Published, Optical Engineering, Special Section on Electronic Holography, 1996 June..

Holographic bandwidth compression using spatial subsampling

Mark Lucente
MIT Media Laboratory

current location:
International Business Machines Corporation
IBM T. J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598 USA
lucente@watson.ibm.com
URL: http://www.watson.ibm.com/people/l/lucente/
tel: 914.945-2980
fax: 914.945-4001

Abstract. A novel electro-holographic bandwidth compression technique, fringelet bandwidth compression, is described and implemented. This technique uses spatial subsampling to reduce the bandwidth and complexity of holographic fringe computation for real-time 3-D holographic displays. Fringelet bandwidth compression is a type of diffraction-specific fringe computation, an approach that considers only the reconstruction process in holographic imaging. The fringe pattern is treated as a spectrum that is sampled in space (as holographic elements or "hogels") and in spatial frequency (as "hogel vectors"). Fringelet bandwidth compression achieves a compression ratio of 16:1 without conspicuously degrading image quality. Further increase in compression ratio and speed is possible with additional image degradation. Fringelet decoding is extremely simple, involving the replication of fringelet sample values. This simplicity enables an overall increase in fringe computation speed of over 3000 times compared to conventional interference-based methods. The speed of fringelet bandwidth compression has enabled the generation of images at nearly interactive rates: under 4.0 s per hand-sized (one-liter) 3-D image generated from a 36-Mbyte fringe.

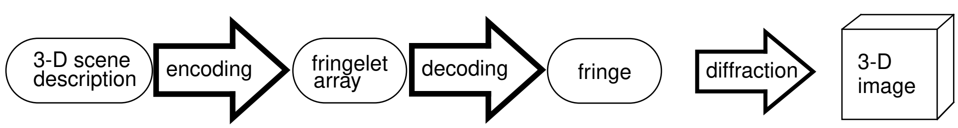
Subject terms: electro-holography, holographic computation, holographic bandwidth compression, diffraction-specific, fringelet, hogel vector

1 Introduction

Electro-holography - also called *holovideo* - is a new visual medium that electronically produces three-dimensional (3-D) 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,[1] promising numerous applications in the creation and manipulation of information, including education, entertainment, telepresence, medical imaging, interactive design, and scientific visualization. Optical holography[2] is used to create 3-D images by recording an interference pattern (*fringe pattern* or simply *fringe*) and subsequently using the fringe to diffract light to form a 3-D image. Both the computation and display of synthetic holographic images are difficult due to huge fringe bandwidths.[3-9] When researchers at the MIT Media Laboratory Spatial Imaging Group created real-time 3-D holography,[10] fringe computation required several minutes for small simple images using conventional computation methods. Recent progress in *diffraction-specific fringe computation* increased speed dramatically using hogel-vector bandwidth compression.[7,8]

A new technique based on diffraction-specific computation[7] called "fringelet bandwidth compression" achieves interactive-rate holographic computation. As demonstrated in this paper, fringelet bandwidth compression (illustrated in Figure 1) reduces computational bandwidth by a ratio of 16:1 and higher, allowing for easier display, transmission, and storage. This technique also achieves fringe computation that is over 3000 times faster than conventional interference-based computation. A background section is followed by a review of diffraction-specific fringe computation, a description of fringelet bandwidth compression, its implementation, and experimental results.

Figure 1. Fringelet Bandwidth Compression



2 Background

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

2.1 Computational Holography

Computational holography[3] begins with a 3-D numerical description of the object or scene to be imaged. Conventional holographic computation imitated the interference of optical holographic recording. Speed was limited by two fundamental properties of fringes: (1) the numerous samples required to represent microscopic features, and (2) the computational complexity associated with the physical simulation of light propagation and interference.

In a computer-generated fringe, the number of samples per unit length (in one dimension) is the pitch, p. To satisfy fringe sampling requirements, the pitch is chosen as $p \geq \frac{4}{\lambda}\sin\frac{\Theta}{2}$ where Θ is the range of angles of diffraction (i.e., the width of the viewing zone) and λ is the wavelength of light used[6]. A typical full-parallax 100 mm $\times$ 100 mm hologram has a sample count (also called "space-bandwidth product," or simply "bandwidth") of over 100 gigasamples. The elimination of vertical parallax provided savings in display complexity and computational requirements without greatly compromising display performance.[6] This paper deals with horizontally off-axis transmission horizontal-parallax-only (HPO) holograms.[2] Such an HPO fringe is commonly treated as a vertically stacked array of one-dimensional holographic lines[4] (called *hololines)*. Fringelet bandwidth compression however can be used to compute full-parallax holograms. (See the reference[7] pp. 143-144).

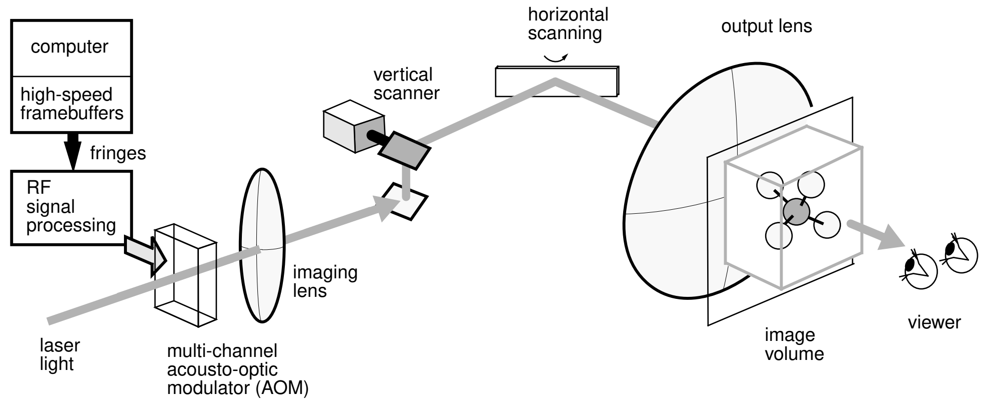
A common approach to the computation of holographic fringes resembles 3-D computer graphics ray-tracing. Complex wavefronts from object elements are summed with a reference wavefront to calculate the interference fringe. Interference-based computation requires myriad complex arithmetic operations (including trigonometric functions and square roots), making rapid computation impossible[6] and bandwidth compression difficult.

