

# Computational holographic bandwidth compression

by M. Lucente

***A novel technique to compute holographic fringe patterns for real-time display is described. Hogel-vector holographic bandwidth compression, a diffraction-specific approach, treats a fringe as discretized in space and spatial frequency. By undersampling fringe spectra, hogel-vector encoding achieves a compression ratio of 16:1 with an acceptably small loss in image resolution. Hogel-vector bandwidth compression achieves interactive rates of holographic computation for real-time three-dimensional electro-holographic (holovideo) displays. Total computation time for typical three-dimensional images is reduced by a factor of over 70 to 4.0 seconds per 36-MB holographic fringe and under 1.0 seconds for a 6-MB full-color image. Analysis focuses on the trade-offs among compression ratio, image fidelity, and image depth. Hogel-vector bandwidth compression matches information content to the human visual system, achieving "visual-bandwidth holography." Holovideo may now be applied to visualization, entertainment, and information.***

**E**lectro-holography—also called holovideo—is a new visual medium that electronically produces three-dimensional (3D) holographic images in real time. Holovideo is the first visual medium to produce dynamic images that exhibit all of the visual depth cues and realism found in physical scenes.<sup>1</sup> It has numerous potential applications in visualization, entertainment, and information, including education, telepresence, medical imaging, interactive design, and scientific visualization. Electro-holography combines holography and digital computational techniques. Holography<sup>2,3</sup> is used to create 3D images using a two-step coherent optical process. An interference pattern (*fringe pattern* or simply *fringe*) is recorded in a high-resolution light-sensitive medium. Once developed,

this recorded fringe diffracts an illuminating light beam to form a 3D image.

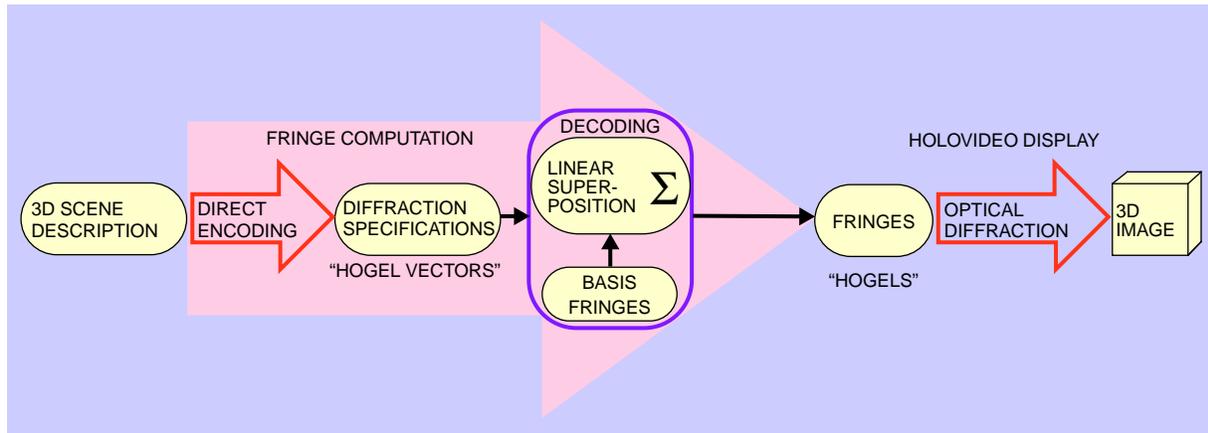
As early as 1964 researchers computed holographic fringes to create images that were synthetic and potentially dynamic.<sup>4-10</sup> Both the computation and display of holographic images are difficult due to huge fringe bandwidths. A computed (discrete) fringe must contain roughly ten million samples per square millimeter to effectively diffract visible light. Interactive-rate computation (about one frame per second or faster) was impossible. In 1989, researchers at the MIT Media Laboratory Spatial Imaging Group created the first display system that produced real-time 3D holographic images.<sup>11</sup> Computation of the 2-MB fringe required several minutes for small simple images using conventional computation methods.

A new diffraction-specific computation<sup>8</sup> technique named *hogel-vector bandwidth compression* achieves interactive-rate holographic computation (see Figure 1). The main features of this technique reported here are:

- Its architecture, based on the discretization of space and spatial frequency
- The use of hogel-vector bandwidth compression to reduce bandwidth by 16:1 and higher, allowing for easier display, transmission, and storage

©Copyright 1996 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

**Figure 1 Hogel-vector bandwidth compression: direct encoding and decoding using superposition of precomputed basis fringes**



- Fringe computation that is over 70 times faster than conventional computation
- The trade-offs among the system parameters of image resolution, image depth, and bandwidth

This paper contains a background section followed by sections that describe hogel-vector bandwidth compression and its implementations, experimental results, and analysis.

## Background

This section includes background information on computational holography, holographic displays, and past work in holographic information reduction.

**Computational holography.** Computational holography<sup>4,5</sup> begins with a 3D numerical description of the object or scene to be imaged. Traditional, conventional holographic computation imitated the interference of optical holographic recording. Speed was limited by two fundamental properties of fringes: (1) the myriad samples required to represent microscopic features (>1000 line-pairs per millimeter [lp/mm]), and (2) the computational complexity associated with the physical simulation of light propagation and interference.

In a computer-generated hologram, I define the number of samples per unit length (in one dimension) as the pitch,  $p$ . To satisfy fringe sampling requirements, a minimum of two samples per cycle of the highest spatial frequency are needed. The pitch is chosen to

be  $p \geq (4/\lambda)\sin(\Theta/2)$  where  $\Theta$  is the range of angles of diffraction (i.e., the width of the viewing zone) and  $\lambda$  is the wavelength of light used.<sup>10</sup> A typical full-parallax 100 mm × 100 mm hologram has a sample count (also called “space-bandwidth product,” or simply “bandwidth”) of over 100 gigasamples. The elimination of vertical parallax provides savings in display complexity and computational requirements<sup>9,10</sup> without greatly compromising display performance. This paper deals with horizontally off-axis transmission horizontal-parallax-only (HPO) holograms.<sup>3</sup> Such an HPO fringe is commonly treated as a vertically stacked array of one-dimensional holographic lines.<sup>6</sup>

A straightforward approach to the computation of holographic fringes resembled 3D computer graphics ray-tracing. The complex wavefront from each object element was summed, with a reference wavefront, to calculate the interference fringe. Interference-based computation requires many complex arithmetic operations (including trigonometric functions and square roots), making rapid computation impossible even on modern supercomputers.<sup>10</sup> Furthermore, interference-based computation does not provide a flexible framework for the development of holographic bandwidth compression techniques.

**Holographic displays.** Holographic displays modulate light with electronically generated holographic fringes. Early researchers employed a magneto-optic<sup>12</sup> spatial light modulator (SLM) or a liquid-crystal display (LCD)<sup>13</sup> to produce tiny planar images. The time-multiplexing of a very fast SLM provides a suitable

## Terms and Abbreviations

<b>basis fringe</b>	An elemental fringe pattern computed to contain a particular spectral profile. Linear summations of basis fringes are used to diffract light. This new name is analogous to mathematical basis functions.
<b>Cheops</b>	A digital image processing platform originally designed to explore scalable digital TV and real-time image encoding and decoding for the Television of Tomorrow (TVOT) consortium at the MIT Media Laboratory.
<b>fringe</b>	The holographic pattern, recorded optically or generated computationally, used to diffract light to form an image.
<b>hogel</b>	Holographic element; a small functionally diffractive piece of hologram representing a spatial sample of the fringe pattern and possessing a homogeneous spectrum.
<b>hogel vector</b>	A sampled hogel spectrum, specifying the diffractive purpose of a hogel.
<b>holovideo</b>	An electronic interactive 3D holographic display system that uses holographic fringes to cause light to diffract and form a 3D image (from Greek <i>holos</i> , whole, and Latin <i>videre</i> , to see).
<b>HPO</b>	Horizontal parallax only. A 3D imaging system, e.g., a holographic one, that provides viewing-zone motion parallax horizontally but not vertically.
<b>image volume</b>	The space that may be occupied by a 3D image.
<b>MAC</b>	Multiplication accumulation; a calculation consisting of one multiplication and one addition.
<b>pitch</b>	Sampling pitch. The number of samples per unit length in a discretized digital fringe pattern.

substitute for an ideal holographic SLM.<sup>11</sup> The research presented in this paper employs the 6-MB color holovideo display<sup>11,14</sup> and the 36-MB holovideo display<sup>15,16</sup> developed by the Spatial Imaging Group at the MIT Media Laboratory. These displays used the combination of an acousto-optic modulator (AOM) and a series of lenses and scanning mirrors to assemble a 3D holographic image in real time. (See Figure 2.) The 6-MB display generated a full-color 3D image with a  $35 \times 30 \times 50$ -mm width by height by depth. The 36-MB (monochromatic) display generated a  $150 \times 75 \times 160$ -mm image. By incorporating the proper physical parameters (e.g., wavelengths, sampling pitch), the fringe computation described in this paper can be used for other holographic displays.

