Diffraction-Specific Fringe Computation for Electro-Holography

by

Mark Lucente

Master of Science, 1989
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Bachelor of Science, 1986
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
Massachusetts Institute of Technology
September 1994

© 1994 Massachusetts Institute of Technology. All Rights Reserved.

Signature of Author ....................................................

Electrical Engineering and Computer Science
August 31, 1994

Certified by ......................

Stephen A. Benton
Allen Professor of Media Technology, Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ......................

Frederic R. Morgenthaler
Chairman, Committee on Graduate Students
Electrical Engineering and Computer Science
Diffraction-Specific Fringe Computation for Electro-Holography

by

Mark Lucente

Submitted to the
Department of Electrical Engineering and Computer Science
on August 31, 1994
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract
Diffraction-specific fringe computation is a novel system for the generation of holographic fringe patterns for real-time display. This thesis describes the development, implementation, and analysis of diffraction-specific computation, an approach that considers the reconstruction process rather than the interference process in optical holography. The primary goal is to increase the speed of holographic computation for real-time three-dimensional electro-holographic (holovideo) displays. Diffraction-specific fringe computation is based on the discretization of space and spatial frequency in the fringe pattern. Two holographic fringe encoding techniques are developed from diffraction-specific fringe computation and applied to make most efficient use of hologram channel capacity. A “hogel-vector encoding” technique is based on undersampling the fringe spectra. A “fringelet encoding” technique is designed to increase the speed and simplicity of decoding. The analysis of diffraction-specific computation focuses on the trade-offs between compression ratio, image fidelity, and image depth. The decreased image resolution (increased point spread) that is introduced into holographic images due to encoding is imperceptible to the human visual system under certain conditions. A compression ratio of 16 is achieved (using either encoding method) with an acceptably small loss in image resolution. Total computation time is reduced by a factor of over 100 to less than 7.0 seconds per 36- MB holographic fringe using the fringelet encoding method. Diffraction-specific computation more efficiently matches the information content of holographic fringes to the human visual system. Diffraction-specific holographic encoding allows for “visual-bandwidth holography,” i.e., holographic imaging that requires a bandwidth commensurate with the usable visual information contained in an image. Diffraction-specific holographic encoding enables the integration of holographic information with other digital media, and is therefore vital to applications of holovideo in the areas of visualization, entertainment, and information, including education, telepresence, medical imaging, interactive design, and scientific visualization.

Thesis Supervisor: Professor Stephen A. Benton
Title: Allen Professor of Media Technology, Program in Media Arts and Sciences
To my ancestors:

Helen Lesnansky Dojonovic
Michael Dojonovic
Almerinda Le Donne Lucente
Emiliano Lucente

and

Mary Pope Lesnansky (1894-1994)
Perché vede più certa la cosa l'occhio ne 'sogni
che colla imaginatione, stando desto?

- Leonardo da Vinci
Table of Contents

1 Introduction ..................................................................................................................................13
  1.1 Overview of Thesis ..................................................................................................................16

2 Background ..................................................................................................................................19
  2.1 Human Visual System ..............................................................................................................19
    2.1.1 Acuity ................................................................................................................................19
    2.1.2 Pupil Size ........................................................................................................................20
    2.1.3 Depth Cues .........................................................................................................................20
  2.2 Three-Dimensional Displays ....................................................................................................21
  2.3 Holography .............................................................................................................................22
  2.4 Computational Holography .......................................................................................................27
  2.5 Holographic Displays ...............................................................................................................28
  2.6 Bandwidth Compression in Holography ....................................................................................31
  2.7 Iterative Hologram Computation Methods .............................................................................34

3 Motivation For Diffraction-Specific Computing ............................................................................37
  3.1 Problems with Interference-Based Fringe Computation ..........................................................37
    3.1.1 Noise ................................................................................................................................38
    3.1.2 Lack of Speed ....................................................................................................................38
    3.1.3 Analytical Image Model Constraint ..................................................................................39
    3.1.4 Need for Encoding ..............................................................................................................40
  3.2 Bipolar Intensity ......................................................................................................................41
  3.3 Precomputed Elemental Fringes ...............................................................................................42
  3.4 Conclusion ................................................................................................................................44

4 Diffraction-Specific Computation .................................................................................................45
  4.1 Recipe for Diffraction-Specific Computation .........................................................................47
  4.2 Discretization of Space and Spatial Frequency: “Hogels” .......................................................49
    4.2.1 Sampling: Concepts .............................................................................................................50
    4.2.2 Spatial Sampling ................................................................................................................52
    4.2.3 Spectral Sampling ..............................................................................................................53
    4.2.4 Introduction of the Hogel ..................................................................................................55
  4.3 Generation of Hogel Vectors .....................................................................................................56
    4.3.1 Diffraction Tables ..............................................................................................................57
    4.3.2 Use of 3-D Computer Graphics Rendering ......................................................................59
    4.3.3 Additional Techniques ......................................................................................................59
4.4 Converting Hogel Vectors to Hogels
4.5 Implementation of Diffraction-Specific Computing
  4.5.1 Cheops Overview
  4.5.2 Normalization
4.6 Image Generation
  4.6.1 Photographing of Images
  4.6.2 Point Images
  4.6.3 Incoherent Illumination Considerations
4.7 Speed
4.8 Conclusion

5 Hogel-Vector Encoding ...............................................................77
  5.1 The Electro-Holographic Communication System
    5.1.1 Information Symbols
    5.1.2 Information Entropy
  5.2 Description of Hogel-Vector Encoding
  5.3 Image Generation
  5.4 Discussion of Point Spread
    5.4.1 Comparison of Theory and Experiment
    5.4.2 Empirical Selection of System Parameters
    5.4.3 Analytical Selection of System Parameters
  5.5 Speed
  5.6 Conclusion

6 Fringelet Encoding ...............................................................103
  6.1 Fringelet Generation
  6.2 Fringelet Decoding
  6.3 Implementation
  6.4 Image Generation
  6.5 Discussion of Point Spread
  6.6 Speed

7 Holographic Encoding: Discussion ...........................................123
  7.1 The Looks and Trends of Encoded Formats
Chapter 1

Introduction

Humans are a visual animal. For thousands of years, humans have created visual media with which to express and communicate ideas, to record and understand our environment, and to manipulate and analyze the nature of real and imagined entities. Electro-holography - also called *holovideo* - is the newest visual medium. A holovideo display produces three-dimensional (3-D) holographic images electronically in real time. Holovideo is the first visual medium to produce dynamic images that exhibit all of the visual sensations of depth and realism found in physical objects and scenes. Perhaps the most exciting feature of holovideo as a new visual medium is its ability to produce tangible-looking images of scenes and objects that do not exist - or cannot exist. Holovideo has numerous potential applications in visualization, entertainment, and information, including education, telepresence, medical imaging, interactive design, and scientific visualization.
The new technical field of electro-holography is essentially the marriage between holography and digital computational technologies. Holography is used to create 3-D images by recording and reproducing optical wavefronts. The conception of holography began less than 50 years ago, and was applied to the recording and subsequent reconstruction of 3-D images beginning only 30 years ago with the arrival of the laser. Optical holographic imaging is a two-step process. Generally, a rigid object is illuminated with a coherent beam of light. A mutually coherent reference beam is aligned to interfere with the light scattered from the object. The resulting microscopic interference pattern (fringe pattern or simply fringe) is recorded in a high-resolution light-sensitive medium such as photographic film. Once developed, this recorded fringe diffracts an illuminating light beam to form a 3-D image that can look identical to the original object. The fringe pattern diffracts lights because its feature size is generally on the order of the wavelength of visible light (about 0.5 μm). Because of this microscopic resolution, the fringe pattern contains an enormous amount of information - roughly ten million resolvable features per square millimeter.

As early as 1964 researchers began to consider the computation, transmission, and use of holographic fringes to create images that were synthetic and perhaps dynamic. These researchers encountered the fundamental problems inherent to computational holography: both the computation and display of holographic images are difficult due to the large amount of information contained in a fringe pattern. Roughly ten million samples per square millimeter are required to compute a discretized (sampled) fringe that matches the resolution (diffractive power) of an optically made hologram. The possibility of computing a ten-billion-sample fringe pattern at a rate of once per second was impossible, and the possibility of modulating a beam of light with such a fringe pattern was beyond any spatial light modulation technologies available at that time. For decades, the enormous size of a holographic fringe prohibited and discouraged the pursuit of real-time electronic 3-D holographic imaging.

In 1989, researchers at the MIT Media Laboratory Spatial Imaging Group created the first display system capable of producing real-time 3-D holographic images. The
images were small but were made possible by information reduction strategies that lowered the number of fringe samples to only 2 MB - the minimum necessary to create an image the size of a golf ball. A modulation scheme based on time-multiplexing an acousto-optic modulator was used to modulate a beam of light with the 2-MB of discretized computed holographic fringe pattern. Computation of the 2-MB fringe pattern still required several minutes for simple images using traditional computation methods that imitated the optical creation of holographic fringes. Speed was limited by two factors: (1) the huge number of samples in the discretized fringe, and (2) the complexity of the physical simulation of light propagation used to calculate each sample value. As display size increased, the amount of information increased too, roughly proportional to the image volume (i.e., the volume occupied by the 3-D image). To achieve interactive holographic computation, a new type of approach would have to be invented. That new approach - called "diffraction-specific computation" - is the subject of this dissertation.

This thesis concentrates on the generation of holographic fringes using a new method called diffraction-specific computation. The architecture of diffraction-specific computation is directed by two primary goals: (1) to produce fringes at a faster rate and (2) to enable holographic encoding schemes to reduce the bandwidth required to display holographic images. Traditional computing methods achieved neither of these goals. Traditional computation imitated the interference occurring in the optical generation of fringes. In contrast, diffraction-specific computation is based on only the diffraction that occurs during the reconstruction of a holographic image. The diffraction-specific approach is a better match to holovideo since the purpose of a real-time holographic display is to generate 3-D images through the modulation and subsequent diffraction of light. The research described in this dissertation demonstrates that moving from interference-based methods and toward diffraction-specific methods increased fringe computation speed.

The application of diffraction-specific computation provides a means for encoding fringes to make the most efficient use of computational power and electronic and opti-
cal bandwidth. Holographic fringes contain far less usable information than is inti-
mated by a simple measure of bandwidth. Roughly ten million samples per square
millimeter is required whether the fringe pattern represents a simple object or a com-
plex scene, and whether the holographic image is to be viewed through a microscope
or by the less acute human visual system (HVS). However, this observation alone does
not provide instruction for the reduction or compression of fringe bandwidth. The
development of holographic encoding begins by taking a closer look at the nature of
holographic fringes. Diffraction-specific fringes are no longer strictly physical but are
synthetically generated by specifying more deliberately their diffractive purpose. This
specification leads to a more efficient use of holographic channel capacity. This disser-
tation documents a progression from interference-based computation towards diffrac-
tion specific fringes, from physical fringes to synthetic fringes, from inefficient use of
channel capacity to a reduction in required bandwidth. Ultimately, the information-
reduction strategies born of the diffraction-specific approach should allow for the
design and construction of more information-efficient holographic displays.

1.1 Overview of Thesis

The following original research is documented in this thesis dissertation:

- A description of the bipolar intensity approach to fringe computation,
  invented to solve the problems of noise and lack of speed in interference-
  based computing.

- A description of a fringe computation approach in which precomputed
  elemental fringe patterns are rapidly weighted and summed during com-
  putation. This approach, invented to provide greater speed, enabled the
  first-ever interactive computation and display of 3-D holographic
  images.

- The architecture and implementation of diffraction-specific fringe com-
  putation based on the discretization of space and spatial frequency in a
  holographic pattern.
• Generation of synthetic "basis fringes" as required by diffraction-specific computation, including the adaptation of simulated annealing for the generation of holographic fringes.

• Demonstration of a holographic encoding technique called "hogel-vector encoding" based on undersampling fringe spectra. Bandwidth is reduced by a ratio of 16 and higher.

• Demonstration of a holographic encoding technique called "fringelet encoding" that successfully increases the simplicity and speed of the decoding algorithm.

• Fringe computation, using fringelet encoding and decoding, that is over 100 times faster than traditional interference-based computation.

• An analysis of the trade-offs among the holographic system parameters of image resolution, image depth, and bandwidth. Diffraction-specific holographic encoding is shown to match the holographic bandwidth to the viewer-useful visual information in an image.

• Future directions, including descriptions of "fringelet" decoding performed electronically or optically.

The following "Background" chapter describes optical and computational holography as well as various information-reduction strategies that have been attempted in the past. Also included are computational techniques that are later applied to this research. The remainder of this dissertation begins with a discussion of the problems of traditional interference-based fringe computation: unwanted noise, lack of speed, restriction to analytical models, and lack of bandwidth compression. Subsequent chapters describe the architecture and implementation of diffraction-specific fringe computation based on the need to solve the problems of traditional computation. Finally, the description of two novel holographic encoding schemes are described and shown to reduce the required bandwidth of a holographic display and to increase computation speed by over an order of magnitude.
A glossary of terms and abbreviations is included in Appendix A. Some of the analytical basis for diffraction-specific computation is discussed Appendix B, "Spectral Decomposition of Diffracted Light." Computational support for diffraction-specific computation is described in Appendix C, "Computation of Synthetic Basis Fringes."
Chapter 2

Background

This chapter includes background information on 3-D displays, optical and computational holography, past attempts at holographic information reduction, and useful computational techniques. These concepts are central to the development of diffraction-specific computation and holographic encoding.

2.1 Human Visual System

The goal of a holographic display is to produce 3-D images for viewing by the human visual system (HVS). Optically produced holographic images generally exhibit resolution far beyond the abilities of the HVS. This is essentially why so much of holographic bandwidth is wasted: the HVS simply cannot make use of the information contained in an image with micron-scale resolution. Diffraction-specific computation and the holographic encoding schemes developed from it exploit the limited capabilities of the HVS. Holovideo should provide only visual information that is useful to the HVS and no more. An information-efficient display system must stimulate the viewer in the most efficiently way possible\(^8\). It is important to begin the development of these novel holographic computation techniques with a discussion of the performance of the human visual system.

2.1.1 Acuity

The lateral and depth spatial resolutions (acuities) of the HVS vary with viewing conditions such as brightness and motion. The values listed here are typical values that were derived empirically\(^8\). The lateral acuity is approximately one minute of arc (1/60 degree of arc) or 290 microradians. A typical viewing situation places the viewer at about 600 mm from the display. At this distance, the HVS can resolve 600 x 0.000290 = 0.175 mm. A point of light that is angularly smaller looks identical
to a 0.175-mm spot. Two small spots confined to less than 0.175 mm appear as a single point of light. Holography is capable of producing spots that are 100 times smaller than 0.175 mm. Therefore, a holographic encoding scheme is allowed to add small amounts of blur to image points, as long as the total blur is imperceptible to the HVS, i.e., less than 0.175 mm.

Depth acuity at a viewing distance of 600 mm is approximately 0.75 mm. Holographic imaging can produce depth resolutions that are 100 times smaller than this. Again, this is wasted bandwidth, and an opportunity for bandwidth compression.

### 2.1.2 Pupill Size

As a viewer moves throughout the viewing zone, the eyes see different views of the image or object. The number of distinguishably different views (called the *parallax resolution*) depends on the size of the pupil of the eye. Typically the diameter of the pupil is 3 mm. At a viewing distance of 600 mm, the useful angular resolution of an image is $\arctan(3/600) = 5$ milliradians = 0.3 degrees. An optically produced hologram provides a continuously varying parallax, i.e., the perspective views of the image vary continuously as a function of viewing angle as does a real object. This is another way in which a holographic image provides too much information for the HVS.

### 2.1.3 Depth Cues

The HVS uses a number of depth cues to resolve the depth, shape, texture, and relative distance of scenes. The oculomotor cues are physiological cues that derive depth through physical movements in the muscles of the eyes. Oculomotor depth cues include accommodation (focusing cues) and convergence (triangulation between the two eyes). Stereopsis is the highly acute sensation of depth that results from the depth cue of binocular disparity, in which each eye sees a slightly different view of the scene. Motion parallax is a depth cue sensed from the apparent change in the lateral displacements among objects in a scene as the viewer moves. Motion parallax allows the viewer to move around the object scene, as is the case when viewing real scenes.
The remaining depth cues are pictorial depth cues: they can be found in two-dimensional (2-D) images and pictures. These are monocular depth cues (as is motion parallax) because they can be sensed with a single eye. The strongest pictorial depth cue - perhaps the strongest depth cue - is occlusion (also called overlap). The HVS derives relative depth information when one part of an image is obstructed by another overlapping part. The HVS uses the kinetic depth cue to derive shape and depth information by observing the relative motions of parts of a scene. The remaining pictorial depth cues (linear perspective, texture gradient, aerial perspective, shading, relative sizes, etc.) are discussed in the book by Okoshi\(^8\).

Understanding the visual depth cues is particularly important when comparing holography to other types of 3-D display technologies. Holographic displays require a large amount of computation and display bandwidth. The value of holovideo must be compared to the relatively easier existing 3-D displays.

### 2.2 Three-Dimensional Displays

Three-dimensional displays\(^8,9\) are generally electronic devices that provide binocular depth cues, particularly convergence and binocular disparity. They also provide some or all of the pictorial depth cues. Some 3-D displays provide additional depth cues such as motion parallax and ocular accommodation. (The reference by McKenna\(^9\) contains a good discussion of the visual depth cues and a detailed evaluation of 3-D display techniques.) A 3-D display enables the viewer to more efficiently and accurately sense both the 3-D shapes of objects and their relative spatial locations, particularly when monocular depth cues are not prevalent in a scene. When viewing complex or unfamiliar object scenes, the viewer can more quickly and accurately identify the content of a scene. Therefore, 3-D displays are important in any application involving the visualization of complex 3-D data, including telepresence, education, medical imaging, computer-aided design, and scientific visualization.
The two types of 3-D displays are stereoscopic and autostereoscopic. A stereoscopic display presents different views of the imaged scene to the left and right eyes. Examples include boom-mounted, head-mounted, and displays using polarizing glasses. Other than pictorial depth cues, these displays fundamentally add only binocular disparity. Motion parallax can be simulated with the addition of head-tracking. An autostereoscopic display is a 3-D display that does not require special viewing aids. Examples include lenticular, parallax barrier, slice stacking, and holography, each supplying different depth cues to the human visual system. A good autostereoscopic display provides motion parallax by presenting more than two views of the imaged scene.

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

2.3 Holography

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

Diffraction-specific computation is developed for computation of all types of holograms. For simplicity of discussion, the focus in this dissertation is on off-axis transmission holograms possessing horizontal parallax only (HPO). It is possible to represent an HPO hologram with a vertically stacked array of one-dimensional (1-D) holographic lines\textsuperscript{34,36}. Consider an HPO hologram made optically using a reference beam with a horizontal angle of incidence. Spatial frequencies are large in the horizontal direction (~1000 lp/mm) and increase with the reference beam angle. In the horizontal dimension, the sampling rate (or pitch) must be high to accurately represent the holographic information. During reconstruction of this hologram, diffraction occurs predominantly in the horizontal direction. Each hololine (a single horizontal line of the fringe) diffracts light to a single horizontal plane to form image points describing a horizontal slice of the image. Therefore, one hololine should contain contributions only from points that lie on a single horizontal slice of the object. Essentially, the 2-D holographic pattern representing an HPO 3-D image can be thought of as a stack of 1-D holograms or hololines.

A computer-generated hologram (CGH) represents a fringe pattern as an array of discrete samples. Given a fixed (HPO) hologram size, the sample count is simply the width times number of samples per unit length - the pitch, \( p \). The relationship between (minimum) sampling pitch \( p \) and angle of diffraction \( \Theta \) is
\[ p = \frac{4}{\lambda} \sin \frac{\Theta}{2}. \]  

(1)

For a hologram located in the image plane (i.e., an \textit{image-plane} hologram), the diffraction angle is equal to the viewing angle. (The image plane is the plane in the middle of the image volume.) Therefore, the image-plane requirement on sample count is determined by the size of the viewing zone. For a hololine of width \( W \), the sample count becomes:

\[ \text{number of samples} = Wp = \frac{4W}{\lambda} \sin \frac{\Theta}{2}. \]  

(2)

Since an HPO CGH contains only a single vertical perspective (i.e., the viewing zone is vertically limited to a single location), spatial frequencies are low (\( \sim 10 \text{ lp/mm} \)) in the vertical dimension. The vertical image resolution is the number of hololines. Eliminating vertical parallax reduces CGH information content by at least a factor of 100 by reducing the vertical spatial frequency content from roughly 1000 to roughly 10 \text{ lp/mm}.

Consider the typical optical holographic set-up in the following illustration. Light scattered from the object, \( E_O \), interferes with reference light, \( E_R \). (Note: In this analysis, optical wavefronts are represented by spatially varying complex time-harmonic electric field scalars. All wavefronts are assumed to be mutually coherent sources of monochromatic light. The units of an electric field amplitude are normalized so that the square of magnitude equals optical intensity. The polarizations are assumed to be identical and for simplicity are not specified.) The total time-harmonic electric field
incident on the hologram is the interference of the light from the entire object and the reference light, \( E_O + E_R \).

The total interference fringe intensity is \( I_T \):

\[
I_T = |E_O + E_R|^2
\]  

(3)

This expression for total intensity expands to:

\[
I_T = |E_O|^2 + |E_R|^2 + 2\text{Re} \left\{ E_O \cdot E_R^* \right\}
\]  

(4)

The total intensity is a real physical light distribution comprising three components.

- **Object self-interference**: The first term, called object self-interference, is a spatially varying pattern that is generated when interference occurs between light scattered from different object points. During image reconstruction, this component of the holographic pattern is unnecessary and often produces unwanted image artifacts. In optical holography, a common solution is to spatially separate the object self-interference artifacts from the reconstructed image by increasing the reference beam angle to
at least three times the angle subtended by the object. However, in computational holography, a large reference beam angle is a luxury that one does not have.

- **Reference bias**: The reference beam intensity represents a spatially nearly invariant (dc) bias that increases the value of the intensity throughout the hologram. This is also an unwanted term because it wastes available dynamic range in the holographic medium.

- **Useful fringes**: The third term is the interference between the object wave front and the reference beam. This fringe pattern contains all of the holographic information that is necessary for image reconstruction.

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

In a horizontal-parallax-only (HPO) image, the usable depth is limited by the tolerance of the human visual system to astigmatism. HPO holograms focus light only in the horizontal direction and not in the vertical direction. In most cases, the eye sees light vertically scattered from the image plane (the plane in the middle of the image volume). Light horizontally focuses to a range of depths within the image volume. The human visual system cannot tolerate astigmatism beyond its normal range of depth of focus\(^8\) which has been empirically measured to be approximately 0.34 diopters (m\(^{-1}\)). Let the distance from the viewer to the image plane be \(D_v\) and the distance to an imaged point of extreme depth be \(D_H\). These parameters are related to the astigmatism constraint as
\[ \left| \frac{1}{D_H} - \frac{1}{D_V} \right| = 0.34 \text{ m}^{-1}. \] (5)

Typically, the viewing distance is 600 mm, making the maximum deviation of \((D_V - D_H)\) between the image plane and the viewer be 100 mm. (In this thesis, the practical depth of points imaged with the MIT holovideo system was closer to 80 mm.)

Full-parallax image depth is often limited by vignetting, i.e., the windowing effect. The usable depth is a function of the location and lateral dimensions of the image plane and of the viewing angle. Typically, this limitation is about as strict as the astigmatism limit imposed on HPO images.

### 2.4 Computational Holography

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

Traditionally, computational holography\(^{21,22}\) was slow due to two fundamental properties of fringe patterns: (1) the enormous number of samples required to represent microscopic features, and (2) the computational complexity associated with the physical simulation of light propagation and interference. A typical full-parallax hologram 100 mm \(\times\) 100 mm in size has a sample count (also called space-bandwidth product or SBWP or simply "bandwidth") of over 100 gigasamples of information. A larger image requires a proportionally larger number of samples. Several techniques have been used to reduce information content to a manageable size. The elimination of ver-
tical parallax provides great savings in display complexity and computational requirements without greatly compromising the overall display performance. Other less desirable sacrifices include reducing the size of the object scene or the size of the viewing zone.