Bipolar intensity was developed to simplify fringe computation. As discussed in the reference,[6] in common circumstances the real part of the object wavefront substitutes for the intensity pattern resulting from interference with reference light. Therefore, linear summation of fringes is possible. Real-valued summation enabled the efficient use of precomputed elemental fringes, an approach that, when implemented on a supercomputer, achieved (for the first time) hologram computation at interactive rates[6].

2.2 Holographic Displays

Holographic displays modulate light with electronically generated fringes. The research presented in this paper employed the 36-Mbyte holovideo display[11,12] developed by the Spatial Imaging Group at the MIT Media Laboratory, which used the time-multiplexing of a fast optical modulator[10]. Two 18-channel acousto-optic modulators (AOMs) and a series of lenses and scanning mirrors assembled a 3-D holographic image in real time. Figure 2 shows a schematic of the MIT holovideo display. Computed fringes traverse the AOM as acoustic waves, phase-modulating a beam of laser light. Two lenses image the diffracted light at a plane in front of the viewer. The horizontal scanning system angularly multiplexes the image of the modulated light. The vertical scanning mirror positions hololines vertically. The viewer sees a (red) real 3-D image, 150 mm wide by 75 mm tall by 160 mm deep, in front of the output lens. Fringe sampling pitch was p=1.7x10^3 samples/mm. By incorporating the proper physical parameters, wavelengths, and sampling pitch, fringelet bandwidth compression has been used for other holographic displays[7,8].

Figure 2. MIT Holovideo Display: Schematic



2.3 Optical Holographic Bandwidth Compression

Holographic fringes contain more information than can be utilized by the human visual system[13,14]. Several researchers attempted to reduce bandwidth in optical holographic imaging. By employing a dispersion plate, Haines and Brumm[15] used a hologram of reduced size to generate an image that was not reduced in size or in viewing zone. However, image quality suffered. Either image resolution or signal-to-noise ratio was reduced. Hildebrand[16] generalized the dispersion-plate approach. Burckhardt and Enloe[14,17] reduced the information recorded in a hologram by exposing an array of small regularly spaced areas. This work was equivalent to spatially subsampling the hologram. The reconstructed image had an annoying "screen" artifact. Techniques to eliminate this picket-fence artifact reduced the resolution of the image. Good images were reconstructed with information reduction factors of six in each lateral dimension. Lin[18] used spatial subsampling of a Fourier-transform hologram - the equivalent of spectral sampling. The subsampled hologram was exposed through an array of regularly spaced small apertures. Multiple exposures, each preceded by small lateral translations, formed a mosaic hologram in which a given region contained replicas of a subhologram. Image fidelity suffered due to decreased image resolution and the presence of artifacts, e.g., graininess and moire-like stripes.

These experiments exploited the redundancy inherent in holographic fringes. Researchers spatially or spectrally subsampled to reduce bandwidth. Image quality suffered: dispersion plates caused graininess and noise, periodic replication caused moire-like stripes. Artifacts were inevitable because researchers could not directly manipulate the recorded fringes. However, computed fringes can be directly manipulated[3]. For example, researchers[19] have used phase manipulation to reduce speckle-like artifacts in computer-generated holograms produced by replicating smaller subholograms. Fringelet bandwidth compression translates optical information-reduction concepts - particularly those of Lin[18] - into computational holography, where they are more realizable.

3 Review of Diffraction-Specific Fringe Computation

Fringelet bandwidth compression is based on diffraction-specific fringe computation. The diffraction-specific approach considers only the reconstruction step in holography. In practice, it is the spatially and spectrally sampled treatment of a holographic fringe. (References[7,8] contain a comprehensive discussion.) Diffraction-specific fringe computation has the following features:

- ***Spatial discretization:*** The fringe is treated as a regular array of functional holographic elements called "hogels." In HPO holograms, a hololine is treated as evenly spaced line-segment hogels of width w_h, each comprising several hundred samples.
- ***Spectral discretization:*** A "hogel vector" is a spectrally sampled representation of a hogel. Each component represents the spectral energy within a small range of spatial frequencies, spaced by Δ_f. A 3-D object scene is converted to an array of hogel vectors.
- ***Basis fringes:*** Precomputed basis fringes combine to convert hogel vectors into (ultimately) physically useful fringes. Each basis fringe represents an independent part of the fringe spectrum and is precomputed with appropriate sample spacings.
- ***Rapid linear superposition:*** The real-valued linear superposition of the precomputed basis fringes, directed by hogel vectors, exploits the bipolar intensity method.[6]

Hogel-vector bandwidth compression, reported elsewhere[8], was based on diffraction-specific fringe computation. This technique used the hogel vectors themselves as an encoded fringe format. Decoding was simply the linear superposition of precomputed basis fringes to compute fringes (hogels). This method achieved a compression ratio of 16:1 with an acceptably small loss in image resolution. It also reduced total computation time for typical 3-D images to less than 4.0 s per 36-Mbyte holographic fringe. Decoding was the slower step, requiring many multiplication-accumulation calculations.

4 Fringelet Bandwidth Compression

Fringelet bandwidth compression is a diffraction-specific fringe computation technique that reduces the number of encoded symbols through spatial subsampling of hogels. In fringelet encoding, each hogel (fringe) is encoded as a spatially smaller fringelet, so named because it looks like a small fringe with a spectrum that is similar to the desired hogel spectrum. An array of fringelets is encoded from the 3-D scene description using the diffraction-specific treatment. The resulting fringelet array is decoded into usable fringes using a sample-replication scheme that is extremely fast. Bandwidth compression, measured by compression ratio (CR), results because the size of fringelet array is only 1/CR the size of the final fringe. One fringelet of width $N=N_h/CR$ samples is decoded into one hogel of width $N_h = pw_h$ samples. Speed results from simplicity and efficiency, especially in the decoding step.