**Information reduction in holography.** Although this paper may be the first published report of *computational* holographic bandwidth compression, several researchers have attempted to reduce bandwidth in *optical* holographic imaging. Holographic fringes contain more information than can be utilized by the human visual system.<sup>17,18</sup> By employing a dispersion plate, Haines and Brumm<sup>19</sup> generated a full-size

image using a reduced-size hologram. However, image quality suffered. Image resolution or signal-to-noise ratio was reduced. Hildebrand<sup>20</sup> generalized the dispersion-plate approach. Burckhardt and Enloe<sup>18,21</sup> reduced the information recorded in a hologram by exposing an array of small regularly spaced areas—the equivalent to spatially sampling the hologram. The reconstructed image had an annoying “screen” artifact or a reduced resolution. Good images were reconstructed with information reduction factors of six in each lateral dimension. Lin<sup>22</sup> used spatial subsampling of a Fourier-transform hologram—the equivalent of spectral sampling. The subsampled hologram was exposed through a mask of regularly spaced small apertures. Image fidelity suffered due to decreased image resolution and the presence of artifacts.

These experiments exploited the redundancy inherent to holographic fringes. Essentially, researchers subsampled (spatially or spectrally) to reduce bandwidth. Image quality suffered, e.g., dispersion plates caused graininess and noise. Such artifacts were inevitable because researchers could not directly manipulate the

Figure 2 Schematic of MIT holovideo display, which presents a real 3D image (in front of the output lens) to the viewer

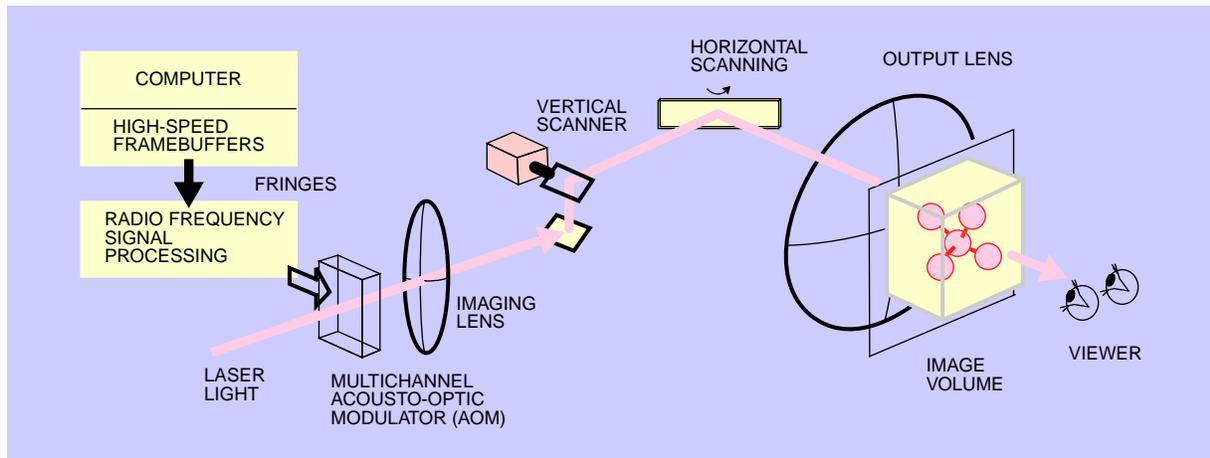
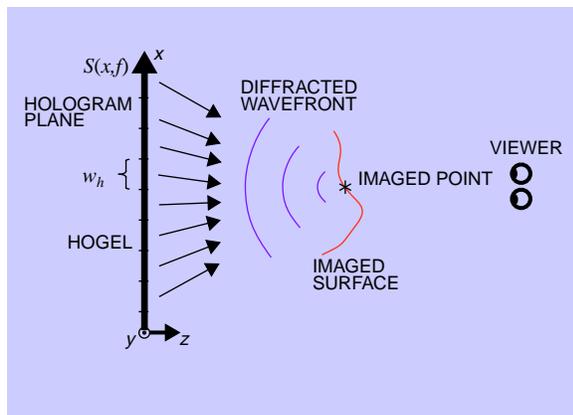


Figure 3 Computed fringe diffracts light to form an image



recorded fringes. *Computed* fringes can be directly manipulated.<sup>5,23</sup> This paper discusses the translation of optical information-reduction concepts into computational holography, where they are more useful and realizable.

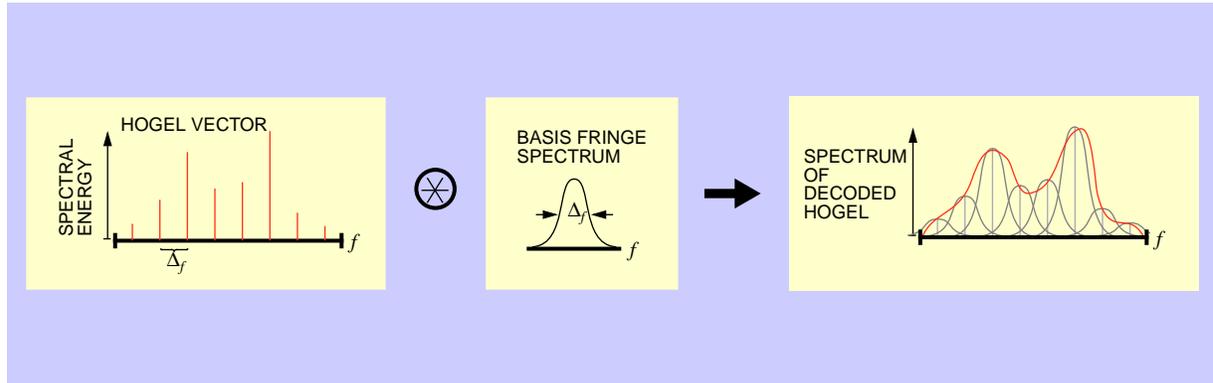
### Hogel-vector bandwidth compression

Hogel-vector bandwidth compression is a diffraction-specific fringe computation technique.<sup>8</sup> Stated simply, the diffraction-specific approach is to consider only the reconstruction step in holography. In practical

terms, it is the spatially and spectrally sampled treatment of a holographic fringe. Although numerical methods can compute diffraction backwards,<sup>24</sup> they are far too slow for interactive-rate computation. Diffraction-specific fringe computation provides a fast means for generating useful fringes through calculations that relate the fringes to the image through diffraction in reverse. It has the following features (see Figure 1):

- *Spatial discretization*—The hologram plane is treated as a regular array of functional holographic elements named *hogels*. In HPO holograms, a horizontal line of the hologram is treated as regular line-segment hogels of width  $w_h$ , each comprising roughly 100 to 2000 samples.
- *Spectral discretization*—A *hogel vector* is a spectrally sampled representation of a hogel. Each component, spaced by  $\Delta_f$ , represents the amount of spectral energy near a particular spatial frequency. A hogel vector is the diffraction specification of a hogel. 3D object scene information is encoded as an array of hogel vectors.
- *Basis fringes*—A set of precomputed basis fringes combine to decode each hogel vector into one hogel-sized fringe. Each basis fringe represents an independent part of the hogel spectrum and is pre-computed with appropriate  $w_h$  and  $\Delta_f$ .
- *Rapid linear superposition*—In the decoding step, hogel vectors specify the linear real-valued superposition of the precomputed basis fringes to generate physically usable fringes.

Figure 4 Spectral characteristics of hogel-vector decoding



By encoding the 3D scene description as diffraction specifications—an array of hogel-vectors—this technique reduces required bandwidth. Speed results from the simplicity, efficiency, and directness of basis-fringe summation in the decoding step.

**Sampling and recovery.** Encoding hogel vectors and decoding them into fringes are based on a spatially and spectrally sampled treatment of the fringe. The spatial and spectral sample spacings are selected to allow the fringe to be recovered from the hogel-vector array and used to diffract light to form the desired image. The first-order diffracted wavefront is the physical entity being represented by a fringe. Diffraction is linear, and the wavefront immediately following modulation by a fringe can be expressed as a summation of plane waves,<sup>25</sup> each diffracted by a spatial frequency component.

Diffraction-specific computation treats a one-dimensional HPO fringe (at some vertical location  $y$ ) as a two-dimensional (2D) localized spectrum  $S(x, f)$ , where  $x$  is the spatial position on the hologram and  $f$  is the spatial frequency. This is a Wigner distribution,<sup>26</sup> possessing a continuously varying amplitude as a function of space and spatial frequency.  $S_{ij}$ —the discrete representation of  $S(x, f)$ —is an array of hogel vectors, i.e., the diffraction specifications for the one horizontal line of the fringe. When sampled and recovered correctly,  $S(x, f)$  causes light to diffract and to form image points throughout one plane of the image volume. (See Figure 3.) To effectively sample  $S(x, f)$ , sufficiently small spatial and spectral sample spacing,  $w_h$  and  $\Delta_f$ , must be chosen. As discussed later, these spacings are selected using an empirically

verified model that relates these parameters to the quality of the reconstructed image.