Traditional holographic computation imitates the interference that occurs when a hologram is produced optically using coherent light. In this dissertation, these traditional methods are classified as "interference-based computation." Early methods made use of the Fourier transform to calculate the phase and amplitude of the object wavefront at the plane of the hologram. A plane of object points is propagated to the plane of the hologram using a Fourier transform to compute $E_O$. At each sample point in the hologram plane, this object light wavefront was added to a reference beam wavefront, and the magnitude squared became the desired fringe pattern. This approach was used to create planar images. Multiple image planes were separately Fourier transformed and combined to produce 3-D images. This multiplicity makes the Fourier-transform approach slow and inefficient for computing 3-D images.

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

### 2.5 Holographic Displays

A holographic display is an electro-optical apparatus that modulates light with a holographic fringe pattern. In the earliest work in computational holography, the computed fringe pattern was recorded (permanently) in a piece of film. The film provided the SBWP sufficient to represent the fringe pattern. In some cases, the film also provided
grayscale (e.g., research involving the kinoform\textsuperscript{29,32}, and the ROACH\textsuperscript{22}). Light passing through this film created a static holographic image. Some researchers used binary printers as output devices, and photographically reduced the printed binary patterns to achieve the required holographic resolutions. Most employed the detour-phase technique\textsuperscript{25,27,28,30} for representing both the amplitude and phase of the computed object wavefront.

To create dynamic holographic images, a dynamic spatial light modulator (SLM) must be used. The SBWP of a holographic SLM must be as high as that of holographic film\textsuperscript{74}. Ideally, a holographic SLM must display over 100 gigasamples. Current SLMs, however, can provide a maximum of 10 megasamples. Examples of SLMs include the flat-panel liquid-crystal display (LCD) and the magneto-optic SLM. These SLMs are capable of displaying a very small CGH pattern in real time. Early researchers employed a magneto-optic SLM with a SBWP of 16384 elements\textsuperscript{61} or a LCD SLM with 10,000 elements\textsuperscript{62} to produce tiny planar images. One group produced a small planar image using a deformable mirror device (DMD) with 16384 binary elements\textsuperscript{63}. More recent work employed LCDs with higher pixel counts\textsuperscript{64,65}, but the images were still very small and essentially two-dimensional. (Note: To be considered a 3-D display, a real-time holographic display must provide at least binocular disparity, the one depth cue that all other 3-D displays provide. Therefore, the viewing zone must be a minimum of about 100 mm to allow for both eyes to see the diffracted light. Otherwise, binocular disparity cannot be supported, and the display is only 2-D.)

An ideal holographic SLM does not yet exist, but time-multiplexing of a very fast SLM provides a suitable substitute\textsuperscript{50,51,57}. The display system that we used in this thesis research was the second generation holovideo display developed by the Spatial Imaging Group at the MIT Media Laboratory\textsuperscript{58}. This display used the combination of an acousto-optic modulator (AOM) and a series of lenses and scanning mirrors to assemble a softball-sized 3-D holographic image at video frame rates. This time-multiplexed SLM approach is sometimes called the “Scophony geometry” after the early contender for television displays\textsuperscript{82,83}. A partial schematic is shown in the figure.
below. A general description follows, and a full description can be found in the references\textsuperscript{57,58}. It is important to note that the diffraction-specific fringe computation described in this dissertation is not limited to use with the MIT display. By incorporating the proper physical parameters, wavelengths and sample size, a hologram generated using this method can be viewed using other holographic displays.

This figure shows a partial schematic diagram (top view) of the MIT holovideo display. The scanning mirror angularly multiplexes the image of the modulated light. A vertical scanning mirror (not shown) positions each hololine vertically. Electronic control circuits synchronize the scanners to the incoming holographic signal.

In the MIT holovideo display, as each line (hololine) of the fringe pattern was read out of a high-speed framebuffer, it passed through a radio-frequency (RF) signal processing circuit and into one channel of the AOM. This display used 18 parallel channels from an 18-channel framebuffer and feeding into an 18-channel AOM that modulated 18 separate beams of red laser light. At any instant, as 18 lines of the holographic pattern traversed the aperture of the AOM in the form of acoustic waves (at a speed of 617 m/s), a portion equal to roughly 1000 samples (in each channel) modulated the
phase of the wavefront of laser light that passed through each channel of the AOM. Two lenses imaged the diffracted light at a plane in front of the viewer. By reflecting the light off of a synchronized horizontally scanning mirror, the apparent motion of the holographic pattern was cancelled. The scanning mirror also acted to angularly multiplex the image of the acoustic wave. It extended the apparent width of the imaged holographic pattern to 256 Ksamples, with each sample representing a physical spacing of 0.6 μm.

The viewer saw a real 3-D image located just in front of the output lens of the system. The image occupied a volume that was 150 mm wide, 75 mm high and 160 mm deep. The size of the viewing zone - i.e., the range of eye locations from which the viewer can see the image - was 30 degrees horizontal. The viewer experienced the depth cue of horizontal motion parallax. This was a horizontal-parallax-only (HPO) image. Vertical parallax was sacrificed to simplify the display. (This restriction does not limit the display’s usefulness in most applications.) Because the holographic image possessed no vertical parallax, there was no need for diffraction in the vertical dimension. The vertical resolution of 144 lines over 75 mm was equivalent to that of a common 2-D display.

### 2.6 Bandwidth Compression in Holography

Holographic fringe patterns contain more information than can be utilized by the human visual system. For example, consider a 1-Msample hololine with one byte per sample. For a typical display system, this hololine may represent an array of up to 1000 points, with ~20 bytes each (x, y, z, brightness, and directionally dependent information), for a total of ~20 KB of actual usable information. Most of the bandwidth is unused: only 20 KB of visual information is communicated through a hologram with a 1-MB channel capacity.

Several researchers have attempted optically to reduce bandwidth in holographic imaging. Haines and Brumm attempted to use a hologram of reduced size to
generate an image with that was not reduced in size or in viewing zone. During optical recording, a dispersion (scattering) plate was positioned between the hologram plane and the object to be imaged. A reduced-size hologram was recorded and repositioned so that diffracted light passed through the dispersion plate. Although the hologram was of reduced size, the viewer saw an image that was as large as an image produced by a full-size hologram in a standard holographic imaging system. However, image quality suffered. Either image resolution or signal-to-noise ratio (SNR) was reduced. The choice of dispersion plate determined which of these degradations was traded off against holographic bandwidth. Bandwidth reduction was as much as 60 times in each lateral dimension. As bandwidth reduction increased, the increase in artifacts overwhelmed the image. Hildebrand\textsuperscript{[40]} generalized the dispersion-plate approach by including time-varying scattering functions. The advantage was increased flexibility in alignment and a decrease (in some cases) in image artifacts. In all of these dispersion-plate methods, the intermediate scattering plate was used as an angular multiplier. The optically produced hologram records light propagating at a continuous range of directions. However, the human visual system does not need such precision. Essentially, the dispersion plate encodes angular information during optical recording by dispersing this information throughout the entire hologram. A reduced-size portion of this hologram contains information about light traveling in all directions. (This is not generally the case in standard optical holography.) During reconstruction, light passing in the opposite direction through the same dispersion plate is “decoded” from a larger image. Light from the small hologram appears to be emerging from a wider range of directions because of the angular multiplication effect of the dispersion plate.

Burckhardt and Enloe\textsuperscript{[76,79]} reduced the information recorded in a hologram by exposing only an array of small regularly spaced areas. As the proportion of the hologram that was exposed decreased, the amount of information recorded in the hologram decreased. This work was equivalent to spatially sampling the hologram. The reconstructed image had an annoying “screen” artifact; the image appeared as if on the other side of a picket fence. Techniques to eliminate this artifact reduced the resolution of
the image. Good images were reconstructed with information reduction factors of 6 in each lateral dimension.

Lin\textsuperscript{77} also used a spatial subsampling to reduce holographic bandwidth. In this work, a Fourier transform the hologram was recorded by placing the hologram plate at the focus of a lens that essentially performed a Fourier transform of the image light. Spatially sampling a Fourier transform hologram was equivalent to spectrally sampling the image light. Similar to the work by Burckhardt and Enloe, the subsampled hologram was exposed through a mask (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 the sample recorded at that location in the hologram plane. In this way, the spectrum was subsampled, and the samples replicated to recover an approximation of the original hologram. (This work was inspirational for the computational holographic encoding schemes developed later in this dissertation.) A bandwidth reduction of 10 in each lateral dimension was achieved. (More information reduction was achieved through eliminating vertical parallax.) As in all of these optical holographic bandwidth reduction experiments, image fidelity suffered due to decreased image resolution and the presence of artifacts, e.g., graininess and moire-like stripes.

All of these experiments in optical holographic information reduction exploited the redundancy inherent in holographic fringes. Generally, these researchers subsampled (spatially and/or spectrally) to reduce information content. Image quality suffered due primarily to the reconstruction process. Dispersion plates caused graininess and noise. Periodic replication caused moire-like stripes. All methods decreased image resolution, though this was acceptable within a certain range. What these methods lacked was the ability to directly manipulate the recorded holographic fringe information to reduce reconstruction artifacts. However, if the fringes had been computed, direct manipulation would have been possible. For example, Wyrowski, Hauck, and Bryngdahl\textsuperscript{45,46} used phase manipulation to reduce speckle-like artifacts in computer-generated hologram produced by replicating smaller subholograms. The advantage of
computational holography is the ability - indeed, the necessity - to specify more precisely the nature of the fringes. Therefore, this thesis translates some of these optical information-reduction concepts into computational holography, where they are more useful and more realizable.

2.7 Iterative Hologram Computation Methods

Several numerical computational algorithms were developed for the purposes of phase retrieval\textsuperscript{44}. These methods were capable of deriving an unknown phase pattern given the amplitude of a signal and its spectrum\textsuperscript{42}. Certain iterative methods were applied to the computation of phase patterns (diffusers\textsuperscript{41,49}, etc.) for optical systems and in some cases actual holographic fringes\textsuperscript{43,48}. These computation methods iteratively applied spatial and spectral constraints while sequentially transforming the signal from the spatial domain to the Fourier (spectral) domain. The constraints in either the spatial or spectral domains were imposed upon the amplitude and the phase of the pattern. The algorithm is as follows:

1. Begin with a random pattern.
2. Transform into the spatial frequency domain.
3. Apply the (spectral) constraints.
4. Inverse transform back to the spatial domain.
5. Apply the (spatial) constraints.
6. Iterate starting at step 2, until pattern converges.
After several iterations, the process converges to an acceptable solution to the constraints.

In the case of holographic fringes, the diffraction integral describing the propagation of light is essentially a Fourier transform integral\(^\text{12}\). Therefore, holographic fringes were computed by using the (2-D) object as a spectral amplitude constraint. The spectral phase was left unconstrained. Spatially, the pattern was left unconstrained, or in some cases was constrained to have a uniform amplitude\(^\text{48}\). After a sufficient number of iterations (~100), the spatial pattern converged to a useful fringe pattern. These holograms were Fourier transform holograms, i.e., the square of the amplitude of the fringe spectrum (the spectral energy) was the desired image.

The method of iterative constraints is used in this thesis to generate synthetic "basis fringes" for diffraction-specific computation. The adaptation of this algorithm is discussed in Appendix C, "Computation of Synthetic Basis Fringes" on page 159.
Chapter 3

Motivation For Diffraction-Specific Computing

The traditional approach to fringe computation is to imitate the interference between object beam and reference beam. Early research attempted to produce fringe patterns that closely resembled the fringes recorded in optical holography. This seemed logical: since optically produced physical fringes diffracted light to form images, then their computed counterparts will do the same. Indeed, the analytical treatment of optical holography dating back to 1948 guaranteed that given certain conditions in the recording and reconstruction setups, an image was faithfully reproduced\(^1\,^2\). Also reproduced, however, are unwanted noise components in the diffracted light. Interference-based computing hung its hopes on achieving the imaging capabilities of optically made holograms, and succeeded in bringing along all of the problems associated with optical holograms.

3.1 Problems with Interference-Based Fringe Computation

This section discusses the four main problems with interference-based fringe computation:

- **Noise**: Included in the final fringe pattern is an unwanted noise term (object self-interference) and an unwanted reference bias (reference beam average intensity), which appear in Equation 4 (page 25).

- **Speed**: It is slow. The huge numbers of complex arithmetic operations (including trigonometric functions and square roots) make rapid computation impossible even on modern supercomputers.

- **Analytical image model**: The model of the object is limited to analytically describable elemental sources, such as point sources.
• **Need for encoding:** The typical spectrum of interference-based fringes is continuous and does not allow for fringe encoding.

As described in the next chapter, diffraction-specific holographic computation was invented to solve these four problems. The primary goals are faster computation and the possibility of holographic encoding for bandwidth compression. Early work for this thesis lead to the creation of a “bipolar intensity method” of hologram computation, which later lead to the use of precomputed elemental fringes. These two novel approaches to fringe computation are briefly discussed following descriptions of each of the four problems with interference-based computing.

### 3.1.1 Noise

Recall from Section 2.3 the expression for total intensity recorded in an optically produced fringe pattern expands to (Equation 4)

\[
I_T = |E_O|^2 + |E_R|^2 + 2Re\{E_O \cdot E_R^*\}
\]  

The first term is the object self-interference, which produces unwanted image artifacts and noise. The second term the reference beam intensity. This basically dc bias adds nothing useful to the image but uses up precious dynamic range. These unwanted terms add noise to the reconstructed image. Traditional interference-based hologram computation generally leaves these noise terms. Diffraction-specific computation does not include such noise terms.

### 3.1.2 Lack of Speed

Interference-based fringe computation is slow due to the large number of calculations required for each hologram sample. Computing \(I_T\) involves dozens of complex-valued mathematical operations for each point or element in the image. When the object is treated as a collection of point sources in 3-D (as discussed in the reference by the
author\textsuperscript{19}, computation for each point requires a minimum of five additions, five multiplications, one division, one square root, and two cosine (or sine) function calls. Also, after the real and imaginary field components are summed, they must be squared and added together, and then the square root must be taken at all samples of $I_T$. Floating-point data representation must be used to employ the transcendental math functions.

For illustration, consider the following example. An object description composed of 10,000 points in 3-D space is to be computed using tradition point-to-point interference-based methods. A hologram that comprises 36 M samples would require a minimum of 5 trillion (i.e., $5 \times 10^{12}$) floating-point mathematical operations. Even for an HPO hologram, the minimum is over 10 billion. On a high-end workstation, this requires over 15 minutes for a single fringe computation.

3.1.3 Analytical Image Model Constraint

Fringe computation begins with a 3-D description of the object or scene to be imaged. Usually this a point-by-point description. Often, a higher level description is useful to represent lines and patches and curved surfaces. Higher-level image elements can enable higher quality and more directly computed images.

Interference-based computation uses Equation 6 to produce a sampled fringe pattern for a given 3-D object. One proviso is that the object wavefront $E_0$ can be represented using an analytical expression. This allows a computation algorithm to be coded to use a uniform set of information about each image element. (If numerical methods are used to add the contributions from each image element, then computation becomes impossibly slow.) Nevertheless, an analytical model allows for the useful 3-D point-source model, as well as several other higher-order image element representations\textsuperscript{35,37,38}, such as line sources or small rectangles of light. These higher order representations require a large amount of additional calculations per element, again sacrificing speed.
3.1.4 Need for Encoding

Interference-based computing does not provide any obvious means for computationally encoding to reduce the information content of the fringe pattern. Encoding schemes that reduce information bandwidth often exploit gaps in the spectrum of a signal. Physical or interference-based fringes generally have a continuous non-zero spectrum, making lossless encoding impossible. Only a fringe computation method that employs discretized spatial frequencies can hope to achieve encoding for bandwidth compression.

The conventional approach to image and data compression is to compute the desired data set, and then to encode the data set into a “compressed” format. This compressed format is subsequently decoded into a replica of the desired data set. Applied to the generation of holographic fringes, such an approach looks like this:

![Diagram of the conventional approach](image)

In general, more total computing power will be used to perform the three computing steps: initial generation, encoding, and decoding. It is therefore more desirable to compute the encoded format directly:

![Diagram of the direct encoding approach](image)

This second method requires only two computation steps: direct encoding and decoding. For holovideo, it is necessary to pursue this direct-encoding approach since computing a whole fringe pattern is often prohibitively slow. Therefore, the two holographic encoding techniques - “hogel-vector” and “fringelet” - are designed to
directly compute the encoded formats. This is an unconventional approach to data compression: all common schemes for compression of images, audio, or data generate the full desired data set first, and subsequently encode to produce the “compressed” format. An example of such an approach in the Lempel-Ziv (L-Z) compression technique, described in Section 7.2.5 on page 131.

The success of a compression scheme depends on two main factors: compressibility and fidelity. The amount of bandwidth compression in an encoded fringe pattern is measured by the compression ratio (CR). This is the ratio between the sample counts of the final (decoded) fringe pattern and of the encoded fringe. When the CR is higher, the required bandwidth is reduced. In holovideo, a lower bandwidth means fewer samples to represent each holographic fringe pattern.

It is important that holographic encoding schemes maintain image fidelity. However, some fidelity can be sacrificed provided that this loss is imperceptible to the human visual system (HVS). In particular, the “hogel-vector encoding” and “fringelet encoding” schemes introduce blur into imaged points. The amount of blur must be kept below the amount perceivable to the human visual system.

### 3.2 Bipolar Intensity

Bipolar intensity was developed to eliminate the problem of noise inherent to interference-based fringe computation. Simply stated, the bipolar intensity method is to compute only the terms of the expression for the total fringe pattern (Equation 6) that actually diffract useful image light. This leaves only the last term of Equation 6, henceforth called the bipolar fringe intensity. The bipolar intensity term results from the interference between the object wave front and the reference beam. This fringe pattern contains the holographic information that is sufficient to reconstruct an image. In the bipolar intensity method of computation, the object self-interference and the reference-bias terms are simply excluded during computation. The dc bias of the reference term ensures that a physical fringe pattern contains only positive definite values, as is
necessary for a real physical intensity. Computed intensities, however, can be bipolar (i.e., can range both positive and negative), making the dc reference bias unnecessary. If a fringe pattern needs to be positive definite, then a dc offset can be added in during the normalization process. Normalization is the numerical process that limits the range of the total fringe pattern by introducing an offset and a scaling factor to tailor the fringe pattern to fit the requirements of a display system.

As discussed in the references by the author, this expression was simplified to involve only real-valued arithmetic, resulting in a computation speed increase of a factor of 2.0. There are many advantages to the use of bipolar intensity computation. There is no object self-interference noise. There is no reference bias - in fact there is no need to specify the reference beam intensity - resulting in a more efficient use of the available dynamic range of the fringe pattern. The most interesting advantage of the bipolar intensity method is that linear summation of elemental fringes is possible, with each elemental fringe representing a single image element. Real-valued summation enabled the efficient use of precomputed elemental fringes, an approach which, when implemented on a supercomputer, achieved CGH computation at interactive rates.

### 3.3 Precomputed Elemental Fringes

The bipolar intensity computation method allows for real-valued linear summation of fringe patterns. To increase computing speed, a large array of elemental fringes were precomputed and stored for later access during actual fringe computation. Each precomputed elemental fringe represented the contribution of a single image element located at a discrete 3-D location of the image volume. Linearity allows for the scaling of a given elemental fringe (to represent the desired brightness of an image point) and then summation at each applicable sample in the fringe. The complexity of computation for each image point was reduced to only one multiplication and one addition. Speed increased by a factor of about 25 when implemented on a Connection Machine Model 2 (CM2) supercomputer with 16K data-parallel processors, and by a factor of over 50 when implemented on a serial computer.
It must also be noted that the holographic patterns computed by the bipolar and precomputed-elemental-fringe approaches were equivalent, with the following exceptions. Use of the bipolar intensity eliminated object self-interference and dc terms, making the CGH brighter and less noisy than when using the complex method. The look-up table approach resulted in a pattern that was identical to the bipolar intensity approach, with the addition of some quantization noise if the precomputed elemental fringes were stored in a highly quantized format. However, for objects of sufficient complexity, this quantization noise was comparable to that of the more straightforward approaches, due to the quantized nature of the output frame-buffer device used in the system.

A more thorough discussion of the bipolar intensity method and the use of precomputed elemental fringes is included in the references by the author. Here, the important points are summarized:

- The bipolar intensity approach eliminated the unwanted noise terms in the fringe pattern.

- The linearity allowed in the bipolar intensity method enabled the linear superposition of precomputed elemental fringes.

- This early attempt at moving away from traditional interference-based fringe computation provided the first reported computation of holographic fringes at interactive rates.

Even though the interactively generated images were small, this early work demonstrated the power of linear summation and the possibility of generating a fringe as a linear combination of precomputed fringes provided. These two concepts provided guidance for the development of diffraction-specific computation.
3.4 Conclusion

The problems inherent to interference-based computation are noise, lack of speed, the analytical constraint, and lack of encoding schemes. Although the use of bipolar intensity and precomputed elemental fringes successfully addressed the problems of noise and (to some degree) speed, there was still no clue to the development of a holographic encoding scheme that can reduce the necessary bandwidth at some or all stages of the computing pipeline. And there is always room for more speed. The fastest computation performed on the 16 Kprocessor CM2 yielded interactive rates, but processors remained idle on average during over 50% of the computing cycle. This was the result of computing too closely to the physical model of optical interference. This was the fundamental limitation to traditional and bipolar intensity methods of fringe computation. To solve all of the problems of interference-based computation required a completely new approach: diffraction-specific computation.
Chapter 4

Diffraction-Specific Computation

The purpose of this thesis is to design, implement, and analyze a new "diffraction-specific" fringe computation method. It solves the four main problems inherent to traditional (interference-based) holographic computation outlined in the previous chapter. Stately simply, the diffraction-specific approach is to consider only the reconstruction step in optical holography. The diffraction-specific method is inspired by the early work in bipolar intensity and precomputed elemental fringes, which together eliminated unwanted noise components in the fringes and increased computation speed by a factor of over 50. The bipolar intensity method eliminates the unwanted terms in the numerical simulation of interference. Diffraction-specific computation eliminates the simulation of interference altogether.

**Interference-Based**

3-D Scene Description → Interference → Fringes → Diffraction → 3-D Image

Traditional interference-based fringe computation imitates the interference step in optical holography. Computed fringes are used to diffract light to form an image.

**Diffraction-Specific**

3-D Scene Description → Diffraction Backwards → Fringes → Diffraction → 3-D Image

In diffraction-specific fringe computation, fringes are generated by considering only the diffraction process that creates the image. Essentially, diffraction-specific computation generates the fringes through a backwards imitation of diffraction.
The origin of diffraction-specific computation lies in the most fundamental question of computational holoxygraphy: What should the fringe pattern do? Early work to answer this question involved summing elemental fringes to construct a hologram of a particular image. Each elemental fringe was precomputed using interference-based methods to represent a particular image element. However, the real job of a fringe pattern is to diffract light in a particular way. Diffraction-specific computation provides a more direct method of computing a fringe pattern: summing precomputed fringes that represent specific diffractive functions rather than image elements. Instead of precomputing an array of elemental fringes where each represents a different image point in space, diffraction-specific computation uses a precomputed set of basis fringes where each basis fringe represents a specific, independent, and useful diffractive purpose. The following diagram shows schematically the general process of diffraction-specific computation. By separating the 3-D scene description from the fringe computation by a special set of instructions called “diffraction specifications,” diffraction-specific computation can be tailored to the most elusive problem of holographic computation, namely, the need to encode fringes to reduce the information content. As this thesis demonstrates, the basis fringes are specially precomputed to allow for encoding of the fringes. Speed results from the simplicity, efficiency, and directness of basis-fringe summation. Non-analytical image elements can be produced using special sets and combinations of basis fringes. The elimination of unwanted interference terms can be achieved by simply not including them in the fringe precomputation.
4.1 Recipe for Diffraction-Specific Computation

The ingredients to diffraction-specific computing are as follows:

- **Spatial discretization**: The hologram plane is treated as a regular array of functional holographic elements (called “hogels”).