Spectrally, a N-sample fringelet contains N independent spatial frequencies. Therefore, the fringelet is synthesized from a hogel vector containing N components, using a set of precomputed basis fringes, each with a width of w_h/CR (or N_h/CR samples) and with energy in a specific region of the spectrum. The resulting fringelet has a spectrum with the desired amount of energy in each region of the spectrum, centered at intervals of

$$\Delta_f = B\frac{CR}{w_h} = \frac{pB}{N} \quad , \tag{1}$$

where B is the spectral bandwidth used (up to 0.5 cycles/sample), and p is the horizontal sampling pitch.

The information content of an encoded fringe, i.e., the number of symbols in the fringelet array, is equal to the product of the number of hogels times the number of components (N) in each fringelet. Therefore, the sample spacings (w_h in space and Δ_f in spatial frequency) determine the compression ratio:

$$\mathrm{CR} = \frac{\text{samples per final hogel}}{\text{samples per fringelet}} \equiv \frac{N_h}{N} = \frac{w_h \Delta_f}{B} \quad . \tag{2}$$

The sampling parameters, w_h and Δ_f, are chosen according to the relation derived in literature on diffraction-specific fringe computation[7,8]:

$$N \geq \frac{Z}{\delta}(\lambda p B \sqrt{2}) \tag{3}$$

where δ is the maximum allowable image point spread, Z is the required image depth, and $N=N_h/CR$ is a measure of bandwidth in symbols/hogel. This expression assumes that the optimal hogel width is chosen:

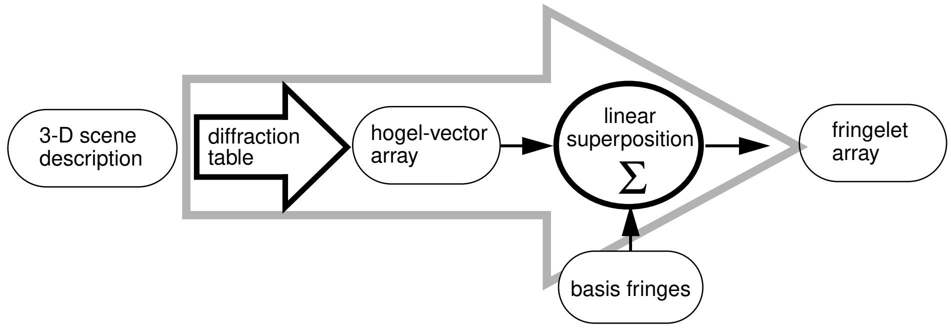
$$w_h = \delta / \sqrt{2} \, . \tag{4}$$

For practical imaging, blur must be below the amount perceivable to humans - about 0.18 mm at a typical viewing distance of 600 mm.[7] Equations 3 and 4 were derived from a model for the imaging point spread due to spatial and spectral discretization. Equations 3 is essentially the best that fringelet encoding can achieve, though, as discussed later, fringelets do add additional image blur. To use Equations 3 and 4, first the hogel width is set by Equation 4. The value of Z, usually limited by the display system, gives a value for N through Equation 3, and subsequently Δ_f through Equation 1.

5 Fringelet Encoding

Fringelet encoding converts a 3-D object scene description (e.g., a list of points) into a fringelet array. A full fringe pattern is not computed until the decoding step, i.e., fringelet encoding (like hogel-vector encoding[8]) is a direct-encoding technique. The fringelet array is generated in two steps, as illustrated in Figure 3. First, a pre-computed "diffraction table" is used to map desired 3-D image elements to hogel-vector components. (This process is the same as hogel-vector encoding.[8]) Second, the hogel vectors combine with precomputed basis fringes through linear superposition to generate fringelets.

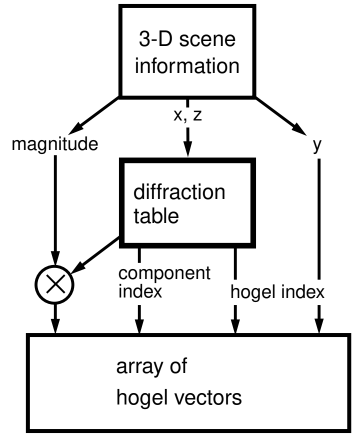
Figure 3. Schematic of Fringelet Encoding



5.1 Diffraction Tables

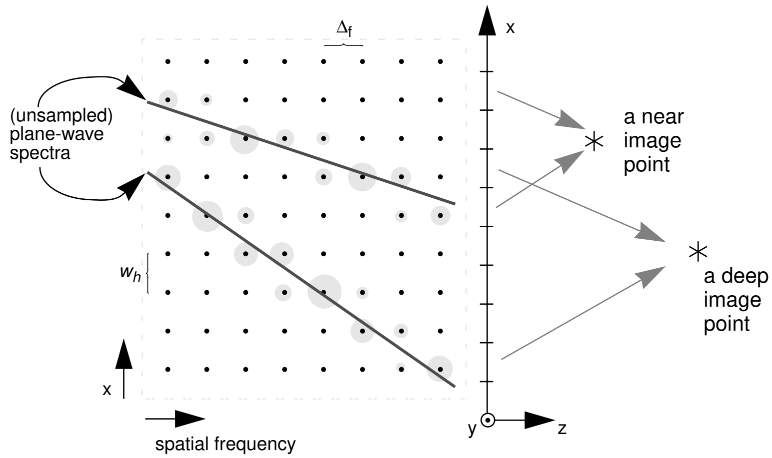
A point at some (x,y,z) location in the image volume can be represented in a sampled manner as components of particular hogel vectors. In an HPO hologram these are on the same hololine, as indexed by the vertical (y) location of the image point. The horizontal and depth locations (x,z) of the point determine which components of which hogel vectors are needed. A *diffraction table* maps this sampled relationship. Each entry of the diffraction table is an amplitude factor that is multiplied by the desired magnitude to determine the amounts of contribution to each component in the hogel-vector array. The magnitude of an image point is the square-root of the intensity calculated from the 3-D scene and lighting information. The use of diffraction tables has been described elsewhere[8] in more detail.

