 11, 2005; revised manuscript received Apr. 10, 2006; accepted for publication Apr. 19, 2006; published online Nov. 29, 2006.

1 Computed Holographic Stereograms

A great deal of information is required to describe a three-dimensional (3-D) scene over a reasonable angle of view, and current electroholographic displays provide limited computation, communication, and modulation bandwidth in which to encapsulate, transmit, and reconstruct that information. In order to generate holograms that can be rapidly updated on these systems, holographic stereograms may be computed as an alternative to physically based interference modeling. Computed holographic stereograms encode a finite set of two-dimensional (2-D) parallax views within an interference pattern. The interference pattern is designed to spatially multiplex these views when illuminated, according to some prescribed reconstruction geometry. Within the viewzone of some electroholographic display device, a static or moving viewer has appropriate and disparate parallax views delivered to each eye and thus receives the impression of scene depth and parallax.

Computed holographic stereograms usually sample and display scene parallax more coarsely than traditional interference-simulated holograms would, and the same interference patterns, or *fringes*, can be used to encode any set of appropriately captured 2-D parallax views. These characteristics imply several notable advantages. First, capturing a scene optically or with computer graphics techniques and encoding it with a precomputed pattern is generally more convenient and much faster than performing more elaborate interference modeling on the scene geometry. Second, with the benefit of modern computer graphic rendering and photographic techniques, carefully or subtly illuminated scenes replete with textures, reflections, and material surface properties are much easier to incorporate. Third, coarsely sampling scene parallax helps to reduce the amount of information in each hologram, and both rendering hardware and image compression techniques can further reduce computation time and overall bandwidth requirements. Finally, and perhaps best of all, dissociating fundamental diffractive behavior and image content allows a single set of diffractive primitives to be precomputed; to update a stereographic frame, only a set of parallax-view pixels need be generated and used to modulate the precomputed diffractive primitives.

Conventionally computed holographic stereograms also present many disadvantages. For instance, conventional stereograms, generated by tiling the hologram with abutting hologram elements (each a superposed set of modulated fringes[1]), have low and fixed angular and spatial resolution, and the discontinuous phase boundaries between abutting elements introduce visible artifacts into their diffracted field. Stereograms' fixed spectral resolution may undersample rapid angular variations, and their fixed spatial resolution may also fail to reconstruct high amplitude variations—both causing aliasing in the reconstructed spatial image. Conversely, stereograms generated with properly bandlimited capture may fail to exhibit the high reconstruction resolution at the heart of holography's great allure.

Further, without a lightness curve correction, the gray-

0091-3286/2006/$22.00 © 2006 SPIE

scale of the final reconstructed image will not match that of the input parallax views; thus subtly shaded parts of a scene may appear flat instead, absent the information about shape, curvature, or material properties that shading can richly convey. And finally, in horizontal parallax only (HPO) stereograms, captured or rendered parallax views are often projected astigmatically; yet view capture or generation is almost always accomplished with a conventional perspective projection in both directions. Such differences between view capture and projection lead to anamorphic distortion in the reconstructed image, an unacceptable result for applications where spatial accuracy is crucial.

The *reconfigurable image projection* (RIP)* technique allows many of these problems of conventional computed holographic stereograms to be addressed, while still encoding the scene using precomputed fringes and rendered 2-D parallax views. RIP holograms offer the ability to vary their spatial and angular resolution in order to satisfy scene sampling requirements, constrained by the properties of a given electroholographic display and the diffraction limit. RIP holograms, the computed analog of optically recorded *ultragrams*,[3] allow the shape and location of one or more *projection surfaces* (where a projection surface corresponds to the single "slit plane" in the holographic stereogram literature) to be specified. Whereas in conventional computed holographic stereograms the slit plane most often coincides with the hologram plane, a RIP hologram's projection surfaces are located off the hologram plane. As a consequence, instead of treating the hologram plane as a tiled set of diffractive apertures, RIP holograms allow the footprints of adjacent hologram elements to be windowed and overlapped on the hologram (although RIP holograms can be computed and assembled in the former fashion if a given display architecture necessitates it). These hologram elements collectively reconstruct a projection surface with *holographic primitives* that can have varying spatial and projective characteristics, as prescribed by the precomputed fringes that reconstruct them. In general, the RIP approach permits important capture and projection parameters to be adjusted in order to improve the quality of the light field perceived by an observer in the hologram viewzone. RIP holograms are produced efficiently; their reconstructed spatial images can have high spatial and angular resolution, and are vivid and realistic, with minimal distortion and diffracted artifacts.

In this paper, we begin with a brief review of how physically based computed holograms, conventional computed stereograms, and hybrid techniques encode and display scene parallax. We then present the hybrid RIP hologram approach and describe a simple embodiment: the method for computing a simple diffractive element and the method of assembling the hologram from modulated instances of it.

2 Designing Diffractive Behavior

In electroholography, computational techniques have largely followed one of three general approaches: physically based techniques (interference modeling of scenes populated by holographic primitives, producing computed patterns similar in nature to optical holograms); holographic stereogram techniques (holographic encoding of 2-D parallax views, producing computed patterns similar in nature to optical holographic stereograms); or diffraction-specific techniques, which can be viewed as a hybrid of the previous two. We have experimented with computing images for the MIT Mark II electroholographic display[4] using all of these techniques. The following sections describe basic background on each general approach and introduce RIP hologram computing, a hybrid technique.

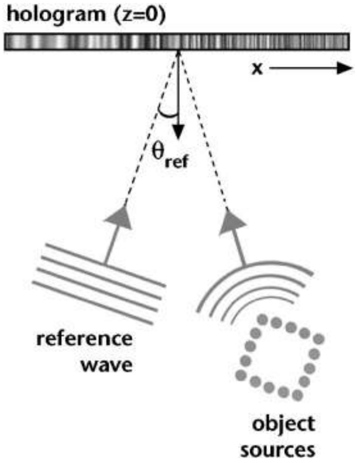

Fig. 1 Physical interference modeling for HPO holograms.

2.1 *Physically Based Interference Modeling*

In physically based techniques,[5,6] three-dimensional scene geometry is represented by a collection of primitives, such as spherical or cylindrical emitters, which populate the edges or surfaces of a model as shown in Fig. 1. Their collective interference with a reference wave is computed at some location in the combined field to generate a hologram of that scene.

Using this technique to compute holograms for HPO systems, the field radiated by a reference wave inclined at an angle θ_{ref} may be modeled as

$$E_{\mathrm{pl}} = E_{\mathrm{pl0}} \exp\left[-j\frac{2\pi}{\lambda}(q_x x + q_z z) - j\varphi_0\right], \quad (1)$$

where E_{pl0} is the wave amplitude; λ is the illumination wavelength; φ_0 is a random initial phase, assigned a uniform deviate between 0 and 2π; and

$$q_x = \hat{\mathbf{q}} \cdot \hat{\mathbf{x}} = \sin\theta_{\mathrm{ref}}, \quad q_z = \hat{\mathbf{q}} \cdot \hat{\mathbf{z}} = \cos\theta_{\mathrm{ref}}. \quad (2)$$

The field radiated by an individual spherical-emitting object source may be given as

$$E_{\mathrm{sph}} = E_{\mathrm{sph0}} \exp\left\{-j\frac{2\pi}{\lambda}[(x-x_0)^2 + (z-z_0)^2]^{1/2} - j\phi_0\right\} \quad (3)$$

with E_{sph0} as the wave amplitude, ϕ_0 as a random initial phase, and the location of the HPO spherical emitter given by

*In Ref. 2 we reported *reconfigurable image surface* (RIS) and *reconfigurable image plane* (RIP) holograms, both renamed in this work for sake of generality to *reconfigurable image projection* (RIP) holograms.

$$\overline{\mathbf{r}}_0 = (x_0, y, z_0). \tag{4}$$

These simplified models are derived from the field equations by ignoring both time dependence and the amplitude attenuation associated with propagation distance, and by assuming that all sources share an identical linear polarization.[6]

Rather than computing the composite field from the complex interference between a reference wave and a collection of object sources, the bipolar intensity method[7] may be used, which considers only the real part of the combined field. This method not only simplifies computation of the intensity pattern, but eliminates object self-interference and reference wave bias as well. A bipolar intensity hologram for a given reference wave and collection of object sources can be computed from

$$I(x,z) \approx 2\ \mathrm{Re}\{E^*_{\mathrm{ref}}E_{\mathrm{obj}}\} = 2\ \mathrm{Re}\left\{E^*_{\mathrm{pl}}\sum_i E_{\mathrm{sph}i}\right\}. \tag{5}$$

For the MIT Mark II electroholographic display, this intensity pattern is computed for each one-dimensional hologram line (hololine) by interfering the subset of object sources located within its yz plane with a collimated reference wave. The distance from a sample within a hololine to its ith contributing spherical source is given by

$$r_i(x) = |\mathbf{r} - \mathbf{r}_{0i}| = [(x - x_{0i})^2 + z_{0i}^2]^{1/2}. \tag{6}$$

Using expressions for plane and spherical waves as given in Eqs. (1) and (3), respectively, the interference pattern within any hololine expands to

$$\begin{aligned} I(x,z) &= 2\ \mathrm{Re}\left\{E_{\mathrm{pl}0}\exp[jkx\sin\theta_{\mathrm{ref}} + j\varphi_0]\right. \\ &\qquad \left.\times\sum_i E_{\mathrm{sph}0i}\exp[-jkr_i(x) - j\phi_{0i}]\right\} \\ &= 2E_{\mathrm{pl}0}\sum_i E_{\mathrm{sph}0i}\cos\{k[x\sin\theta_{\mathrm{ref}} - r_i(x)] + \psi_{0i}\}, \end{aligned} \tag{7}$$

where $k=2\pi/\lambda$, $z=0$ in the hologram plane, $\hat{\mathbf{q}}\cdot\hat{\mathbf{x}}=\sin\theta_{\mathrm{ref}}$, and ψ_0 describes the initial phases of the reference and i'th object waves bundled together. Finally, by setting $E_{\mathrm{pl}0}$ to eliminate any uniform bias, the final intensity pattern computed within each hololine is given by

$$I(x,z=0) \approx \sum_i E_{\mathrm{sph}0i}\cos\{k[x\sin\theta_{\mathrm{ref}} - r_i(x)] + \psi_{0i}\}, \tag{8}$$

and the complete set of hololines collectively contains the HPO interference pattern generated by the entire scene.