- **Diffraction specifications**: 3-D object scene information is first converted into a description of the diffractive purpose of each “hogel” constituting the fringe pattern. These specifications, called “hogel vectors,” are based on sampling the fringe spectrum.

- **Basis fringes**: A set of basis fringes are precomputed and used to map each diffractive specification (“hogel vector”) to the corresponding fringe pattern (“hogel”) contribution.

- **Rapid linear superposition**: The diffraction specifications (“hogel vectors”) are combined with the precomputed basis fringes to generate physically usable fringes.

The basis fringes are the most important ingredient in designing diffraction-specific fringe computation. The requirements on a basis fringe are many. Each basis fringe performs a specific diffractive duty. Consider diffraction in the local region of a fringe
pattern (shown in the following figure). The angles of the incident and (first order) diffracted beams are related by the grating equation:

\[ f \lambda = \sin \Theta_0 - \sin \Theta_i \]  

(7)

where \( \Theta_i \) is the angle of the incident light and \( \Theta_0 \) is the angle of a diffracted beam, \( \lambda \) is the wavelength of light, and \( f \) is a spatial frequency component. (For a more detailed discussion of diffraction as a function of fringe spatial frequency, see Appendix B, "Spectral Decomposition of Diffracted Light.") If the fringe pattern contains a specific spatial frequency component, then light is diffracted in a specific direction. Using the grating equation in reverse, if the fringe is required to diffract light to a specific direction, then the fringe must contain a specific frequency component. The spatial frequency content, i.e. spectrum, is linked directly to the diffractive properties of a fringe pattern. Fringes produced physically or through interference-based computation generally have continuous spectra since they diffract light in continuous ranges of directions. However, because the acuity of the human visual system (HVS) is finite, it is possible to discretize the spectrum of a fringe pattern without visible image degradation. To accommodate bandwidth compression, the basis fringes should have spatial frequency characteristics that allow for encoding of the fringes using fewer samples. Finally, because the fringe pattern is treated as a regular array of spatially discrete units, each basis fringe must occupy a spatially finite region.

The many constraints on each basis fringe make their computation intractable using interference-based or any kind of analytical approach. To satisfy all of these requirements, numerical methods must be used. As described in Appendix C, "Computation of Synthetic Basis Fringes," the basis fringes are computed using a numerical iterative
constraint approach in which the specific spectral and spatial characteristics of each basis fringe are alternately applied as constraints until a satisfactory fringe is derived.

At the heart of diffraction-specific computation is the spatial and spectral discretization of the fringe pattern. The generation of diffraction specifications ("hogel vectors") and their conversion into fringes are based on the discretized model of the fringe. The following Section 4.2 describes the spatial discretization of the hologram plane and the spectral discretization of each functional diffractive holographic element ("hogel"). Section 4.3 is a description of the conversion of 3-D scene information into diffractive specifications called "hogel vectors." Section 4.4 is a description of the last step of the computation process, in which "hogel vectors" are combined with the precomputed basis fringes to produce a full usable fringe pattern. Section 4.5 describes the implementation of diffraction-specific computation in this thesis work. The final sections include pictures of images generated using diffraction-specific fringes as well as a discussion of image quality and computation speeds.

4.2 Discretization of Space and Spatial Frequency: "Hogels"

The purpose of diffraction-specific hologram computation is to generate a fringe pattern that diffracts light in a specified manner. Given a fringe pattern, the law of diffraction can be applied to determine how it will diffract light. In a sense, this is a backwards method of computing because it only determines if what was computed is correct, but does not allow for direct computing of the fringe pattern. Appendix C, "Computation of Synthetic Basis Fringes," shows how numerical methods can compute fringes using of this backwards approach, i.e., a fringe pattern can be computed given a specified image. However, these methods are far too slow to be implemented for full on-the-fly hologram computation.

The solution - the diffraction-specific fringe computation method - is to precompute elemental fringe patterns (called basis fringes) that can be composited to form a specific image. Because diffraction is linear, each of these precomputed basis fringes can
be selected, weighted, and summed to form a part of the holographic fringe pattern. This is similar to the bipolar intensity method (described in Chapter 3) in which the rapid linear superposition of elemental fringe patterns builds up a fringe one image element at a time. In the diffraction-specific approach, however, the fringe pattern in a small region of the hologram is built up independently of the others, and the component fringes correspond not to specific image points but to the spectral characteristics of the desired fringe. (See Appendix B, “Spectral Decomposition of Diffracted Light” on page 153.) The first step is to divide the fringe into pieces of equal width, and to divide the spectrum within each piece into evenly spaced discrete steps. In other words, the fringe pattern is sampled in space and in spatial frequency. The sample spacing must be selected in a way that allows for the fringe pattern to diffract light to form the desired image.

4.2.1 Sampling: Concepts

Sampling and subsequent reconstruction of a signal is common in communication systems involving continuous physical properties (e.g. acoustic signals, images, metrological data). The sample spacing must be sufficiently small to capture and represent all of the important features of the continuous signal. In computational holography, one does not seek to sample a continuous physical fringe, but rather to compute a fringe specified by its diffraction function using the smallest numbers of samples possible.

The wavefront of the first-order diffracted wavefront is the actual physical entity being represented by a fringe pattern. As detailed in Appendix B, “Spectral Decomposition of Diffracted Light,” the wavefront immediately following diffraction by a fringe is expressed as a summation of plane waves, each diffracted by a different spatial frequency component of the fringe. Diffraction-specific computation is therefore a spectrum-specific approach: a fringe spectrum must be computed so that the fringe diffracts the specified light.
Consider an HPO fringe pattern with a spectrum \( S(x,f) \), where \( x \) is the lateral position on the hologram and \( f \) is the spatial frequency at that lateral position. The spectrum \( S(x,f) \) is an instantaneous local spectrum at each sample \( x \) in the hologram. This may seem odd: how can a single sample possess a spatial frequency content? Nevertheless, sampling theory\(^7\) says that one spatial sample does contribute to the spatial frequency content of the fringe that is being physically represented.

In general, \( S(x,f) \) does not need to vary rapidly between two adjacent fringe samples. Indeed, the spatial frequency does not change rapidly from one sample of the fringe to the next. It is not necessary to sample \( S(x,f) \) as finely as a discretized fringe pattern. The first goal in sampling \( S(x,f) \) is to determine a sufficiently small sample spacing for the dimension \( x \), which is called \( w_h \). The second goal in sampling \( S(x,f) \) is to determine a sufficiently small spectral sample spacing \( (\Delta f) \) for the frequency dimension \( f \). The one-dimensional fringe pattern is treated as a 2-D spectrum that is a spatial array of sampled spectra. When performed correctly, the sampled and recovered \( S(x,f) \) causes light to diffract and to form a 3-D image:

![Diagram of hologram and fringe pattern](image)

Sampling theory\(^7\) guarantees that a signal can be retrieved if it is sampled (in each dimension) at more than twice per period of its highest Fourier component. In particular, let \( S_{ij} \) be the spectrum \( S(x,f) \) sampled in space and spatial frequency at intervals of
\( w_h \) and of \( \Delta f \). According to the sampling theorem, the spectrum of one spatially limited fringe (of width \( w_h \)) at \( x_i \) can be recovered from \( S_{ij} \) through a convolution with a sinc function of first-node full-width \( 2/w_h \):

\[
S(x_i,f) = \sum_j S_{ij} \text{sinc}(j-fw_h) \tag{8}
\]

Furthermore, \( S(x,f) \) can be fully recovered through a convolution with a sinc function of width \( 2/\Delta f \):

\[
S(x,f) = \sum_i \sum_j S_{ij} \text{sinc}(j-fw_h) \text{sinc}(i-x \Delta f) \tag{9}
\]

These convolutions are equivalent to performing a low-pass filtering to the fringe pattern, i.e., to the diffracted wavefront. For the spatial dimension \( x \), the requisite low-pass filtering is provided by the process of diffraction and by the imaging function of the viewer’s eye. For the spectral dimension \( f \), the convolution is performed (in part) by the weighted summation of basis fringes, where each basis fringe represents one of the spectral regions \( j \). Essentially, each basis fringe fills in a specific region of the fringe spectrum. In reality, a sampled signal cannot be processed using ideal low-pass filtering. For example, when basis fringes have gaussian spectral shapes rather than sinc-function shapes, the resulting spectral cross-talk adds some noise to the image.

### 4.2.2 Spatial Sampling

To sufficiently sample \( S(x,f) \), the most rapid variations in \( S(x,f) \) must be determined as separate functions of \( x \) and of \( f \). These are determined by the imaging requirements of a holographic imaging system. Given one spatial frequency \( f_0 \), the spatial sample spacing \( w_h \) must be small enough so that the most rapid amplitude variations in \( S(x,f_0) \) can be reconstructed. Consider a holographic imaging system in which a fringe consisting of a single constant spatial frequency diffracts light to a single location at the viewing zone:
In this case, this spatial frequency diffracts light to a viewer's eye at a particular location. The magnitude of the $f_0$ component must vary to represent what the viewer sees from this one point of view. What is the smallest variation in an image that the human visual system can see? The HVS can laterally resolve approximately 1" of arc or approximately 290 milliradians. For a typical viewing distance of 600 mm, the calculation of $w_h$ becomes $w_h = (0.000290) \times 600$ mm $= 0.175$ mm.

### 4.2.3 Spectral Sampling

Next, consider a fringe region of width $w_h$ centered at position $x_0$. The spatial frequency sample spacing $\Delta_f$ must be sufficiently small to reconstruct the most rapid variations in $S(x_0f)$ as a function of $f$. Each discrete spatial frequency diffracts light to a specific location in the viewing zone:
Therefore, the variations in $S(x_0 f)$ depend on the most rapid variations in light at the viewing zone. View-dependent image qualities (e.g. parallax, specular highlights, occlusion) give rise to these viewing zone variations. The ability of the human eye to detect view-dependent variations is limited by the size of the pupil which is typically 3 mm. The angle subtended by the human pupil therefore determines the maximum allowable $\Delta f$. The grating equation (Equation 7) relates the diffracted angle $\Theta_0$ to the incident angle $\Theta_i$ of the illuminating light. Taking the derivative of both sides with respect to $\Theta_0$:

$$\frac{\partial f}{\partial \Theta_0} = \frac{\cos \Theta_0}{\lambda}.$$  \hspace{1cm} (10)

For a viewing distance of 600 mm and a $\lambda=0.633 \mu$m (and allowing the $\cos$ term to be its maximum of 1), the calculation of $\Delta f$ becomes

$$\Delta f = \frac{(3mm) / (600mm)}{0.633 \mu m} = 8 \hbox{ \mu m}^{-1}.$$ \hspace{1cm} (11)

In a small holographic fringe of width 1024 samples, where each sample is spaced by $1/p=0.6 \mu$m, the spatial frequencies range from 0 to 850 mm$^{-1}$ with a spacing of 0.8 mm$^{-1}$ - a factor of 10 times less than the calculated value for $\Delta f$. Under these typi-
cal holovideo conditions, not only is the representation of a spatially and spectrally sampled fringe adequate for the reconstruction of images, but actually contains roughly tens times more information than is required by the HVS. The fringe can be encoded in such a way as to compress the necessary bandwidth, but preserve the useful image information. Holographic encoding schemes developed to exploit the oversampled nature of fringe spectra are the subject of the latter chapters of this dissertation.

4.2.4 Introduction of the Hogel

By sampling a fringe pattern at regular intervals of width $w_h$, the fringe is treated as an array of small functionally diffractive elements. Each discrete element is given the new name of hogel for holographic element. A hogel is simply a small region of a fringe pattern. In HPO holograms, a hololine is divided into evenly spaced hogels, each representing a small line-segment region of the hologram plane. (For a full-parallax hologram, a hogel is the fringe pattern in a rectangular region, regularly dividing the hologram plane vertically and horizontally.)

A hogel possesses two important qualities: (1) a homogenous spectrum; (2) a size small enough to appear (to the viewer) as a point. The possession of a homogenous spectrum is equivalent to stating that the hogel is a sampling of $S(x,f)$ at intervals of $w_h$ and $\Delta f$. This allows for the entire hogel to be described simply by describing its spectrum. The second quality (small size) is equivalent to choosing $w_h$ to be smaller than is resolvable by the HVS. Also, wide hogels do not sufficiently sample the curvature of the wavefront to be diffracted - as encoded in $S(x,f)$ - leading to degradations in the image. If the hogel is too wide or too narrow, then aperture effects cause the image to appear to be blurry.

A hogel is sufficiently described by its spectrum. (See Appendix B.) For a given hogel the set of spectral components that describe the hogel spectrum is given the new name hogel vector. A hogel vector is simply a small vector of weighting factors in which each component represents the amount of spectral energy within a small range of the spectrum. The components constituting a given hogel vector specify the diffractive
duty of that hogel. Thus, a hogel vector is the diffraction specification for a hogel, and the hogel-vector array \( S_{ij} \), the sampled version of \( S(x,f) \) — is the diffraction specifications for the entire hologram.

### 4.3 Generation of Hogel Vectors

The first step to using diffraction-specific computation is to compute the array of hogel vectors. The 3-D description of the scene to be imaged must be converted into the hogel-vector array using essentially a ray-tracing algorithm. Traditional computergraphics ray-tracing uses geometric optics to numerically “propagate” light from the objects in a scene to the rendering window\(^8\). In the case of hogel-vector generation, geometric optics is employed to calculate the amount of light that a particular hogel must diffract in each discretized direction. As discussed in Appendix B, this diffraction-direction information is related directly to the spectral regions of the hogel. Therefore, a holographic ray-tracing algorithm was developed to map desired image features directly to components of each hogel vector.

For illustration, consider the case of a point source located at the holoplane. (The holoplane is the plane containing the fringe pattern.) To diffract light to form this point, the hogel in which this point is located must diffract light in all directions. Therefore, a point source on the holoplane contributes an equal amount to all of the components of a single hogel. Each of the basis fringes must be weighted by the appropriate amplitude and summed together to form the hogel (fringe) that will diffract light to produce the image of this point source.

Next, consider a point source located at a distance in front of the hologram, i.e., between the holoplane and the viewer. To diffract light to form this point, light must be diffracted in slightly different directions from each adjacent hogel in a finite linear region of a hololine. One or more components of each hogel vector must include a contribution for this point. Therefore, one or more basis fringes must be weighted and summed at each of the affected hogels.
4.3.1 Diffraction Tables

Generally, an image point or element at some \((x,y,z)\) location in the image volume contributes to particular components of particular hogel vectors. In an HPO hologram these hogels are all on the same line, 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 receive contributions, i.e., which basis fringes must be weighted and accumulated to produce the image element. For this thesis research, the mapping from \((x,z)\) to hogel position and vector component (basis index) was precalculated using simple geometric optics and stored in a table. This table — called a diffraction table — rapidly maps a given \((x,z)\) location of a desired image point to the appropriate hogel-vector components in the array of hogels. As shown in the figure below, this table selects which basis fringes are to contribute to each hogel. The diffraction table must also include an amplitude factor for each entry. This factor is multiplied by the desired amplitude (taken from the 3-D scene information) to determine the precise hogel-vector component contributions. These amplitude factors are necessary to maintain energy conservation. Consider the following two imaging cases: (1) to image a point at a particular depth \(z_0\), the appropriate component (as dictated by
the diffraction table) of each hogel vector is incremented by one unit; (2) to image a point at a depth of \( z_0/2 \), two components per hogel vector must be incremented. If each hogel-vector component in Case 2 is incremented by one unit, the resulting hogel has twice the amplitude of the hogel in case 1. Therefore, the diffraction table must request that hogel-vector components are incremented by only half a unit in case 2. The amplitude factors also allow for directionally dependent qualities (e.g., specular highlights) when a diffraction table is used to represent more complex image elements.

The amplitude of an image element is determined from its desired brightness. This brightness is represented as an intensity (or energy) that is equal to the square of its amplitude. This amplitude is used in calculations as the amplitude of the image element. As an additional subtlety, holovideo display nonlinearities may require that each brightness is mapped to an adjusted value to create more accurate imaged results. (The MIT second generation holovideo display required a large amount of nonlinear processing of image brightness values.) This nonlinear mapping is accomplished using a look-up table, often generated using an exponential function. (Such an exponential mapping is often referred to as "gamma correction" when applied to mapping for 2-D display on a CRT.)
Each scene element is processed using a diffraction table, and the resulting array of hogel vectors contains all of the contributions necessary to compute the fringe pattern. Since speed of computation is a primary motivation for diffraction-specific computing, it is important to note that the use of the diffraction table is very fast.

4.3.2 Use of 3-D Computer Graphics Rendering

Another approach to performing the ray-tracing calculations described above is to begin with standard 3-D computer graphics rendering software\(^8\). The rendered views provide directional information that are then converted into hogel-vector information using a modified diffraction table. To implement hogel-vector generation using a rendering algorithm, the view window (the plane upon which the scene is projected) must be coincident with the hologram plane. The view-point must be at \(z \to \infty\); each rendered view is an orthographic projection of the scene as seen from a particular view direction. The picture element spacing in the 2-D rendering of the scene should be smaller by at least a factor of two than the spacing of the hogels and hololines. This allows for subsampling during the conversion of rendered views into hogel vectors. This subsampling is necessary to generate the amplitude factors from the diffraction table and also reduces image artifacts.

The advantages of the use of existing computer graphics algorithms are numerous. Many powerful and highly developed 3-D rendering packages exist, each capable of providing advanced image properties, such as specular reflections, texture mapping, advanced lighting models, scene dynamics, and viewer interactivity. Specialized rendering engine hardware also exists, making the generation of complex scenes very fast. In addition, rendering engines can be utilized in the conversion of hogel-vectors into hogels\(^9\).

4.3.3 Additional Techniques

Diffraction-specific computation provides an easy solution to the problem of analytically constrained image elements (discussed in the previous chapter, Section 3.1.) To
employ higher-level image or scene elements, multiple diffraction tables are used. For example, if line segments or polygons of various sizes are useful for assembling the image scene, then a diffraction table are used to map location, size, and orientation of the element to the proper hogel-vector contributions.

Finally, the issue of color holovideo images must be addressed. The most straightforward approach is to compute 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\textsuperscript{54}. Three separate sets of basis fringes are precomputed and used for the three respective fringe computations. Each of these separate fringes can be computed independently. However, it is interesting to note a single diffraction table can be used for all three wavelengths. The diffraction table is wavelength-independent. All of the physics of diffraction, including the wavelength dependence, is incorporated in the basis fringes. Basis-fringe selection via hogel-vector components can proceed using a single diffraction table, provided that the shorter wavelengths are limited to a smaller range of diffraction directions. Therefore, to produce a full-color fringe pattern, three separate hogel-vector arrays are generated using the same diffraction table, and each is converted into three separate fringes using the linear summation of three separate sets of basis fringes.

4.4 Converting Hogel Vectors to Hogels

The second step in diffraction-specific fringe computation is to convert the array of hogel vectors into an array of hogels, i.e. fringes. This is a straightforward though time-consuming operation involving weighting and summing basis fringes at each hogel location. As shown in the following figure, to compute a given hogel, each component of the hogel vector is used to multiply the corresponding basis fringe. The weighted basis fringe is then added to the hogel. The accumulation of all the weighted basis fringes is the resulting hogel (fringe). This is the more time-consuming step in diffraction-specific fringe computation. 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 a high speed.

**Conversion of Hogel Vectors to Hogels**

Integrated into the set of basis fringes is the exact imaging function of the holovideo display to be used. These fringes (hogels) diffract light according to the equation (Appendix B, Equation B13) used to map diffraction angle to spatial frequency:

\[ f = \frac{\sin \Theta_o}{\lambda} - \frac{k_x^i}{2\pi} \]  \hspace{1cm}  \text{(12)}

where \[ k_x^i = \frac{2\pi}{\lambda} \sin \Theta_i \]  \hspace{1cm}  \text{(13)}

is the \(x\)-component of the effective direction of light incident on the modulator of the display system, and \(\Theta_o\) is the angle of a diffracted ray of light. This parameter, as a function of hogel position in \(x\), was empirically derived by measuring the directions in which light is imaged by fringes consisting of constant spatial frequencies. The basis
fringes were computed (see Appendix C) incorporating the measured optical characteristics of the display system. If changes are made to the display, or if another display is used, the basis fringes must be regenerated. However, for simple changes in the display system, e.g., changes in $k_x^i$, the generation of the hogel-vector array is the same; only a different set of basis fringes must be used. The precomputed basis fringes incorporate the parameters of a display and can be used “locally” by a given display. Therefore, a hogel-vector array computed for one holovideo display can be used on different displays. For large changes in size or in image depth (relative to the holoplane), the hogel vectors should be computed with the specific parameters of the display. Nevertheless, it is an interesting feature of diffraction-specific fringe computation that the intermediate hogel-vector fringe description is display-independent within reasonable bounds. This display independence, called “interoperability,” is discussed further in Section 7.2.1 on page 127.

### 4.5 Implementation of Diffraction-Specific Computing

Diffraction-specific computing was implemented and used to generate holographic fringes. The general data flow is diagramed below. There are two computation steps

![Diffraction-Specific Fringe Computation Diagram](diagram.png)

(represented by the two arrows). The first step, generation of the hogel vectors, was implemented on an SGI Onyx workstation, a standard high-end serial processing computer. This step involved a wide range of calculations, but was relatively fast. The second computation step, the conversion of hogel vectors into hogels, was implemented in the “Cheops” framebuffer system used to drive the MIT holovideo display. Because
this step involved a large number of simple calculations, it was necessary to implement it on specialized Cheops hardware to obtain rapid computation speeds. The Cheops framebuffer system is described briefly in the following subsection.

4.5.1 Cheops Overview

The Cheops Image Processing system is a compact, block data-flow parallel processor designed and built for research into scalable digital television. The preceding figure shows a block diagram of the Cheops system used in this thesis research. For the MIT holovideo display, the primary function of Cheops is to support 18 parallel 2-MB framebuffers. This function is performed by six Cheops output cards, each providing three channels of high-speed analog information to the MIT holovideo display system. Cheops also contains a processor card, the P2, that manages data transfers and is capable of performing computations. It contains an Intel i960 RISC microprocessor and 32 MB of read-write RAM. Data is communicated between the P2 and output cards using either a slow Global Bus or a fast Nile Bus. The P2 communicates to the outside world
(in this case, the SGI Onyx workstation) via a SCSI link with limited bandwidth. (SCSI stands for Small Computer Standard Interface.) The P2 also supports a special stream-processing superposition daughter card called the "Splotch Engine" that performs weighted summations of arbitrary one-dimensional basis functions. The Splotch Engine performs the many multiplication-accumulation (MAC) operations required for the conversion of hogel vectors into hogels.

For speed comparison, the CM2 was also used to perform the conversion of hogel vectors to hogels. Also for comparison, the entire diffraction-specific computation pipeline was implemented on the SGI Onyx serial workstation. This allowed for comparisons of different computation techniques within a fixed computational platform.

4.5.2 Normalization

Normalization prepares the final computed fringe to be displayed using a particular holovideo display system. Consider the second generation MIT holovideo display. The Cheops output card VRAM stores each fringe sample as an 8-bit unsigned integer value. (VRAM is video RAM, i.e., fast dual-ported read-write random-access memory.) Computed fringes must be numerically processed so that they will fit within these 256 values. Normalization generally involves adding an offset and multiplying by a scaling factor. The offset ties the minimum normalized value to 0, and the scaling ensures that no normalized value exceeds 255. In diffraction-specific computing, normalization is built into the computational pipeline. For example, when using the Cheops Splotch Engine to perform the final computation step, the basis fringes and hogel-vector components are engineered to have values that produce useful fringes in the higher 8 bits of the 16-bit result field. The low byte of the result is ignored, and the high byte is sent to the output card VRAM.
4.6 Image Generation

Diffraction-specific fringe computation was performed on the Onyx/Cheops computational platform. Computed fringes were fed to the MIT second-generation holovideo display and used to generate 3-D holographic images. The process began with a 3-D image scene description generally consisting of about 0.5 MB of information or less. After performing the appropriate scaling, rotations, lighting, and shading, this 3-D information was used to generate an array of hogel vectors consisting of 36 MB. The array of hogel vectors was downloaded to the Cheops P2 where it was converted into hogels using the Splotch Engine. Finally, the hogel array, consisting of 36 MB, was sent to the output cards.