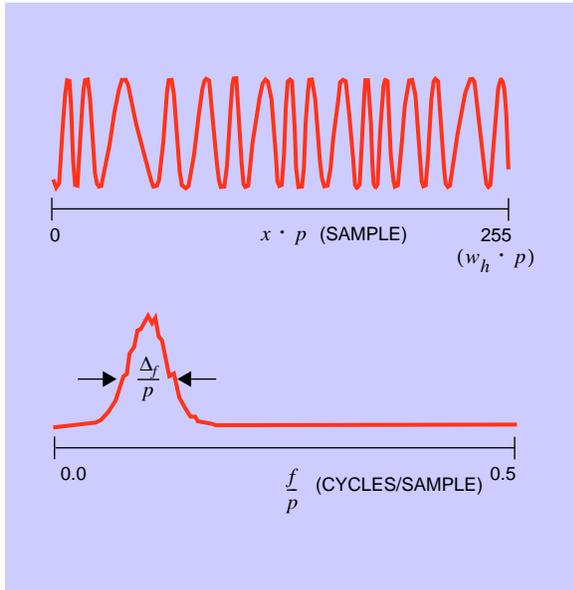
$S(x, f)$  is recovered from  $S_{ij}$  through convolutions with sinc functions (as per the sampling theorem<sup>27</sup>):

$$S(x, f) = \sum_i \sum_j S_{ij} \text{sinc}(j - fw_h) \text{sinc}(i - x\Delta_f) . \quad (1)$$

For the spectral dimension  $f$ , the convolution is performed in the spatial domain by the weighted summation of basis fringes, where each basis fringe represents one of the spectral regions indexed by  $j$ . These convolutions are equivalent to performing a low-pass filtering. For the spatial dimension  $x$ , the sinc function is approximated by the rectangular envelope of the basis fringe combined with the low-pass process of diffraction.<sup>28</sup> In practice, no ideal low-pass filter exists. In this paper, the basis fringes had Gaussian spectral shapes rather than sinc-function shapes. The resulting spectral cross-talk theoretically added some noise to the image, though little additional noise was observable. Spectrally Gaussian basis fringes were used because they produced superior image results. A properly decoded (recovered) fringe has a smooth continuous spectrum, as shown in Figure 4.

**Bandwidth compression: spectral subsampling.** *Hogel-vector bandwidth compression* reduces the number of encoded symbols through subsampling of hogel spectra. The information content of an encoded fringe, i.e., the number of symbols in the hogel-vector array, is equal to the product of the number of hogels times the number of components ( $N$ ) in each hogel vector. Therefore, the sample spacings ( $w_h$  in space

**Figure 5** A basis fringe and its spectrum



and  $\Delta_f$  in spatial frequency) determine the information content of the encoded fringe. Hogel-vector encoding reduces the number of symbols by sampling hogel spectra in large frequency steps. The amount of bandwidth compression in an encoded fringe is measured by the compression ratio, CR:

$$CR = \frac{\# \text{ samples in final hogel}}{\# \text{ symbols per hogel vector}} \equiv \frac{N_h}{N} = 2w_h\Delta_f \quad (2)$$

where the spectrum is assumed to range from 0.0 to the sampling limit of 0.5 cycles/sample. A  $CR > 1$  means a reduction in required bandwidth: fewer symbols are required to represent a fringe because each symbol represents a larger portion of a hogel spectrum. The spectrum of a hogel comprising  $N_h$  samples is encoded as  $N = N_h / CR$  symbols. (Although the uncompressed spectrum has  $N_h$  samples for magnitude and  $N_h$  for phase, there are only  $N_h$  independent samples because the spectrum of the real-valued fringe has even conjugate symmetry,  $S(f) = S^*(-f)$ .)

Conventional image and data compression starts with the desired data and then encodes the data into a “compressed” format.<sup>29</sup> This compressed format is subsequently decoded into a (sometimes approximate) replica of the desired data. This approach can

be applied to holographic fringes. However, total (model-to-fringe) computation speed is increased by computing the encoded format directly. This “direct-encoding” approach involves only two computation steps: direct encoding and decoding. Hogel-vector encoding is a *direct-encoding* technique, giving it the speed necessary for holovideo interactivity.

**Basis fringes.** Each basis fringe is used to contribute spectral energy with a Gaussian profile centered at  $((i + 0.5) / N)\Delta_f$  for  $i = [0, N - 1]$ , and with a  $1/e^2$  full-width of  $\Delta_f$ . The spectral phase is uncorrelated among the basis fringes to make effective use of dynamic range. Spatially, a basis fringe has a uniform magnitude of 1.0 within the hogel width, and zero elsewhere. The spatial phase contains the diffractive information. The inter-hogel phase continuity is assured by constraining the endpoints of each basis fringe. The many constraints on basis fringes make their computation intractable using analytical approaches. For the present research, nonlinear optimization was used to design each basis fringe to have the desired spectral characteristics.<sup>8,23,24</sup> The synthetic basis fringes for given sampling spacings ( $w_h$  and  $\Delta_f$ ) and a given display (i.e., pitch,  $p$ ) were precomputed and stored for use in hogel-vector decoding. Figure 5 shows a typical basis fringe and its spectrum, which was constrained to a Gaussian region centered at 0.06 (cycles/sample). A final optimization step further refined the Gaussian spectral profile.

**Direct encoding.** Hogel-vector encoding converts a given 3D object scene into a hogel-vector array.

*Diffraction tables.* The mapping from image element ( $x, z$ ) to hogel position and vector component was pre-calculated and stored in a *diffraction table*. Indexing by the horizontal and depth locations ( $x, z$ ) of each element, the diffraction table contains a spatially and spectrally sampled representation of the fringe that diffracts light to form the image element. The table lists which nonzero components of which hogel vectors are needed to generate the correct fringe. In an HPO hologram, each line of the fringe pattern is treated independently, indexed by the vertical ( $y$ ) location of the image element.

During hogel-vector encoding (shown in Figure 6), the diffraction table rapidly maps a given ( $x, z$ ) location of a desired image element (e.g., a point) to components in the hogel-vector array. The diffraction table includes an amplitude factor at each entry. This factor is multiplied by the desired magnitude (taken

from the 3D scene and lighting information) to determine the amounts of contributions to each hogel-vector component. The magnitude of an image element is determined from its desired brightness. This brightness is represented as an intensity that is equal to the square of its magnitude. Therefore, in theory, the square roots of desired brightness values are used when calculating hogel vectors. In practice, however, nonlinearities in the MIT holovideo display systems necessitated a brightness correction approximately equivalent to squaring magnitudes—cancelling out the need to calculate square roots.

During computation of hogel vectors for a particular 3D object scene, each image point is used to index the diffraction table. The contents of the table—for each indexed hogel-vector component—are summed to calculate the total hogel-vector array for this object. This direct-encoding step is fast because it involves only simple calculations.

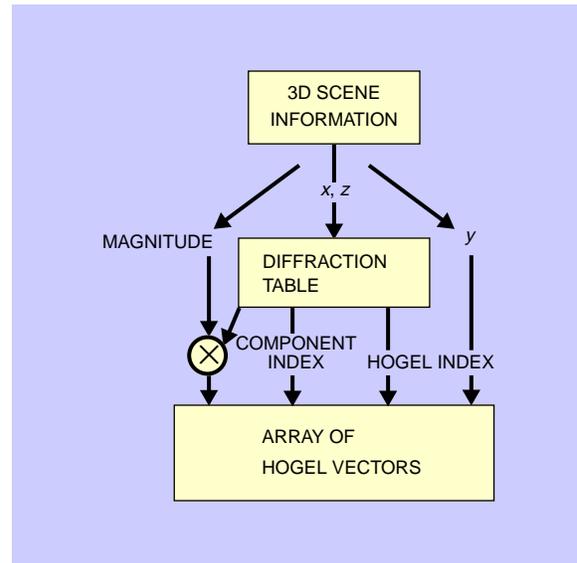
A diffraction table is computed by spatially and spectrally sampling (with spacings  $w_h$  and  $\Delta_f$ ) the continuous spectral (Wigner) distribution at the plane of the hologram ( $z = 0$ ). The spectrum is related to an imaged element through optical propagation (i.e., diffraction).<sup>7,8,25</sup> The Wigner distribution of the desired image element is back-propagated from  $z_p$  to  $z = 0$  using the transport equation of free space.<sup>26</sup> For point image elements—the most common case—the spectral distribution of the fringe is a uniform distribution with

$$\frac{x - x_p}{z_p} = \frac{(f - f_r)}{\sqrt{k^2 - (f - f_r)^2}} \quad k \equiv \frac{2\pi}{\lambda} \quad (3)$$

where  $f_r$  is the center spatial frequency (typically  $p/4$ ). The  $x$ -location of this spectral energy (i.e., which hogels are to contain this spectral energy) is a function of the  $x$ -position ( $x_p$ ) of the image point, and its spread is a function of image point depth ( $z_p$ ). The continuous spectrum was sampled to calculate the value at each discretized location in the spatial and spectral dimensions, i.e., each hogel-vector component. A sampled value was calculated using a bi-Gaussian kernel with  $1/e^2$  full widths of  $w_h$  and  $\Delta_f$  in the spatial and the spectral dimensions.