The images produced by interference modeling have high angular and spatial resolution and can include occlusion,[8] texture mapping, and smooth shading. The images can also be free of visible diffraction artifacts and are strikingly realistic in appearance. To display lines and surfaces that appear solid and continuous, model geometry can be populated at densities that approach the arcminute resolution limit optimal for human visual acuity[9] (within the limit diffraction imposes on the size of a single point).

However, while a hologram computed in this fashion exhibits high resolution and accurate reconstruction throughout the scene volume, its computation can be slow, and the resulting diffractive structure cannot be repurposed to project any other scene geometry. Rather, the full computation must be done afresh each time changes occur in the scene texture, shading, position, orientation, scale, or geometry. Algorithms have been described to make this computation more efficient, among them the use of table lookup,[7] difference calculations,[10] and incremental modification of the hologram.[11]

2.2 *Modeling Diffractive Projection for Computed Stereograms*

Computed holographic stereogram techniques offer a set of diffractive behaviors that can be used to multiplex 2-D computer-generated or optically captured parallax views of many scenes. Stereograms achieve their great economy by using precomputed *basis fringes*, each with spectral content capable of diffracting input light in a different direction and uniformly through some small angular extent. In general, a set of N_b basis fringes can be used to angularly multiplex a set of N_p parallax views in a variety of ways: for instance, each view may be projected in a different and collimated direction, or views may be projected according to a more specialized prescription.

In the parlance of the literature on optically recorded HPO holographic stereograms, it is instructive to describe computed holographic stereograms as having a *hologram plane* (the plane on which the hologram elements are written), a *slit plane* (the exit pupil plane through which view pixels are projected), a *focal plane* (the plane at which the pixel information is focused), and a *viewer plane* (the plane at which a human observer is located). Computer-generated holographic stereograms can arrange these four planes in a variety of formats, as illustrated in a top view in Fig. 2, where they are marked H, S, F, and V, respectively. However, in computed holography it is most common to collocate the hologram plane and the slit plane as shown in Fig. 2(c). Conventional computed holographic stereograms employ this arrangement, often using a number of parallax views equal to the number of basis fringes (N_b=N_p). In conventional stereograms, the focal plane is often colocated with the hologram plane. However, in hybrid HPO techniques, the horizontal and vertical focus planes may differ, introducing some degree of astigmatism in the final image.

2.2.1 *Conventional computer-generated holographic stereograms*

In HPO displays, the input parallax views and the final computed stereogram have their vertical resolution limited by the number of hololines, N_v, on the electroholographic display. In the horizontal dimension, however, conventional computed stereograms are composed of an array of tiled chunks, each of which corresponds to a slit on the hologram plane and contains the same number of samples as each basis fringe. Each chunk, or hologram element (often called a *hogel*[1]), is a linear combination of basis fringes, each weighted by an appropriate view pixel value. The hogel is designed to project a set of view pixels—each in a different direction—throughout the hogel's spatial extent. Resulting hogels are tiled on the hologram plane in an abutting fashion, and each is seen to "light up" with a different

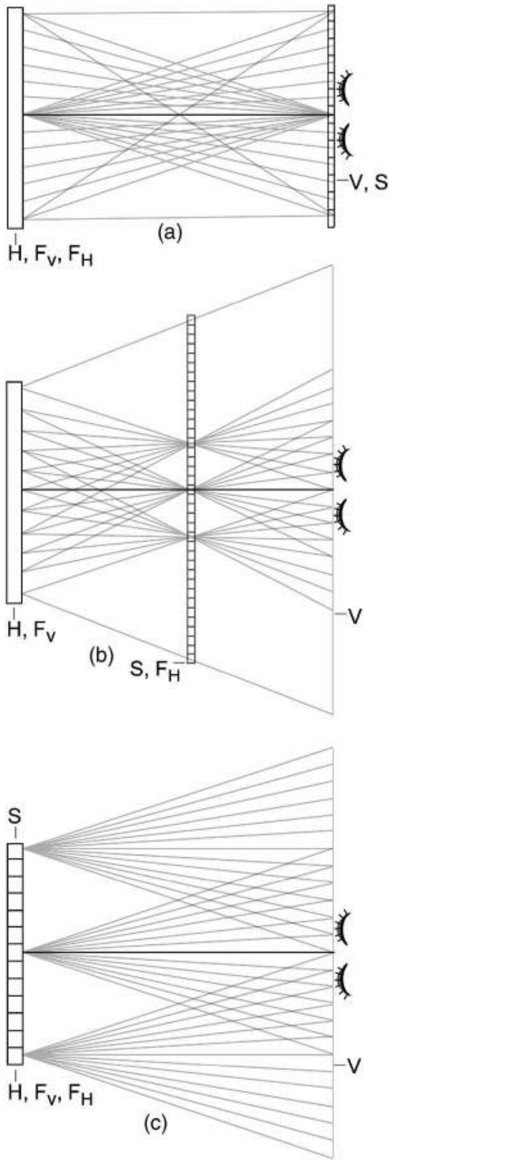

Fig. 2 Computer-generated holographic stereogram formats (top view): (a) slit plane colocated with the viewer plane; (b) slit plane offset from viewer and hologram planes; (c) slit plane colocated with the hologram plane.

pixel value when reconstructed and viewed from different angles within the viewzone.

To deliver the 2-D view pixels well, so that there are no gaps in the viewzone and the image appears equally bright throughout, the basis fringes are designed to satisfy a set of spatial and spectral constraints for a given reconstruction geometry. Their design details, as well as a nonlinear optimization approach used to generate them, are described in Ref. 12

2.2.2 *Ideal stereogram resolution for the human viewer*

To generate output optimized for a human viewer, an HPO computed hologram must adequately sample (both spatially and spectrally) a continuous one-dimensional interference pattern $I(x,z,f)$, where z is the plane of the hologram, x denotes a location on the hologram, and f is the spatial frequency represented there. For holographic stereograms, which combine parallax views with an interference pattern that projects them, both the input parallax views and their angular projection are sampled. These sampling rates are not independent; thus determining optimal spatial and angular sampling rates often involves some compromise between scene depth and resolution.[13]

When the slit plane is located on the hologram plane, the parallax-view sample spacing is equal to the hogel spacing w_h. Assuming that the most rapid perceptible amplitude variation within a parallax view is 1 arcmin and using a geometric analysis, the width of a view pixel (equal to the width w_h of a hogel in a conventional computed holographic stereogram) should be no larger than

$$w_h = p_{\mathrm{opt},x} = D_v \tan\left(\frac{1}{60}\ \mathrm{deg}\right) \tag{9}$$

in order to be optimal for the human viewer positioned a distance D_v from the hologram. Since the sampling frequency must be more than twice the highest spatial frequency in the parallax views, they must be bandlimited using a filter with cutoff frequency $f_{\mathrm{max,amplitude}}$ given by

$$f_{\mathrm{max,amplitude}} \leq \frac{1}{2w_h} \tag{10}$$

to satisfy the Nyquist limit. This bandlimiting[14,15] of the parallax views may be accomplished during view capture by using a real or synthetic camera aperture at least $2w_h$ in diameter.

Within any hogel, the spectral sampling increment Δf used to generate the set of basis fringes must be chosen to encode the varying diffracted angular output of that region. We assume that a viewer's ability to perceive angularly varying phenomena (like changing occlusion relationships or specular reflections) is limited by their pupil diameter d_p. To determine a Δf optimized for a human viewer, the grating equation may be used to relate the angle of diffracted output θ_{out} to the spatial frequency and the angle of incident illumination, θ_{ill}:

$$f\lambda = \sin\,\theta_{\mathrm{out}} - \sin\,\theta_{\mathrm{ill}}. \tag{11}$$

For an input illumination $\theta_{\mathrm{ill}}=0$ and a maximum angular deviation $\Delta\theta_{\mathrm{out}}$ given by

$$\Delta\theta_{\mathrm{out}} = \tan^{-1}\left(\frac{d_p}{D_v}\right), \tag{12}$$

a $\Delta f_{\mathrm{opt,spectral}}$ that will meet the spectral sampling requirements for a viewer positioned at D_v is given by

$$\Delta f_{\mathrm{opt,\ spectral}} = \frac{d_p}{\lambda}(d_p^2 + D_v^2)^{-1/2}. \tag{13}$$

The number of basis fringes required to achieve this angular resolution across the stereogram's field of view θ_{view} is $\theta_{\mathrm{view}}/\Delta\theta_{\mathrm{out}}$. If each basis fringe is to angularly multiplex unique parallax information into the viewzone, then $\Delta f_{\mathrm{opt,spectral}}$ also implicitly defines an upper bound for the intercamera spacing Δ_c; thus,

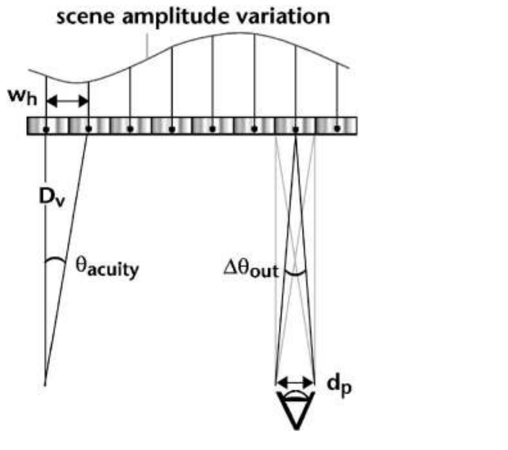

Fig. 3 Optimal spatial and spectral sampling for a human viewer.