Digital photographs were taken of images and close-ups of images to analyze image quality. Pictures of individual image points were used to determine the resolution of images generated using diffraction-specific computation.

The following figure shows a digitally photographed picture of a 3-D image produced on the MIT holovideo display. This particular image was of a Honda EPX concept car, modeled using a computer-aided design system. The design database was used to compute the holographic fringe pattern.

![Image of a Honda EPX Concept Car](image_url)
4.6.1 Photographing of Images

Several technical points must be described before proceeding with the analysis of generated images:

- Images produced on the MIT second generation holovideo display consisted of 144 discrete hololines, and measured roughly 150 mm by 75 mm (by 160 mm in depth).

- The discrete hololines often produced visible artifacts due to imbalances and nonlinearities in the RF signal-processing electronics of the display system. Note (in the previous picture) that the individual hololines are evident in the horizontal streaks and bands of light and dark appearing in the photograph. The 18-channel display system was not correctly balanced to provide signals of equal strength over their full operating range. The resulting nonlinearities produced the horizontal artifacts which are merely a nuisance and are not important to this dissertation.

- Images generated by this display can be as deep as 80 mm in front of or behind the holoplane. Deeper images suffer from astigmatism in this HPO display. Thus, \( |z| = 80 \) mm is the worst-case imaging scenario. For this reason, a point imaged at \( z = 80 \) mm was used to test the worst-case image resolution for the various computation methods presented in this dissertation.

- Digital photographs of full images were acquired using a Kodak DCS 200 camera consisting of a Nikon N8008s body and 105-mm lens and a CCD (charge-coupled device) backplane array of 1524x1012 tricolor pixels. Exposure times were 1 to 4 seconds, with apertures ranging between \( f/8 \) and \( f/22 \). Exposure time and aperture size were held constant for comparable images. Image quality suffered from lack of depth of focus, from speckle, and from artifacts present in the display. Moire patterns are evident in some pictures due to the periodically spaced CCD elements and holographic image elements.
• Digital photographs of close-ups of images were acquired using a Sony CCD array placed directly at the plane of the image. The CCD array measured 768x494 tricolor pixels, with approximately 8-μm spacing horizontally. Each frame was grabbed and digitized using an SGI Sirius Video system. Frame integration time (exposure time) was approximately 0.03 s.

• In each case, the red separation of each digital photograph was selected, downsized to fit the size of this document, and half-toned to allow for binary printing and copying. The half-tone screen adds some slight artifacts to the pictures.

• A horizontal point-image profile (see, for example, the figure on page 68) was obtained from each digitized close-up photograph by vertically integrating over the vertical extent of the image. An effective width was calculated from each profile by horizontally integrating the profile and calculating the narrowest range where half of the energy is contained. This simple measure of spotsize is used to determine the imaging resolution for various computational methods.

4.6.2 Point Images

The figure on page 68 shows a point focused at 80 mm in front of the hologram plane (i.e., focused between the hologram plane and the viewer) using traditional computation methods and using diffraction-specific computation methods. The point image generated using interference-based computation methods has an effective width of 0.144 mm. The diffraction-specific point is blurred to twice this width. This additional blur is caused primarily by the spatial sampling of the hologram plane, namely, the width of the hogel in this case is $N_h=512$ or $w_h=0.3$ mm. Although diffraction-specific computation has added some blur to the image, notice that there is no additional noise or other artifacts. Consider also that the diffraction-specific computation was more than twice the speed of the interference-based computation when implemented on the CM2.
Point imaged at $z=80$ mm

Traditionally Computed Point

Effective width = 0.144 mm

Diffraction-Specific Point

$w_h=0.300$ mm

Effective width = 0.288 mm

Top: A point imaged at $z=80$ mm in front of the hologram plane. This point was generated using traditional interference-based computation. The graph shows a cross-section of the focused point.

Bottom: A point computed using diffraction-specific computation. The hogel width was $N_h=512$ samples, or $w_h=0.3$ mm. The graph shows a cross-section of the focused point.
The increase in point spread due to diffraction-specific computation is discussed in the following chapter (Section 5.4). For now, the empirical behavior of point spread is illustrated in the figure on page 70. Selecting $w_h=0.3$ mm gives the best performance. A larger $w_h$ blurs the imaged point, and a smaller $w_h$ blurs the imaged point for the deeper point at $z=80$ mm.

4.6.3 Incoherent Illumination Considerations

Holovideo displays have used coherent laser light, but the general requirement is that the light be quasi-monochromatic and not necessarily coherent. The effects of incoherent illumination on the diffraction of light must be considered when using the diffraction-specific approach. Generally, the coherent analysis of spectral decomposition of diffracted light (Appendix B) puts more strict requirements on the fringe computation than does the incoherent treatment. The incoherent results can be interpreted directly from the coherent results.

Diffraction-specific fringe computation, as implemented in this thesis, assumes that the light is quasi-monochromatic with a maximum coherence length $L_c$ of less than half of a hogel width, i.e., $L_c\sim0.100$ mm or less. This ensures that the summation of spectral components (basis fringes) can be performed linearly, and that the diffracted light adds linearly with intensity. In practice, the holovideo display that was used to generate real-time images from the computed fringes used coherent laser light. The effective coherence length of this illumination source was reduced due to the scanning and modulating performed by the display system. Nevertheless, significant coherence remained, estimated to be $\sim2.0$ mm. This significant coherence resulted in speckle in the holographic image. Instead of diffracted light incoherently adding intensities, the partially coherent light exhibited coherent summation of complex amplitudes. The resulting speckle artifact appeared as annoying variations in image intensity at infinity.

Several techniques were used to successfully reduce the speckle effect. The most effective means was to essentially introduce into each hogel a random set of phases in each basis fringe. This had the effect of reducing the correlation among rays of light.
This figure shows the effects of spatial quantization. Shown here are a point at \( z = 80 \) mm and a point at 40 mm, generated using diffraction-specific computing and a variety of hogel widths \( (w_h) \). For the point at \( z = 40 \) mm, point blur increases roughly linearly with hogel width. The point at \( z = 80 \) mm is more blurry for \( w_h = 0.15 \) mm than for the larger \( w_h = 0.30 \) mm. This is primarily the contribution of spectral blur.
diffracted by each hogel. To implement these random phase components, a large set of
different but spectrally equivalent basis fringes were precomputed. During the conver-
sion step from hogel vector to hogel, a particular basis fringe was selected at random
from the set of basis fringes, and computation proceed as usual.

4.7 Speed

It is difficult to compare the computing times involved in the interference-based versus
diffraction-specific methods. Both scene complexity and implementation hardware
vary among computing tasks. In particular, selection of scene complexity is arbitrary.
Diffraction-specific computation is independent of image scene complexity. On the
other hand, the computing time for interference-based ray-tracing computations varies
roughly linearly with image scene complexity. And the range of complexities in which
an object can be useful to the viewer is a function of image volume which is in turn
related to the number of samples in the fringe pattern. Also, different computational
platforms have varying computational power and transfer bandwidth. For comparison,
these transfer times are not included, except where noted. In analyzing and comparing
computation speeds, an effort is made to make benchmarks as equivalent as possible.

In diffraction-specific computation, most time was spent converting the hogel vectors
to the hogels. This was due to the large number of samples in the final fringe pattern
and the large number of basis fringes to be summed. To convert an \( N_h \)-component
hogel vector to an \( N_h \)-sample hogel requires calculating their inner product, i.e., \( N_h^2 \)
multiplication-accumulation operations (MACs). For example, for a hogel width of
\( N_h=512 \) (\( w_h=0.3 \)), each sample required 512 MACs, independent of object complex-
ity. Although this is a large amount of calculations per sample (~\( 10^6 \) per hogel), not all
of these basis fringes are necessary to produce a reasonable image. Chapter 5 demon-
strates that the high degree of redundancy present in this type of calculation can be
eliminated, leading to a factor of 10 speed increase.
On the CM2, conversion of hogel vectors to hogels was very efficient. No processors remained idle during fringe computation. To convert a hogel-vector array into a 36-MB fringe on the CM2 required 350 s for $N_h=512$. (Over 40% of this time was spent passing the hogel-vector array into the CM2.) Hogel-vector direct encoding required 20 s, for a total of 370 s. For comparison, traditional interference-based computation required 790 s (~13 minutes) using the same fairly complex image of 20,000 discrete points (roughly 128 imaged points per hololine). Diffraction-specific computation is a factor of 2.1 times faster than the traditional interference-based approach when implemented on the CM2.

Hogel-based diffraction-specific computation makes efficient use of computing power. The advantage of diffraction-specific computing is that the slowest step – the conversion of hogel-vectors to hogels – is independent of image content and complexity. The number of MACs per fringe sample required to compute a hogel is $N_h$, and these calculations account for nearly all of the computation time in diffraction-specific computation, independent of image content. For example, an image composed of 100,000 discrete points (five times the previous example) required over an hour on the CM2 using traditional methods. This image required 350 s using the diffraction-specific method - the same time required for the simpler image.

The hogel-vectors are essentially the fringe pattern in encoded form, and their conversion to fringes is a process that is not only independent of image content but is simple enough to be implemented in specialized hardware. The Splotch Engine on the Cheops framebuffer system was used to perform the conversion step. The Cheops Splotch Engine converted hogel vectors to hogels at a rate of 0.27 ms/hogel for a hogel width of $N_h=512$ ($w_h=0.30$ mm). For a 36-MB fringe pattern, a single Splotch Engine required about 310 s to compute a fringe pattern. This is the worst case for the most complex image scene. In general, typical image scenes produced hogel vectors that were completely zero or that contained many zero components. In these cases, some second-order optimization resulted in speed-ups of a factor of two or three. This optimization consisted simply of skipping zero-valued hogel-vector components. (Such an
optimization was impractical on the CM2.) It is also important to note that when these speed benchmarks were measured, the Splotch Engine was required to make twice the number of passes intimated by the number of MACs required. In other words, the Splotch was running at roughly half of its potential speed. The Cheops/Splotch system is currently being reworked to eliminate this inefficiency. Once solved, the above example should give a computing time of 160 s to compute the entire 36-MB hogel array. This is faster than the time benchmarked on the CM2.

The following listing summarizes the timings for diffraction-specific computation:

- Diffraction-specific on Cheops/Splotch: \(310 \text{ s} + 20 \text{ s} = 330 \text{ s}\)
- Diffraction-specific on CM2: \(350 \text{ s} + 20 \text{ s} = 370 \text{ s}\)
- Traditional interference-based on CM2: \(790 \text{ s}\)
- SCSI transfer time: add 45 s.

The first two numbers (in the diffraction-specific cases) indicate times for the generation of the hogel-vector array and for their conversion to hogels. These two times add to the total computing time (excluding transfer time).

4.8 Conclusion

Diffraction-specific computation successfully generates fringe patterns for the display of 3-D holographic images. Although it is difficult to compare computation speeds between the two fundamentally different approaches, diffraction-specific computing is faster on all accounts than traditional interference-based computing. Although it is a multi-step process, it makes most efficient use of computing power at each step. Most of the computational burden is placed at the final step, in which hogel vectors are converted into fringes. Therefore, this conversion step was implemented using specialized hardware, providing a potential computing time of about three minutes. This speed is still too slow. However, the following chapters describe two holographic encoding
schemes based on diffraction-specific computation that can reduce computation time to as low as one second.

If the hogel vectors are considered to be an encoded form of the fringe pattern, then the conversion of hogel vectors to hogels is equivalent to decoding. The discussion of spectral discretization (Section 4.2.3) indicated that using a full bevy of hogel vector components to encode the hogel fringe was redundant by roughly a factor of ten. Eliminating this redundancy is the cornerstone to the two holographic encoding schemes developed from diffraction-specific computation to provide bandwidth compression. The first encoding scheme is described in Chapter 5, “Hogel-Vector Encoding,” which begins with a discussion of electro-holography as a holographic communication system. The second encoding scheme, built upon hogel-vector encoding, is described in Chapter 6, “Fringelet Encoding.”
As a final observation regarding diffraction-specific fringe computation as developed thus far, there are three important features that characterize the diffraction-specific hogel vector description.

- The computation of a particular hogel can proceed using nothing more than the information contained in a hogel vector.

- No two hogels computed from different hogel vectors give rise to the same viewer stimulus (i.e., what is seen by the viewer).

- (Therefore) no two different hogel vectors give rise to the same viewer stimulus.

These observations are useful in the discussion to follow on information reduction schemes.
Chapter 5

Hogel-Vector Encoding

Diffraction-specific fringe computation provides a foundation for the development of two holographic encoding schemes. In this chapter and the next, hogel-vector encoding and "fringelet encoding" are described and demonstrated. Both are capable of achieving compression ratios of 16 and more, and both dramatically reduce the total time required to generate holographic fringes. These encoding schemes are the fulfillment of the most elusive goal of diffraction-specific computation.

Diffraction-specific fringe computation is developed around the ideas of sampling. A holographic pattern is represented by a sampling of its spectrum as a function of both space and spatial frequency. The information content of such a sampled fringe pattern is equal to the product of the number of hogels times the number of components in each hogel vector. Therefore, the sample spacing in the two sampled dimensions (\(w_h\) in space and \(\Delta f\) in spatial frequency) determine the information content of the sampled fringe representation. To reduce the number of samples, the spectrum of each hogel is sampled in larger frequency steps. This reduction in the total number of samples comprising the hogel-vector array provides a savings of bandwidth at the stages of computation before the hogel-vector array is converted to the final fringe pattern. The final fringe pattern must have a minimum number of samples, as dictated by the physics of diffraction. However, if the hogel-vector description of the fringe pattern contains fewer samples, then the array of hogel vectors can be thought of as an encoded form of the fringe pattern, one that is compressed in terms of information bandwidth.

It is important to compress the huge information content of a fringe pattern so that it can be easily displayed, transmitted, or stored. In particular, because the Cheops framebuffer system (used to drive the second generation MIT holovideo display) is designed to read in data over a relatively low-bandwidth SCSI link, compression is
central to avoiding this bottleneck. To understand the nature of a holographic encoding scheme, fringe computation is next discussed in terms of communication systems. Such a discussion helps to frame the issues of information content in the realm of electro-holographic imaging.

5.1 The Electro-Holographic Communication System

This section deals with the concepts of information coding in a communication system. The communication system to be discussed - the electro-holographic display of 3-D images - is one that converts 3-D digital data (the 3-D scene description) into holographic images. The information is converted from one format to another along the way: hogel vectors, hogels, diffracted light, etc. The amount of information used to represent each of these formats (the required channel capacity or bandwidth) changes, though the amount of useful information never exceeds the amount contained in the original 3-D data source. One of the primary purposes for the development of diffraction-specific computation is to minimize the bandwidth required to represent the holographic information. Only for the purpose of actually diffracting light must the encoded fringe be converted to the less efficient, higher bandwidth fringe.

A typical communication system is diagramed in the succeeding figure. Data begins at the source, and the goal is to reproduce the data at the destination as accurately as possible.\textsuperscript{71} Noise is introduced in the channel. This model, adapted to apply to holovideo computation and display, becomes a holographic communication system.
In the general case of holovideo, the source is a description of the 3-D object scene to be displayed. The encoder converts this information into fringes. The display system acts as the channel, converting the fringe information into a holographic image. The channel includes the intermediate storage of the fringes in a framebuffer, and any signal processing that occurs in the analog domain on the way to the display. The channel must also include the propagation of the diffracted light to the viewing zone. The decoder is essentially the stimulation of the human visual system (HVS). The destination is the brain of the viewer. When discussing holographic encoding schemes, the encoder is the computational process that generates the encoded fringe representation, and the channel includes the decoding process, which often contributes additional noise to the overall system. An effective encoding scheme reduces the required bandwidth (capacity) of the channel. Bandwidth compression allows for more rapid trans-
mission (display) of holographic information, and allows for faster computation at the source.

5.1.1 Information Symbols

In an information content analysis, it is necessary to define a set of information-bearing symbols. Discrete samples of continuous signals are commonly used as the information-bearing symbols. The final fringe pattern that is used to diffract light in the display system is measured in samples. In the current MIT holovideo display, the number of 8-bit samples is 36 M per fringe pattern. It is possible to compute a large number of different 36-MB fringe patterns to generate the same holographic image. For example, because the acuity of the HVS is limited, every point or element in an image can be moved by an imperceptible amount before computing the fringe pattern using traditional interference-based methods. This can result in thousands of different fringe patterns that produce essentially the same image. Clearly, the 36 M samples are not being put to full use in an information bearing sense: these samples are dedicated to producing microscopic imperceptible differences in images. If an encoded format is to represent this fringe pattern with a reduced number of symbols, then most of these 36 M samples must be culled from the fringe. This observation, however, gives no clue as to how bandwidth reduction can be performed. This is one of the reasons that until now no research has been done to develop a fringe description that allows for the unique and therefore most information-efficient specification system.

Diffraction-specific computation uses a sampled spectrum to encode holographic information. The encoding formats developed from diffraction-specific computation are formulated by specifying the diffractive duty of each hogel only to a degree that matches the requirements of a typical human visual system.

5.1.2 Information Entropy

The concept of entropy is used as a measure of efficiency (versus redundancy) in an encoding scheme. The entropy of the occurrence of a symbol is essentially a measure
of its uncertainty and is a function of its probability \( p \) of occurring. Simply stated, the entropy of a symbol is the useful information conveyed by the occurrence of a particular symbol. Each symbol has an entropy of

\[
\text{entropy} = -p \log_2 p.
\]

(14)

The higher the entropy, the more information "valuable" a symbol is. (For practical systems, \( 0 \leq p \leq 0.5 \).) An important axiom in information theory is that when multiple symbols (different fringes) exist containing the same information (the same hogel vector component), the entropy in a symbol transmission underutilizes the channel capacity. To most efficiently use bandwidth (the number of samples), each symbol must convey unique information.

Consider hogel vectors. There are a large number of fringe patterns that diffract light from a particular hogel to a particular part of the imaged scene. The actual usable information in any one of these possible fringes is the same, yet only a single hogel vector component is required to convey the information that light is to diffract is this manner. The components of hogel vectors are a more efficient set of coding symbol, and have an entropy that is roughly ten times higher than the individual samples in a traditionally computed fringe pattern.

Recall that hogel vectors, in representing the diffractive duties of a hogel, are related directly to the spectrum of a hogel. Consider the number of different spatial frequencies in a hogel. For \( N_h \) samples, this number is \( N_h \). Although the spectrum has \( N_h \) samples for the magnitude and \( N_h \) for the phase, the real-valued fringe (physically, a real-valued intensity) has a spectrum with even conjugate symmetry (i.e., \( S(f) = S^*(-f) \)). Therefore, the \( N_h \) samples describing spectral phase carry no useful information from the point of view of the holographic system. The useful information content of a set of spatial frequencies depends on the number of spatial frequencies that give rise to distinguishably different viewer stimuli. For example, a hogel containing \( N_h = 1K \) discrete spatial frequencies can diffract light in \( 1K \) different directions. However, the viewer is capable of resolving only a fraction of these. In the development of the diffraction-spe-
cific method of computation (see Section Chapter 4) it is remarked that the compression ratio achievable in HPO holographic fringes should be ten or more without destroying image quality. Therefore, to reduce bandwidth, efficient holographic encoding schemes must discretize the spectrum in larger steps.

The focus of the remainder of this dissertation is on the two encoding schemes developed on top of diffraction-specific computation. The first - called “hogel-vector encoding” - is based on undersampling the spectrum of hogels as represented by hogel vectors. The second - called “fringelet encoding” - is also based on undersampling the spectrum, but uses an encoding format that is a step closer to an actual fringe representation but using only a fraction of the number of samples.

5.2 Description of Hogel-Vector Encoding

The first type of holographic encoding is called “hogel-vector encoding” because it essentially undersamples hogel vectors to reduce information content. The array of hogel vectors is treated as the encoded fringe format. To achieve bandwidth compression, the hogel spectrum is discretized in larger steps. For larger spectral sampling, fewer samples are required, increasing the compression ratio (CR). The decoding step in hogel-vector decompression is the process of superposing basis fringes weighted by the appropriate hogel-vector components to produce a decoded hogel (fringe). But because hogel-vector encoding undersamples the spectrum, it must use a special set of basis fringes. For a higher CR each basis fringe must represent a larger portion of the hogel spectrum. Each basis fringe must be tailored to cover a proportionally larger region of the spectrum. The decoded hogels each contain information about the entire spectrum. The difference between these hogels and the ones generated without compression (i.e., minimum spectral sampling) lies in the specificity and uncertainty in the final spectrum.
In hogel-vector encoding, a more coarsely sampled spectrum represented by a hogel vector with fewer discrete components contains a less specific description of the desired diffraction. Since a single component of a hogel vector includes all contributions within its enlarged region of the spectrum, information is lost: it is uncertain from whence in the spectral range a unit of energy in a particular vector component arose. However, from the point of view of the human visual system, the fully sampled hogel-vector array contains redundant information about the hogel spectra. Therefore, hogel-vector encoding eliminates redundancy in holographic fringes and injects a tolerable amount of ambiguity.

Stated in terms of the holographic communication system, the entropy contained in each information-bearing symbol (hogel-vector component) is increased. For example, a hogel vector that is not undersampled contains \( N_h \) components. Generally, each is equally like to occur, making each symbol have a probability

\[
p = \frac{1}{N_h}
\]

of occurring and an entropy of
\[
\text{entropy} = -\frac{1}{N_h} \log_2 \left( \frac{1}{N_h} \right)
\]

(16)

For \(N_h=512\), the entropy per symbol is \((1/512)\times9 = 0.0176\). If the spectrum is under-sampled by a factor of \(CR=8\), each symbol has a probability and entropy of

\[
p = \frac{CR}{N_h}
\]

(17)

\[
\text{entropy} = -\frac{CR}{N_h} \log_2 \left( \frac{CR}{N_h} \right).
\]

(18)

For \(N_h=512\), the entropy per symbol in this compressed case is \((8/512)\times6 = 0.0938\).

The channel of the holographic communication system is used over 5 times more efficiently in terms of usable information conveyed per symbol.

Hogel-vector encoding is somewhat similar to certain 2-D image encoding schemes that divide the 2-D image into blocks. In particular, hogel-vector encoding is similar to discrete cosine transform (DCT) encoding\(^{67}\). The important difference is that hogel-vector decoding does not use constant frequency sinusoids as basis functions. The basis fringes used to perform the hogel-vector decoding must utilize the whole spectrum available to the hogel. If the decoded fringe has gaps in its spectrum (as in the following illustration) the dropouts severely degrade the image, producing a picket fence of drop-outs across the viewing zone, giving the image the appearance of being behind bars. Gaps in the spectrum do not fulfill the second step in sampling theory, which requires that an appropriate low-passing be performed on the samples to recover the continuous signal. In hogel-vector encoding, the decoding step performs the proper low-pass filtering of the sampled spectrum by convolving each component with the spectrum (rectangular for purposes of illustration) of the corresponding basis fringe. A properly decoded fringe has a continuous spectrum (as in the following illustration), and the desired spectrum is reproduced with only the added ambiguity as a possible artifact.
5.3 Image Generation

The computation steps in hogel-vector encoding are the same as the hogel-vector generation described for (uncoded) diffraction-specific computation. The two-step computation was implemented on the Oryx/Cheops computation platform. The hogel-vector array was generated on the Onyx workstation. The hogel-vector array was downloaded to the Cheops P2 card, where it was then decoded using the Splotch Engine. The decoded 36-MB fringe pattern was subsequently loaded into the VRAM of the Cheops output cards. (Due to concern for reliability, for some experiments the hogel-vector decoding was performed on the Onyx, and the decoded fringes were downloaded directly to the Cheops output cards. The computed fringes were identical, byte-for-byte, to those computed using the Splotch engine.) For a CR of 1 (no encoding), the hogel-vector array comprises 36 MB. For larger values of compression ratio, this number is proportionally smaller: a CR=16 gives a 2.25-MB hogel-vector array.