Figure 4. Use of Diffraction Table to Calculate Hogel Vectors



A diffraction table is precomputed by spatially and spectrally sampling (with spacings w_h and Δ_f) the theoretical spectral content of a diffracted wavefront. The spectrum is related to an imaged point through optical propagation. (This relation is treated in more detail in the reference[7] pp. 153-158, and in other references[5].) For point image elements, the spectral content immediately following diffraction is a roughly linear distribution of plane-wave spectra. A bi-gaussian (with $1/e^2$ full-widths of w_h and Δ_f) is used to sample the continuous spectrum. Figure 5 illustrates this process with two examples. The regular dot grid indicates the discretized spectrum, i.e., a hogel-vector array. The diagonal lines each represent the continuous unsampled plane-wave spectrum. The size of the circular region around a dot indicates the magnitude of that hogel-vector component required to create the image point.

Figure 5. Diffraction Table For Image Point Elements: Two Examples



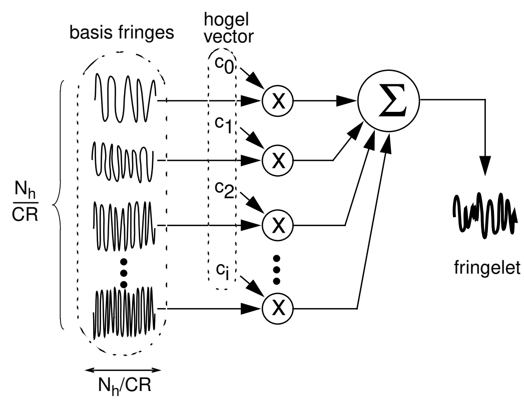
Diffraction table entries are precomputed for all possible values of image point x and z (horizontal and depth locations). During computation of hogel vectors for a particular 3-D object scene, each image point indexes the diffraction table, and the magnitudes are summed to calculate the total hogel-vector array for this object scene. This step is fast because it involves only simple calculations.

Another approach to generating the intermediate hogel-vector array employs standard 3-D computer graphics rendering software. This technique was described previously for hogel-vector encoding[8]. It facilitates advanced image properties, such as specular reflections, texture mapping, and advanced lighting models.

5.2 Conversion to Fringelets

The second process in fringelet encoding is the conversion of hogel vectors to fringelets. This process is similar to hogel-vector decoding, i.e., a linear summation of precomputed basis fringes using the hogel vector as weights. As shown in Figure 6, the fringelet is the accumulation of all the weighted basis fringes. Looking at the array of precomputed basis fringes as a two-dimensional matrix, the linear summation is an inner product between the basis-fringe matrix and a hogel vector. This is the only significantly slow step in fringelet bandwidth compression. However, the simplicity and consistency of this step enables it to be implemented on specialized hardware and performed rapidly.

Figure 6. Converting Hogel Vectors to Fringelets



5.3 Precomputed Basis Fringes

Each basis fringe is responsible for contributing 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 Δ_f. In a N-sample fringelet spectrum, these spectral gaussians are narrow though not singular. The spectral phase must be uncorrelated among the basis fringes to make effective use of dynamic range. Spatially, the basis fringe must have a uniform magnitude of 1.0 within the fringelet width, w_h/CR. The calculation of basis fringes is intractable using analytical approaches. Instead, the basis fringes were computed by cascading two iterative numerical techniques: the iterative constraint method[20,23] followed by a novel simulated annealing method. (These techniques are described in the reference[7], pp. 159-168). The synthetic basis fringes for given sampling spacings (w_h and Δ_f) and a given display (i.e., pitch, p) were precomputed and stored for use during interactive fringe computation.

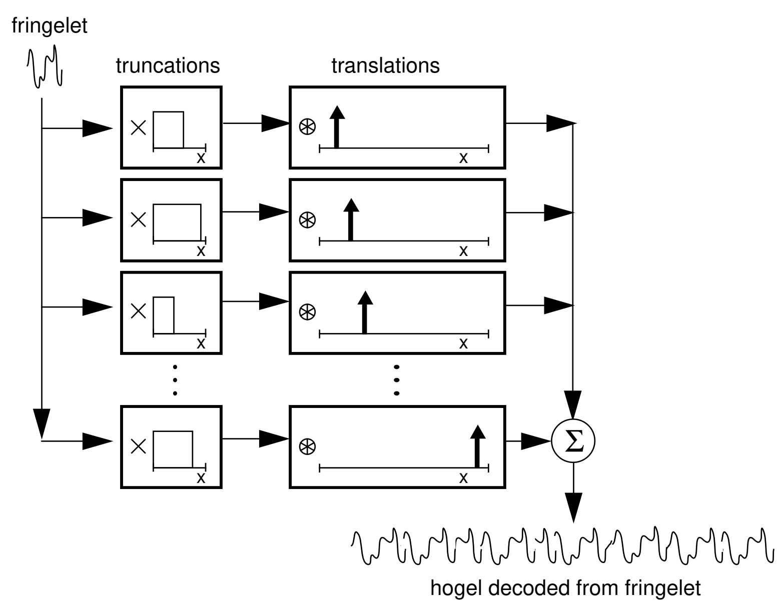
6 Fringelet Decoding

Fringelet decoding converts a fringelet of width N_h/CR into a hogel of width N_h while preserving the encoded spectral information. The decoding step consists of replicating fringelet sample values. Each fringelet sample is mapped to a number of hogel values. Each hogel sample receives exactly one fringelet sample. Therefore, a mapping table ("Table") of width N_h is precalculated for this purpose. The following pseudocode describes the fringelet decoding algorithm using Table[i], where i ranges from 0 to N_h-1.

```
for each Fringelet: {
    for each Hogel sample: {
        Hogel[i] = Fringelet[Table[i]]
    }
    load Hogel into appropriate location
}
```

The only operations are (1) fetching the value from the Table, (2) using that value to fetch the appropriate fringelet sample, and (3) copying the fringelet sample into the indexed hogel sample location. Fringelet decoding is very fast due to this extreme simplicity.