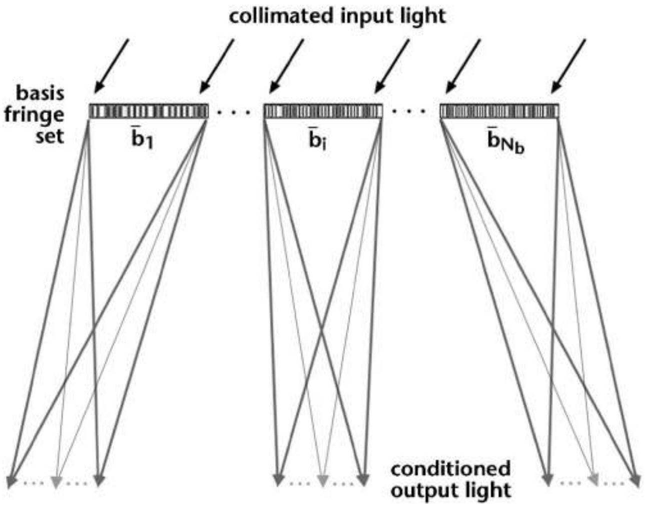

Fig. 4 Basis fringes used to project uniformly through a set of $N_b = N_p$ directions.

$$2w_h \leq \Delta_c \leq d_p. \tag{14}$$

These optimal spatial and spectral sampling requirements for the human viewer are illustrated in Fig. 3. It is against these human factors benchmarks that the modulation transfer function (MTF) of a computed stereogram should be evaluated.

2.2.3 *Computing conventional holographic stereograms*

In the common implementation in which $N_b = N_p$, each hogel, with aperture size and spectral content to satisfy the conditions in Sec. 2.2.2, is generated as a linear combination of pixel-weighted basis fringes: corresponding pixels from each of the N_p parallax views (viz., the first pixel from the first row in each image) form a vector called a *hogel vector*. Each element of a hogel vector multiplies a basis fringe in the precomputed set, and their sum is accumulated. The entire HPO hologram has N_v hologram lines containing a series of N_h hogels, each given by

$$\mathbf{h}_j = \sum_{i=1}^{N_p} p_{i,j}\mathbf{b}_i \quad (1 \leq j \leq N_v N_h), \tag{15}$$

where $p_{i,j}$ are the pixel weights expressed in the hogel vector and $\mathbf{b}_i$ are basis fringes.

For the MIT Mark II electroholographic system, with viewing distance $D_v = 600$ mm, view angle $\theta_{\text{view}} = 30$ deg, hologram width $w_H = 150$ mm, and illumination wavelength $\lambda = 633$ nm, and assuming an average pupil diameter $d_p = 3$ mm, an ideal hogel width w_h would be 0.1745 mm, and an ideal spectral sampling increment $\Delta f_{\max}$ would be about 7.9 mm^{-1}. These parameters imply that each hololine would be populated with an array of about 860 hogels, and that each hogel would be composed of pixels from more than 100 parallax views scaling that many basis fringes.

This prescription for conventional computed holographic stereograms is impractical for many reasons. First, the hogel aperture width is physically related to the diffracted resolution of its output; as noted previously, a compromise that minimizes hogel aperture size while maximizing angular resolution during the synthesis of basis fringes is required. Second, with the limited bit depth in a display's framebuffer, not enough dynamic range may exist to represent so many superposed basis fringes. Consequently, this kind of stereogram is likely to support lower spatial and angular resolution than is optimal for a human viewer. An analysis of the MTF of conventional computed holographic stereograms under coherent illumination is necessary to precisely determine their optimum slit width and angular resolution, but that analysis is beyond the scope of this paper.

In early work[12] implemented for the MIT Mark II electroholographic system, the number of basis fringes was equal to the number of parallax views as in Eq. (15), and each view was projected from the hologram plane in a different direction and in a collimated fashion. Hogels 0.59 mm wide and $N_b = N_p = 32$ basis fringes were used, as illustrated in Fig. 4, each with $B = 1024$ samples. A total of N_p monochromatic parallax views were generated as input to stereogram computation, each of dimension $w_i = 256$ by $h_i = 144$ pixels. Hogel vectors, formed from these parallax views, and basis fringes were used to assemble the 256 × 144 hogels that composed the computed holographic stereogram.

This work produced images with landmark speed from a wide a variety of photographic and computer-rendered input. However, when a scene's spectrum was not properly bandlimited during capture, the coarse spatial and angular sampling caused visible aliasing artifacts in the reconstruction. By comparison, this technique generated images with noticeably lower resolution than those produced by interference modeling.

These images exhibited other shortcomings as well. First, phase discontinuities at the hard boundaries between adjacent hogels introduced diffracted artifacts in the reconstructed light field, which manifest themselves as distracting dark vertical banding. And second, given the astigmatic optical projection of the component parallax views and the HPO nature of the display, correctly captured parallax views would employ an orthographic projection in the horizontal direction and a perspective projection in the vertical direction to match the hologram viewing geometry (Fig. 5).

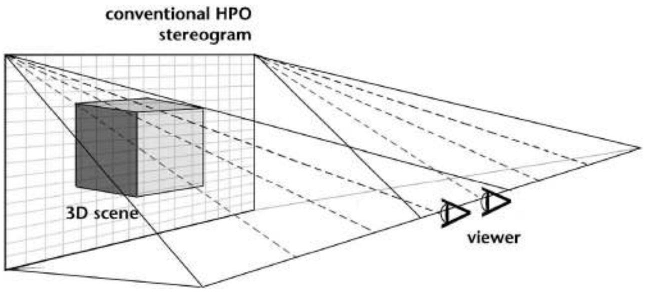

Fig. 5 Conventional HPO stereogram's astigmatic projection.

However, standard perspective rendering is usually employed rather than this unconventional capture geometry, resulting in output that can appear hyperstereoscopic and anamorphically distorted.

Nonetheless, the synergy between computer graphics rendering, digital capture techniques, and holographic stereogram computing remains potentially great. The basic multiply-and-accumulate operations at the heart of stereogram computation are fast and adaptable to hardware implementation[16]—suggesting that in principle, a platform for generating stereograms can be efficient enough for use in real-time applications.

2.3 *Diffraction-Specific Computing: A Hybrid Form*

Computing conventional holographic stereograms can be considered a special case of a general approach called *diffraction-specific computing*.[12] This approach can be applied to generate a range of computed holograms, varying from conventional holographic stereograms to physically based interference-modeled holograms, and may thus be regarded as a hybrid approach.

Diffraction-specific computing proceeds by dividing the hologram surface into contiguous hologram elements, also called hogels, and assembles the hogels by combining precomputed sets of basis fringes modulated by the amplitudes contained in hogel vectors as previously described. However, diffraction-specific computing also uses the (x,y,z) locations of object *elements* within a scene as well as factors that scale the amplitude contributions from hogel vector elements.

The technique begins by processing a given scene to determine the information to be stored in the diffraction table, and the pixel information that will be used to generate hogel vectors. Then for each object element represented in the table, all hogels that contribute to reconstructing it are determined, as well as which basis fringe(s) within that hogel should be involved in its projection, and which corresponding hogel vector components should scale those basis vectors.

Each hogel is assembled by accumulating the basis fringes that have been prescribed for it, and each basis fringe is multiplied by an appropriate scaled hogel vector element. Depending on the particular basis fringes used and the object elements represented in the diffraction table, the light field projected by the hologram may range from being similar to one projected by a conventional holographic stereogram, to being similar to one projected by a physically based interference-modeled hologram. Finally, the hogels are arrayed in the hologram.

Notably, diffraction-specific techniques as described in the literature construct the hologram plane using abutting hogels. While this tiled arrangement of hologram elements can prove useful for some parallelized computational pipelines and display architectures,[17,18] it is not strictly necessary unless the slit plane is located on the hologram plane. When the slits are located elsewhere in space, abutting hogels are in fact a limiting construct. RIP hologram computing, another hybrid approach, allows individual diffractive elements to be windowed to minimize diffraction artifacts at their boundaries, modulated by view pixels, and overlapped on the hologram plane. The superposed hogels collectively produce a slit plane or more complicated projection surfaces in space, off the hologram plane. This new approach is highly flexible and can offer many advantages: it combines the tremendous speed, efficiency, and flexibility of stereogram computing with image quality that approaches the metric accuracy and realism of interference-modeled holograms.

3 RIP Hologram Computing

3.1 *RIP Holograms: A Flexible Hybrid Form*

RIP holograms project one or more series of parallax views of a 3-D scene through one or more holographically reconstructed projection (slit) surfaces. Projection surfaces are populated by a collection of holographic primitives reconstructed at some (x,y,z)-distance offset from the hologram surface. Each holographic primitive behaves as a kind of projector that can relay parallax-view information to a sweep of locations in the viewzone. Holographic primitives may be points, lines, microfacets, or other primitives with spatial and projective characteristics that vary according to their modeling.[19–21]

A holographic primitive is encoded within the hologram by an instance of a precomputed diffractive element, called a projector fringe, or *pfringe*. Pfringes are analogous to basis fringes, and may be generated by any optimization technique, or by simulating interference between an object wave with the primitive's desired projective behavior and a reference wave that matches the reconstructing wave. If a simple isotropic spherical emitter (a point) is interfered with a plane-wave reference, for instance, its reconstructed holographic primitive will radiate uniformly through some angle of view [Fig. 6(a)]. The spatial and angular resolutions of a projection surface populated by these holographic primitives can match the surface population density and diffractive resolution found in interference-modeled holograms.