Images generated using hogel-vector encoding showed an increase in imaged point spread as a function of compression ratio (CR). In many cases, this point spread was not perceivable. Hogel-vector encoding also added a noticeable speckle-like appearance to the image. The figure on page 86 shows results using hogel-vector encoding. The 3-D holographic image of a VW Beetle* car was generated from a polygon data-

* Historical note: This VW Beetle database was originally measured by hand by Ivan Sutherland et al. [11]
Image of a VW Beetle

Top: A digitally photographed picture of a 3-D image computed using diffraction-specific computation.

Bottom: The same image, computed using hogel-vector encoding, CR=16, N_{i}=1024. The discrete image points that form the image are blurred, and a slightly speckle-like appearance is added to the image.
base comprising 1079 polygons. The image was converted into over 10,000 discrete elements in the image volume using a simple lighting model.

To analyze the effect of hogel-vector encoding on point spread, a series of experiments were performed using a single imaged point. To make the blurring effect as pronounced as possible, the point was imaged to $z=80$ mm, the maximum usable depth of the MIT display. The point was also imaged to other depths, such as $z=40$ mm, $z=20$ mm, etc. In each case, the imaged point was generated with a fringe pattern (a single hololine) computed using hogel-vector encoding. Compression ratios ranged from CR=1 to CR=64, in powers of two. Typical hogel widths used and their corresponding number of samples per hogel, are listed in the following table. (Fringe sampling pitch was $1.7 \times 10^3$ samples/mm.)

<table>
<thead>
<tr>
<th>$w_h$</th>
<th>$N_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.150 mm</td>
<td>256</td>
</tr>
<tr>
<td>0.300 mm</td>
<td>512</td>
</tr>
<tr>
<td>0.600 mm</td>
<td>1024</td>
</tr>
<tr>
<td>1.200 mm</td>
<td>2048</td>
</tr>
</tbody>
</table>

To capture the imaged spot's cross-section, the small 768x494 CCD array was placed in the beam path within the image volume at the location of an imaged point. The detected cross-section was digitized and stored. A measure of effective point spread was calculated for each using the half-energy convention. Many of these cross-sections and point-spread data are gathered on the succeeding pages.

Image resolution is limited by the blur added using hogel-vector encoding. Point spread is a function of image depth and of encoding parameters, as shown in the illustrations on pages 88-90. The succeeding subsection discusses the point spread caused by hogel-vector encoding.
Hogel-Vector Encoding:

Point imaged at z=80 mm, \( w_h = 0.300 \text{ mm} \) (\( N_h = 512 \))

![Unencoded Image with Eff. width: 0.144 mm]

![CR: 1 Image with Eff. width: 0.288 mm]

![CR: 4 Image with Eff. width: 0.304 mm]

![CR: 8 Image with Eff. width: 0.400 mm]

![CR: 16 Image with Eff. width: 0.672 mm]

![CR: 32 Image with Eff. width: 1.152 mm]

![CR: 64 Image with Eff. width: 2.048 mm]

This figure shows a series of cross-sections of a point imaged at z=80 mm for a range of compression ratios. The point blurs (horizontally) as CR increases. For CR of 8 or lower, the point is still relatively sharp. For compression of CR=16, the point begins to blur to a width that is easily seen by a human viewer. However, in many cases this blur is an acceptable trade-off for increased computation speed. Speed of computation is directly proportional to CR. The CR=16 point, though blurred to 0.672 mm (less than 5 times the unencoded point width) required 1/16 the computation time. Hogel width for all of these images was \( w_h = 0.300 \text{ mm} \), or \( N_h = 512 \) samples.
Hogel-Vector Encoding:
Point imaged at z=80 mm, \( w_h=0.600 \) mm (\( N_h=1024 \))

This figure shows a series of cross-sections of a point imaged at \( z=80 \) mm. This is similar to the previous figure, except that the hogel width in this case was \( w_h=0.600 \) mm, or \( N_h=1024 \) samples (twice the previous figure). This figure illustrates that setting \( w_h=0.600 \) blurs the point more for low compression ratios. However, for \( CR=16 \) and higher the point was less blurred. Clearly, for a \( CR=16 \), the choice of \( w_h=0.600 \) mm is better than a hogel width half this size. Although point spread of roughly 0.5 mm limits image resolution to less than the acuity of the HVS, recall that this is the worst-case image. Points nearer to the holoplane (i.e., \( |z|<80 \) mm) have smaller point spread.
This figure shows a series of point images focused at z=40 mm. To further illustrate the effect of choosing the correct hogel width, the point is shown for hogel widths of 0.300 mm and 0.600 mm over the range of compression ratios. The measured effective widths (in mm) are listed in the lower right corner of each cross-section. Notice that the point spread is generally less at z=40 mm than for points at z=80 mm. Although the two values of $w_h$ are more similar than in the case of the z=80 mm point, the choice of $w_h=0.600$ mm gives less point spread at high values of CR.
5.4 Discussion of Point Spread

The blur added when using hogel-vector encoding is the result of several processes:

- spectral sampling blur due to more coarsely sampling hogel spectra;
- aperture effects;
- aberrations in the display;
- quantization and other noise.

The total increase in point spread is equivalent to a reduction in image resolution.

Contribution to blur from spectral sampling is a function of the spectral sampling width,

\[ \Delta_f = p BW \frac{CR}{N_h} \]  \hspace{1cm} (19)

and is approximated by the following expression:

\[ \text{blur}_{\text{spectr.sampl.}} = z \lambda p BW \frac{CR}{N_h} \]  \hspace{1cm} (20)

where BW is the total spectral bandwidth (in cycles/samples), \( z \) is the distance (in mm) of the point from the hologram plane and \( p \) is the fringe sampling pitch in samples/mm. The expression \( N_h/CR \) the number of components in a hogel vector, and the expression \( CR/N_h \) is the fraction of the spectrum represented by each hogel vector component. As \( CR \) increases or as \( N_h \) decreases, spectral sampling increases, and each component carries information about a wider region of the hogel spectrum. Light is diffracted in a larger range of angles, limiting the achievable image resolution. Here, then, is the trade-off between bandwidth and image resolution. System bandwidth is reduced proportionally as the number of symbols per hogel vector decreases, but the trade-off is the decrease in image resolution.
Aperture effects are simply the results of the finite extent of a hogel (i.e., spatial sampling). For a beam of light diffracted from an aperture of width \( w_h \), the width of the beam at a depth of \( z \) (mm) is

\[
\text{blur}_{\text{aperture}} = \left( w_h^2 + \left( \frac{z \lambda}{\pi w_h} \right)^2 \right)^{1/2}.
\]  

(21)

Near the holoplane, the aperture effect is approximately constant, making the minimum spot size equal to the hogel width \( w_h \). For larger values of \( z/w_h \), spreading due to diffraction by the aperture becomes significant, adding to blur for deep points.

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. Combining, the model for point spread (minimum spot size) becomes

\[
\text{blur}_{\text{total}} = \left( w_h^2 + \left( \frac{z \lambda}{\pi w_h} \right)^2 + (z \lambda \rho BW \frac{CR}{N_h})^2 + \text{blur}_{\text{display}}^2 \right)^{1/2}.
\]  

(22)

Additional contributions to point spread are for now neglected. Nevertheless, the following subsection demonstrates that this point-spread model matches the experimentally measured image point spread.

### 5.4.1 Comparison of Theory and Experiment

The figures on pages 93-94 compare the point-spread model (Equation 22) to the values measured from points displayed on the MIT display.
Measured and Theoretical Point Spread vs. Compression Ratio:
Point at z=80 mm

The top graph is a plot of the measured point width versus compression ratio for a point imaged at z=80 mm. The bottom graph is a plot of the values derived from the theoretical model for point spread. Each plotted line is a different hogel width.
**Measured and Theoretical Point Spread vs. Compression Ratio:**

**Point at z=40 mm**

The top graph is a plot of the measured point width versus compression ratio for a point imaged at z=40 mm. The bottom graph is a plot of the values derived from the theoretical model for point spread. Each plotted line is a different hogel width.
Qualitatively, the point spread model fits very well to the measured data. In absolute terms, there appears to be an additional source of blur for some values of CR. To illustrate this deviation more clearly, the following graph compares the measured and theoretical point spread values for a point imaged at \( z=80 \) mm and a hogel width of \( w_r=0.600 \) mm (\( N_h=1024 \)).

![Graph showing point spread vs. compression ratio](image)

For small values of CR, this additional blur is most likely the result of quantization noise. These fringes resulted from the superposition of a large number of basis fringes. For \( N_h=1024 \) and \( CR=1 \), the number of basis functions to be accumulated is 1024. For \( CR=4 \), this number is still 256. Since the final fringe pattern was quantized to 8 bits (256 levels), the superposition of over 128 basis fringes means that each is represented by only 2 levels. For an essentially binary waveform, the signal-to-noise ratio (SNR) is only 20:1. These experiments were performed for worst-case conditions. In a typical
image, however, roughly 1/3 or the hogel vector components are zero or nearly zero, making better use of the 8-bit dynamic range of the display system.

Another likely source of the additional blur is an additional aperture effect caused by the AOM of the holovideo display. As a hogel traverses the aperture of the AOM in the form of an acoustic wave, it is clipped at the beginning an end of its path. This clipping adds to the aperture blur, and is more prominent for larger hogel widths. AOM clipping is most likely the cause of the additional blur found in hogel widths of $N_h=1024$ and $N_h=2048$ in the measured point-spread data graphed on pages 93-94.

5.4.2 Empirical Selection of System Parameters

The optimal parameters can be selected for a given application. If CR=16 is required, then the model of point spread can be used to select the proper hogel width. A small hogel width causes significant blur for large $z$ values due to spectral blur. A large hogel width keeps spectral blur at bay, but minimum spot size is limited to the hogel width. The following figure illustrates the variation in spot size as a function of hogel width. For deep points, a hogel width of 0.600 mm provides a reasonable spot size.
The following graph is a subset of the measured effective point width for a point imaged at depths of \(z=20\) mm, \(z=40\) mm, and \(z=80\) mm. The hogel width that ensures reasonable spot size for any depth is \(w_h=0.600\) mm, i.e., \(N_h=1024\). One additional feature is that for the parameters of this display, \(w_h=0.600\) mm has a roughly constant spot size for a wide range of depths. Note in this graph that the worst case of the 80-mm deep point can be unacceptably blurred. Depth is the primary struggle when compressing bandwidth. It is relatively easy to achieve high values of CR and good image fidelity when all points are near the holoplane. Though this may seem like an obvious statement, it is good to see that all of this work to compress holographic bandwidth (as opposed to 2-D image bandwidth) is actually necessary.
\[ \delta^2 = Z\lambda \left[ \sqrt{c} + \frac{1}{\pi^2 \sqrt{c}} + \frac{BW \cdot CR^2}{\sqrt{c}} \right] \]  

(23)

\[ c \equiv \frac{1}{\pi^2} + (BW \cdot CR)^2 \]  

(24)

The expression simplifies under the assumption that the compression ratio (CR) is somewhat greater than one:

\[ BW \cdot CR \gg \frac{1}{\pi} \]  

(25)

\[ \Rightarrow \quad \delta = \frac{Z}{N} \left( 2 \sqrt{2} \sin \frac{\Theta}{2} \right) \]  

(26)

given that the hogel width has been chosen as

\[ w_h = \delta \sqrt{BW} \]  

(27)

(This assumption is equivalent to setting equal the contributions to point spread from the aperture and spectral sampling effects. Also, for this analysis, other sources of blur are treated as negligible.) Note that the parameters of pitch \((p)\) and wavelength \((\lambda)\) have been substituted by the expression containing the size of the viewing zone \((\Theta)\). The parameter \(N=N_p/\text{CR}\) is simply a measure of bandwidth in symbols per hogel.

Equation 26 is an analytical expression relating the most important parameters of a holovideo system, namely the image resolution, the image depth, the bandwidth \((N)\) and the size of the viewing zone \((\Theta)\). To design a holovideo system, an image resolution is first chosen, setting the parameter \(\delta\). Next, the hogel width is fixed by Equation 27. Finally, minimum bandwidth \(N\) and maximum depth \(Z\) are adjusted for optimal performance using Equation 26. This analysis shows quite clearly the trade-off between image depth and required bandwidth. These concepts are explored further in Section 7.3 “Engineering Trade-Off: Bandwidth, Depth, Resolution” on page 134.
5.5 Speed

Using hogel-vector encoding, total computation time consists of the initial direct-encoding step to produce the hogel-vector array on the Onyx, the hogel-vector decoding step done with the Cheops Splotch Engine, and the time to transfer the encoded fringes (the hogel-vector array) to the Cheops system. The first step - generation of the hogel-vector array - was very fast. For simple wireframe objects, typical times were less than 1.0 s for a CR of 16. The downloading of the hogel-vector array over the SCSI link is slow, even with a 2.25-\(\text{MB}\) hogel-vector array (CR=16), the data transfer required approximately 2.0 s. This time should reduce to a negligible amount with the future introduction of a high-speed data link.

The decoding step required the greatest amount of computing time. Each fringe sample in the resulting hogel requires \(N_h/\text{CR}\) MACs. For compression of 16 times (i.e., CR=16), the decoding step requires that for each of 144x512 hogels, 32 basis fringes of length \(N_h=512\) bytes be multiplied and accumulated to produce a 36-\(\text{MB}\) fringe. This is approximately 1.2 GMACS, i.e., over 1 billion multiplies and 1 billion adds.

When implemented on a single Splotch Engine, hogel-vector decoding time was 20 s. (These timings were worst case, measured using a fully non-zero hogel-vector array. Typical test images were closer to 9 s owing to their more sparse hogel vectors.) Compared to the (unencoded) diffraction-specific computation speed in the previous chapter, hogel-vector encoding was a factor of 16.0 times faster. This was due to the reduction by a factor of 1/CR in the number of time-consuming MAC calculations required for each fringe sample. Since the number of MACs decreases with higher compression ratios, the speed of hogel-vector decoding increases linearly with CR.

Faster speeds can be achieved by sacrificing image quality for a reduction in information content.

As previously noted (Section 4.7), the Splotch Engine should be able to achieve a factor of two speed increase once it has been reconfigured to handle hogel-vector decoding. Also, because the Cheops P2 board can contain three Splotch Engines, decoding
time can in the future be reduced by two thirds. This potentially brings total computation time down to about 5 s total, including all transfer times, using hogel-vector encoding implemented on the Onyx/Cheops/Splotch platform. For comparison, a Connection Machine Model 2 with 16 Kprocessors required 21.9 s for the computation task in the previous paragraph (not including the transfer time from the CM2 to Cheops). In other words, for hogel-vector decoding, a single Cheops P2/Splotch combination provides computing power equivalent of a CM2 - all on one convenient and portable board.

The following listing summarizes the timings for hogel-vector encoding and decoding. The first two numbers indicate times for direct-encoding and for decoding. These times sum to total computing time (excluding transfer time).

- Onyx → Cheops Splotch Engine: 1 s + 20 s = 21 s
- CM2: 1 s + 22 s = 23 s
- SCSI transfer time: add 2 s

### 5.6 Conclusion

Hogel vectors are a natural choice for a fringe encoding format. Recall the three important features characterize an information-efficient diffraction-specific hogel description: (1) the computation of a hogel can proceed using nothing more than a hogel vector; (2) no two hogels computed from different hogel vectors give rise to the same viewer stimulus; (3) no two different hogel vectors give rise to the same viewer stimulus. Hogel vectors increase the amount of entropy per symbol.

Redundancy is inherent in physically useful fringes. The fringe pattern, encoded as the hogel-vector array, contains reduced redundancy and increased symbol entropy. Thus, the bandwidth is reduced until the final computation step, decoding.
Speed is still a problem. Although speed has increased by over an order of magnitude, the fastest typical time of 9 s is still beyond the reach of interactive computing. The decoding step accounts for most of the total computing time. In the next chapter, a second holographic encoding scheme - "fringelet encoding" - is developed specifically to increase the simplicity and speed of decoding.
Chapter 6

Fringelet Encoding

The major drawback to using hogel-vector encoding is that the decoding step requires an enormous amount of computing power. A second type of encoding scheme - "fringelet encoding" - is designed to decrease decoding time by using an encoded format that more closely resembles the final fringe pattern. In fringelet encoding, the encoded format for a given hogel is called a fringelet because it looks like a small fringe, with a spectrum that is closely related to the desired hogel spectrum. An array of fringelets is computed directly from the set of hogel vectors. The resulting fringelet array is subsequently decoded into usable fringes using an aperiodic replication scheme that is extremely fast. The decoding step is fast because a fringelet contains the sample values that are to appear in a hogel. Decoding involves no arithmetic, only sample replication and reordering.

Overview of Fringelet Encoding and Decoding

3-D Description → Hogel vector generation → Hogel Vectors → Linear Superposition $\Sigma$ → Special Basis Fringes → Fringelet Array → Decoding → Hogels (Fringes)
Essentially, fringelet encoding and decoding (see overview, above) substitutes for the conversion of hogel vectors to hogels. Each hogel vector is converted into a fringelet - an encoded description of the hogel, but one that occupies only a fraction of the bandwidth (i.e., the number of samples). Fringelets are converted to hogels in the fringelet decoding step. The following sections describe the encoding and decoding steps, including a description of the special spectral qualities of fringelets. Also included in this chapter are implementation details, pictures of fringelet-encoded images, and an analysis of fringelet performance.

6.1 Fringelet Generation

The essence of fringelet encoding is to construct a fringelet that possesses the desired hogel spectrum - or a spectrum that is closely related - using only $N_h/CR$ samples. The driving concept was to make a fringelet look more like a hogel fringe so that decoding would be simpler and more rapid. The width of each fringelet is only a fraction of the width of the hogel for which it is encoded. Their relative widths determine the compression ratio (CR). For a hogel of width $N_h=1024$ samples, the fringelet width is 64 samples for a CR=16.

The fringelet for a particular hogel is computed from a hogel vector but using a special set of basis fringes. Each basis fringe has a width of $N_h/CR$ samples, and is specially computed to contain spectral energy in a specific region. (See Appendix C.) Fringelet generation (direct-encoding) converts a hogel vector into a fringelet using the same type of superposition used to convert hogel vectors into hogels. The resulting fringelet has a spectrum with the desired amount of energy in each discretized region of the spectrum, centered at intervals of $\Delta_f=p\text{BW}(CR/N_h)$, where BW is the spectral bandwidth used (up to 0.5 cycles/sample). The important difference is that the fringelet has only $N_h/CR$ samples. But a signal with only $N_h/CR$ samples can have only $N_h/CR$ distinguishable spatial frequencies. It is important to use basis fringes that leave some empty spaces in the spectral regions in between non-zero regions. This allows for energy compaction, i.e., bandwidth compression by eliminating useless information.
symbols. The fringelet was originally conceived as a “sparse-spectrum” hogel, i.e., a truncated piece of a hogel generated using basis fringes that would give the hogel a sparse spectrum. If such a hogel is truncated to only $N_p/CR$ samples, the separate contributions in each region of the spectrum broaden but remain independent for reasonable values of CR.

The fringelet spectrum is related to the desired spectrum as shown in the figure on page 107. A hogel vector represents a piece-wise continuous spectrum, with the amount of power in each region of the spectrum being proportional to the components of the computed hogel vector. The special set of basis functions are generated (using simulated annealing) to occupy a more sparse spectrum. The gaps in the spectrum allow for bandwidth compression without loss of spectral information. The spectrum of the fringelet still contains the same relative powers in each region of the spectrum.

6.2 Fringelet Decoding

The goal of fringelet decoding is to use a fringelet of width $N_p/CR$ to create a hogel of width $N_h$ - a usable fringe - that possesses the desired spectrum. This desired spectrum is encoded in the fringelet, the main difference being that some space exists between adjacent spectral regions. The decoding process must broaden the spectrum of each of the $N_p/CR$ regions to produce a continuous spectrum. This is equivalent to the low-passing process required by the sampling theory to recover the sampled spectrum. The display process, including the diffraction of light, contributes to this desired spectral blur. The decoding process must also expand the $N_p/CR$-sample fringelet into a usable $N_h$-sample hogel. Fringelet decoding must perform the spectral broadening and the expansion without using complicated and time-consuming mathematical processing. Fringelet decoding is engineered to solve both of these problems.

The process of fringelet decoding is illustrated on page 108. Replicas of the fringelet are truncated and translated by convolution with a series of stochastically spaced impulses. This analytical model looks complicated, but because the impulses have
unity amplitude, there is no need for multiplication. In fact, if the replicated truncated fringelets do not overlap, then there is no need for any mathematical calculations at all! The only operations are the copying of fringelet values to hogel sample locations.
Fringelet and Hogel Spectra

This figure illustrates the spectra involved in fringelet decoding. At left is the desired spectrum. This spectrum was encoded in the fringelet, and subsequently decoded to produce a hogel. (The “sparse-spectrum hogel” is not part of the encoding-decoding progression, but is included for illustration.) The decoded spectrum contains the correct amount of spectral energy in each region. Notice that the decoded spectrum is continuous, whereas the desired spectrum (the information contained in the hogel vector used to compute the fringelet) is piecewise constant. A continuous spectrum is more desirable, since the diffracted wavefront will not have jumps that can lead to image artifacts. Thus, fringelet decoding actually improves the quality of the holographic information by spreading the spectral energy to produce a continuous spectrum that more accurately diffracts light to generate the intended wavefront.
Fringelet Decoding: Block Schematic

As shown in this figure, fringelet decoding is performed by convolving a truncated replica of the fringelet with a series of stochastically spaced impulses of unity height. Randomly truncating (windowing) a fringelet has the desirable effect of spreading the spectrum in each of the regions. Convolution with a series of impulses has the desirable effect of expanding the width of the fringelet to match the width of the hogel.
The most important ingredient in fringelet decoding is the statistics of the truncations and replications. The truncations are relatively simple, giving rise to the correct spectral broadening for any random set of truncations. The mean value of the sequence of truncation widths determines the amount of spectral broadening. For simplicity, the truncation widths are set to match the width between each impulse in the convolution impulse sequence. This means that each replicated fringelet is truncated by precisely the same amount by which it is to be translated. This simplifies the decoding algorithm.

The statistical properties of the convolution impulse sequence are crucial to maintaining image fidelity. The impulses must have a spectrum that is as flat as possible. Recall that the convolution of the impulse sequence with the fringelet in the spatial domain acts to multiply the spectrum of the fringelet by the spectrum of impulse sequence. The spectrum of the impulses must be uniform to recover the desired spectrum. If the spectrum of the impulses has gaps or sharp peaks, then the spectrum of the hogel decoded from the fringelet will have noise and lead to image artifacts. To create an impulse sequence that satisfies all of these constraints, the simulated annealing algorithm (described in Appendix C) was adapted to this purpose. The constraints on the spatial and spectral characteristics of the impulse sequence are here illustrated and summarized:

- Spatial amplitude: unity amplitude impulses (zero in between).
• Spatial amplitude: The impulses must occur at random (not periodic) intervals of $s_i$.

• Spatial phase: constant (zero).

• Spectral amplitude: impulse sequence must have a uniform spectrum across the spectral region of interest, and should be zero elsewhere.

• Spectral phase: unconstrained.

The starting point to the simulated annealing algorithm was a randomly distributed impulse with an average spacing that produced the desired spectral spreading. For each iteration, an impulse was chosen from the sequence at random and moved either forward or backward. This modification was either kept or rejected based on the probability function described in Appendix C. The desired impulse sequence converged after about 10000 iterations. Although this was a slow process (over 15 minutes), it only needed to be performed once for a given set of parameters. A separate replication sequence was computed in this way for all useful combinations of hogel width $N_h$ and compression ratio CR. The separations $s_i$ between the impulses is the amount by which the fringelet will be truncated and translated to decode it into a hogel. Once computed, this replication sequence was built into the decoding algorithm.