Figure 7. Model of Fringelet Decoding Algorithm

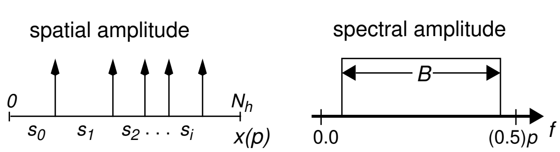


In fringelet decoding, the mapping operation is based on a nonlinear processing algorithm, as illustrated in Figure 7. Replicas of the fringelet are truncated and translated by convolution with a series of stochastically spaced impulses of unity height. This process acts to transfer the fringelet spectrum into the hogel spectrum with a predicable amount of broadening, i.e., a convolution with a narrow, roughly gaussian kernel.

The mapping table is computed from the sequence of truncations and replications in the model shown in Figure 7. The truncation widths are set to match the interval, s_i, between each impulse in the convolution impulse sequence. A replicated fringelet is truncated and translated by the same amount, s_i. The mean value of the sequence s_i determines the spectral broadening. Spatial convolution of the fringelet with the impulse sequence acts to multiply the fringelet spectrum by the spectrum of impulse sequence which must therefore be very flat. If the impulse spectrum has gaps or sharp peaks, then the spectrum of the decoded hogel will have variations that lead to image artifacts. The impulse sequence must be calculated to produce the correct amount of spectral broadening while maintaining image fidelity.

A simulated annealing[24] algorithm was adapted to precompute the impulse sequence. As illustrated in Figure 8) the goal was to generate a sequence of unity amplitude impulses at aperiodic intervals of s_i, with zero spatial phase (real-valued), and possessing a uniform spectral energy only in the region of interest (of width B).

Figure 8. Spatial and Spectral Characteristics of the Impulse Sequence



The simulated annealing algorithm was seeded with a sequence of randomly distributed impulses with an average spacing of (0.75)N. For each iteration, an impulse was randomly chosen from the sequence and moved laterally by a random amount. If it decreased the root-mean-squared spectral error, this modification was kept. If it increased error, it was rejected if a probability function was not greater than a random number:

$$exp(\frac{-E}{kT}) \leq random[0,1] \tag{5}$$

where E is the change in spectral error and kT is a “temperature” parameter that was slowly “cooled” after each iteration. During the annealing, no interval was allowed to exceed N. The desired impulse sequence converged after about 10,000 iterations. This precomputation required approximately 15 minutes on a massively parallel supercomputer. The separations s_i between the impulses was used to precompute the mapping table. Mapping tables were precomputed for all useful combinations of hogel width N_h and compression ratio CR, i.e., values of w_h and Δ_f.

7 Implementation

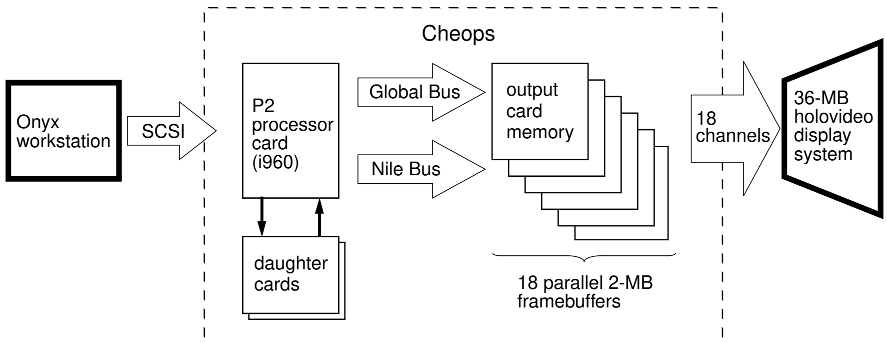
Fringelet encoding and decoding were implemented on three computational platforms: an SGI Onyx workstation for both encoding and decoding; the Onyx workstation for encoding and Cheops for decoding (see Figure 9); and an IBM RS6000 workstation (for both). Fringelet encoding began with a 3-D image scene description generally consisting of about 0.5 Mbyte of data. After the appropriate transformations (e.g., rotations, translations) and lighting, it was encoded using the precomputed set of basis fringes to produce a fringelet array with size 36/CR (Mbyte), represented as 8-bit bytes. This fringelet array was either decoded within the same workstation or (for practical imaging) was downloaded over the SCSI link to Cheops.

7.1 Implementation on Cheops

An advantage of fringelet decoding is its simplicity, which allows for its implementation on practically any hardware. Fringelet decoding was implemented on a Cheops image processing system, a compact, block dataflow parallel processor designed and built for research in scalable digital television[21,22]. As shown in Figure 9, six Cheops output cards provided 18 parallel analog-converted channels of computed fringes to the MIT 36-

Mbyte holovideo display. Cheops contained a processor card, the P2, that manages data transfers. Data is communicated between the P2 and output cards using the faster Nile Bus. The P2 communicates to the host via a SCSI link at roughly 1-Mbyte/s.

Figure 9. Fringelet Decoding on Cheops



The fringelet array is downloaded to the Cheops P2 card, where it is decoded using an Intel i960 microprocessor. A highly optimized looping algorithm decoded each fringelet using a mapping table loaded during initialization. Three hololines full of decoded hogels were transferred at one time to the Cheops output cards over the fast Nile Bus. The 36-Mbyte decoded fringe is used by the MIT 36-Mbyte holovideo display to generate 3-D images.

7.2 Implementation on Serial Workstations

For development and for speed comparison, fringelet bandwidth compression was implemented on either serial workstation. The process for generating a 36-Mbyte fringe was the same as described before, except that a simple linear loop used a mapping table to decode each fringelet.