As in conventional stereogram computing, sampled instances of these pfringes (pfringe vectors) can be modulated by a parallax-view pixel vector describing one or a set of view pixels' values across their angle of view—thus modulated pfringe vectors are analogous to hogels. Depending on the method of view generation or capture, the view pixel vector may be assembled from a "slice" through the volume of captured parallax views or from a more complicated indexing scheme, or it may be generated directly by a specialized renderer. The resulting hogel reconstructs

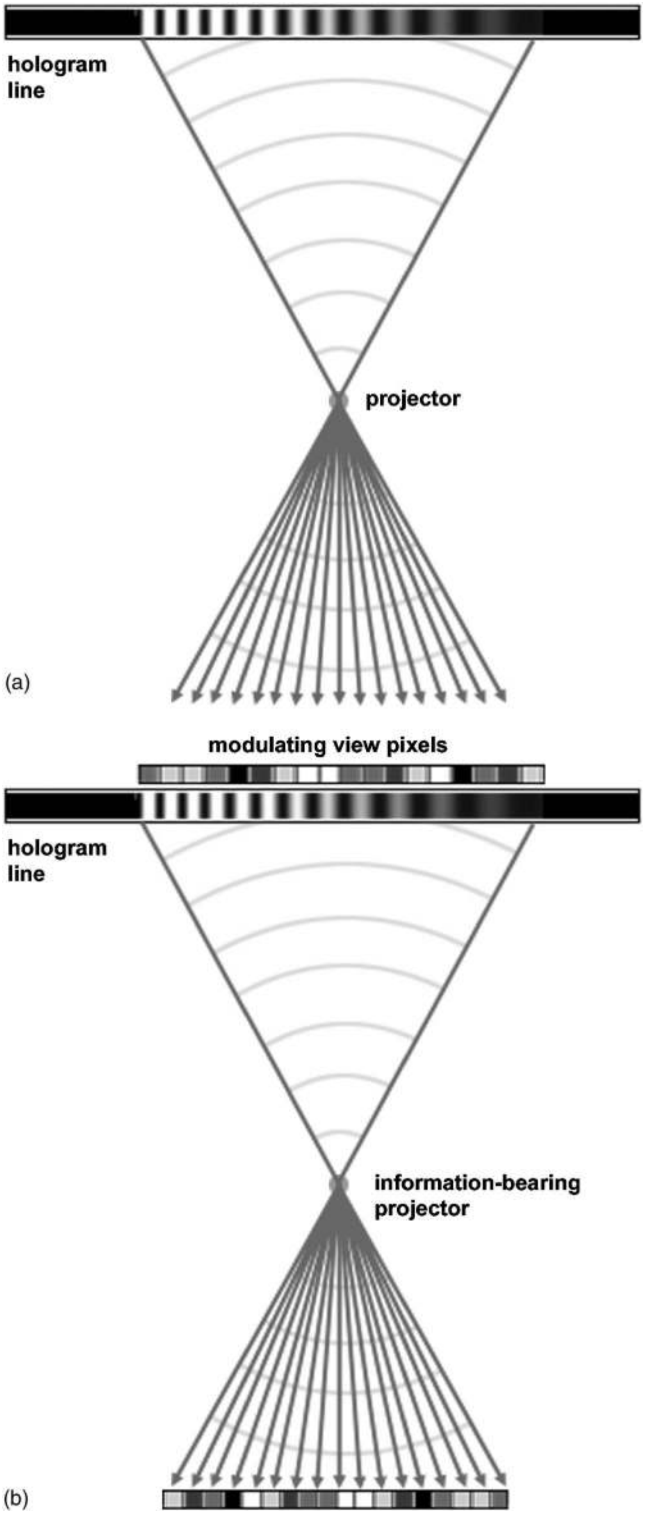

Fig. 6 Using fringes as a simple projectors of parallax: (a) unmodulated fringe; (b) modulated fringe.

an information-bearing holographic primitive that relays the view pixels back out along the direction of original capture [Fig. 6(b)]. Hogels are subsequently accumulated into the final RIP hologram, so their manner of assembly and their angular multiplexing behavior are much like those of conventional stereograms too. On reconstruction, a static or moving observer within the viewzone sees with each eye an appropriate and disparate view of the reconstructed scene. This technique borrows from both interference-modeled holograms and conventional computed stereograms to provide a high-quality and configurable light field projector. Within the limits of the MTF, the spatial and angular resolution of the output can be more appropriate for the human observer than when the slits are affixed to the hologram plane.

3.2 *Hybrid Display*

Through their flexible embodiment, RIP holograms represent a kind of continuum between stereograms and true volumetric displays. To illustrate this continuum—the particulars of parallax-view generation aside—first consider the case in which the projection surface describes a single plane, which can be produced using multiple instances of a single pfringe vector. Each holographic primitive populating the plane projects parallax views, which were captured using the same projection geometry via computer graphics, or optical capture. This process of view capture and display is stereographic in nature. Next, one can imagine the projection surface being shaped as a hemicylinder or hemisphere, wrapped around the scene's perimeter. Instances from a larger set of pfringes are required to populate the surface in depth, and view capture must proceed in a different fashion; but the stereogram-style assembly is similar in principle. A more specialized projection surface may be shaped to "shrink-wrap" the scene, so that holographic primitives populate object surfaces (in principle, the same input as interference modeling would require). One way to approximate this "shrink-wrap" would populate the display volume with many projection planes, finely stacked in depth; such a configuration is volumetric in nature and could effectively display transparent or refractive volumes and/or opaque and occluding surfaces. This generalized continuum of RIP hologram embodiments is shown in Fig. 7.

3.3 *RIP Diffractive Projection*

Each in the set of HPO pfringes $f_i(x)$ used for RIP hologram computing in this work is derived from modeling the interference of an inclined plane wave and a spherical wave whose source is located at $x=0$ and a distance $z=d_{0i}$ from the hologram plane. We use the same prescription given in Eq. (8), with amplitude $E_{\mathrm{sph}0i}=1.0$ and initial phase $\psi_{0i}=0.0$, to describe the signal-bearing part of this interference pattern:

$$f_i(x)=\cos\frac{2\pi}{\lambda}[r_i(x)-x\sin\theta_{\mathrm{ref}}]$$

$$\text{for}\quad d_{0i}\tan\theta_{\mathrm{out,max}}\le x\le d_{0i}\tan\theta_{\mathrm{ill}},\qquad(16)$$

where $r_i(x)$ is given in Eq. (6) for $z=d_{0i}$. The expression is computed with $\theta_{\mathrm{ill}}=\theta_{\mathrm{ref}}$, and for values of x that allow the pfringe's spatial frequency to vary from 0 to some $f_{\max}$. To prevent visible diffractive artifacts generated by abrupt phase discontinuities at its edges, the i'th pfringe $f_i(x)$ may be filtered, for instance, by a Kaiser window of width

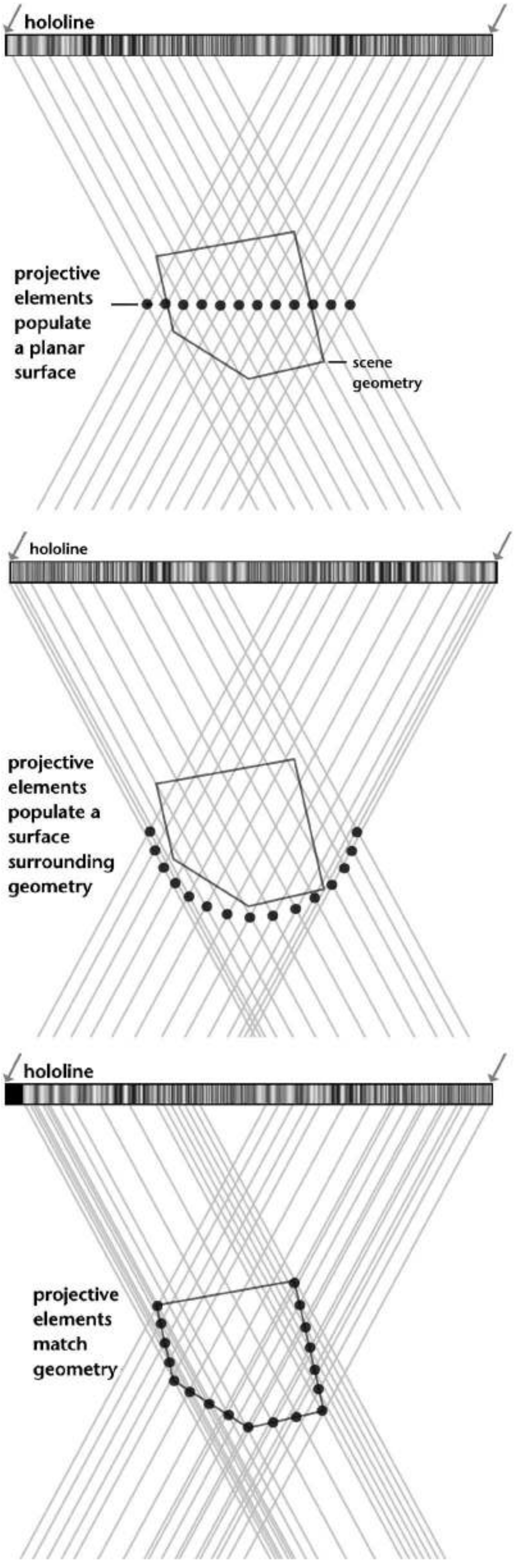
hololine
projective elements populate a planar surface
scene geometry
hololine
projective elements populate a surface surrounding geometry
hololine
projective elements match geometry

Fig. 7 Generalized continuum of RIP hologram embodiments.

$$W_i = d_{0i}(\tan \theta_{\text{ill}} + \tan \theta_{\text{out,max}}). \quad (17)$$

The i'th sampled windowed pfringe, the pfringe vector $\mathbf{f}_{\text{win } i}$, will reconstruct an image of a point at $z=d_{0i}$ from the hologram, projecting into the viewzone through an angle

$$\theta_{\text{pfringe}} = \theta_{\text{out,max}} + \theta_{\text{ill}}. \quad (18)$$

This image, a holographic primitive, or *holoprimitive*, acts as a simple light projector, and a densely populated collection of such holoprimitives constitutes the projection surface. If the projection surface is planar and located at $z=d_0$, then translated instances of a single pfringe vector, accumulated into the hologram, can be used to populate holoprimitives over one entire projection plane. If the desired projection surface is tipped with respect to the hologram plane, is multiplanar, or is nonplanar, then instances from a set of precomputed pfringe vectors will be required. Though a simple isotropic spherical-wave projector has been used in the work reported here, pfringes that model holoprimitives with other spatial and projective characteristics may be used as well.