To help to visualize the actual fringelet decoding process, consider the example of a hogel width of $N_h=1024$, a compression ratio of CR=16, and a fringelet width of $N_h/CR=64$. To begin fringelet decoding, the 64-byte fringelet was copied into the first 64 hogel samples. (The replication sequence $s_i$ was also constrained to make the first impulse spacing be equal to $N_h/CR$.) Next, the fringelet is replicated and truncated by $s_1$, the next value in the replication sequence. This truncated fringelet is then translated to fill hogel samples 64 through $64+s_1-1$. Next the fringelet is replicated and truncated by $s_2$ and translated to fill hogel samples $64+s_1$ to $64+s_1+s_2-1$. This recursive aperiodic
truncation and translation continues until the full hogel width of \( N_h = 1024 \) is filled, as illustrated in the figure below. Each fringelet is subsequently decoded in this same way.

![Fringelet Encoding Diagram](image)

64 samples

\( N_h = 1024 \)

Decoded Hogel Fringe

1024 samples

6.3 Implementation

Fringelet encoding and decoding were performed using the same computational platform as hogel-vector encoding. The first computational step was to generate the fringelet array on the Onyx workstation using the precomputed set of special basis fringes. This array of fringelets, with size \((36 \text{ MB})/\text{CR}\), was downloaded over the SCSI link to the Cheops P2 board. The i960 microprocessor on the P2 performed the fringelet decoding. The hogel array decoded from the fringelet array was then transferred to the Cheops output cards for display.

In the fringelet decoding step, the aperiodic truncation and translated replica of the fringelet is equivalent to taking each sample of the fringelet and copying it to several hogel sample locations. Each fringelet sample maps to a set of hogel samples, and each mapping is mutually exclusive and fully exhaustive. Therefore, this entire mapping process was stored in an indirection table on the P2. This indirection table was \( N_h \) entries wide, and each location contained the appropriate fringelet sample number.

Let the indirection table be \( \text{INDtable}[i] \), where \( i \) ranges from 0 to \( N_h - 1 \). The following pseudocode describes the fringelet decoding algorithm using the indirection table.
For each Fringelet: {
    For each Hogel sample: {
        Hogel[i] = Fringelet( INDtable[i] )
    }
    Load Hogel into appropriate output card location.
}

The only operations are (1) fetching the value from the INDtable, (2) using that value to fetch the appropriate fringelet sample, and (3) copying the fringelet sample value into the indexed hogel sample location. Each decoded hogel value is decoded using no math and only a few memory operations. As discussed later, the speed of fringelet decoding is very fast due to this extreme simplicity.

The implementation of fringelet encoding and decoding included two esoteric details. First, the lowest spatial frequency contained in any fringelet was nonzero, limited to allow the fringelet to contain at least one period of this frequency. (For typical fringelet parameters, only 3 per cent of the available bandwidth was wasted as a result.) Second, each fringelet basis fringe was constrained such that its average value was equal to the average between the minimum and maximum values. This eliminated a significant source of noise generated in the decoding step.

### 6.4 Image Generation

To illustrate the effect of fringelet encoding on holographic image quality, several images were digitally photographed, both close-up and full image. Several examples were documented, using images at different depths, and using different values of hogel width $N_h$ and compression ratio $CR$. The digitally photographed images were gathered together and shown on pages 113-116. The profile of each focused point was also acquired by sampling a cross-section of the central region of each data set. The illustrations on page 116 show an example of a cluster of imaged points, photographed in the same way as the single points, showing the interaction between neighboring points on a hololine.
Rocket Engine Fuel Intake

Each of these illustrations is a photograph of a 3-D image of the fuel intake system of a Space Shuttle rocket engine. The unencoded image (top) shows the discrete points of light that compose the image. This sharpness is lost when the image is fringelet compressed with a CR=16 (bottom). Nevertheless, the blur added to the image does not severely degrade image fidelity.
Fringelet Encoding:

Point imaged at $z=80$ mm, $w_h=0.600$ mm ($N_h=1024$)

This figure shows a series of cross-sections of a point imaged at $z=80$ mm for a range of compression ratios. This is the worst-case situation: 80 mm is the maximum depth used in this display. The point blurs (horizontally) as CR increases. For CR of 8 or lower, the point is still relatively sharp. For compression of CR=16, the point begins to blur to a width that is easily seen by the human viewer. Hologram width for all of these images was $w_h=0.600$ mm, or $N_h=1024$ samples.
This figure shows a series of point images focused at \( z=40 \) mm. To illustrate the effect on point spread of different hogel widths, the point is shown for hogel widths of 0.300 mm and 0.600 mm over the range of compression ratios. The measured effective widths (in mm) are listed in the lower right corner of each cross-section. Although the two values of \( w_h \) are more similar than in the case of the \( z=80 \) mm point, again the choice of \( w_h=0.600 \) mm gives less point spread at high values of CR.
Fringelet Encoded Image: Close-up

This figure shows close-ups of a full image (of a rocket engine) computed using fringelet encoding. The CCD array was placed in a region at z=40 mm containing a surface (represented by an array of imaged points) as well as regions of black (no image elements). In the unencoded image, the array of imaged points is clearly visible. For the fringelet encoded images (w_h=0.600 mm, N_h=1024), the blur of each points has the desirable effect of joining the discrete image points together to form a continuous surface.
To measure the image resolution achievable using fringelet encoding, a series of point images was acquired and their point spread calculated. For the worst-case point at z=80 mm, fringelet encoding produces point spread that is unacceptable for high values of CR. The graph at the top of page 118 plots the measured point spread for a range of compression ratios and hogel widths. For CR=16, a hogel width of $w_h=0.600$ mm ($N_h=1024$ samples) produces reasonable results, even in this worst-case image depth.

Fringelet encoding performs much better for points that lie within the more commonly used regions of the image volume. The lower graph on page 118 shows the measured point spread for a point imaged at z=40 mm. Point spread is reduced by a factor of 2 as compared to the point imaged at z=80 mm. This indicates that the primary source of the blur is spectral, which causes a spreading that increases linearly with diffracted distance. As for the worst-case image point, the best results for CR=16 were for $N_h=1024$, $w_h=0.600$ mm.
6.5 Discussion of Point Spread

Fringelet encoding adds more point spread than does hogel-vector encoding. The primary cause is spectral sampling blur and crosstalk. The number of spectral regions to be encoded in a fringelet is $N_p/CR$, the same number of samples in the fringelet. However, the spectrum of each component becomes broad for small values of $N_p/CR$. The resulting spectral cross-talk is the cause of the additional blur. Because of this crosstalk, fringelets do not allow for completely independent control over separate spectral regions. However, for reasonable values of CR, this crosstalk adds only a small amount of image noise. Notice in the graphs on page 118 that the point spread decreases dramatically as CR decreases or as hogel width ($N_h$) increases. For these larger values of $N_p/CR$, cross-talk is limited to nearest neighbors only. For smaller values, i.e., fewer spectral samples, cross-talk begins to affect broader ranges of the spectrum. This cross-talk essentially places a maximum limit on the entropy per symbol in the holographic communication system.

As was the case in hogel-vector encoding, choice of hogel width is important to encoding performance. The figure below shows the effect of different hogel widths, $w_h$. A point is imaged at $z=40$ mm and at $z=80$ mm using fringelet encoding for a range of hogel widths. The measured effective widths (in mm) are listed in the lower right corner of each. As previously noted, the selection of $w_h=0.600$ mm ($N_h=1024$) provides optimal performance. Notice also in this figure that the shape of the imaged point has some structure. These variations are caused by imperfections in the fringelet decoding process. the impulse sequence does not have a perfectly smooth spectrum. Each focused point is essentially multiplied by an envelope function that is a representation of the actual spectrum of the impulse sequence.

6.6 Speed

The use of fringelet encoding increased holographic fringe computation speed by a factor of 2.8 without requiring specialized hardware. Decoding a fringelet involves no
mathematical calculations. Using the indirection table described earlier, only memory access and byte replication is necessary. It can be performed as quickly as data can be transferred within the P2 board. Total fringe calculation time using fringelet encoding was typically 9 seconds, distributed as follows: 5 seconds for the fringelet computation (direct encoding), 2 seconds for the SCSI download, and 2 seconds for the decoding including the time to transfer to the VRAM. Transfer to the VRAM of the output cards used the Nile bus. Fringelet decoding accounted for 1.2 s and Nile transfer of the 36-MB decoded fringe pattern accounted for 0.8 s. These times are for fringelet decoding implemented on the i960 microprocessor on the Cheops P2 board. Fringelet decoding is fast even without the use of specialized hardware.

Note that the decoding time for generating 36-MB fringes was 1.2 s compared to 20 s for hogel-vector decoding implemented on special hardware (the Splotch Engine). To
more accurately compare the two decoding schemes, each was implemented on the Onyx using only standard serial C code. On the same computational platform, fringelet decoding is consistently over 100 times faster than hogel-vector decoding. The trade-off is that the first computational step - generation of the fringelets - requires more time than the generation of hogel vectors. (The fringelets are computed from the hogel-vector array, so some additional computation is always required to generate the fringelets.) Nevertheless, total computation time is down by a factor of more than 2.0 compared with hogel-vector encoding.

The following listing summarizes the timings for fringelet encoding and decoding. The first two numbers indicate times for direct-encoding and for decoding. These times sum to total computing time (excluding transfer time).

- Onyx → Cheops P2 → VRAM: \[ 5 \text{ s} + 2 \text{ s} = 7 \text{ s} \]
- SCSI transfer time: add 2 s

The goal of fringelet encoding has been achieved: the decoding algorithm is so simple that it can produce a 36-MB fringe in under two seconds - fast enough for interactive computation. A system designed to convert hogel-vectors to fringelets and fringelets to hogels can be implemented on specialized hardware to reduce speeds to interactive rate. Chapter 8 discusses the possibilities of implementing fringelet decoding in digital hardware, in the analog RF electronics, or optically in a specially designed holographic "fringelet" display system. First, however, Chapter 7 further compares and contrasts the performance of hogel-vector versus fringelet encoding, to determine which is more appropriate for a given imaging requirement.
Chapter 7

Holographic Encoding: Discussion

This chapter compares the results of hogel-vector and fringelet encoding schemes. To put these holographic encoding schemes into perspective, this chapter begins with a discussion of the trade-offs and features in encoding schemes designed for bandwidth compression. Section 7.2 describes a number of additional features that determine the usefulness of holographic encoding schemes. The final section describes a standard data compression technique for the purpose of comparison.

7.1 The Looks and Trends of Encoded Formats

The encoded version of a hogel may resemble one of two extremes: the desired hogel (i.e., fringes) or the data source (i.e., a 3-D description). For example, a list of 3-D points is essentially a representation of the fringes in a very compact format. It does not resemble a fringe pattern at all, but it does look like the original image scene to be displayed. This highly “encoded” description of the desired fringes requires a maximum amount of “decoding” time. The other obvious extreme is the fringe pattern itself represented in digital form. It is not encoded at all, but requires no decoding.

Different encoding schemes provide different degrees of compressibility and speed. There is a general trend in compressibility: encoded fringes that resemble the 3-D description are more compressed (have a higher CR), whereas encoded fringes that more closely resemble the final fringes are less compressed (have a lower CR). Computing speed is also a function of the “look” of an encoded format, i.e., its resemblance to one or the other extremes. The initial encoding time (direct generation) and the decoding time are both functions of the encoding scheme. An encoded fringe pattern that resembles the 3-D description requires little time for the initial generation of encoded fringes, but requires much time to decode. An encoded fringe that resembles
the final fringe can be decoded quickly, but requires much time for the initial generation. There is a general trend toward rapid decoding as the encoded format more closely resembles the fringes themselves. Although it is important to minimize the total computation time, it is often more important that the decoding step be quick and simple. This arrangement avoids prohibitive bandwidth bottlenecks, as is the case in the current MIT holovideo display in which the Cheops computation platform is used to perform the decoding step.

Hogel-vector encoding “looks” like both the 3-D image description and the final fringes. The hogel vector components represent 3-D image information, giving the hogel-vector array a semblance to the image. However, hogel vectors are arrayed in a regular grid, giving them at least some semblance to the final decoded hogels (fringes). As predicted by the general trend, the hogel-vector array can be quickly generated because it resembles the 3-D scene, but requires much decoding time as a result.

Fringelets are designed to resemble the final fringes. In keeping with the trend, fringelet decoding is extremely fast. A fringelet array looks nothing like the original 3-D scene description. Initial generation of the fringelet array can be slow, but not nearly as slow as hogel-vector decoding.

The two general trends in compressibility and decoding time are plotted in the following graph. The more compressed an encoded format is, the more time is required to decode it. The points for hogel-vector encoding and fringelet encoding are for typical images and optimal values of hogel width. Hogel-vector encoding generally provides an additional factor of 2 savings in bandwidth, but fringelet decoding is many times faster.
In the general trend relating compression ratio and decoding time, fringelet encoding stands out as possessing the best overall performance. The primary reason is that the number of calculations required to produce one sample of one hogel using fringelet encoding is a minimum. To go from a hogel vector to a hogel using fringelet encoding requires \( (N_h/CR)^2 \) MACs per hogel of width \( N_h \). (The decoding step, \( N_h \) byte replications, is considered negligible.) Amortized over the hogel, the calculations per fringe sample is \( N_h/CR^2 \) MACs. In contrast, hogel-vector encoding requires \( N_h^2/CR \) MACs per hogel, or \( N_h/CR \) MACs per fringe sample. Therefore, fringelet computing requires \( 1/CR \) the number of calculations per fringe sample when compared to hogel-vector encoding, and \( 1/CR^2 \) the number of calculations when compared with unencoded diffraction-specific computation. (These results are summarized in the table below.) It is difficult to imagine a holographic computing scheme that can generate \( N_h \) fringe samples in fewer than \( (N_h/CR)^2 \) calculations.

<table>
<thead>
<tr>
<th>Computation Method</th>
<th>MACs/sample</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unencoded</td>
<td>( N_h )</td>
<td>--</td>
</tr>
<tr>
<td>Hogel-Vector</td>
<td>( N_h/CR )</td>
<td>CR</td>
</tr>
<tr>
<td>Fringelet</td>
<td>( N_h/CR^2 )</td>
<td>( CR^2 )</td>
</tr>
</tbody>
</table>
The actual (worst-case) computation times are listed in the following table. These times are for the typical parameters of \( N_h=1024 \) samples (\( w_h=0.6 \) mm) and \( CR=16 \). They do not include transmission times to Cheops or to the Cheops output cards. The transfer times are approximately 40 s for the unencoded computation and 2.0 s for encoded computation. Although fringelet encoding is the fastest, recall that it is not implemented in specialized hardware. Both unencoded diffraction-specific computation and hogel-vector encoding rely upon the Cheops Splotch Engine for speed. When implemented on the same computer, speeds are commensurate with the number of operations, as listed in the previous table.

<table>
<thead>
<tr>
<th>Computation Method</th>
<th>Computation Time</th>
<th>Improvement</th>
<th>Improvement vs. Traditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unencoded</td>
<td>330 s</td>
<td>--</td>
<td>2.4 x</td>
</tr>
<tr>
<td>Hogel-Vector</td>
<td>21 s</td>
<td>16 x</td>
<td>38 x</td>
</tr>
<tr>
<td>Fringelet</td>
<td>7 s</td>
<td>47 x</td>
<td>114 x</td>
</tr>
</tbody>
</table>

- Note: SCSI transfer times add: 45 s for unencoded, 2 s for encoded.

Fringelet encoding provides a minimum of calculations per fringe sample. Although it is fast and can achieve reasonable compression ratios, it does not preserve image fidelity as well as hogel-vector encoding, mainly due to the crosstalk among spectral components. Future work in fringelet encoding will involve decreasing point spread and eliminating sources of image artifacts. Meanwhile, as discussed in the following section, there are several important features that characterize the viability of encoding schemes.

### 7.2 Features of Encoding Schemes

The primary purpose of an encoding scheme is to reduce required bandwidth. The holographic encoding schemes developed in this thesis have also achieved a secondary
goal: an increase in overall computation speed. This feature is rarely found in image encoding and data encoding schemes\textsuperscript{67}. Holographic encoding schemes based on diffraction-specific encoding increase computation speed by decreasing the required number of calculations per fringe sample. This reduction is possible only because hogel-vector encoding and fringelet encoding are direct-encoding schemes, i.e., they do not generate a fringe pattern before generating an encoded fringe.

Hogel-vector and fringelet encoding are designed for speed as well as for bandwidth compression. In general, encoding schemes are designed to incorporate features besides compression. To fully judge each encoding scheme, this section enumerates and describes these features (interoperability, extensibility, scalability, manipulability, second-order encodability, and 2-D compatibility) and discusses which holographic encoding schemes possess these features.

7.2.1 Interoperability

A useful feature of some encoding schemes is that the encoded information can be used for a number of different purposes simply by altering the decoding algorithm. This quality, interoperability\textsuperscript{68}, is found in subband coding and pyramid encoding schemes used for image bandwidth compression\textsuperscript{66,67}. In the same way, holographic encoding should allow for encoded fringes to be usable by different displays, either through alterations to the decoding algorithm or through preprocessing of the encoded fringes. The size of the viewing-zone and the fringe sampling pitch are the two main parameters that vary among holographic displays. For example, if an encoded fringe pattern is computed for a wide viewing zone, then it is useful to be able to decode to produce a fringe pattern that is viewable on a display with a smaller viewing zone. The size of the image volume is another parameter that varies among holographic display systems. (This feature is discussed below under “Extensibility.”)

A holographic fringe pattern, computed using traditional or diffraction-specific methods, can be altered to suit a different display geometry. To reduce viewing zone size, the fringe must be band-pass filtered, a computationally intensive process that requires
nearly as much time as computing a new fringe pattern. To account for a reduction in sampling pitch, a computationally intensive interpolation process is required. Clearly, fully computed fringes are not easily display-commutable.

Hogel-vector encoding exhibits display interoperability. As discussed in Section 4.4, the set of basis fringes contain the necessary information about the display geometry. To adapt to a display with a smaller viewing zone, decoding proceeds using a subset of the components in each hogel vector. Fringe sampling pitch is not an issue with a hogel-vector array. The basis fringes are specific to a given display system. These local basis fringes contain the proper sampling pitch and spectral characteristics to decode the diffraction-specific information encoded in the hogel-vector array into usable fringes. The only obvious short-coming is the inability to scale up the viewing zone. This is simply the result of only encoding a limited range of directional information. If the hogel-vector array is generated for a large (>100-degree) viewing zone, then individual displays can select only the applicable encoded information by selecting the appropriate subset of hogel-vector components.

Fringelet encoding exhibits weak display interoperability. Fringelet decoding is not well-suited to changing the size of the viewing zone or the size of the sampling pitch. The same computationally intensive processing by the fully computed fringe pattern must be used.

7.2.2 Extensibility

As holovideo displays grow in size, it is important that an encoded format be able to provide the increased information capacity. This feature is called extensibility\(^\text{68}\). Once extended, it is important to be able to maintain a back compatibility with smaller displays. Changing the size of a 2-D image is a common image-processing task. Fully computed holographic fringes cannot easily be scaled to produce larger or smaller images. It is therefore desirable that a holographic encoding scheme allow for increasing or decreasing image size in all three dimensions.
Hogel-vector encoding exhibits strong extensibility. To change lateral image size, hogel vectors can be processed before decoding. For example, to reduce image size by a factor of two in the horizontal dimension, each component in pairs of adjacent hogel vectors are averaged to produce a half-size hogel-vector array. This array can then be decoded normally. In an HPO hologram, adjacent vertical pairs can be averaged in this same way to reduce vertical size by a factor of two. Scaling in depth is automatic: downsizing horizontally by a factor of two reduces depth by the same proportion. Such extensibility is practically impossible with fully computed fringes.

Fringelet encoding also exhibits strong scalability. For example, to reduce image width by a factor of two, each fringelet is decoded to fill half of the original hogel width. This calculation-free approach can be coded into a special indirection table. As in hogel-vector encoding, the dimension of depth scales linearly with lateral scaling.

### 7.2.3 Scalability

Related closely to interoperability and extensibility is scalability - the ability of an encoded format to supply as much or as little information as is required for a particular use. For example, during interactions that necessitate negligible refresh times, a subset of the encoded information can be used to produce a "quick and dirty" image that can subsequently be replaced by the full-fidelity image generated from the entire encoded format. Holographic encoding should provide a means for selecting image resolutions arbitrarily.

Fully computed holographic fringes do not allow for scalability. For example, if a lower (lateral) resolution image was required, the transformation of the fringe would require more time than the initial computation.

Hogel-vector encoding exhibits strong scalability. For example, to quickly generate an image with half of the intended resolution, only every other hogel is decoded and sent to the display. The remaining portion is subsequently decoded and used to generate the full-fidelity image. A better approach to generating quick-and-dirty interactivity is to
subsample the hogel-vector components, equivalent to a further subsampling of the hogel spectra. A reduced-resolution image from (for example) every forth component of each hogel vector can be decoded and displayed. The remaining hogel-vector components can subsequently be decoded and added to the first-pass decoded hogels.

Fringelets also exhibit some scalability. Similar to hogel-vector encoding, an image can be decoded using every other fringelet to generate a reduced-resolution image. The remaining fringelets are subsequently decoded. However, the slightly more desirable spectral subsampling approach is not possible. Nevertheless, fringelet decoding is so fast that the quick-and-dirty approach is seldom useful.

7.2.4 Manipulability

A fringe pattern in encoded form can be used for more than just decoding. For example, it may be useful to add two such encoded fringes together to form an encoded fringe that when decoded produces the images from both of the original fringes. It is desirable for a holographic encoding scheme to allow for all of the common 2-D image processing manipulations, including adding, subtracting, attenuating (changing relative brightness), etc.

Adding together two holographic images is impossible with fringes computed using traditional interference-based computation. However, fringe addition is possible with fringes generated using the bipolar intensity method or diffraction-specific computation. However, what if it becomes necessary to subtract away rather than to add images? It is difficult or impossible to subtract part of a holographic image by manipulating its fully computed fringes.

Hogel-vector encoding exhibits strong manipulability due to the orthogonality of hogel-vector components. Adding, subtracting, and attenuating are realized simply by performing hogel-vector-component-wise additions, subtractions, and multiplications. For example, if a hogel-vector array is computed from a scene containing a car and a house, and it becomes necessary to remove the car, then a second hogel-vector array
computed from the same car is subtracted from the first hogel-vector array. The subtraction is performed for each component of each hogel vector. This process is fast, and is intuitive to persons familiar with image processing and computer graphics since it parallels the manipulation of 2-D image pixels (picture elements).

Fringelet encoding exhibit equally strong manipulability. Additions, subtractions, and multiplications are performed fringelet by fringelet, sample by sample. One stipulation is that the fringelet arrays must all be computed using the same fringelet basis fringes to preserve orthogonality. Fringelet encoding also offers calculation-free alternatives, similar to the example of scalability. For example, two fringelet arrays can be added together during decoding by randomly selecting between the two source fringelets at a particular array location.

7.2.5 Second-Order Encodability

Encoded fringes allow for further compression using existing image-compression and data-compression techniques. Additional compression is possible in most cases.

Consider an array of hogel vectors. The 3-D scene is generally composed of objects that do not rapidly vary as functions of space or viewing direction. The spatial correlation of the scene gives rise to correlations between respective components in adjacent hogel vectors. The often slow dependence on viewing direction is equivalent to a spectral correlation, causing the components within a hogel vector exhibit correlation. Therefore, hogel-vector encoded fringes can be further encoded using existing multi-dimensional encoding schemes that take advantage of this correlation\textsuperscript{67}. Hogel-vector arrays can also benefit from the interpolation and other manipulations associated with these multi-dimensional encoding schemes.

For the purposes of quantifying second-order bandwidth compressibilities, adaptive Lempel-Ziv coding (Unix compression) was applied to fully computed fringe patterns and hogel-vector-encoded fringes. Adaptive Lempel-Ziv (L-Z) coding is a lossless compression scheme, so the size of the L-Z compressed data gave a rough measure of
required bandwidth. First the 36-MB fringe pattern was computed using traditional interference-based method, and L-Z compression was used to compress the 36-MB fringe pattern into a format of about 10 MB. Clearly, traditionally computed fringes are not amenable to bandwidth compression. A 36-MB fringe pattern computed using hogel-vector encoding (CR=16) compressed to only 3 MB using L-Z compression. This is over 3 times smaller than the traditionally computed case. The size of the L-Z compressed file is a first order measure of non-redundant information content. Therefore, hogel-vector encoding produces fringes that have most of the redundancy ironed out. The hogel-vector decoded fringes were compressed by L-Z to nearly as small a symbol count as the hogel-vector array from which it was generated (in this case, 2.25 MB). Application of L-Z compression to the 2.25-MB hogel-vector array yielded a data file of only 0.8 MB. This example of second-order compression was possible by the correlation among hogel-vector components since they are generated from typical physically sensible scenes.