Figure 7 illustrates the function of a diffraction table for image points. The plane at left represents the spatially varying fringe spectral content. The regular dot grid indicates the discretized spectrum, i.e., a hogel-

**Figure 6** Generating hogel vectors using a diffraction table

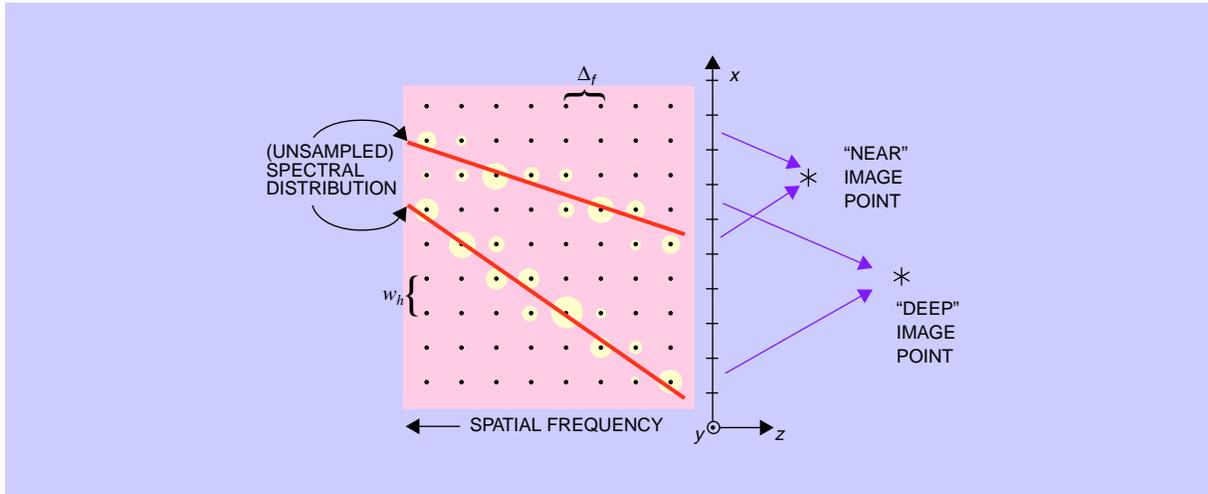


vector array. Each dot is the location of one hogel-vector component. Each diagonal line represents the continuous (approximately linear) Wigner spectrum corresponding to the diffraction of light to an image point. The size of the circular region around a dot indicates the amount of that hogel-vector component required to create the image point. The “deep” image point has a wide range of nonzero hogel-vector contributions. The “near” image point has a narrow range of contributions.

Hogel-vector encoding provides higher-level image elements. For example, if line segments of various sizes are useful for assembling the image scene, then a diffraction table is used to map location, size, and orientation of the desired segment to the proper hogel-vector contributions. Furthermore, the amplitude factors in the diffraction table allow for directionally dependent qualities (e.g. specular highlights) when a diffraction table is used to represent more complex image elements.

*Use of 3D computer graphics rendering.* Another approach to performing direct-encoding employs 3D computer graphics rendering software.<sup>30</sup> This facilitates advanced image properties, such as specular reflections, texture-mapping, advanced lighting models, and scene dynamics. A series of views are ren-

Figure 7 Precomputation of a diffraction table: typical spectral distribution for image points at different depths



dered, each in the direction corresponding to the center of one of the spectrally sampled regions. The view window (the plane upon which the scene is projected) must be coincident with the plane of the hologram ( $z = 0$ ), and the viewpoint must be at  $z \rightarrow \infty$ . Each rendered view is an orthographic projection of the scene from a particular view direction. The rendered views provide a discrete sampling of space and spectrum. These views are converted into a hogel-vector array using either a modified diffraction table or filtering, depending on hardware. In the modified-table method, the picture element (pixel) spacing in the 2D rendering of the scene is half the hogel spacing. This allows for subsampling. A special diffraction table uses view direction and rendered pixel location to select hogel-vector components, providing a sampling of the spatial-spatial-frequency space that matches  $w_h$  and  $\Delta_f$ . The second method uses the features of advanced rendering hardware. Anti-alias filtering combined with nearest-neighbor spatial linear interpolation (as part of a texture-mapping) gives a roughly Gaussian sampling.

*Color.* Full-color holovideo images are produced by computing three separate fringes, each representing one of the additive primary colors—red, green, and blue—taking into account the three different wavelengths used in a color holovideo display.<sup>14</sup> Three separate hogel-vector arrays are generated, and each is decoded using the linear summation of one of three sets of precomputed basis fringes. Basis-fringe selec-

tion via hogel-vector components proceeds using a single diffraction table, with the shorter wavelengths limited to a smaller range of diffraction directions.

**Decoding hogel vectors to hogels.** Hogel-vector decoding is the conversion of each hogel vector into a useful fringe in a hogel region. Decoding performs the convolutions of Equation 1 through a linear summation of basis fringes using the hogel vector as weighting coefficients. To compute a given hogel, each component of its hogel vector is used to multiply the corresponding basis fringe. The decoded hogel is the accumulation of all the weighted basis fringes (as shown in Figure 8). Looking at the array of precomputed basis fringes as a two-dimensional matrix, hogel decoding is an inner product between the basis-fringe matrix and a hogel vector.

Decoding is the more time-consuming step in hogel-vector bandwidth compression. However, the simplicity and consistency of this step means that it can be implemented on specialized hardware and performed rapidly. Various specialized hardware exists to perform multiplication-accumulation (MAC) operations at high speeds.

## Implementation

Direct-encoding of the hogel vectors was implemented on a Silicon Graphics, Inc., Onyx workstation, a high-end serial computer. Encoding involved a

wide range of calculations, but was relatively fast. The second computation step, the decoding of hogel vectors into hogels, was implemented on three systems: the Cheops framebuffer system used to drive the MIT 36-MB holovideo display; the Onyx workstation; and a Silicon Graphics, Inc., RealityEngine2 graphics subsystem.

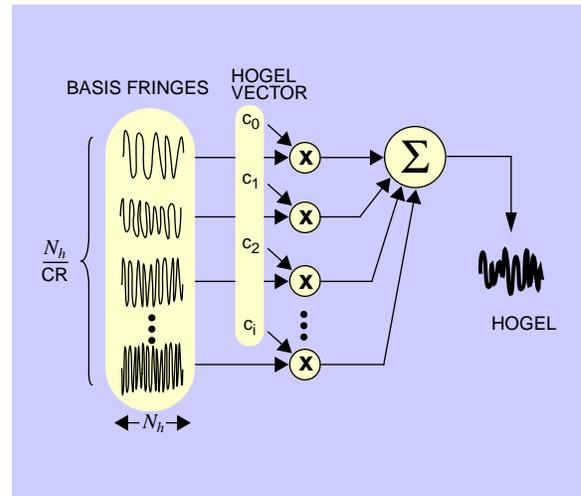
**Implementation on Cheops.** Hogel-vector encoding begins with a 3D image scene description generally consisting of about 0.5 MB of information or less. After the appropriate transformations (e.g., rotations, translations) and lighting, it is direct-encoded as a hogel-vector array. For a compression ratio (CR) of 1:1 (no bandwidth compression), the hogel-vector array comprises 36 MB. For larger compression ratios, this number is proportionally smaller, e.g., a CR = 16:1 gives a 2.2-MB hogel-vector array.

The Cheops image processing system (see Figure 9) is a compact, block data-flow parallel processor designed and built for research in scalable digital television.<sup>31,32</sup> The P2 processor card communicates to the host via a SCSI (small computer standard interface) link with ~1-MB/s bandwidth. Six Cheops output cards provide 18 parallel analog-converted channels of computed fringes to the MIT 36-MB holovideo display. The P2 communicates data to the output cards using the fast Nile Bus. The P2 also supports a type of stream-processing superposition daughter card (the *Splotch Engine*) that performs weighted summations of arbitrary one-dimensional basis functions. The Splotch Engines (two were used in this research) perform the many MAC operations required for the decoding of hogel vectors into hogels.<sup>32</sup> The hogel-vector array was downloaded to the Cheops P2 card, where it was decoded using two Splotch Engines.

The Cheops output cards store each fringe sample as an 8-bit unsigned integer value. Computed fringes are normalized to fit within these 256 values. Normalization generally involves adding an offset and multiplying by a scaling factor. In the hogel-vector technique, normalization is built into the computational pipeline. For example, when using Cheops, the hogel-vector components are pre-scaled to produce useful fringes in the higher 8 bits of the 16-bit result field. Only this high byte is sent to the output cards.

The decoded 36-MB fringe was transferred to the Cheops output cards and used by the MIT 36-MB holovideo display to generate images. Figure 2, seen earlier, shows a general schematic of the MIT holo-

**Figure 8** Decoding hogel vectors to hogels



video displays. After radio frequency (RF) processing, computed fringes (in the form of acoustic waves) traversed the aperture of an AOM (acousto-optical modulator), which phase-modulated a beam of laser light. Two lenses imaged the diffracted light at a plane in front of the viewer. The horizontal scanning system angularly multiplexed the image of the modulated light. The vertical scanning mirror positioned diffracted light to the correct vertical position in the hologram plane. Electronic control circuits synchronized the scanners to the incoming holographic signal.