8 Images

Fringelet bandwidth compression successfully generated 3-D holovideo images. Imaged point blur increased as the CR increased. However, for CR=16:1 and smaller, blur was small enough to be unnoticeable to the human visual system. To illustrate the effect of fringelet encoding on holographic image quality, several images were digitally photographed. Several examples were documented, using images at different depths, and using different values of hogel width and compression ratio. Two are shown in Figure 10. The image derives from a 3-D model of the fuel intake system above a Space Shuttle rocket engine. Some sharpness is lost for CR=16:1. Nevertheless, the blur added to the image does not severely degrade image fidelity. Unlike the physical periodic replication scheme used by Lin[18], fringelet decoding did not produce vertical stripes or other image artifacts. Imbalances and nonlinearities in the electronics of the display system produced the unwanted horizontal streaks and bands of light and dark seen in Figure 10.

Figure 10. Fringelet Encoded Image: Rocket Engine Fuel Intake

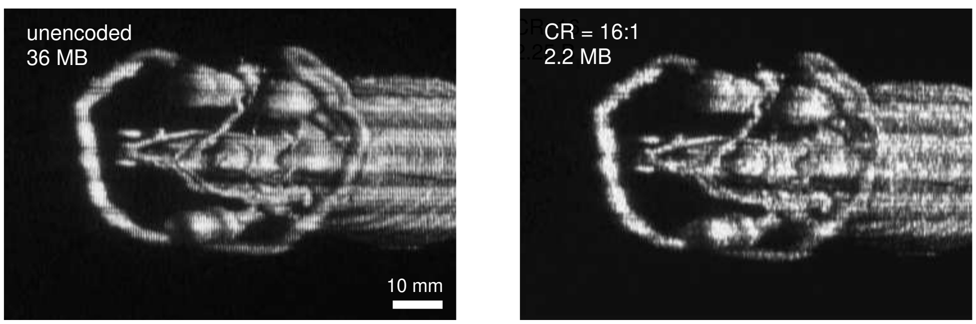
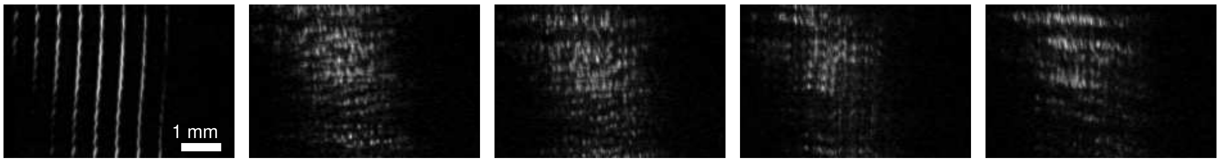


Figure 11 shows an example of a cluster of imaged points, showing the interaction between neighboring points on a hololine. These are close-ups of a fuel pipe of the rocket engine - a small curving surface represented by the presence of imaged points. These digital photographs were grabbed from a small tricolor 768x494 CCD array placed at z=40 mm - coincident with the image of the fuel pipe. In the unencoded image, the array of imaged points is clearly visible. For the fringelet encoded images (w_h=0.6 mm, N_h=1024), the blur of each point has the (desirable) effect of joining the discrete image points together to form a continuous surface.

Figure 11. Fringelet Encoded Image: Close-up



To measure the image resolution achievable using fringelet bandwidth compression, a series of point images were recorded. A CR=16:1 (w_h=0.6 mm, N_h=1024) produces reasonable results, even in the worst-case image depth of z=80 mm. Point spread is less acceptable for higher values of CR, but images are still useful at CR=32:1. The measured imaged point spread using fringelet bandwidth compression roughly matches the model used for hogel-vector bandwidth compression:[7,8]

$$blur_{total} = \left(w_h^2 + \left(\frac{z\lambda}{\pi w_h} \right)^2 + \left(z\lambda p B \frac{CR}{N_h} \right)^2 + blur_{display}^2 \right)^{1/2} \quad . \tag{6}$$

Fringelet bandwidth compression adds more blur, due to the spectral cross-talk that occurs in a fringelet. This added image blur cannot be avoided entirely, and the construction of the decoding process was crucial to minimizing it. Nevertheless, for reasonable values, i.e., CR=16:1 or smaller, this cross-talk adds only a small amount of blur.

For large compression ratios (CR> 8:1), fringelet bandwidth compression added a noticeable speckle-like pattern to the images, appearing as brightness variations at infinity. Fringelet encoding loses visually useless information that in a fully computed fringe pattern acts to control speckle that results primarily from the use of coherent light. Diffraction-specific fringe computation assumes that light is quasi-monochromatic with a coherence length $L_c < w_h/2 \sim 0.1$ mm. In practice, the effective coherence length of light in the holovideo displays was approximately 2.0 mm. To reduce the speckle artifact, a random set of phases was introduced into each hogel via the basis fringes, reducing the correlation of light diffracted by each hogel. During fringelet encoding, each basis fringe was selected at random from the set of 16 equivalent basis fringes. The reduction of inter-hogel phase correlations was furthered through the randomization of fringe phase during decoding. Sixteen different mapping tables were used, each precomputed to perform virtually identical fringelet decoding but each with different sets of intervals. A given fringelet is decoded using a mapping table that is selected at random. These efforts at reducing inter-hogel correlations successfully reduced speckle artifacts to acceptable levels.

Selection of bandwidth affected image quality. In particular, fringelet decoding cannot faithfully reproduce spatial frequencies below CR/w_h. Fringelet spectrum was therefore constrained to be non-zero only for f>CR/w_h. This represents only a 6% loss in usable bandwidth for CR=16:1.

9 Speed

Computing times were measured on three computational platforms: the Onyx workstation (alone); the Onyx (for fringelet generation) and the Cheops (for decoding); and the RS6000 workstation (alone). The results from two computational benchmarks are described: (1) a conventional interference-based technique, and (2) fringelet compression with CR=32:1 (with N_h=1024). The speed of fringelet bandwidth compression is basically independent of image scene complexity, whereas the computing time for interference-based ray-tracing computations varies roughly linearly. When comparing computing times, scene complexities were chosen to ensure equivalent benchmarks.