Another source of second-order compressibility is the quantizability of hogel vectors as compared to fully computed fringes. Hogel-vector components can be quantized to fewer than 8 bits with predictable and tolerable image degradations. Any decrease in sample quantization in a fully computed fringe pattern produces unacceptable levels of image noise. This is also true for fringelets.

Fringelets are not as amenable to second-order compression as are hogel vectors. Fringelets resemble fringes. They do not contain readily accessible orthogonal components that correspond to the 3-D scene description. In general, they are only as compressible as fully computed fringes. The application of L-Z compression, as described above, compressed a 2.25-MB fringelet array into a data format that typically contained 1.6 MB. Also, fringelets cannot be quantized in the way that hogel vectors can. Quantization of fringelets results in the same kind of image noise that occurs when quantizing fully computed fringes.
7.2.6 2-D Compatibility

Although it would be a horrible waste to use encoded fringe patterns simply to produce 2-D images, in reality most electronic displays in 1994 are 2-D. It is useful to view the scene content of an encoded fringe pattern on a 2-D display, either for performing diagnostics or as a quick and cheap preview. Therefore, another desirable feature of an encoded fringe format is that it be very simply converted to a 2-D version of the 3-D image that it represents.

Consider a fringe pattern computed using traditional methods. The time required to convert this fringe back to a 2-D image is as much (or more) as the time to compute the original fringe. Such a process requires a convolution or Fourier transform to numerically perform the diffraction that occurs in a holovideo display.

Hogel-vector encoding supports 2-D compatibility. Simply selecting a single component from each hogel vector provides an orthographic projection of the 3-D scene encoded in the hogel-vector array.

Fringelet encoding can support 2-D compatibility. For diagnostic purposes, the fringelet basis fringes have one previously unmentioned constraint: the first sample of all but one of the basis fringes has a zero value. The one basis fringe with a non-zero value in the first sample location corresponds to a diffraction direction that was roughly normal to the hologram plane. Therefore, selecting the first sample of each fringelet in the fringelet array provides a 2-D image similar to the hogel-vector case.

7.2.7 Summary of Features

The following chart is a summary of the features discussed in this section. A blank entry signifies little or no practical applicability, and "****" signifies strong applicability. Included are the speeds for total computation (excluding transfer times) and for decoding. Note that fully computed fringes are already "decoded" and therefore do not apply to decoding speed.
<table>
<thead>
<tr>
<th></th>
<th>Fully Computed Traditional</th>
<th>Fully Computed Diff.-Spec.</th>
<th>Hogel-Vector Encoded</th>
<th>Fringelet Encoded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Speed</td>
<td>*</td>
<td>**</td>
<td>****</td>
<td>****</td>
</tr>
<tr>
<td>Decoding Speed</td>
<td>**</td>
<td>**</td>
<td>****</td>
<td>****</td>
</tr>
<tr>
<td>Compressibility</td>
<td>****</td>
<td>***</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Interoperability</td>
<td>***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extensibility</td>
<td>****</td>
<td>****</td>
<td>****</td>
<td>****</td>
</tr>
<tr>
<td>Scalability</td>
<td>*</td>
<td>****</td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>Manipulability</td>
<td>*</td>
<td>**</td>
<td>****</td>
<td>***</td>
</tr>
<tr>
<td>2nd-ord. encodability</td>
<td>*</td>
<td>****</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>2-D compatibility</td>
<td>****</td>
<td>****</td>
<td>***</td>
<td></td>
</tr>
</tbody>
</table>

In general, hogel-vector encoding is the fringe computation method that allows for the best performance. Its only drawback is a time-consuming decoding step. However, this drawback may be less important than the scalability, manipulability, and other features to which hogel vectors are strongly amenable. Fringelet encoding is the computing method of choice when speed is the most important concern.

### 7.3 Engineering Trade-Off: Bandwidth, Depth, Resolution

The analysis of point spread caused by holographic bandwidth compression (Section 5.4) resulted in a useful expression relating the important parameters of a holovideo imaging system. An important feature of this relation (Equation 26) is that it relates the image resolution and image depth to bandwidth. This is a unique relationship: no previous analysis of a holographic imaging system has managed to account for all of these parameters in one simple expression.
Recall from Section 5.4.3 that point spread is approximately minimized when the hoogle width $w_h$ is chosen to balance the contributions from aperture blur and spectral-sampling blur, which is (Equation 27) equivalent to choosing

$$w_h = \delta \sqrt{BW} .$$  \hspace{1cm} (28)

Typically, the full sampling bandwidth is used: BW is hereafter in this analysis assumed to be $BW=0.5$ cycles/sample. (In general, BW can be reintroduced into these analytical expression by making the substitution $CR \rightarrow 2BW \cdot CR$.) Therefore, the first step to designing a bandwidth-efficient holovideo system is to choose the hoogle width to be

$$w_h = \frac{\delta}{\sqrt{2}} .$$  \hspace{1cm} (29)

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 26) is

$$\text{Bandwidth: } N \text{ in symbols/hogel} \rightarrow \quad N \geq \frac{Z}{\delta} \left(2\sqrt{2} \sin \frac{\Theta}{2}\right) .$$  \hspace{1cm} (30)

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

$$\text{Depth: } Z = \text{max. image depth (mm)} \rightarrow \quad Z \leq \delta N \left(2\sqrt{2} \sin \frac{\Theta}{2}\right)^{-1} .$$  \hspace{1cm} (31)

$$\text{Resolution: } \delta = \text{point spread (mm)} \rightarrow \quad \delta \geq \frac{Z}{N} \left(2\sqrt{2} \sin \frac{\Theta}{2}\right) .$$  \hspace{1cm} (32)

Another way to approach holovideo system design is to calculate what compression ratio (CR) can be achieved given the other system parameters. In this way, the band-
width compressibility of an existing system can be determined. Recalling that 
\( N = N_h / CR \) and Equation 30 gives the expression

\[
CR = \frac{\delta^2}{\lambda Z}.
\]

(33)

This is a fascinating result. It indicates that bandwidth compression can be achieved as
the square of the tolerable image point spread. Sacrificing image resolution has a dramatic impact on compressibility. Also, limiting image depth has a direct effect on compressibility.

7.3.1 Encoding Efficiency: Visual-Bandwidth Holography

The expression for calculating bandwidth given other holovideo parameters
(Equation 30) gives a measure of success for holographic encoding schemes. Electronic holography is difficult because it traditionally requires a huge bandwidth that is
tied to the physics of optical diffraction. To diffract light to form 3-D images requires a
high sampling pitch, and therefore a large number of samples for each hologram.
However, the 3-D image cannot contain as much useful visual information as the traditionally computed holographic fringes, i.e., holographic bandwidth is wasted due to
the limited performance of the human visual system. Equation 30 gives a measure of
the bandwidth required for a holovideo system employing one of the holographic
encoding schemes born of diffraction-specific computation. If they are efficient, then
the minimum required bandwidth should be roughly the same as the amount of useful
information contained in the 3-D image, where useful information means useful to the
human visual system. This is indeed the case: given the acuity of the human visual sys-
tem, Equation 30 yields a bandwidth commensurate with the information content of a
3-D image when considering the number of volume elements (voxels) that it contains.

To illustrate the efficiency of diffraction-specific holographic encoding, consider the
following example. Let \( \Theta = 30 \) degrees, \( Z = 80 \) mm, and \( \delta = \sqrt{2} \cdot 0.300 \) mm
\( = 0.424 \) mm. Equation 30 requires that \( N = 138 \) symbols/hogel. This is the minimum
bandwidth required using diffraction-specific holographic encoding. (For comparison, a traditionally computed fringe pattern requires 491 samples per each width of 0.300 mm.) One way to compute the useful visual information in the image volume is to divide it into voxels with lateral and depth resolutions that match the acuity of the human visual system. (See Section 2.1.1.) Since image volume is proportional to the number of hogels, the amount of useful visual information in this image volume is 213 voxels/hogel. This is within a factor of two of the minimum bandwidth requirement. (In fact, it is lower since the selected image point spread of $\delta = 0.424$ (mm) is somewhat larger than can be seen with the human visual system.)

The preceding example illustrates that diffraction-specific holographic encoding does indeed provide a minimum bandwidth requirement in a visual information sense. Of course, the statistical correlations among the elements of a particular image allow for second-order bandwidth compression. Nevertheless, in a general sense, hogel-vector or fringelet encoding match the required holographic bandwidth to the useful visual information that can be detected by the human visual system. No longer is the holographic bandwidth dictated by the physics of optical diffraction, but is instead dictated (as it ought to be) by the information content of the image as seen by the human visual system. Diffraction-specific holographic encoding schemes are therefore properly place in a new category of holographic imaging: "visual-bandwidth holography."

If diffraction-specific holographic encoding is indeed visual-bandwidth holographic imaging, then it should be capable of providing 2-D images using only the bandwidth dictated by the 2-D image content. Two-dimensional images are usually discretized into a 2-D array of pixels. When properly discretized, each pixel (as each hogel in a fringe) is a size that matches the lateral acuity of the human visual system (HVS). Therefore, a diffraction-specific hologram of a 2-D image contains a hogel count that is equal to the image pixel count. The minimum required holographic bandwidth should be approximately one symbol per hogel. To apply Equation 30 to the case of a 2-D image, the proper value for maximum image depth is $Z=0.375$ mm. Because the HVS can only distinguish the flatness of a surface down to the depth acuity of the HVS
(approximately 0.75 mm), the maximum depth of a 2-D holographic image is half of the depth acuity. Using the lateral acuity (at a viewing distance of 600 mm) of \( \delta = 0.175 \) mm and the viewing zone size of \( \Theta=30 \) degrees, Equation 30 gives a minimum required bandwidth of 1.6 symbols/hogel. This number is quite close to the expected value of 1.0. (To be fair, the size of the viewing zone should be smaller than \( \Theta=30 \) degrees since a 2-D image has no view-dependent visual information and therefore should only be seen from the range where it appears to be approximately orthographic. Indeed, such a value of \( \Theta=20 \) degrees gives \( N=1 \).) This is a fascinating result: diffraction-specific holographic encoding requires the bandwidth contained in the image whether the image is 2-D or 3-D. Bandwidth does not exceed visual information content, regardless of image depth or size.
Chapter 8

Future Directions

This chapter describes some future research directions based on diffraction-specific fringe computation.

8.1 Specialized Fringelet Decoding

Fringelet decoding is so simple that it can be implemented in a number of ways. The following are three different suggested implementations of fringelet decoding: (1) in special digital hardware, (2) in RF analog processing, and (3) optically in the display.

8.1.1 Digital Fringelet Decoding

Digital signal processing (DSP) technology can be used to implement fringelet decoding. As shown in the figure below, a hybrid output card that stores fringelets rather than fringes could perform the decoding on the way to the digital-to-analog converter (DAC) as each hololine is needed.

The primary advantage of such an approach is that the bandwidth going into the framebuffer card is on 1/CR the full fringe bandwidth, where CR is the compression ratio. Such a framebuffer card stores fringelets and decodes them into fringes only when necessary. This “fringebuffer” approach allows CR times the storage using the
same amount of VRAM. For example, in a standard framebuffer, 2 MB of VRAM can hold a fully computed 2-MB fringe. In a fringebuffer, using CR=18, the same VRAM can hold 2 MB of fringelets that represent a 36-MB fringe pattern. The main disadvantage of such a fringebuffer is that it must be specially designed and constructed. Also, the requirements on this fast digital hardware support for a holovideo display are not conceptually alleviated. This fringebuffer still must ultimately generate the full (36-MB) fringe signal at video frame rates.

8.1.2 Analog Electronic Fringelet Decoding

An analog RF electronic decoding method is similar to the digital implementation described above, but the decoding is done to the analog signal coming out of the framebuffer DAC. A fringelet is read out, followed by sufficient blanking to allow for decoding into a hogel before the next fringelet comes along. Decoding is performed with a series of delays and a combiner.

This approach has similar advantages to the digital fringelet decoding implementation. In this case, however, a standard framebuffer reads out the fringelets to the analog decoder. The requirements on the fast digital hardware support for the holovideo display are reduced by a factor of CR. Fringelet decoding occurs in real time. As in the fringebuffer case, 2 MB of VRAM provide fringelets for a 36-MB fringe pattern for a CR=18. The disadvantage of this approach lies in its analog design: it is subject to noise, drift, and nonlinearities.
8.1.3 Optical Fringelet Decoding

A fringelet is shift invariant within the hogel region. An optical fringelet decoding scheme exploits the shift-invariant nature of a fringelet. This shifting can be accomplished temporally. Although an HOE can be used to optically replicate light diffracted from a single fringelet, but it is simpler to perform the replication temporally.

Specifically, fringelets are fed into the AOM of a scophony-type display\(^5\) with mismatched horizontal scanning. Mismatching the horizontal scanning system causes the image of a fringelet to be swept across the hogel region. If each fringelet is read out padded by sufficient blanking, the viewer sees a decoded version of the fringelet. The best way to produce this sweeping effect is to reverse the direction of the horizontal scanner. For example, if the scanner is set to match exactly the motion of the acoustic signal within the AOM, reversing scanner sweeps the signal across twice its width in the opposite direction.
Typical numbers are as follows: hogels are 4K samples wide, with fringelets of 512 samples. (Compression ratio is therefore 8.) Using this method of optical fringelet decoding in a viewer-plane geometry, the current 36-MB MIT display can produce an image that is 8 times larger (in a combined image-volume and view-angle sense) without requiring additional bandwidth. In general, optical replication works best for viewer-plane geometry since hogel widths can be larger.

This approach has many advantages. Conventional framebuffers can be used since blanking is not only tolerated but is indeed required. Also, the engineering trade-offs\textsuperscript{60} used in designing the scophony-geometry holovideo display are relaxed: the horizontal scanning system needs only to scan at the appropriate data rate. Its scanning rate no longer needs to be geometrically matched to exactly compensate for the motion of the acoustic signal within the AOM. This advantage manifests itself in many possible ways. For example, the horizontal scanners can be slower, and larger displays can be built. As another example, other high-bandwidth SLMs, particularly a deformable mirror device (DMD) can be used to modulate the light in a fringelet display. Finally, the saving provided in this “fringelet display” allows for a full-parallax display to be built with only an order of magnitude increase in required bandwidth. The application of holographic encoding to full-parallax holovideo is discussed in the following section.

To summarize the three specialized fringelet decoding schemes, the analog electronic and optical approaches make use of blanking and therefore can utilize conventional framebuffer systems. In these two cases, a framebuffer system is no longer required to store and read out an entire decoded fringe pattern. Instead, multiple channels can be represented on a single framebuffer card, adding to the speed and further reducing communication bottlenecks. In addition, optical fringelet decoding allows the construction of a “fringelet display” that alleviates the burden on the digital and the analog support electronics. The optical decoding approach is best done in a viewer-plane geometry due to a reduction in the achievable image resolution that would result in other geometries.
8.2 Extension to Full Parallax Holovideo

Diffraction-specific fringe computation and the holographic encoding schemes born of it are applicable to full parallax holographic imaging. Although this thesis focused on horizontal-parallax-only (HPO) holovideo, full parallax can be implemented by treating the vertical dimension in the same way as the horizontal dimension. Diffraction is not only linear, but it is separable into the two lateral dimensions. (See Appendix B.) A full-parallax diffraction-specific approach is outlined in this section. The application of hogel-vector and fringelet encoding decreases required bandwidth and computation time by the roughly the square of the reductions found in the HPO case.

A full-parallax fringe pattern has a vertical sampling pitch that is the same order of magnitude as the horizontal sampling pitch. In general, the fringe pattern is sampled with different pitches in each dimension. Hogels take the form of rectangles rather than line segments; each hogel has a height as well as a width. The vertical size of the hogel is roughly the same as the hololine spacing in the HPO case. The vertical spectrum of the hogel is sampled and treated separately from the horizontal hogel spectrum. Each hogel is encoded as two vectors: one for the horizontal spectra and one for the vertical. A set of rectangular basis fringes must be precomputed, with one set for the horizontal spectra and one set for the vertical spectrum. Decoding precedes as in the HPO case, the difference being that both the horizontal and vertical components of the hogel vector are used.

For full-parallax hogel-vector encoding, bandwidth compression is achieved independently in the vertical and horizontal dimensions. The spectrum of each rectangular hogel is subsampled (by generally independent amounts) in each of the two dimensions. If each full-parallax hogel has a height of \( N_h^V = 256 \) samples and is encoded with a compression ratio of \( CR^V = 16 \), then the addition of vertical parallax increases the computational task of hogel-vector decoding by a factor of \( N_h^V / CR^V = 16 \). For comparison, traditional interference-based computation would require roughly 256 times the computation in such a case. Hogel-vector encoding decreases information
content and computation time by $N_h^H/CR^H$ in the horizontal dimension and by $N_h^V/CR^V$ in the vertical dimension. The total compression ratio achievable for a full-parallax fringe pattern is $N_h^H/CR^H \times N_h^V/CR^V$. For typical values, this total compression ratio would be about 256. The implications are propitious: a typical full-parallax fringe computed using hogel-vector encoding can be computed in the same time as a full-parallax fringe computed using traditional interference-based methods. The number of samples in the full-parallax hogel-vector array would be roughly equal to the number of sample in an HPO fringe computed using traditional methods.

Fringelet encoding promises the same improvements outlined for hogel-vector encoding. However, the decoding scheme must be considered carefully. A full-parallax fringelet is rectangular, with dimensions $N_h^H/CR^H \times N_h^V/CR^V$. The indirection table (used for the implementation of HPO fringelet decoding) must map each 2-D hogel sample location to one of the 2-D fringelet samples. The indirection table is generated using a separate set of truncation-translation statistics in the vertical dimension. This full-parallax indirection table represents aperiodic replications in both the horizontal and vertical dimensions.

In summary, the holographic encoding schemes developed from diffraction-specific computation allow for the generation of full-parallax fringe with just over an order of magnitude increase in required bandwidth and computation time. The application of these holographic encoding schemes may for the first time make possible a full-parallax holovideo display.
Chapter 9

Conclusion

Diffraction-specific fringe computation has been designed, implemented, and used to generate complex holographic images for real-time display. The discretization of space and spatial frequency in a holographic pattern has yielded this faster, more direct method of computation. By treating a hologram as an array of hogels, the computational process is streamlined and generalized. Although it is a two-step process, diffraction-specific computation uses computing resources more efficiently than traditional interference-based methods. The intermediate diffraction specifications (hogel vectors) can be transmitted, stored, and manipulated analogous to 2-D image processing.

Early attempts to eliminate the four inherent problems of traditional interference-based computation led to the bipolar intensity method and the linear summation of precomputed basis fringes. As described in Chapter 3, the bipolar intensity method eliminated the noise components inherent to traditional interference-based computation, but did not provide a means for bandwidth compression. The use of precomputed fringes provided dramatic speed increases, and enabled the first-ever interactive computation of holographic images. But bipolar intensity computation was still fundamentally an interference-based approach. Computation followed an equation that is derived from the process of interference.

Diffraction-specific computation automatically eliminates the noise components found in traditional interference-based computation. It also provides a direct means of incorporating higher-level, non-analytical image elements. More important, it increases speed and allows for two diffraction-specific holographic encoding schemes. The two holographic encoding schemes born of diffraction-specific computation allow a bandwidth compression ratio of 16 times without severely degrading image fidelity. The
use of diffraction-specific holographic encoding eliminates transmission bottlenecks caused by the enormous bandwidth requirements of traditionally computed holographic fringes.

Hogel-vector encoding and fringelet encoding are direct-encoding schemes. They generate the encoded fringe directly. As a result, the total time to compute them and to decode them was far less than the time to compute fringes directly. Therefore, not only are these methods useful for bandwidth compression, but they also provide a superior means of generating holographic fringes – even if no bandwidth bottleneck exists.

Computation speeds are increased, in the case of fringelet encoding and decoding, by a factor of over 100 due to the reduced number of calculations per fringe sample. Already, fringelet encoding allows for computation of complicated image scenes in under 10 s. The implementation of fringelet decoding in simple specialized hardware or in optical systems should allow for another order of magnitude increase in speed. Fringelet decoding is fast and easy, so it can be incorporated into an output framebuffer card. In such a case, there was no longer a bandwidth bottleneck in a holovideo display system. The full fringe pattern never exists until it was generated on-the-fly by the output module - each hololine as it is needed. This specialized fringelet decoding can be implemented in digital or analog electronic hardware. However, the most promising application of specialized fringelet decoding is an optical decoding scheme in which the holovideo display is specially constructed to perform the simple fringelet decoding step. Such a “fringelet display” may achieve an 8-fold increase in image volume size without increasing the requirements on computation or on electronics hardware.

Another important result of these novel holographic encoding techniques is the three-way trade-off between bandwidth, image resolution, and image depth. For example, given an upper limit on the amount of blur that is imperceptible by the viewer, a finite bandwidth limits the depth of the image. The conclusion is simply this: 3-D is more difficult than 2-D. This observation is expected, since a 3-D image contains more
visual information than does a 2-D image. The bandwidth of an encoded holographic fringe representing a softball-sized image is still roughly a factor of 100 times larger than the equivalent size 2-D image. If holographic images need only to be flat, then higher achievable compression ratios reduce the bandwidth to be roughly equal to the 2-D image case. If the image must be very deep, then lower compression ratios must be used lest image quality should suffer. However, as long as a holovideo imaging system is not required to produce images with infinite depth, holographic encoding can provide bandwidth compression.

The engineering trade-off analysis of Section 7.3 is a powerful result. This is the first time that the analysis of a holographic system has resulted in a simple expression relating the fundamental system parameters of bandwidth, image resolution, maximum image depth, and viewing zone size. More significant is the result that diffraction-specific holographic encoding enables a reduction of bandwidth to roughly match the useful visual content of an image. This accomplishment makes diffraction-specific computation the first “visual-bandwidth holography.” Diffraction-specific holographic encoding has closed the gap between the amount of information the human visual system can utilize and the huge amount of information required to effect holographic diffraction. For the first time, holographic fringe patterns can be encoded and manipulated without being bound to the enormous bandwidths required by the physics of the diffraction of light. Instead, a stream of holographic information contains the same amount of visually useful information as does the image. Bandwidth is determined by the abilities of the human visual system, not by the physics of the diffraction of light.

By making holographic computation faster and more practical, diffraction-specific computation promises to create many applications for holovideo. The encoding schemes – particularly hogel-vector encoding – allows for the manipulation of encoded fringes using techniques that are similar to existing (2-D) image processing techniques. For the first time, encoded holographic fringes can be treated along with 2-D image data and other digital information. Diffraction-specific fringe computation
has made holographic information more miscible with digital information from video, audio, images, text, etc. The new medium of electro-holography can be integrated into digital multimedia technologies. Holovideo is, after all, a genuinely digital medium. Unlike photography, video, or optical holography, which are essentially analog media, holovideo was invented as a digital medium. Its digital nature makes it an intensively computational medium; diffraction-specific computation minimizes the computation requirements and makes holovideo compatible with other digital media.

Diffraction-specific computation, though invented for real-time interactive holovideo, can be applied to other holographic applications. For example, fringes generated using diffraction-specific computation can be recorded onto film to produce static holographic images. (As part of the debugging process in this thesis research, computed fringes were often printed (using a 600 dots/inch laser printer) directly onto a transparency. These fringes created 3-D, sometimes full-parallax images.) The “fringe printer” approach to holographic hardcopy would require that large fringe patterns be computed on relatively standard computational platforms. In this case, diffraction-specific computation could provide the advantages of speed and bandwidth reduction. A type of holographic hardcopy called binary optics\textsuperscript{39} records computed fringes in various optical materials using microlithography. Again, diffraction-specific computation provides a means for rapid, flexible fringe computation. Whether by printing, photoreduction, or binary optics techniques, hardcopy holograms can be used as holographic optical elements (HOEs) that can be used as complex optical components. Finally, the engineering trade-offs of a holographic memory system may require the use of diffraction-specific holographic encoding to make the most efficient use of system components. The same may be true in holographic metrology.