**Implementation on a serial workstation.** The entire diffraction-specific computation pipeline was also implemented on the SGI Onyx serial workstation. The process for generating a 36-MB fringe was the same as above, except that a simple linear loop performed the decoding step. The computed fringe was downloaded to Cheops to generate images on the 36-MB holovideo display.

**Implementation on a graphics subsystem.** The SGI RealityEngine2 (RE2) is a computer graphics subsystem generally used to render images. The rapid texture-mapping function and the accumulation buffer were used to perform rapid multiply-accumulate operations.<sup>33</sup> The RE2 rendered directionally dependent 2D views of the object scene. These rendered views were converted into a hogel-vector array that was then

Figure 9 Hogel-vector decoding on the Cheops modular framebuffer and image processing system

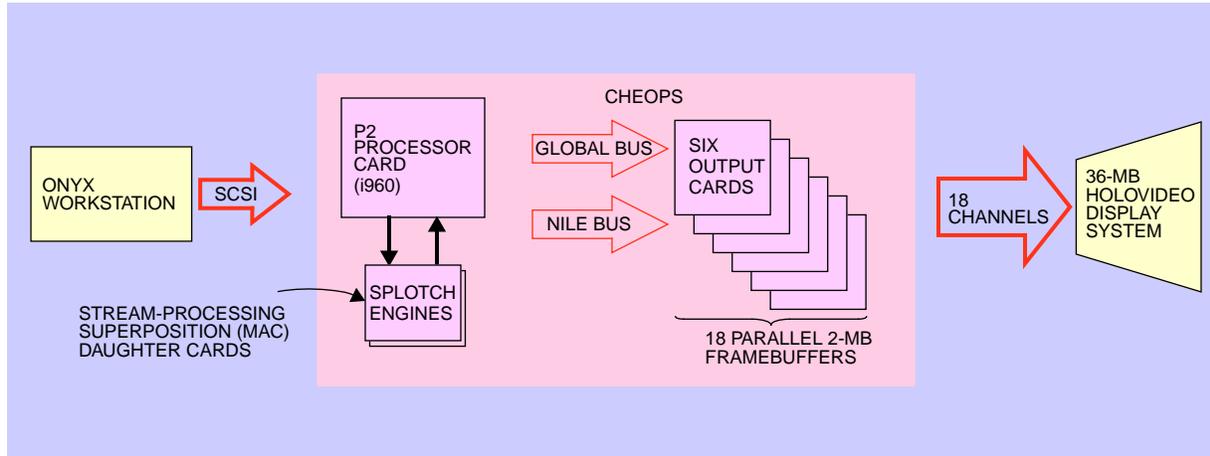
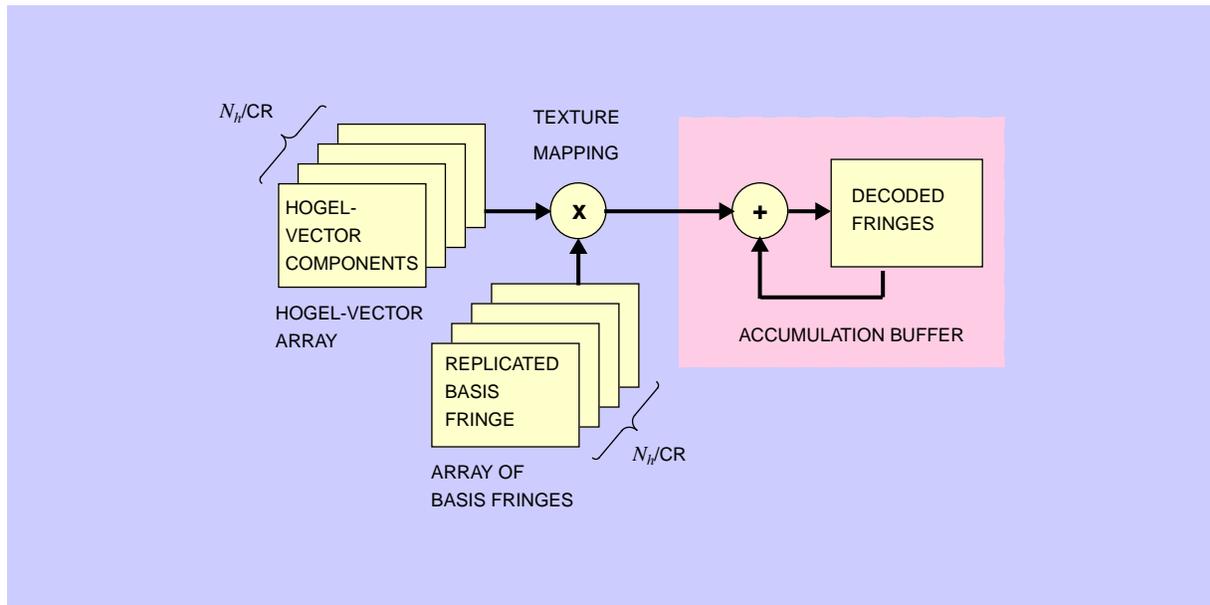


Figure 10 Hogel-vector decoding on the graphics subsystem



decoded in the RE2, as shown in Figure 10. The texture-mapping function rapidly multiplied a component from each hogel vector by a replicated array of a single basis fringe. This operation is repeated  $N_h/CR$  times, once for each hogel-vector component, accumulating the result in the accumulation buffer. Transfer times were negligible because all computations occurred inside the graphics subsystem that included

the framebuffer to drive the display. For  $CR = 32:1$ , a 2-MB fringe was decoded from a 64-KB hogel-vector array for each of three colors.

### Model of point spread

A fringe generated using hogel-vector bandwidth compression generally loses some of its ability to pro-

duce a sharp image. A given image point appears slightly broadened or blurred. The increased point-spread results from several processes:

- Aperturing due to spatial sampling,  $blur_{spatial}$
- Spectral sampling blur due to sparsely sampled hogel spectra,  $blur_{spectral}$
- Aberrations in the display,  $blur_{displ}$
- Quantization and other noise

The first blur component is the width of a beam of light diffracted to  $z$  (mm) from a  $w_h$ -wide aperture:

$$blur_{spatial} = \left( w_h^2 + \left( \frac{z\lambda}{\pi w_h} \right)^2 \right)^{1/2}. \quad (4)$$

Contribution to blur from spectral sampling varies with  $\Delta_f = pB/N$ , and is approximated by

$$blur_{spectral} = z\lambda pB \frac{CR}{N_h} \quad (5)$$

where  $B$  is the total spectral bandwidth in cycles/sample, and  $p$  is the fringe sampling pitch in samples/mm.

As the number of symbols per hogel  $N = N_h/CR$  decreases, spectral sample spacing increases; each hogel-vector component carries information about a wider region of the hogel spectrum, limiting the achievable image resolution.

The spectral sampling blur and aperture blur add geometrically with other sources of blur. Blur caused by the display was measured for various  $z$  locations in the image volume. Additional contributions to point spread are for now neglected. Combining gives

$$blur = \left[ w_h^2 + \left( \frac{z\lambda}{\pi w_h} \right)^2 + \left( z\lambda pB \frac{CR}{N_h} \right)^2 + blur_{displ}^2 \right]^{1/2} \quad (6)$$

as the model for total blur or point spread.

**Experimental verification of model.** To verify the point-spread model of Equation 6, a series of experiments was performed. Pictures of individual imaged points were used to measure point spread as a function of the spatial and spectral sampling parameters,  $w_h$  and  $\Delta_f$ , and to compare these data to the model for point spread. For the deepest points at  $z = 80$  mm and for smaller depths, the point-spread model fit very well to the measured data.