3.4 *Parameters for Hologram Computation*

To illustrate the specification of compatible hologram projection and parallax view capture geometries, we consider a RIP hologram that projects a simple planar projection surface (using a single pfringe vector to reconstruct each holoprimitive on that surface) and employs a simple shearing and recentering capture scheme to collect scene parallax. The geometric characteristics of hologram projection are based on a given display's hologram width H_w and height H_h, some minimum hologram sample spacing $w_s=1/f_s$, an illumination angle θ_{ill}, and, for HPO systems, a viewing distance $z=D_v$ and number of hologram lines N_v. First, a maximum spatial frequency that a pfringe may contain is determined in order to satisfy the Nyquist criterion:

$$f_{\text{max,amplitude}} \leq f_s/2, \quad (19)$$

and the display's maximum diffracted output angle $\theta_{\text{out,max}}$ can be determined from Eq. (11) with $f=f_{\text{max,amplitude}}$. The projection plane to be populated by holoprimitives may be positioned in depth some distance $z=d_v \neq D_v$ from the viewer's eye, and therefore a nonzero distance $z=d_0=D_v-d_v$ from the hologram plane. In this case, holoprimitives located on the hologram plane are undefined.

Positioning a pfringe vector within any HPO hologram line causes the holoprimitive to reconstruct at a height geometrically related to the hologram line's vertical height. Horizontally translating the pfringe vector within a hologram line moves the reconstructed point along with it. Superposing M pfringe vectors within a hologram line reconstructs M holoprimitives. Thus, a composite pattern H_{raw} describing a uniformly bright, unmodulated projection surface can be assembled from a set of M appropriately positioned and accumulated instances of the unmodulated, windowed pfringe vector $\mathbf{f}_{\text{win}}$ on each hologram line, according to

$$H_{\text{raw}} = \sum_{j=1}^{N_v} \sum_{i=1}^{M} \mathbf{f}_{\text{win } i\times\Delta h_x, j\times\Delta h_y}, \quad (20)$$

where Δh_x and Δh_y are the horizontal and vertical holoprimitive spacings on the projection plane, respectively. In practice, since all holoprimitives are represented by an instance of the same windowed pfringe vector, a visible artifact will be generated by their self-interference during reconstruction. This artifact can be effectively eliminated by randomly perturbing the regular spacing between them by a small amount.

To populate the projection plane well, with no visible gaps, a value of Δh_x should be chosen with respect to the observers' visual acuity and the size of a pfringe vector's diffraction-limited focus. For ideal point foci of infinitesimal size, two adjacent holoprimitives will be just distinguishable to the human viewer when their horizontal separation $\Delta h_{x,\max}$ on the projection surface positioned at a depth d_v is given by

$$\Delta h_{x,\max} = d_v \tan \frac{1}{60}. \tag{21}$$

A reasonable lower bound on the horizontal spacing is given by the diffraction limit as

$$\Delta h_{x,\min} = \frac{\lambda}{\theta_{\text{pfringe}}}. \tag{22}$$

The vertical holoprimitive spacing Δh_y in an HPO system is fixed by the vertical raster of a given display and the choice of the projection plane's location relative to the viewing distance d_v:

$$\Delta h_{y,\max} = \frac{H_h}{N_v}\frac{d_v}{D_v}. \tag{23}$$

Locating a projection plane at d_0 also geometrically constrains its useful horizontal and vertical extents $I_{w,\max}$ and $I_{h,\max}$:

$$I_{w,\max} = H_w - d_0(\tan \theta_{\text{out},\max} + \tan \theta_{\text{ill}}),$$

$$I_{h,\max} = H_h \frac{d_v}{D_v}. \tag{24}$$

Finally, given the projection plane width I_w, the viewzone width V_w can be computed as

$$V_w = I_w + d_v(\tan \theta_{\text{out},\max} + \tan \theta_{\text{ill}}). \tag{25}$$

Note that since each holoprimitive's angle of projection θ_{pfringe} is limited by $f_{\max}$, a viewer will be able to consume an individual hogel's projected information only in the part of the viewzone it addresses. The desired angular sampling will target the angle a viewer's pupil subtends from the projection plane, and will be constrained by the capture camera aperture and the MTF of a coherently illuminated hogel. The resulting projection geometry is shown in Fig. 8.

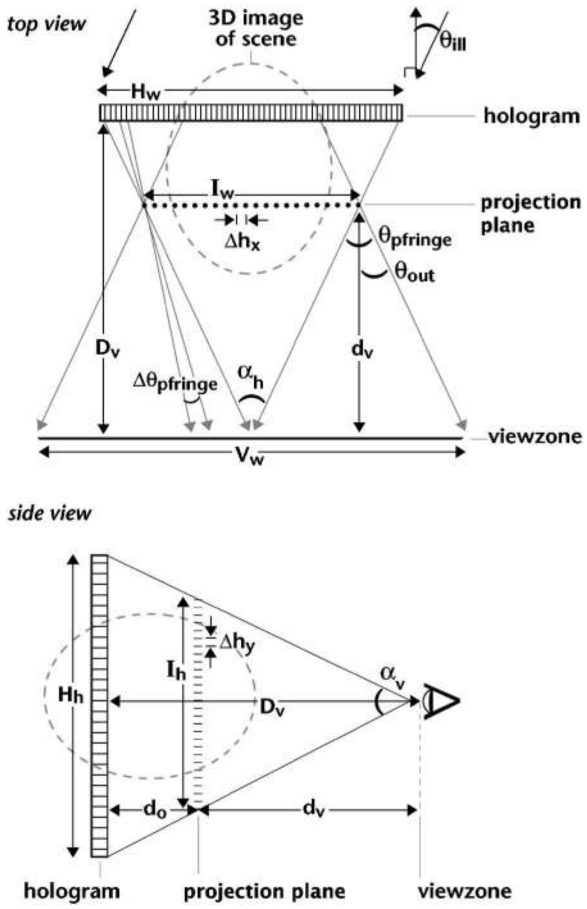

Fig. 8 HPO projection geometry: top view and side view.

3.5 *Parameters for Compatible View Capture*

The capture geometry can now be designed to match the projection geometry to within a scale factor s. Many different capture schemes can be used to acquire the requisite parallax views; here we consider a shearing-recentering capture technique in which the frustum is sheared as a function of camera translation. The camera distance D_c from the capture plane, horizontal and vertical field of view α_h and α_v, and capture track width T_w are specified as

$$D_c = s d_v,$$

$$\alpha_h = 2 \tan^{-1}\left(\frac{I_w}{2d_v}\right),$$

$$\alpha_v = 2 \tan^{-1}\left(\frac{I_h}{2d_v}\right),$$

$$T_w = s V_w. \tag{26}$$

The shear angle α_{shear}, which fixes the rendering window in the center of the scene as the capture camera translates along the track, is given as

$$\alpha_{\text{shear}} = \tan^{-1}\left(\frac{d_c}{D_c}\right), \tag{27}$$

where d_c is the positive distance of the camera from the track center.

In this particular embodiment, where holoprimitives function as slits with very narrow extent, we let the view pixel spacing $\Delta p_x \, \Delta p_y$ in each rendered view match the vertical and horizontal holoprimitive spacing in the projection plane, to within a scale factor s. Notably, this need not be the case; a holoprimitive with broader spatial extent and

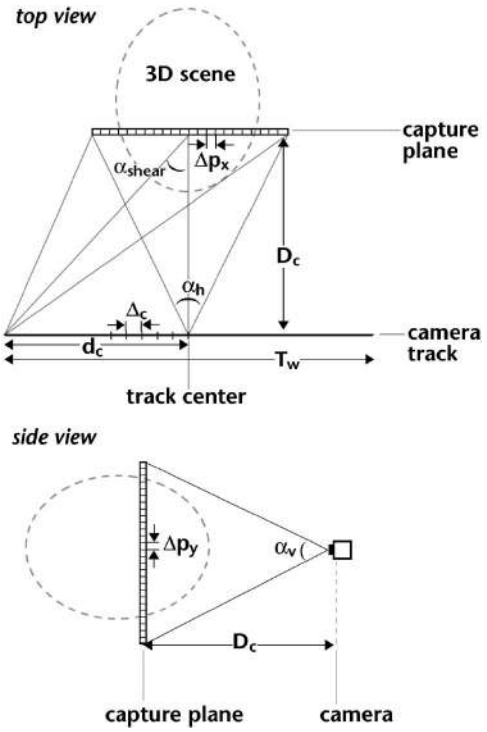

Fig. 9 Corresponding capture geometry.