9.1 Speed on Onyx Workstation

Both benchmarks were implemented on a two-processor SGI Onyx workstation. For fringelet bandwidth compression with CR=32:1, a 36-Mbyte fringe required only 6.1 s. Encoding was 5.0 s, and decoding was 1.1 s. This represents a speed increase of over 3700 times compared to the conventional benchmark. This does not include data transfer times, which were much faster for fringelet bandwidth compression.

The conventional interference-based method was to sum the complex wavefronts from all object points, plus the reference beam, to generate the fringe. A fairly complex image of 20,000 discrete points (roughly 128 imaged points per hololine) was used. A 36-Mbyte fringe required 23,000 s (over 6 hours) on the Onyx. This timing was extrapolated by computing a representative 2-Mbyte fringe.

9.2 Speed on Onyx/Cheops

Encoding on the Onyx and decoding on Cheops required only 5.9 s. Fringelet decoding on the Cheops P2 i960 was 0.9 s for CR=32:1. Because fringelet decoding uses only memory access, it operates as quickly as data can be copied within the P2 board. Additional transfer times accounted for 1.5 s, for a total model-to-image time of 7.4 s. The downloading over the SCSI link using a 1.1-Mbyte fringelet array was less than one second. Because the conventional interference-based method involved complex-valued, floating-point precision calculations, it was not implemented on the Onyx/Cheops platform.

9.3 Speed on RS6000 Workstation

The same benchmarks were implemented on an IBM 42T RS6000 workstation with a single PowerPC 604 microprocessor. Fringelet bandwidth compression required only 3.9 s. Fringelet encoding was 3.2 s, and decoding was 0.7 s. This represents a speed increase of over 3100 times when compared to the conventional interference-based approach which required 12,000 s.

9.4 Analysis of Speed

The benchmarks for fringelet bandwidth compression are all worst case, i.e., a most complex image and a fully non-zero fringelet array. In practice, typical image scenes produced many zero-valued fringelet components. These trivial fringelets can be ignored with a simple code, eliminating the need to transfer or to decode them. Typical test images (N_h=1024 and CR=32:1) on the Onyx/Cheops platform gave times of 4 s for encoding and 0.6 s for decoding, for a total model-to-image time of 6.1 s. This speed will increase by using faster workstations for the slower encoding step. The single-processor RS6000, using the same microprocessor used in some personal computers, achieved typical encoding-decoding times of only 3.0 s. Furthermore, the simplicity of fringelet decoding should allow for its implementation on very fast specialized hardware such as a digital signal processing (DSP) chip. Fringelet bandwidth compression was also implemented to drive a smaller 6-Mbyte holovideo display, achieving total model-to-image computation times under 0.8 s.

TABLE 1. Computation Times for Different Platforms and Techniques

		bandwidth compression, CR=32:1	
	conventional	**hogel-vector**	**fringelet**
Onyx workstation	23,000 s	300.5 s	6.1 s
Onyx/Cheops	--	6.5 s	5.9 s
RS6000 workstation	12,000 s	180 s	3.9 s

Transfer times are not included. Bandwidth compression times are worst case, and include encoding and decoding.

In fringelet bandwidth compression, encoding required the majority of computing time. To convert an N-component hogel vector into an N-sample fringelet requires $(N_h/CR)^2$ multiplication-accumulation operations (MACs). For example, for a 36-Mbyte fringe, a hogel width of N_h=1024, and a CR=32:1, the encoding step requires 36 MMACs. Because a sample in the fringelet-decoded fringe is the result of N_h/CR^2 MACs (plus the very fast replication operations), the speed of fringelet bandwidth compression increases as the square of CR. Faster overall speeds can be achieved by sacrificing image quality.

Table 1 summarizes the speed benchmarks and includes a comparison to hogel-vector compression[8]. Fringelet bandwidth compression is much faster than hogel-vector bandwidth compression when implemented on identical platforms, e.g., 46 times faster on the RS6000. Fringelet bandwidth compression requires 1/CR the number of time-consuming MACs compared to hogel-vector bandwidth compression. Hogel-vector decoding requires N_h/CR MACs per fringe sample[7,8]. (These results are summarized in Table 2.) On the Onyx/Cheops platform, hogel-vector decoding was accelerated by using specialized hardware - two Splotch Engine[22] daughter cards. If used for fringelet encoding, the combined time of 5.9 s should drop to approximately 1.5 s.

TABLE 2. Calculations Needed for Holographic Bandwidth Compression

	MACs/sample	improvement
CR=1:1, either method	N_h	--
hogel-vector bandwidth compression	N_h/CR	CR
fringelet bandwidth compression	N_h/CR^2	CR^2

Fringelet bandwidth compression was over 3000 times faster than conventional interference-based computation. Beyond the speed-up due to the dramatic reduction in operations per final fringe sample, the use of integers rather than floating-point values provided speed increases on many platforms. It is impractical to implement conventional fringe computation using integers because of the precision required when calculating complex wavefronts.

10 Additional Applications and Conclusion

A "fringelet display" can optically decode fringelets to produce a CR-times greater image volume without increased electronic bandwidth. Optical decoding exploits a fringelet's shift-invariant nature to essentially replicate fringelets after they have modulated a beam of light. Lateral shifting can be accomplished temporally or by using an optical element. Current research demonstrates the feasibility of a fringelet display. Other applications include the use of fringelets to encode holographic movies for storage or transmission over networks. Finally, fringelet bandwidth compression provides the speed and portability required to compute large fringes for a "fringe printer" to generate printed holograms.

Fringelet bandwidth compression has been implemented and used to generate complex 3-D holographic images for real-time display. Built on top of diffraction-specific fringe computation and hogel-vector bandwidth compression, it employs sampling of the fringe spectrum in the spatial and spatial-frequency domains, achieving bandwidth compression through spatial subsampling. A bandwidth compression ratio of 16:1 produced images without obvious degradation. Transmission bottlenecks were eliminated. The fringelet was conceived as an encoded fringe format that "looks" more like a fringe, facilitating fast decoding. Fringe bandwidth compression was over 3000 times faster than conventional fringe computation. The decoding step is particularly fast, designed to involve only sample replications. Fringelet (and hogel-vector) bandwidth compression promises flexibility and interactivity for the future of electro-holography, especially as the size of the image volume increases and the variety of content grows.