Figure 11 A point imaged at  $z = 80$  mm with CR = 1:1

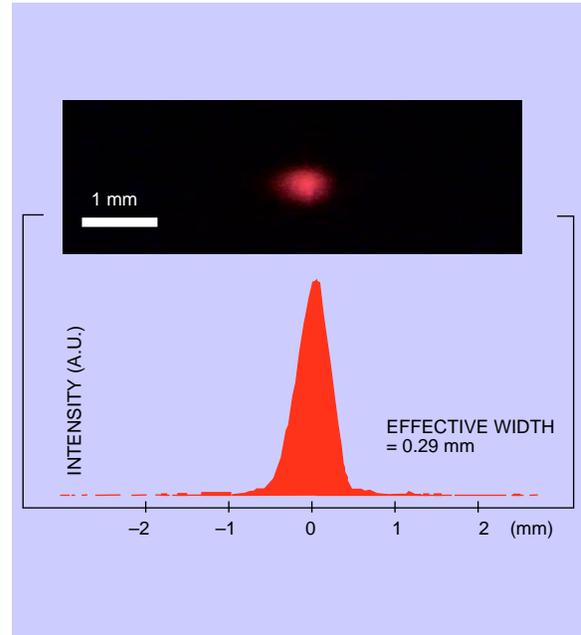
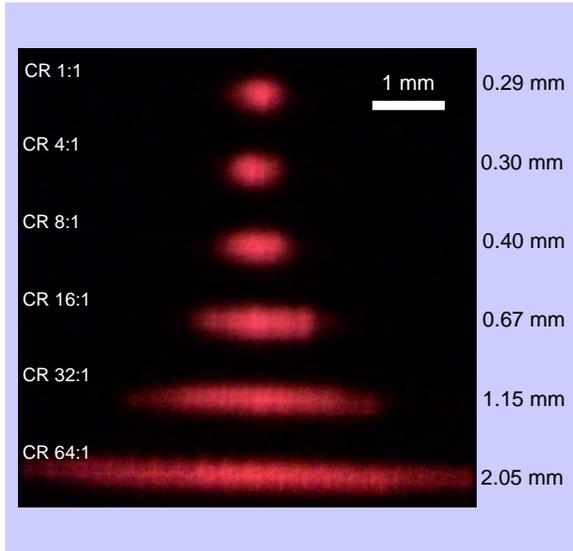


Figure 11 shows a point image focused at 80 mm in front of the plane of the hologram, i.e., focused between the hologram plane and the viewer. The fringe used to generate this image was computed using hogel-vector bandwidth compression (CR = 1:1, hogel width  $N_h = 512$  samples or  $w_h = 0.3$  mm). When compared to a conventionally computed point image, there is no additional noise or other artifact. The graph shows a cross section of the focused point, grabbed using a small tricolor CCD (charge coupled device) array placed at the location of an imaged point. A horizontal profile was obtained from the digitized photograph by integrating over the vertical extent of the red image component. The effective (horizontal) width of the imaged point was measured by calculating the narrowest horizontal region containing half of the total energy.

The point shown in Figure 11 was imaged using the MIT 36-MB display. (Fringe sampling pitch was  $p = 1.7 \times 10^3$  samples/mm.) Points imaged at  $z = 80$  mm to test the worst-case resolution of the MIT 36-MB display are shown in Figure 12. These (and many other test points) were profiled and measured (using the same half-energy method) for a range of compression ratios. Blur increases with increasing CR, i.e.,

**Figure 12 Hogel-vector bandwidth compression: effective width (horizontal point spread) vs CR**



with  $\Delta_r$ . For CR = 8:1 or less, the point is still sharp. For CR = 16:1, the effective width increases such that it is easily seen by a human viewer. Hogel width was  $N_h = 512$  samples or  $w_h = 0.3$  mm.

**Model-based selection of system parameters.** The model for point spread, Equation 6, provides an analytical expression that relates the various parameters of the holovideo system. This expression is used to select certain parameters—such as hogel width and compression ratio—given other parameters, such as desired image resolution  $\delta$  and image depth  $Z$ . Let  $\delta$  be the maximum allowable image point spread.  $N = N_h / CR$  is a measure of bandwidth in symbols/hogel. In Equation 6, point spread can be minimized as

$$\delta^2 = Z\lambda \left[ \sqrt{c} + \frac{1}{\pi^2 \sqrt{c}} + \frac{(B \cdot CR)^2}{\sqrt{c}} \right] \quad (7)$$

$$c \equiv \frac{1}{\pi^2} + (B \cdot CR)^2$$

when an optimal value of  $w_h$  is chosen. Because full sampling bandwidth is often utilized,  $B = 0.5$  (cycles/sample) is assumed. Assuming  $B \cdot CR \gg 1 / \pi$  (CR is usually greater than one), Equation 7 simplifies to

$$\delta \approx \frac{Z}{N} \left( \frac{\lambda p}{\sqrt{2}} \right) \quad (8)$$

given the selection of an optimal hogel width

$$w_h = \delta / \sqrt{2}. \quad (9)$$

For practical imaging, blur must be below the amount perceivable to humans<sup>8</sup>—about 0.18 mm at a typical viewing distance of 600 mm.

**Trade-offs: bandwidth, depth, and resolution.**

Equation 8 relates the salient parameters of a holovideo imaging system: bandwidth, image depth, and image resolution. It can be used to design a bandwidth-efficient holovideo system. First, an optimal hogel width is chosen according to Equation 9. Next, one of the fundamental system parameters can be calculated given the other system parameters. For example, if the image resolution  $\delta$  and maximum image depth  $Z$  are fixed, the minimum required bandwidth, recalling Equation 8, is

$$\text{bandwidth: } N \geq \frac{Z}{\delta} \left( \frac{\lambda p}{\sqrt{2}} \right) \quad (10)$$

Alternately, if image depth or image resolution are the unspecified parameters, they can be calculated from the given parameters using the following expressions:

$$\text{depth: } Z \leq \delta N \left( \frac{\sqrt{2}}{\lambda p} \right) \quad (11)$$

$$\text{resolution: } \delta \geq \frac{Z}{N} \left( \frac{\lambda p}{\sqrt{2}} \right) \quad (12)$$

Furthermore, the attainable compression ratio can be predicted given the system parameters. The equality of Equation 10 can be recast as

$$CR = \frac{\delta^2}{\lambda Z}. \quad (13)$$

This result has several implications. First, sacrificing image resolution allows for a dramatic increase in compressibility. Second, deeper images allow for less compressibility. Finally, a shorter wavelength of light allows for higher compressibility—commensurate with its higher diffraction-limited resolution that wastes more bandwidth.

**Visual-bandwidth holography.** Traditionally, electro-holographic bandwidth is wasted due to the limited performance of the human visual system (HVS). However, Equation 10 shows that hogel-vector encoding compresses bandwidth to match the amount of 3D image information that is useful to the human visual system. A simple measure of useful visual information is the number of volume elements (*voxels*) contained in the image volume. To illustrate, let  $Z = 80$  mm,  $\delta = \sqrt{2} \cdot 0.2 = 0.28$  mm, and  $\lambda p = 1.1$ . The hogel-vector bandwidth (Equation 10) is  $N = 220$  symbols/hogel. By dividing the image volume into voxels with lateral and depth resolutions that match the acuity of the HVS, the maximum amount of useful visual information is 213 voxels/hogel. Thus, hogel-vector encoding achieves a minimum bandwidth in a visual information sense. By matching bandwidth to the abilities of human eyes, hogel-vector bandwidth compression achieves a new type of holographic imaging: *visual-bandwidth holography*.

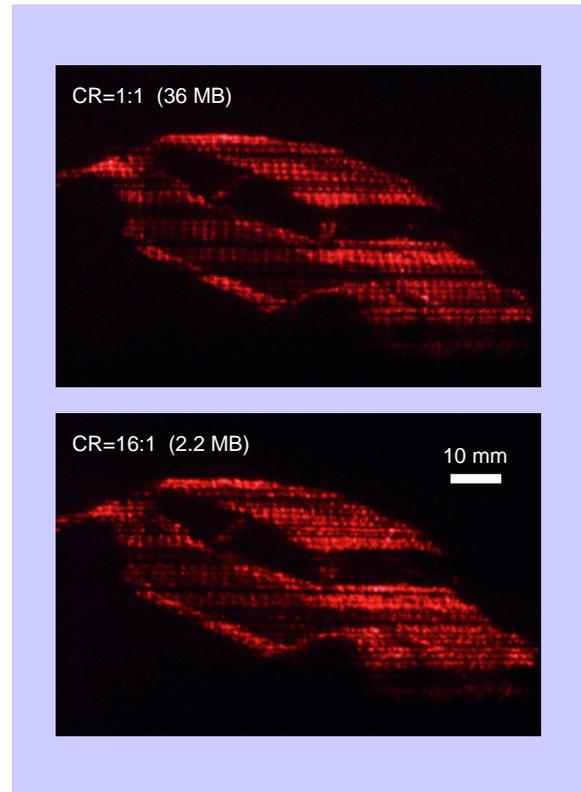
### Imaging results

Fringes were computed using hogel-vector bandwidth compression and used to generate a variety of 3D images on the MIT holovideo displays. Speeds were measured. Digital photographs were taken of images.

Images were generated for a variety of values of  $w_h$  and CR. For properly selected values, point spread was not perceivable. Figure 13 shows a typical result. The upper picture was generated from a 36-MB fringe decoded ( $N_h = 1024$ ) from a 36-MB hogel-vector array, i.e., CR = 1:1. The 3D image (a Volkswagen Beetle car) was generated from a polygon database comprising 1079 polygons. The image was converted into about 10000 discrete image points using a simple lighting model. The lower picture shows the same object, computed using 2.2-MB hogel-vector array, i.e., CR = 16:1. The use of only 1/16 the bandwidth causes only slight changes in the image. The discrete image points are just visibly blurred, and a slightly speckle-like appearance is added. Note that picture quality suffered from artifacts present in the display. Imbalances and nonlinearities in the RF signal-processing electronics of the display system produced the unwanted horizontal streaks and bands of light and dark.