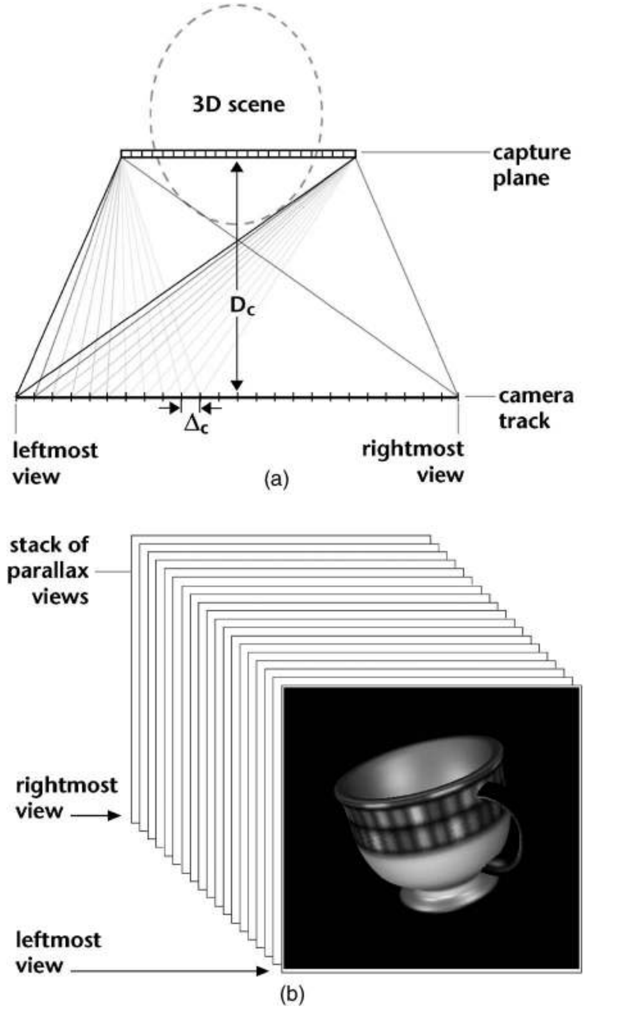

Fig. 10 Conventional rendering of teacup: (a) shearing-recentering capture scheme; (b) resulting parallax view volume.

different diffractive behavior could relay parallax information about more than one view pixel; in that event, the capture plane would not geometrically correspond to the RIP hologram's projection plane. And since Δp_x need not equal Δp_y, the resulting capture pixel aspect ratio might not be square.

To capture a set of $K = T_w / \Delta_c$ parallax views, the intercamera spacing Δ_c is chosen to support the scene angular sampling and bandlimiting requirements. Now this shearing-recentering capture geometry is fully specified to be compatible with the projection geometry as shown in Fig. 9. In this setup, the camera is positioned at one extreme of the capture track, and moved in increments of Δ_c until all parallax views are captured. These views are then formatted into a parallax view volume having $M \times N_v \times K$ pixels, by stacking one view atop the next as shown in Fig. 10. In this simple capture scheme, more view pixels are rendered than can be actually used in projection, due to the limited angular projection of each holoprimitive.

3.6 *Combining Vectors with Parallax Views and Accumulating Them into the Hologram*

Once parallax views are rendered, they can be used to modulate pfringe vectors, which are in turn accumulated into a RIP hologram. In this process, the pfringe vector corresponding to each of the holoprimitives on the projection plane is modulated by an appropriate view pixel vector **p**, which is extracted from the $M \times N_v \times K$ parallax view volume. The method of selecting the appropriate pixels to modulate each pfringe vector depends on way the parallax view volume is formatted, and on the view capture or generation method that generated it. If the parallax view volume created by the shearing-recentering camera is considered, we first determine the number of samples, n, within any pixel vector **p** to be extracted from the volume as a function of the projection and capture geometries:

$$n = \frac{D_c}{\Delta_c}(\tan \theta_{\text{out,max}} + \tan \theta_{\text{ill}}). \tag{28}$$

To modulate a pfringe vector corresponding to the holoprimitive location (i,j) for

$$0 \le i \le M-1,$$

$$0 \le j \le N_v - 1, \tag{29}$$

we must index the corresponding (i,j) location in the parallax view volume and retrieve the appropriate n view pixels between k_{min} and k_{max} (where $k=0$ corresponds to the leftmost captured view and $k=K-1$ corresponds to the rightmost captured view). For any (i,j), k_{min} and k_{max} are given by

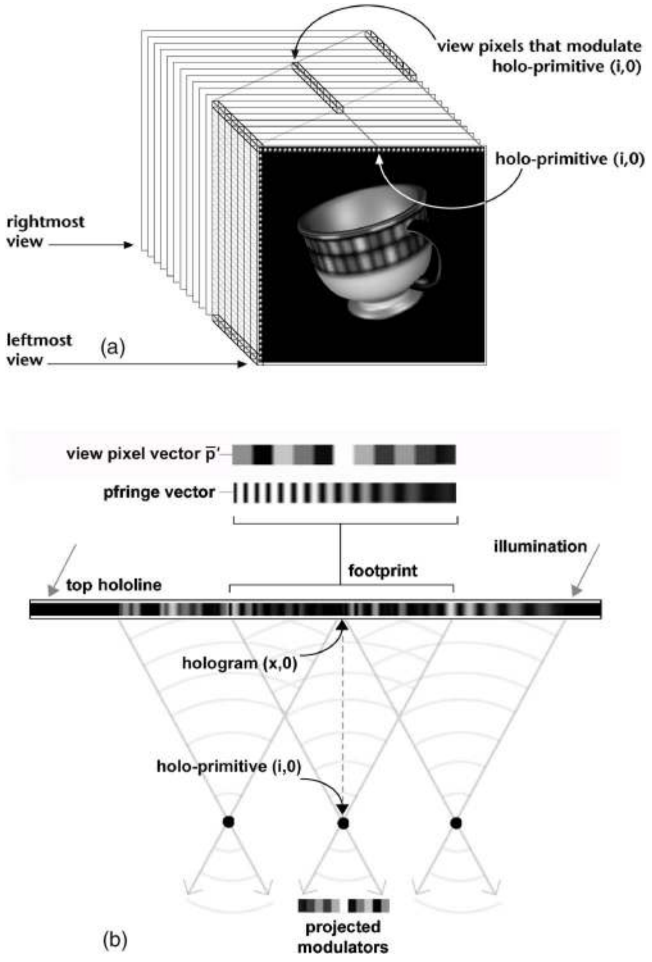

Fig. 11 RIP hologram assembly: (a) extracting appropriate view pixels from parallax view volume; (b) creating hogel and accumulating (top view).

$$k_{\min} = \text{round}\left[\left(\frac{K-n}{M}\right)i\right],$$

$$k_{\max} = \text{round}\left[\left(\frac{K-n}{M}\right)i\right] + n, \tag{30}$$

and their values undergo discrete jumps as i increases.

Once an appropriate $\mathbf{p}$ has been extracted from the view volume, its values are interpolated to fill a vector $\mathbf{p}'$ with the same number of samples as in the pfringe vector it will multiply. As shown in Fig. 11, the hogel that will reconstruct the primitive at (i,j) is created by modulating the pfringe vector with $\mathbf{p}'$ and accumulating it into the final hologram

$$H_{\text{RIP}} = \sum_{j=1}^{N_v} \sum_{i=1}^{M} \bar{\mathbf{f}}_{\text{win } i\times\Delta h_x, j\times\Delta h_y} \mathbf{p}'_{i,j}. \tag{31}$$

Once an entire pattern is finally assembled in this multiply-and-accumulate fashion, it must be normalized to fit the bit depth of the system's framebuffer. When written to the display, each hogel will project the parallax information it encodes, and a viewer can observe appropriate scene parallax as she moves through the viewzone.

As mentioned, the ability of RIP holograms to place directional projectors throughout the object or viewer's space permits the creation of display geometries that span the regime between true volumetric displays and simple stereograms. This flexibility can, for instance, be used to trade off between higher-quality and higher-performance imaging. The RIP geometry is also compatible with many specialized rendering schemes, such as multiple-viewpoint rendering (MVR),[22] which exploits interperspective, or double-frustum rendering,[23] which can be used to define nonplanar or multiplanar projection surfaces.

4 Implementation and Results

4.1 *MIT Second-Generation Holographic Video*

All of our implementation has been designed for the MIT Mark II electroholographic display.[7] The Mark II is a real-time display system that diffracts light by means of eighteen parallel cross-fired shear-mode TeO_2 acousto-optic modulators (AOMs) and passes the result to a chain of optics and scanning mirrors to produce a monochromatic HPO image. The output image volume is 150 mm wide, 75 mm high, and 160 mm deep, is visible over a range of 30 deg, is refreshed 30 times per second, and has a vertical resolution of 144 lines.

The Mark II system is optically very similar to the Scophony[24] television display of the 1930s, differing mostly in the use of multiple parallel AOMs in place of the latter's single AOM; the signals differ in that the Scophony system (which was displaying a single 2-D raster) amplitude-modulated a fixed-frequency fringe pattern with the intensity of the video image, while in Mark II both the amplitude and instantaneous phase of the fringes are varied. Also, Mark II, through its use of cross-fired AOMs, is able to use both the forward and retrace horizontal scans for active video, a technique sometimes called a *boustrophedonic* scanning pattern; this feature has implications for the generation of video signals, and for the way holograms are formatted in the framebuffer.

4.2 *New Computational Subsystem Using Commodity Hardware*

Since its construction in the 1990s, computation for the Mark II system has been performed by a combination of an SGI Onyx workstation and a Cheops imaging system,[25] a compact, block data-flow computing system optimized for real-time parallel computations on streams of data. The Cheops system has many key characteristics that made it a good choice for electroholography: it drives up to six genlocked video output cards whose video timing parameters can be freely configured to meet unusual requirements; it is easily interfaced to a host workstation by SCSI (and HIPPI, if higher speed is required); and it contains sufficient random-access memory.

We have recently reported a new compute architecture[26,27] that uses commercial PC graphics cards in place of the older Cheops-based platform. This new system eliminates the need for an expensive, nonstandard computing hardware and rendering software system, conveniently moving the programming environment to OpenGL running under Linux. The Cheops system was configurable to generate a video signal that was an exact match to the Mark II's needs: 30-Hz refresh rate, eight active lines of 262,142 (256 K) pixels each, with a horizontal blanking interval and sync pulse at the end of each line, and a vertical blanking interval and sync pulse each one line long at the end of the frame.