Acknowledgments

Much of this research was performed at the Media Laboratory at the Massachusetts Institute of Technology. Research on the MIT holographic video system 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-C-0022) 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 US West Advanced Technologies Inc.; by Honda Research and Development Co., Ltd.; by NEC Corporation; by International Business Machines Corp.; by General Motors; and by Thinking Machines, Inc.

The RS6000 workstation and PowerPC microprocessor were manufactured by the International Business Machines Corporation, Armonk, New York. The Onyx workstation and the RealityEngine2 graphics framebuffer system were manufactured by Silicon Graphics, Inc., Mountain View, CA. The i960 microprocessor was manufactured by Intel Corporation, Santa Clara, CA.

The author gratefully acknowledges the support of Stephen A. Benton and researchers in the MIT Spatial Imaging Group and in the MIT Media Laboratory past and present: Pierre St.-Hilaire (display optics and electronics, helpful interactions), Wendy J. Plesniak (image processing, rocket engine model), Michael Halle (image processing, computational support), and the Cheops artisans: Carlton J. Sparrell, Shawn Becker, John Watlington, Brodi Beartusk, V. Michael Bove, Jr.

References

1 M. McKenna and D. Zeltzer, "Three dimensional visual display systems for virtual environments," *Presence: Teleoperators and Virtual Environments.* vol. 1 #4, pp. 421-458, 1992.

2 P. Hariharan, *Optical Holography*. Cambridge University Press, Cambridge, 1984.

3 W. J. Dallas, “Computer Generated Holograms,” in *The Computer in Optical Research*, B. R. Frieden editor, Vol. 41 of Springer Series *Topics in Applied Physics*, pp. 291-366, Springer Verlag, Berlin, 1980.

4 D. Leseberg, “Computer-generated holograms: display using one-dimensional transforms,” *J. Optical Society of America A*, vol. 3, #11, pp. 1846-1851, November 1986.

5 D. Leseberg, “Computer-generated three-dimensional image holograms,” *Applied Optics*, vol. 31, #2, pp. 223-229, 10 Jan. 1992.

6 M. Lucente, “Interactive computation of holograms using a look-up table,” *Journal of Electronic Imaging*, vol. 2, #1, pp. 28-34, Jan 1993.

7 M. Lucente, “Diffraction-Specific Fringe Computation for Electro-Holography,” Ph. D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Sept. 1994.

8 M. Lucente, “Computational holographic bandwidth compression,” *IBM Systems Journal*, vol. 35, no. 3, 1996 May.

9 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, 1995, ACM SIGGRAPH, pp. 387-394.

10 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*, vol. 9, #11, pp. 1969 - 1977, Nov. 1992.

11 M. Lucente, P. St.-Hilaire, S. A. Benton, D. Arias, and J. A. Watlington, “New approaches to holographic video,” in *SPIE Proceedings #1732 Holography '92*, (SPIE, Bellingham, WA, 1992), pp. 377-386.

12 P. St.-Hilaire, “Scalable optical architecture for electronic holography,” *Optical Engineering.*, vol. 34 #10, pp. 2900-2911, Oct. 1995.

13 E. Leith, J. Upatnieks, K. Hildebrand and K. Haines, “Requirements for a wavefront reconstruction television facsimile system,” *J. SMPTE*, vol. 74, pp. 893-896, 1965.

14 C. B. Burckhardt, “Information reduction in holograms for visual display,” *Journal of the Optical Society of America*, vol. 58, #2, pp. 241-246, Feb. 1968.

15 K. A. Haines and D. B. Brumm, “Holographic data reduction,” *Applied Optics*, vol. 7, #6, pp. 1185-1189, June 1968.

16 B. P. Hildebrand, “Hologram bandwidth reduction by space-time multiplexing,” *Journal of the Optical Society of America*, vol. 60, #2, pp. 259-264, Feb. 1970.

17 C. B. Burckhardt and L. H. Enloe, “Television transmission of holograms with reduced resolution requirements on the camera tube,” *Bell System Tech. Jour.*, Vol. 48, pp. 1529-1535, May-June 1969.

18 L. H. Lin, "A method of hologram information reduction by spatial frequency sampling," *Applied Optics*, vol. 7, #3, pp. 545-548, March 1968.

19 F. Wyrowski, R. Hauck, and O. Bryngdahl, "Computer-generated holography: hologram repetition and phase manipulations," *J. Optical Society of America A*, vol. 4, #4, pp. 694-698, Apr. 1987.

20 O. Bryngdahl and F. Wyrowski, "Digital holography - computer-generated holograms," in *Progress in Optics*, E. Wolf, ed. (North-Holland, Amsterdam, 1990), vol. 28, pp. 1-86.

21 V. M. Bove Jr. and J. A. Watlington, "Cheops: a modular processor for scalable video coding," in *Proceedings of SPIE Visual Communications and Image Processing '91 Conference*, vol. 1605 (SPIE, Bellingham, WA, 1991), pp. 886-893, Nov. 1991.

22 J. A. Watlington, Mark Lucente, C. J. Sparrell, V. M. Bove, and I. Tamitani, "A hardware architecture for rapid generation of electro-holographic fringe patterns," in *Proceedings of SPIE #2406 Practical Holography IX,* 2406-23 (SPIE, Bellingham, WA, 1995), pp. 172-183.

23 J. R. Fienup, "Iterative method applied to image reconstruction and to computer-generated holograms," *Optical Engineerin*g, vol. 19 #3, pp. 297-305, March 1980.

24 W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.

Mark Lucente is a member of the Imaging Science and Technology group at the IBM T.J. Watson Research Center. 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 3-D imaging at Columbia University in New York where he is Adjunct Professor of Computer Science. For six years, Dr. Lucente worked in the Massachusetts Institute of Technology 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 S.B., S.M., and Ph.D. (in 1994) from the Department of Electrical Engineering and Computer Science at the MIT, working in a variety of fields: 3-D 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 the Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.