The MIT 6-MB color holovideo display was used to generate images computed using hogel-vector bandwidth compression. Hogel-vector direct-encoding and decoding were performed by the RE2, using three sets

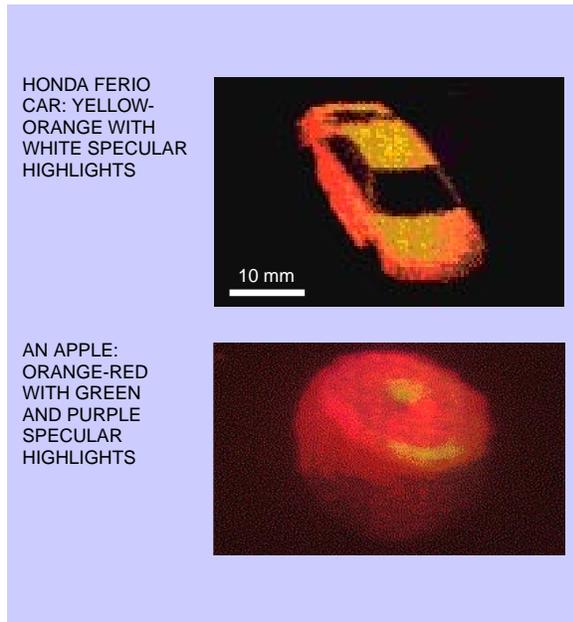
**Figure 13** 36-MB holovideo image: two compression ratios



of basis fringes, each precomputed for the specific wavelength used in the full-color display ( $p = 1000$  mm<sup>-1</sup>). Three 2-MB fringes were decoded from three 64-KB hogel-vector arrays (one per color). The resulting images—computed at interactive rates—showed good quality and possessed the full range of lighting features, e.g., specular highlights and transparency. Figure 14 shows photographs of typical full-color images, computed with a hogel width of 256 samples (0.250 mm). These images were sharp despite the high CR = 32:1.

Earlier work<sup>33</sup> reported the use of the RE2 to compute stereogram-style holograms. The stereogram images had noticeable blur and artifacts, especially when computed rapidly. In comparison, hogel-vector bandwidth compression maintained image fidelity, even at the high compression ratios (CR = 32:1) necessary to achieve computation at interactive rates. These images did not exhibit the artifacts of vertical dark

**Figure 14** 6-MB full-color holovideo images: car and apple



stripes or jumps. And, of course, images computed using the diffraction-specific technique produced real 3D images.

**Spatial coherence.** Hogel-vector bandwidth compression added a noticeable speckle-like appearance to the image. These brightness variations at infinity likely resulted from the use of coherent light in the display. Diffraction-specific fringe computation assumes that light is quasi-monochromatic with a coherence length  $L_c < w_h/2 \sim 0.1$  mm. This ensures that the diffracted light adds linearly with intensity. In practice, the effective coherence length of light in the holovideo displays was approximately 2.0 mm. To reduce the speckle effect, a random set of phases was introduced into each hogel via the basis fringes. This reduced interhogel correlation. Implementation employed a set of 16 different but spectrally equivalent precomputed basis fringes. To decode a given hogel vector, each basis fringe was selected at random from the set of equivalent basis fringes.

### Speed

Hogel-vector bandwidth compression achieved an increase in speed by a factor of over 70 compared to

conventional interference-based methods. Computing times were measured on three platforms: the Onyx workstation (alone); the Onyx (for hogel-vector generation) and the Cheops with two Splotch Engines (for decoding); and the Onyx/RE2. The results from three computational benchmarks are described: a conventional interference-based technique, and two diffraction-specific cases in which hogel-vector compression ratios are CR = 1:1 and CR = 32:1.

Using the hogel-vector technique, total computation time consists of the initial direct-encoding step, the time to transfer the hogel-vector array to the decoding system, and the hogel-vector decoding step. The first step—generation of the hogel-vector array on the Onyx workstation—was very fast. For most objects, typical times were 10 seconds for CR = 1:1 and 0.5 seconds for CR = 32:1. The downloading of the hogel-vector array over the SCSI link was slow. However, the use of hogel-vector bandwidth compression (CR = 32:1) reduced data transfer of the 1.1-MB hogel-vector array to only 1.0 second.

Appropriate scene complexities were chosen to ensure equivalent benchmarks. For hogel-vector bandwidth compression, speed is basically independent of image scene complexity, whereas the computing time for interference-based ray-tracing computations varies roughly linearly.

**Speed on a serial workstation.** The conventional interference-based method used was to sum the complex wavefronts from all object points (plus the reference beam) to generate the fringe. Because it involved complex-valued, floating-point precision calculations, it was not implemented on either of the specialized hardware platforms (Cheops/Splotch, or the RE2). A fairly complex image of 20000 discrete points (roughly 128 imaged points for each line of the fringe) was used. Implemented completely on the Onyx workstation, a 36-MB fringe required 23000 seconds (over 6 hours). This timing was extrapolated by computing a representative 2-MB fringe.

For comparison, hogel-vector bandwidth compression was implemented on the Onyx workstation. For a 36-MB fringe computed using CR = 1:1 (and  $N_h = 1024$ ), total time was 9600 seconds. For CR = 32:1, the time was reduced to only 300 seconds. Including the time for hogel-vector generation (0.5 seconds), this represents a speed increase of 74 times compared to the conventional computing method. Another advantage of hogel-vector bandwidth compression is that the

**Table 1 Computation times for different hardware and techniques**

Platform, Fringe Size	Conventional (seconds)	Hogel-vector Bandwidth Compression	
		CR = 1:1 (seconds)	CR = 32:1 (seconds)
Workstation, 36 MB	23000	9600	300.5
Cheops/2-Splotch, 36 MB	—	200	6.5
RE2, 6 MB	—	28	0.9

Transfer times are not included. Hogel-vector times are worst-case, and include encoding and decoding.

simplicity of the slower decoding step allows for its implementation on very fast specialized hardware.

**Speed on Cheops.** When implemented on the Cheops system containing two Splotch Engines, hogel-vector decoding time was 190 seconds for CR = 1:1 and only 6 seconds for CR = 32:1. These timings are worst case, i.e., most complex image, measured using a fully nonzero hogel-vector array. In practice, typical image scenes produced many zero-valued hogel-vector components. Skipping zero-valued components resulted in faster decoding times. Typical test images ( $N_h = 1024$ , CR = 32:1) were closer to 3 seconds.

The total hogel-vector encoding and decoding time in the case of CR = 32:1 was 6.5 seconds, worst case. Although it is not quite fair to compare this to the conventional method implemented only on the workstation, the relative speed increase of over 3500 times is made possible by the simplicity and efficiency of the hogel-vector decoding algorithm.

**Speed on a graphics subsystem.** When implemented on the RE2 graphics subsystem, for a 6-MB (2-MB per color) fringe, hogel-vector decoding time was 28 seconds for CR = 1:1 and only 0.9 seconds for CR = 32:1. The texture-mapping function of the RE2 graphics subsystem and its accumulation buffer performed rapid multiplications and additions.<sup>33</sup> The 0.9 seconds is a total time, from model to image, since the MIT color holovideo display was driven directly by the RE2. Conventional methods cannot utilize this specialized hardware nor achieve interactive-rate fringe computation.

**Analysis of speed.** In hogel-vector bandwidth compression, the decoding step required the great majority of computing time. To decode an  $N_h / CR$ -

component hogel vector to an  $N_h$ -sample hogel requires calculating an inner product:  $N_h^2 / CR$  multiplication-accumulation operations (MACs). For example, for a 36-MB fringe, a hogel width of  $N_h = 1024$ , and a CR = 32:1, the decoding step requires 1.2 GMACs, i.e., over 1 billion multiplies and adds. Note (from Table 1) that the speed increase from CR = 1:1 to CR = 32:1 is about 32 times. This was due to the reduction by a factor of  $1 / CR$  in the number of time-consuming MAC calculations required. Because each fringe sample requires  $N_h / CR$  MACs, the speed of hogel-vector decoding increases linearly with CR. Faster speeds can be achieved by sacrificing image quality.

Further speed increase during interactivity was achieved by exploiting the scalability of a hogel-vector array, i.e., its ability to supply variable degrees of precision as required. For example, in one interactive demonstration of the 6-MB display, a subset of the hogel-vector array was decoded to produce a “quick and dirty” image that was subsequently replaced by the full-fidelity image when interactivity ceased.

## Conclusion

Hogel-vector bandwidth compression, a diffraction-specific fringe computation technique, has been implemented and used to generate complex 3D holographic images for interactive real-time display. The application of well-known sampling concepts to the localized fringe spectrum has streamlined, generalized, and greatly accelerated computation. Hogel-vector bandwidth compression makes efficient use of computing resources. The slower decoding step is essentially independent of image content and complexity, and simple enough to be implemented in specialized hardware.

Hogel-vector bandwidth compression is the first reported technique for computational holographic bandwidth compression. It achieves a bandwidth compression ratio of 16:1 without conspicuous degradation of the image, thereby eliminating transmission bottlenecks. Fringe generation was 74 times faster than conventional computation. This technique provides a superior means for generating holographic fringes—even if bandwidth compression is not needed. Hogel-vector encoding provides a foundation for “fringelet encoding” which (as reported elsewhere<sup>8,34</sup>) achieves further speed increases. It can also be applied to full-parallax holographic imaging.<sup>8</sup>

Hogel-vector bandwidth compression can be applied to other tasks. For example, holographic movies can be hogel-vector-encoded for digital recording and transmission over networks or television cable. As another example, diffraction-specific fringes have been recorded onto film to produce static holographic images. Hogel-vector bandwidth compression provides the speed and portability required to generate large fringes for a holographic “fringe printer.”