For power-consumption and bus-bandwidth reasons we configured our new PC-based system as three dual-head cards, each in a separate inexpensive (sub-$500) PC, with a fourth PC providing a graphical user interface to control the system. The Quadro FX 3000G video chip is not quite as flexible as Cheops. In particular, the line length is limited to 4096 pixels. Solving this problem by splitting a hololine among multiple frame buffer lines means that the video chip's horizontal blanking cannot be used to generate our horizontal blanking interval; instead we set the horizontal blanking parameter as small as possible (to minimize gaps in our fringe patterns) and add the needed blanking pixels to our active line length, filling them with black. To achieve the desired number of samples, each hololine is formatted as either 178 vertical lines of 2048 pixels or 89 vertical lines of 4096 pixels. Since 4096 is the maximum line length the software drivers will accept, to allow for future increases in the number of samples in a hololine, we chose to use 178 vertical lines at 2048 samples per line. In this configuration, we fill the first 128 lines of each 176 with fringe values and the remaining 50 with black to generate the hololine blanking interval. Since the software drivers do not allow the horizontal blanking on each 2048-sample line to be set to zero, the last 16 samples of each 2048 are blanked out. This creates small gaps in our diffraction fringes, but we have noted no visible artifacts as a result. We also add an external divide-by-178 counter between the horizontal sync output and the input of Mark II. To meet the vertical timing requirements we place 356 blank lines at the top of the frame and 8 vertical sync lines at the end.

Finally, each hologram line within a triplet of hologram lines (i.e., 0,1,2 or 18, 19, 20) is stored in the R, G, or B framebuffer channel, respectively. Then the three dual-head video cards together generate the video signals for (6 video outputs)× (3 channels per output)×(8 hologram lines per channel)=144 hologram lines.

4.3 *RIP Implementation*

Lucente and Gaylean[1] reported an algorithm to compute holographic stereograms that used a high-end graphics workstation's hardware-accelerated accumulation buffer and texture units. In their work, the texture units were used to modulate basis fringes by view pixel information; modulated fringes were written to the framebuffer and summed together using the accumulation buffer. Petz and Magnor[28] describe the use of inexpensive graphics chips that used the texture units with limited bit depth to perform summation; they organized the basis fringe summations into a hierarchy of texture summations to preserve accuracy. Our implementation uses pixel blending within each channel of the framebuffer rather than using the accumulation buffer, as noted below.

We have programmed a nonoptimized OpenGL implementation of the RIP algorithm for this new commodity-hardware compute architecture. Our software configuration positions a 75.4×56.5-mm projection plane 4 mm in front of the hologram plane. The projection plane is populated by 383×144 holoprimitives (spherical emitters), with a horizontal interpoint spacing of 0.19 mm (prior to spatial perturbation). The HPO viewzone is 383 mm wide and 590 mm in front of the projection plane, into which 140 parallax views are projected with an interview spacing of 2.7 mm. The projection plane's population density and the angular sampling are chosen to match the requirements of the human visual system.

The computation takes place in two phases. First, in an initialization step, six processes, each assembling 24 of the 144 hologram lines and rendering them to one of the framebuffers, parse an XML description of the hologram geometry and the scene to be displayed. From this description, both hologram projection and parallax view rendering geometries are configured, scene lighting is specified, model material and texture properties are assigned, and geometry display lists are constructed. Each process computes the interference pattern for each pfringe used to reconstruct the projection plane, windows the pfringe vectors with a Kaiser window to prevent sharp discontinuities at the apertures, and represents them as 1-D textures. Next, in a combined parallax-view and hologram-rendering phase, each process renders the sweep of 140 parallax views of the scene and represents it as a 3-D texture. Each view is rendered at a resolution of 383×144 pixels with a nonsquare pixel aspect, according to the capture geometry specified in the initialization step. Then, the RIP hologram rendering and assembly is accomplished by superposing the set of hogels that reconstruct all view-modulating points in a given framebuffer's portion of the hologram.

In this work, hologram rendering and assembly are implemented to utilize NVIDIA's OpenGL hardware support for multitexturing. Using an orthographic projection, each view-modulated projector fringe is rendered as a rectangle with the height of one framebuffer line and width of the projector fringe's footprint on the hologram, positioned within the rendering window to reconstruct a holoprimitive at the correct location on the projection plane, and modulated by both the fringe texture and an appropriate portion of the parallax view texture. Since the extent of a hogel's footprint was wider than the rendering window, it must be made to span multiple framebuffer lines, beginning on the next line exactly at the sample that was clipped on the previous. This is accomplished most efficiently by rendering multiple versions of the same textured rectangle, shifted by appropriate amounts to position them correctly within each framebuffer line spanned. In addition, since the hogels for adjacent points will partially overlap in a RIP hologram, their superposition is achieved by enabling pixel blending and defining an appropriate blend function that scales each contribution so that the final blended result fits within the dynamic range of each 8-bit channel in the framebuffer.

When all six framebuffers' holograms are rendered and displayed, the pattern they produce reconstructs a projection plane populated by holopoints, and each holopoint projects angularly varying view information to reconstruct the captured scene's light field. Once a hologram frame has been displayed, each process can modify the scene and handshake with a server to synchronize update and display of the next frame.

In side-by-side comparisons of the images generated by the PCs with RIP hologram images generated by the previous SGI and Cheops-based compute architecture, only one significant difference was apparent. Since the screen raster is made up of simultaneous outputs from multiple video cards, any timing variation between the cards is visible as jaggedness in vertical edges. The Cheops images display no such effect, while the three groups of lines in the PC images have a small degree of misalignment, suggesting that the card-to-card synchronization is not quite as precise as needed. The relative misalignment changes each time the cards are initialized but appears stable over time; we are currently compensating by means of a variable shift in the images on the framebuffers, but we are also currently determining the cause of the underlying problem. In side-by-side comparisons of the unmodulated diffracted field of RIP holograms and that of conventional holographic stereograms, the RIP hologram's field appeared uniformly bright and without any visible diffraction artifacts. In the conventional stereogram's diffracted field, numerous diffraction artifacts were present, appearing as dark vertical bands located behind the hologram plane.

Rendering simple texture-mapped objects using the conventional stereogram algorithm (256 hogels per hologram line, 32 basis fringes, and 32 parallax views) for this display architecture, we observed update rates of approximately 2 frames/s, the fastest stereogram computing rate we have ever achieved. By comparison, preliminary results from the nonoptimized RIP renderer have given us an update rate of approximately 1.2 frames/s (0.84 s/frames) for scenes of similar complexity. A sample of RIP images of different scenes are shown in Fig. 12; the visible dark horizontal bands are due to a missing optical channel and optical misalignments in the Mark II system at the time the photographs were recorded, and are not artifacts of the computing method.

5 Conclusions and Future Work

We believe that this new algorithm implemented with hardware acceleration makes a significant stride toward combining pictorial realism with interactive update rates in the field of electroholography. Though the prototype implementation described in this work projects a scene's light field through a simple planar surface, RIP holograms are highly flexible, and capable of producing a range of outputs, from interference-modeled volumetric images to simpler stereographic ones. Many conventional and specialized parallax-view-rendering techniques are compatible with the assembly of RIP holograms, and the spatial and angular sampling during scene capture can be selected to satisfy viewer and scene requirements rather than being strictly constrained by the spatial and spectral characteristics of a basis fringe set. Since the footprints of adjacent superposed hogels overlap instead of abut in RIP holograms, the hard boundaries of their apertures may be conditioned to minimize visible diffracted artifacts in the field.

Since the implementation reported here has not yet been optimized to the extent possible using OpenGL and the NVIDIA's hardware support, we expect that the frame rates reported at the time of this writing will improve as the implementation is refined.

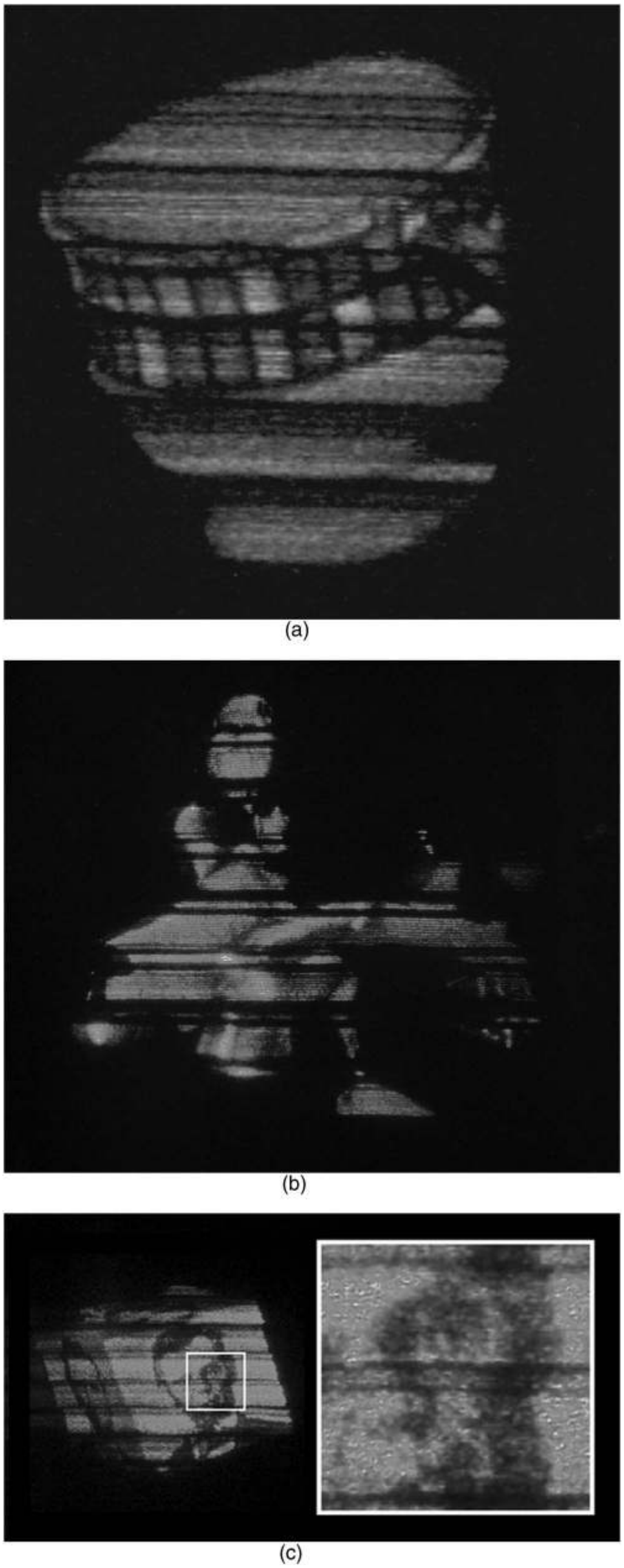

Fig. 12 RIP holograms: (a) teacup produced from 140 MVR-rendered parallax views; (b) ventricles and multiple sclerosis lesions in the brain from 50 rendered parallax views; (c) cube texture mapped with photograph of Abraham Lincoln; 140 parallax views, 383×144 pixels each.