The analysis of hogel-vector bandwidth compression has revealed a simple expression relating the fundamental system parameters of bandwidth, image resolution, and maximum image depth. Hogel-vector encoding attains “visual-bandwidth holography”—it frees holographic fringes from the enormous bandwidths required by the physics of optical diffraction. Bandwidth is instead matched to the abilities of the human visual system.

## Acknowledgments

This research was supported in part by the Defense Advanced Research Projects Agency (DARPA) through the Rome Air Development Center (RADC) of the Air Force System Command (contract F30602-89-C0022) and through the Naval Ordnance Station, Indian Head, Maryland (contract N00174-91-C0117); by the Television of Tomorrow (TVOT) research consortium of the Media Laboratory, MIT; by Honda Research and Development Co., Ltd.; by NEC Corporation; by International Business Machines Corp.; and by General Motors.

The “Onyx” workstation and the “RealityEngine2” graphics framebuffer system were manufactured by Silicon Graphics, Inc., Mountain View, California.

The author gratefully acknowledges the support of Stephen A. Benton and researchers in the Spatial

Imaging Group and in the MIT Media Laboratory past and present: Pierre St.-Hilaire (display optics and electronics), Wendy J. Plesniak (image processing and authoring), Michael Halle (image processing and computational support), Tinsley A. Galyean (computational collaboration), and the Cheops artisans: Carlton J. Sparrell, John Watlington, V. Michael Bove, Jr., and Shawn Becker. Thanks also to the reviewers.

## Cited references

1. M. McKenna and D. Zeltzer, “Three-Dimensional Visual Display Systems for Virtual Environments,” *Presence: Teleoperators and Virtual Environments* **1**, No. 4, 421–458 (1992).
2. D. Gabor, “Microscopy by Reconstructed Wavefronts,” *Proceedings of the Royal Society A* **197** (1949), pp. 454–487.
3. P. Hariharan, *Optical Holography*, Cambridge University Press, Cambridge, MA (1984).
4. B. R. Brown and A. W. Lohmann, “Complex Spatial Filtering with Binary Masks,” *Applied Optics* **5**, No. 6, 967–969 (June 1966).
5. W. J. Dallas, “Computer Generated Holograms,” *The Computer in Optical Research*, B. R. Frieden, Editor, Vol. 41 of Springer series *Topics in Applied Physics*, Springer-Verlag, Berlin (1980), pp. 291–366.
6. D. Leseberg, “Computer-Generated Holograms: Display Using One-Dimensional Transforms,” *Journal of the Optical Society of America A* **3**, No. 11, 1846–1851 (November 1986).
7. D. Leseberg, “Computer-Generated Three-Dimensional Image Holograms,” *Applied Optics* **31**, No. 2, 223–229 (10 January 1992).
8. M. Lucente, *Diffraction-Specific Fringe Computation for Electro-Holography*, Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA (Sept. 1994).
9. M. Lucente, “Optimization of Hologram Computation for Real-Time Display,” *SPIE Proceedings #1667 Practical Holography VI*, SPIE, Bellingham, WA (1992), pp. 32–43.
10. M. Lucente, “Interactive Computation of Holograms Using a Look-up Table,” *Journal of Electronic Imaging* **2**, No. 1, 28–34 (January 1993).
11. P. St.-Hilaire, M. Lucente, and S. A. Benton, “Synthetic Aperture Holography: A Novel Approach to Three Dimensional Displays,” *Journal of the Optical Society of America A* **9**, No. 11, 1969–1977 (November 1992).
12. D. Psaltis, E. G. Paek, and S. S. Venkatesh, “Optical Image Correlation with a Binary Spatial Light Modulator,” *Optical Engineering* **23**, No. 6, 698–704 (1984).
13. 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* **11**, No. 11, 748–750 (November 1986).
14. P. St.-Hilaire, S. A. Benton, M. Lucente, and P. M. Hubel, “Color Images with the MIT Holographic Video Display,” *SPIE Proceedings #1667 Practical Holography VI*, SPIE, Bellingham, WA (1992), pp. 73–84.
15. M. Lucente, P. St.-Hilaire, S. A. Benton, D. Arias, and J. A. Watlington, “New Approaches to Holographic Video,” *SPIE Proceedings #1732 Holography '92*, SPIE, Bellingham, WA (1992), pp. 377–386.
16. P. St.-Hilaire, “Scalable Optical Architecture for Electronic Holography,” *Optical Engineering* **34**, No. 10, 2900–2911 (October 1995).

17. E. Leith, J. Upatnieks, K. Hildebrand, and K. Haines, "Requirements for a Wavefront Reconstruction Television Facsimile System," *Journal of the Society of Motion Picture and Television Engineers* **74**, 893–896 (1965).
18. C. B. Burckhardt, "Information Reduction in Holograms for Visual Display," *Journal of the Optical Society of America* **58**, No. 2, 241–246 (February 1968).
19. K. A. Haines and D. B. Brumm, "Holographic Data Reduction," *Applied Optics* **7**, No. 6, 1185–1189 (June 1968).
20. B. P. Hildebrand, "Hologram Bandwidth Reduction by Space-Time Multiplexing," *Journal of the Optical Society of America* **60**, No. 2, 259–264 (February 1970).
21. C. B. Burckhardt and L. H. Enloe, "Television Transmission of Holograms with Reduced Resolution Requirements on the Camera Tube," *Bell System Technical Journal* **48**, 1529–1535 (May-June 1969).
22. L. H. Lin, "A Method of Hologram Information Reduction by Spatial Frequency Sampling," *Applied Optics* **7**, No. 3, 545–548 (March 1968).
23. O. Bryngdahl and F. Wyrowski, "Digital Holography—Computer-Generated Holograms," *Progress in Optics* **28**, 1–86, E. Wolf, Editor, North-Holland Publishing Co., Amsterdam (1990).
24. J. R. Fienup, "Iterative Method Applied to Image Reconstruction and to Computer-Generated Holograms," *Optical Engineering* **19**, No. 3, 297–305 (March 1980).
25. P. C. Clemmow, *The Plane Wave Spectrum Representation of Electromagnetic Fields*, Pergamon Press, New York (1966).
26. M. J. Bastiaans, "Use of the Wigner Distribution Function in Optical Problems," *Proceedings of SPIE #492 Optical Systems and Applications* (1984), pp. 251–262.
27. A. J. Jerri, "The Shannon Sampling Theorem—Its Various Extensions and Applications: A Tutorial Review," *Proceedings of the IEEE* **65**, No. 11 (November 1977), pp. 1292–1304.
28. J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, New York (1968).
29. A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press, New York (1988).
30. J. D. Foley, A. Van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Co., Reading, MA (1990).
31. V. M. Bove, Jr. and J. A. Watlington, "Cheops: A Modular Processor for Scalable Video Coding," *Proceedings of SPIE #1605 Visual Communications and Image Processing '91 Conference*, SPIE, Bellingham, WA (1991), pp. 886–893.
32. J. A. Watlington, M. Lucente, C. J. Sparrell, V. M. Bove, and I. Tamitani, "A Hardware Architecture for Rapid Generation of Electro-Holographic Fringe Patterns," *Proceedings of SPIE #2406 Practical Holography IX* (1995), pp. 172–183.
33. M. Lucente and T. A. Galyean, "Rendering Interactive Holographic Images," *Proceedings of SIGGRAPH 95*, Los Angeles, CA (August 6–11, 1995). In *Computer Graphics Proceedings*, Annual Conference Series, ACM SIGGRAPH (1995), pp. 387–394.
34. M. Lucente, "Holographic Bandwidth Compression Using Spatial Subsampling," *Optical Engineering* **35**, No. 6, 1529–1537 (June 1996).

*Accepted for publication April 8, 1996.*

**Mark Lucente** IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: [lucente@watson.ibm.com](mailto:lucente@watson.ibm.com); URL: <http://www.watson.ibm.com/people/l/lucente/>). Dr. Lucente joined the IBM Thomas J. Watson Research Center in 1995 as a member of the

Imaging Science and Technology group. He combines knowledge of optics, spatial light modulation, computation, visual perception, and communication systems to develop electro-holography into a practical medium and to explore "visualization spaces," i.e., new paradigms for scientific visualization and information management. Dr. Lucente teaches synthetic 3D imaging at Columbia University in New York where he is adjunct assistant professor of computer science. He worked for six years in the MIT Media Lab Spatial Imaging Group, where he and his colleagues invented holovideo. His innovative computational algorithms enabled the world's first interactive holographic imaging in 1990. He earned the degrees of Ph.D., S.M., and S.B. from the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, working in a variety of fields: 3D imaging systems, lasers in high-bandwidth communication systems, ultrashort laser pulses, and semiconductor device physics using nonlinear optical phenomena. Dr. Lucente is a member of ACM, OSA, SPIE, Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.

Reprint Order No. G321-5610.