Our continuing work is proceeding along three paths: improving the image quality (quantifying the MTF of RIP holograms and generating high-quality light fields of complicated scenes using a variety of RIP configurations); increasing the computation speed (optimizing the rendering scheme, hologram computation, and client-server interaction); and developing analytical extensions of the RIP algorithm to include arbitrary surfaces and advanced primitives.

Acknowledgments

This work was supported by the Honda R&D Co., by the Digital Life, Things That Think, and CELab Consortia at the MIT Media Laboratory, and as part of the Visualization Core of the Neuroimage Analysis Center at the Brigham and Women's Hospital, NCRR grant P41RR13218. The authors would like especially to thank Tyeler Quentmeyer for his design and engineering of the XML scene parser and OpenGL renderer used in this work and his software support at the outset of this project. Also invaluable to this project were Stephen A. Benton's guidance, enthusiasm and support for the MIT Holographic Video Project.

References

1. M. Lucente and T. Galyean, "Rendering interactive holographic images," in *Proc. SIGGRAPH'95*, pp. 387–394, ACM SIGGRAPH, New York (1995).
2. W. Plesniak, M. Halle, S. D. Pieper, W. Wells, III, M. Jakab, D. S. Meier, S. A. Benton, C. R.G. Guttmann, and R. Kikinis, "Holographic video display of time-series volumetric medical data," in *Proc. IEEE Visualization 2003*, pp.78–85 (2003).
3. M. W. Halle, S. A. Benton, M. A. Klug, and J. S. Underkoffler, "The Ultragram: a generalized holographic stereogram," in *Practical Holography V*, S.A. Benton, Ed., *Proc. SPIE* **1461**, 142–155 (1991).
4. P. St.-Hilaire, S. A. Benton, M. Lucente, J. D. Sutter, and W. J. Plesniak, "Advances in holographic video," in *Practical Holography VII*, S. A. Benton, Ed., *Proc. SPIE* **1914**, 188–196 (1993).
5. J. S. Underkoffler, "Development of parallel processing algorithms for real-time computed holography," SB Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (1988).
6. J. S. Underkoffler, "Toward accurate computation of optically reconstructed holograms," SM Thesis, Media Arts and Sciences Section, Massachusetts Institute of Technology (1991).
7. M. Lucente, "Interactive computation of holograms using a look-up table," *J. Electron. Imaging* **2**(1), 28–34 (1993).
8. J. S. Underkoffler, "Occlusion processing and smooth surface shading for fully computed synthetic holography," in *Symp. on Electronic Imaging, Practical Holography XI*, S.A. Benton, Ed., *Proc. SPIE* **3011**, 19–30 (1997).
9. F. W. Campbell and D. G. Green, "Optical and retinal factors affecting visual resolution," *J. Physiol. (London)* **181**, 576–593 (1965).
10. H. Yoshikawa, S. Iwase, and T. Oneda, "Fast computation of Fresnel holograms employing difference," *Opt. Rev.* **8**(5), 331–335 (2001).
11. W. Plesniak, "Incremental update of computer generated holograms," *Opt. Eng.* **42**(6), 1560–1571 (2003).
12. M. Lucente, "Diffraction-specific fringe computation for electroholography," PhD Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (1994).
13. P. St.-Hilaire, "Modulation transfer function and optimum sampling of holographic stereograms," *Appl. Opt.* **33**(5), 768–774 (1994).
14. M. W. Halle, "The generalized holographic stereogram," SM Thesis, Media Arts and Sciences Section, Massachusetts Institute of Technology (1991).
15. M. W. Halle, "Holographic stereograms as discrete imaging systems," in *Practical Holography VIII*, S.A. Benton, Ed., *Proc. SPIE* **2176** (1994).
16. J. Watlington, M. Lucente, C. J. Sparrell, V. M. Bove, Jr., and I. Tamitani, "A hardware architecture for rapid generation of electroholographic fringe patterns," in *Practical Holography IX*, *Proc. SPIE* **2406**, 172–183 (1995).
17. C. W. Slinger, R. W. Bannister, C. D. Cameron, S. D. Coomber, I. Cresswell, P. M. Hallett, J. R. Hughes, V. Hui, J. C. Jones, R. Miller, V. Minter, D. A. Payne, D. C. Scattergood, D. T. Sheering, M. A. Smith, and M. Stanley, "Progress and prospects for practical electroholographic display systems," *Proc. SPIE* **4296**, 18 (2001).
18. C. W. Slinger, C. D. Cameron, S. D. Coomber, R. J. Miller, D. A. Payne, A. P. Smith, M. G. Smith, M. Stanley, and P. J. Watson, "Recent developments in computer-generated holography: toward a practical electroholography system for interactive 3D visualization," *Proc. SPIE* **5290**, 27 (2004).
19. C. Frere, D. Leseberg, and O. Bryngdahl, "Computer-generated holograms of three-dimensional objects composed of line segments," *J. Opt. Soc. Am. A* **3**(5), 726–730 (1986).
20. D. Leseberg, "Computer-generated holograms: display using one-dimensional transforms," *J. Opt. Soc. Am. A* **3**(11), 1846–1851 (1986).
21. D. Leseberg, "Computer generated holograms: cylindrical, conical, and helical waves," *Appl. Opt.* **26**(20), 4385–4390 (1987).
22. M. Halle, "Multiple viewpoint rendering," in *Proc.25th Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 243–254, ACM SKGRAPH (1998).
23. M. Halle and A. Kropp, "Fast computer graphics rendering for full parallax spatial displays," in *Proc. IS&T/SPIE's Symp. on Electronic Imaging, Practical Holography XI*, S.A. Benton, Ed. (1997).
24. H. W. Lee, "The Scophony television receiver," *Nature (London)* **142**, 59–62 (1938).
25. V. M. Bove, Jr., and J. Watlington, "Cheops: a data-flow processor for real-time video processing," *IEEE Trans. Circuits Syst. Video Technol.* **5**(2), 140–149 (1993).
26. T. Quentmeyer, "Delivering real-time holographic video content with off-the-shelf PC hardware," SM Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2004).
27. V. M. Bove, Jr., W. J. Plesniak, T. Quentmeyer, and J. Barabas, "Real-time holographic video images with commodity PC hardware," in *Stereoscopic Displays and Applications*, *Proc. SPIE* **5664A**, 255–262 (2005).
28. C. Petz and M. Magnor, "Fast hologram synthesis for 3D geometry models using graphics hardware," in *Practical Holography XVII and Holographic Materials IX*, T.H. Jeong and S.H. Stevenson, Eds., *Proc. SPIE* **5005**, 266–275 (2003).

Wendy Plesniak is currently a visiting scholar at the MIT Media Laboratory and a research fellow at the Surgical Planning Laboratory, Brigham and Women's Hospital, a teaching affiliate of Harvard Medical School. She works on the creative design and engineering of systems that visualize complicated information for processing and analysis, and for communication and expression. She has developed new methods for computing holograms for the MIT second-generation holographic video display system, and co-developed the first interactive haptic hologram. Plesniak also works professionally as a designer/engineer of interactive systems and information displays. She is a co-founder of Thinkcycle.org, and a co-founder of Carnegie Mellon University's Studio for Creative Inquiry. She holds a PhD and MS from MIT, and a BSEE and BFA from CMU.

Michael Halle is a visiting scholar at the MIT Media Laboratory and Director of Technology Development at the Surgical Planning Laboratory, Brigham and Women's Hospital, a teaching affiliate of Harvard Medical School. He received a SB in computer science, a SM in visual studies, and a PhD in media technology, all from MIT. His professional interests include scientific visualization and three-dimensional display.

V. Michael Bove, Jr., holds an SBEE, an SM in visual studies, and a PhD in media technology, all from the Massachusetts Institute of Technology, where he is currently head of the Object-Based Media Group at the Media Laboratory and directs the consumer electronics program CELab. He is the author or co-author of more than 50 journal and conference papers on digital television systems, video-processing hardware/software design, multimedia, scene modeling, visual display technologies, and optics. He holds patents on inventions relating to video recording, hardcopy, interactive television, and medical imaging, and has been a member of several professional and government committees. He is on the board of editors of the *Journal of the Society of Motion Picture and Television Engineers*, and associate editor of *Optical Engineering*. Bove is a fellow of the Institute for Innovation, Creativity, and Capital and of the Society of Photo-Optical Instrumentation Engineers.

James Barabas received his BS degree in computer science from Cornell University in 2000. As a research associate at the Schepens Eye Research Institute, he developed computer graphics, motion tracking, and haptic interfaces for a walking simulator, and conducted experiments to evaluate mobility in people with visual impairments. He is currently a graduate student at the MIT Media Laboratory studying distributed image capture for holographic display. His research interests include distributed image processing, information visualization, visual psychophysics, and medical imaging.

Ravikanth Pappu is a co-founder of ThingMagic, Inc. He received his PhD in 2001 from MIT for the invention of physical one-way functions. He also holds a BS in electronics and communication engineering from Osmania University, India, an MS in electrical engineering from Villanova, and an MS in media arts and sciences from MIT. While at MIT, he co-created the first dynamic holographic video system with haptic interaction and was a co-founder of Thinkcycle.org. His technical expertise is in physical cryptography, optical engineering, and display technology, and he is also interested in low-cost computing for developing countries.