An important application of diffraction-specific holographic encoding is in the storage of holographic information. Already, HPO fringes have been reduced by a factor of 16, and full-parallax fringes can be encoded for a bandwidth compression of 256. Further “second-order” encoding, based on the statistical correlations among hogel-vector components should allow for additional reduction in required bandwidth. Sufficient
compression may allow holographic movies to be recorded on digital media and transmitted over high-bandwidth networks or television cable.

One missing piece of the holovideo pipeline is an input device. To create a true "televi-
sual" medium, holovideo displays must have a source of 3-D descriptions of real scenes. Work being performed at the MIT Media Lab involves developing an "intelli-
gent camera" that sees not just patches of light and color but instead derives the 3-D structure of a scene\textsuperscript{69}, paralleling the behavior of the human visual system. Once a 3-D model of a scene is extracted it is well suited to be converted into holographic fringes for real-time display using diffraction-specific computation.

As computation power increases and as spatial light modulation technologies improve, holovideo is likely to find applications in visualization, education, design, entertain-
ment, and communication. Most of these applications require interactivity. Holo-
graphic fringes must be computed in negligible time so that the user is only aware of the visible incarnation of the scene, and not the many layers of processing and calculation within the holovideo system. Diffraction-specific computation promises to pro-
vide increases in speed and manipulability as the size of the image volume increases and the variety of content grows.
Appendix A

Glossary of Terms and Abbreviations

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

CM2
Connection Machine Model 2, massively parallel supercomputer manufactured by Thinking Machines, Inc. of Cambridge, MA USA.

CGH
computer generated hologram, specifically, the set of computed fringe patterns in digital form.

Cheops
a digital image processing platform originally designed to explore scalable digital TV and real-time image encoding and decoding for the TVOT Consortium at the MIT Media Laboratory.

diffraction-specific computation
the new method of holographic fringe computation described, implemented, and analyzed in this dissertation.

diffraction table
a table used to map the location and brightness of an image element to a sequence of hogel-vector component contributions.

DSP
digital signal processing.

fringe
the holographic pattern that is either recorded optically or generated computationally and used to diffract light to form an image.

HPO
horizontal parallax only. A 3-D imaging system, e.g., a holographic one, that provides motion parallax in the horizontal viewing-zone direction but not in the vertical direction.

HVS
human visual system.

hogel
holographic element; a piece of hologram that has homogeneous diffraction properties and is small enough to appear as a single point to the viewer.

hololine
a line of a holographic pattern.

holoplane
the plane containing the fringe pattern during recording, reconstruction, or computation.
holovideo
a 3-D holographic electronic display system that impresses a holographic fringe pattern upon a beam of light, causing the light to diffract and form a 3-D image. [from Greek holos whole + Latin videre to see]

hogel vector
a compact set of weights specifying the diffractive purpose of a hogel.

image-plane hologram
a hologram that is located amid the image that it reconstructs.

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

K-
1024. The suffix K- is used to signify 1024 items, e.g., 1 Ksamples = 1024 samples, 1 KB = 1024 bytes.

M-
1048576. The suffix M- is used to signify 1024*1024 items, e.g., 1 Msamples = 1048576 samples, 1 MB = 1048576 bytes.

MAC
multiplication accumulation; a numerical calculation consisting of one multiplication and one addition.

parallax resolution
the number of distinguishably different views that the HVS can see in a given image.

SCSI
small computer standard interface; a somewhat low-bandwidth interface for computers and their peripherals.

SLM
spatial light modulator; a device which modulates a beam of light.

SNR
signal-to-noise ratio.

viewer-plane hologram
a hologram that coincides with its viewing zone, i.e., a hologram designed to be located at the eyes of the viewer.

viewer stimulus
that which is seen by the viewer, i.e., the qualities of a displayed image, as perceived by the viewer.

VRAM
video random-access memory: fast read-write (generally) dual-ported digital memory used in framebuffers.
Appendix B

Spectral Decomposition of Diffracted Light

This section derives an expression for the relationship between the light diffracted by a holographic fringe pattern and the fringe spectrum. In keeping with the philosophy of the diffraction-specific approach, the derivation is backwards: it begins with the desired image and works backward to the fringe spectrum. First the diffracted light is decomposed into a summation of plane-wave components at the plane of the hologram. Each of these plane waves is diffracted by a particular spatial frequency component of the fringe pattern.

The goal of this derivation is to determine what light must be diffracted by a fringe to generate a specified 3-D image \( u(r) \). To derive the light distribution at the plane of the hologram (at \( z=0 \)), first consider the 3-D Fourier transform of this image:

\[
U(k) = \int \int \int u(r) e^{-ik \cdot r} \, dr
\]

(B1)

where

\[
r = x + y + z
\]

\[
k = k_x + k_y + k_z
\]

(B2)

are the spatial and spatial-frequency coordinates. The complementary 3-D inverse Fourier transform is

\[
u(r) = \kappa^3 \int \int \int U(k) e^{ik \cdot r} \, dk
\]

\[
\kappa \equiv (2\pi)^{-1}
\]

(B3)
Thus, a 3-D object can be represented as a summation of spatial frequency components. Each of these components is a plane wave in space with a propagation vector of \( \mathbf{k} \). Plane waves are the eigenmodes of the free-space propagation of light\(^{13} \). Therefore, since the image is represented as a summation of plane waves, there is no need to apply the laws of diffraction to determine the light distribution at a particular plane. The requirements for propagation are

\[
k_x^2 + k_y^2 + k_z^2 = k_0^2 \equiv 2\pi / \lambda \]

\[
k_0 > k_x^2 + k_y^2 \tag{B4}
\]

for propagation in the positive \( z \) direction. (The free-space wavelength of light is \( \lambda \).) Combining the last two equations, the light distribution that must be diffracted by the fringe at the hologram plane \( z=0 \) is

\[
u_D(x, y) = \kappa^2 \int_{-\infty}^{\infty} \int U_O(k_x, k_y) e^{i\kappa x} e^{iyk_y} e^{-iz_0\sqrt{k_0^2 - k_x^2 - k_y^2}} dk_x dk_y \tag{B5}
\]

and

\[
U_O(k_x, k_y) = \int_{-\infty}^{\infty} \int u_O(x, y) e^{-i\kappa x} e^{-iyk_y} dx dy \tag{B6}
\]

where \( u_O(x, y) \) is the specified image in a plane at a distance \( z=z_0 \) from the hologram plane and \( U_O(k_x, k_y) \) is the 2-D Fourier transform of the specified image\(^{38} \).

This dissertation generally discusses horizontal-parallax-only (HPO) holograms. Because a single HPO holograms (hololine) diffracts light only in the \( x \) and \( z \) direction, the above expressions can be simplified to eliminate \( y \) dependence:
\[ u_D(x) = \kappa \int_{-\infty}^{\infty} U_O(k_x) e^{ixk_x} e^{iz\sqrt{k_0^2-k_x^2}} \, dk_x \]  
\[ U_O(k_x, k_y) = \int_{-\infty}^{\infty} u_O(x, y) e^{-ixk_x} \, dx \]  

(For full-parallax holography, the \( y \) dependence can be carried throughout the remaining analysis.) Note that this equation is correct for points at all depths: \( z_p < 0, \ z_p > 0, \ z_p = 0 \). In general, image points or elements lie in front of, behind, and on the hologram plane.

Equation B7 states that a weighted sum of plane waves compose the diffracted distribution. The weighting factors are the Fourier transform of \( u_O \). An additional phase term is included in each plane-wave component. Notice that this phase factor is not itself a function of \( x \). For a given \( x \) location, the superposition (in Equation B7) used to construct \( u_O \) uses the same phase factor for each plane-wave component. To see this more clearly, consider the example of a point image at \((x_p, z_p)\). The Fourier transform of a spatial impulse is

\[ U_p(k_x) = e^{-ix_p k_x} \]  

which gives, at the plane of the hologram,

\[ u_p(x) = \kappa \int_{-\infty}^{\infty} e^{i k_x (x-x_p)} e^{iz_p \sqrt{k_0^2-k_x^2}} \, dk_x. \]  

In this case, the “weights” are uniform, leaving only phase factors in the superposition of plane waves. The role of the phase term is to add a fixed phase shift for each plane.
wave. It represents a second order component of the diffracted wavefront, namely a curvature that is a function of $z_p$.

Consider the role of the phase term in Equation B10. From the point of view of a single sample of the fringe (at a single $x$ location), each spectral component of the diffracted light must include a particular ($x$-independent) phase shift. And for a given hogel, each spectral component needs to include such a phase shift. As implied by the dependence of the phase term on $z_p$, a particular spatial frequency component of the hogel must include different phase factors for contributions to image elements at different depths. Diffraction-specific computation ignores this phase term, lumping all of the diffractive function of a particular spatial frequency into a single basis fringe. This is equivalent to fixing the wavefront curvature to zero independent of $z_p$ and therefore reduces the image resolution. However, the basis fringes are designed to diffract light into a range of directions, not just into a single plane wave. Statistically, the effect of the phase term is small since $k_x$ is not deterministic as is independent of $z_p$. Furthermore, the phase term becomes less important if spatially incoherent light is used. Equations B7 and B8 assume spatially coherent monochromatic illumination, and the superposition involves summing the complex amplitudes of plane waves. However, in the case where incoherent light is used, the intensities of the plane-wave components add. Therefore, the phase term is no longer important in the relationship between fringe spectrum and light diffraction.

So far this analysis has shown that an image can be constructed by diffracting plane waves. The remaining task is to relate a particular spatial frequency component $f$ to each diffracted plane wave. The direction of diffraction $k_x$ is a function of the direction in which light is incident upon the fringe pattern. In optical holography, this means the illumination beam. In holovideo, it means the effective direction of light incident on the modulator in the display system. (This parameter was empirically derived by measuring the directions in which light is imaged by fringes consisting of constant spatial frequencies.) When light incident at the hologram with direction $k^l = (k_x^l, k_y^l, k_z^l)$ is
modulated by a fringe with spatial frequency component $f$, the boundary condition of phase-matching$^{15}$ requires that

$$ k_x = k_x^0 + 2\pi f $$  \hspace{1cm} (B11)  

where $k_x$ is the $x$-component of the directional vector $k$ of the (first-order) diffracted light. The point image example (Equation B10) becomes:

$$ u_p(x) = \int_0^{0.5p} e^{i2\pi f(x-x_p)} e^{i\phi(f)} df $$  \hspace{1cm} (B12)  

where $p$ is the fringe sampling pitch (in samples/mm) and $\Phi(f)$ is the phase term represented as simply a function of $f$. Finally, Equation B10 shows that a superposition of spatial frequencies can be used to construct a fringe that diffracts light from a given $x$ location to form a point image. Relating $k_x$ to geometric optics, $k_x = k_0 \sin \Theta_O$ combined with the phase-matching criterion (Equation B11) gives the useful expression

$$ f = \frac{\sin \Theta_O}{\lambda} - \frac{k_x^i}{2\pi} $$  \hspace{1cm} (B13)  

which is essentially the grating equation. This expression was used in most of this thesis research for determining which spatial frequency must be used to diffract light in a particular direction.

As a final note on the linear superposition of fringe patterns, it is assumed in this thesis research that the modulation technique used by the display system to diffract light is linear. Amplitude modulation of light allows for linearity since it simply scales the wavefront by a real scalar between the values of 0.0 and 1.0. Phase modulation is also approximately linear for a modulation depth of $\Delta \phi < \pi/4$. The MIT holovideo display system using a weak phase modulation, so the condition $\Delta \phi < \pi/4$ is satisfied.
Appendix C

Computation of Synthetic Basis Fringes

This section describes two related methods for generating diffraction-specific basis fringes: the iterative constraint method (discussed in the Background section) and a novel simulated annealing approach. These numerical methods are necessitated by the complicated set of spatial and spectral constraints on a given basis fringe. Spatially, the basis fringe must represent a single hogel at a single location in the fringe pattern. It must therefore have a finite width, and must possess homogeneous characteristics across its width (i.e., a virtually constant amplitude). Spectrally, the basis fringe has multiple often conflicting characteristics.

The spectrum of a hogel is divided into sampled increments of width $\Delta \nu$. Each basis fringe is responsible for contributing energy in a particular portion of the spectrum and in no others. This corresponds to diffracting light in a finite range of directions. If the spectrum has a bandwidth of $BW=0.5$ cycles/sample and is divided into $N$ regions of width $\Delta \nu=0.5/N$, then each basis fringe $i$ must possess a spectrum that is non-zero in the range $[i\Delta \nu, (i+1)\Delta \nu]$ for $i=[0,N-1]$. The shape of the spectrum is a truncated sinc function as required by the sampling theory to recover the continuous spectrum. Other shapes – gaussian, rectangular, triangular – were also used to determine their effects on image quality. Each spectrum has a specific width.

An analytic solution to these many constraints is practically impossible. Attempts at deriving closed-form analytical solutions leads to further complexity. More importantly, these analytic approach are mired in interference-based computation. They are prone to the same problems associated with interference-based computation.

Several numerical methods can be applied to generate basis fringes. These numerical methods produce a fringe that is synthesized from a set of constraints rather than from interference-based analysis. Therefore, these numerically generated fringes are called
synthetic fringes. The numerical methods that worked best for this thesis were the method of iterative constraints, and a novel application of simulated annealing. Both of these algorithms were implemented on the Connection Machine Model 2 with 16K data-parallel processors.

C.1 Method of Iterative Constraints

As discussed in Section 2.7 "Iterative Hologram Computation Methods" on page 34, a typical iterative constraint algorithm involves using the forward and inverse Fourier transforms and alternately applying spatial and spectral constraints. The figure on the next page illustrates this algorithm schematically and in a step by step description.
Algorithm for Method of Iterative Constraints

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

where:

\[
\begin{align*}
    u(x) &= \text{computed spatial pattern} \\
    v(f) &= \text{spectral pattern} \\
    u'(x) &= \text{modified spatial pattern} \\
    v'(f) &= \text{modified spectral pattern}
\end{align*}
\]
As an example of the application of iterative constraints to the synthesis of basis fringes, consider the case of a basis fringe that is to have spectral energy in a rectangular region ranging from 0.0 up to 0.5/8=0.0625 (cycles/sample). The spectral constraint on magnitude is that it be 1.0 within the region \([0.0,0.0625]\) and 0.0 elsewhere. The spatial constraint is that it have a uniform magnitude of 1.0. Phase is unconstrained in both the spatial and spectral domains. For a length of \(N_h = 256\) samples, computation begins with a randomly generated spatial phase and a uniform magnitude of unity. After transforming into the spectral domain, the magnitude is forced to equal the desired spectrum, namely 1.0 in the region \([0.0,0.0625]\) and 0.0 elsewhere. After inverse transforming into the spatial domain, the magnitude is forced to be uniform. This cycle iterates. After 20 iterations, a reasonable solution to the constraints is generally obtained, after which very little changes. This example was run using an iterative constraint algorithm implemented on the CM2. After 100 iterations, the resulting real part of the spatial pattern is shown as the top basis fringe in the following figure:

![Eight Basis Fringes](image)

At its right is its spectrum. Included are an additional seven basis fringes, each with a spectrum that is rectangular with width \(\Delta f=0.5/8=0.0625\), each non-zero in the range \([i\Delta f,(i+1)\Delta f]\) for \(i=[0,7]\). These basis fringes evenly divide the spectrum into eight equal parts. Such a set of basis fringes was used to perform hogel-vector decoding for
the parameters \( N_h = 256 \) samples and CR=32. For most cases, the basis fringes used had a gaussian profile, with \( 1/e^2 \) spectral full-width of \( \Delta_f \).

The advantage of the iterative constraint algorithm is that it can generate synthetic basis fringes using a wide variety of constraints. It is fast, though speed is not important in precomputation of basis fringes since these basis fringes need only to be computed once and then integrated into diffraction-specific computation algorithms. The more important issue is one of closeness to the desired constraints. Notice that in the example of eight basis fringes shown in the previous figure, the spectra are not perfect rectangles. The ripples and drop-outs in basis fringe spectra lead to speckle-like artifacts in the holographic images. The disadvantage is that this method sometimes stagnates at local minima in the error function\(^ {48} \). The error function is the difference between the desired spectral characteristics and those in the spectrum of the calculated pattern. One solution was to introduce a small amount of noise into the spatial pattern with each iteration. The amount of noise decreased with each iteration. Still, the synthesized basis fringes were still lacking in spectral quality. The solution was to implement a simulated annealing algorithm.

### C.2 Simulated Annealing

Simulated annealing\(^ {47} \) is a numerical optimization algorithm used for a variety of applications. Just as physical annealing (the process of slowly cooling a liquid into a solid) seeks to decrease the global energy state of a system, simulated annealing seeks to minimize the error between certain qualities of a numerical system and their targeted constraints. To further the analogy, simulated annealing sometimes increases error based on a probability function that resembles Boltzmann’s probability distribution:

\[
Prob(E) = \exp\left(-\frac{E}{kT}\right)
\]  

\[\text{(C3)}\]
which relates the probability of a system being in a particular energy state $E$ given that its thermal energy is $kT$. As a physical system cools, it generally moves toward a lower energy state. However, there is always a finite probability that it will move (temporarily) to a higher energy state. In simulated annealing, a random test change is made to the numerical system. Let $\Delta E$ be the difference between the system "error" with the test change and the system without it. If the resulting change in error $\Delta E$ is negative, then the change is allowed. If $\Delta E$ is positive, it may be kept or rejected, depending probabilistically on Equation C3. The system always moves toward a potential decrease in error, but does sometimes move toward a potentially higher error state. This occasional addition of error prevents the system from stagnating at a local minimum (as does the iterative constraints algorithm). The parameter $kT$ must be chosen to allow for occasional increase in error, and must be slowly cooled (reduced) as the system converges to its targeted constraints.

A simulated annealing algorithm was implemented for this thesis on the CM2. The process of randomly altering one sample of a basis function is slow. It is important to begin the annealing with a good guess. Therefore, the basis fringe generated using the method of iterative constraints was used as the initial guess (seed) for the simulated annealing algorithm. The first step in the annealing algorithm is randomly to select a sample of the spatial pattern and to change it, within the spatial constraints. After a test change has been made, the spectral energy is calculated by applying a Fourier transform. The error function after a given iteration was calculated as

$$E = \{ \frac{1}{N_h - 1} \cdot \sum_{i=0}^{N_h-1} \frac{1}{2} \left[ |v(f_i)|^2 - |v_c(f_i)|^2 \right] \}$$  \hspace{1cm} (C4)

where $v_c(f_i)$ is the targeted spectral magnitude constraint and $f_i$ is the $j$ location of sample $i$ in the basis fringe spectrum. This error function is therefore the root-mean-squared (RMS) energy difference between the targeted spectral constraint and the
spectra of the pattern at each iteration. The simulated annealing algorithm is illustrated on page 166.

With each iteration of the simulated annealing algorithm, the basis fringe sometimes is left unchanged, sometimes moves toward a lower error, and occasionally moves toward a higher error. To illustrate this process, page 167 shows the basis fringe and its spectrum as it progresses generally toward a more satisfactory solution to the constraints.

The combination of the iterative constraint method followed by simulated annealing produced extremely precise basis fringes. The only disadvantage to simulated annealing is that it is slow. However, speed is not important to the precomputation of basis fringes. The advantages of simulated annealing are many. Besides being capable of producing superior basis fringes with tightly constrained spectra, this algorithm has also been applied to the generation of fringes with other constraints. For example, simulated annealing was applied to the synthesis of fringes with highly quantized phase—in some cases binary phase. Results were good, though the cooling schedule must be more carefully engineered in these nonlinear constraint applications. For comparison, in these cases the method of iterative constraints generally stagnated before a satisfactory fringe was generated.
The Simulated Annealing Algorithm

This figure shows a schematic of the simulated annealing algorithm as implemented for the synthesis of basis fringes. Essentially, the fringe pattern gradually moves toward its targeted spectral constraints through a probabilistic decision to keep or reject randomly made changes. The annealing precedes until the error function $E$ decreased to less than one quantization level per sample.

* $\text{Rnd} = a$ random number, $(0.0,1.0]$
Simulated Annealing: An Example

This figure shows the process of simulated annealing applied to the synthesis of a basis fringe. At left is the real part of the spatial pattern. At right is the energy of the spectrum. The top line is the initial seed spatial pattern and its spectrum. Each line represents a change that was made and kept. The total number of changes attempted was 10000, and the number that were kept was 440. The bottom line is the annealed basis fringe and its spectrum. This basis fringe was targeted to have a uniform spectrum of 1.0 in the range [0.0625, 0.1250] and no spectral energy outside this range. The initial seed pattern was generated by the method of iterative constraints, which left some deviations from the targeted spectrum (even after 100 iterations). Notice the progression of the spectrum: it began with deviations in the form of dark regions within the region where it was targeted to be uniform. These variations virtually melted away as the annealing progressed. The error function decreased by roughly a factor of 10. Total time for 10000 iterations was two minutes.
References

Holography


3-D and Imaging


Optics


Holovideo Computing


20 Mark Lucente and Tinsley A. Galuycan, "Rendering holovideo," IEEE Computer Graphics and Animation. Accepted for review.

Computer-Generated Holography


Iterative Computing


MIT Holovideo


2-D Holographic Displays


Image Processing


**Information Theory**


**Holographic Information**


**Scophony**


**Cheops**


Acknowledgments

Research on the MIT holographic video system has been supported in part by the Defense Advanced Projects Research Agency (DARPA) through the Rome Air Development Center (RADC) of the Air Force System Command (under contract No. F30602-89-C-0022) and through the Naval Ordnance Station, Indian Head, Maryland (under contract No. N00174-91-C0117); by the Television of Tomorrow research consortium of the Media Laboratory, MIT; by US West Advanced Technologies Inc.; Honda Motors, IBM, NEC, GM, Thinking Machines, Inc.

The "Connection Machine" supercomputer was manufactured by Thinking Machines, Inc., Cambridge, MA, USA.

The "Onyx" computer graphics workstation and the "Reality Engine" graphics framebuffer systems were manufactured by Silicon Graphics, Inc., Mountain View, CA, USA.

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. This work was made possible thanks to many: Pierre St.-Hilaire (display optics and electronics), Carlton J. Sparrell (Cheops and display hardware), Wendy J. Plesniak (image processing and authoring), Michael Halle (image processing and computational support), Shawn Becker (Cheops software czar, discussions on encoding), John Watlington (Cheops designer), V. Michael Bove, Jr. (Cheops and TVOT leader), Brett Granger (Cheops software), John Underkofler (image processing and computational support), Michael Klug (optical hardware), Tinsley Galyean (computational collaborator), Paul Hubel (color collaborator), H. John Caulfield (enlightening discussion on holographic computation), Hiroshi Yoshikawa (Nihon University, Japan), Ravikanth Pappu, John D. Sutter, Derrick Arias, Dave Sheppard, Henry Holtzman, John Y. A. Wang, Edward H. Adelson, Kathy Nelson, Scott Wilcox, Jeff Breidenbach, Uli Friedman, Kevin J. Whitcomb, Brad Edelman, Rajeevan Amirtharajah, Bill Farmer, Michele Henrion, Bill Burling, Linda Conte, Melissa Yoon, Marilyn Pierce.

Thanks also to Barry Arons, John Ashley, Janet Cahn, Tracy Clark, Christopher J. Colbert, Cristina Dolan, Steve Drucker, Neil Gershenfeld, Karen Halliday, Mike Hawley, Alexandra MacLeod, Carlos Sanchez Moreno, Alex Sherstinsky, Olin Shivers, Megan J. Smith, Chon and Kathy at Poppa & Goose (>10^6 Kcal), L. v. B., W. A. M., P. I. T., J. S. B., A. L. W., J. V., and God.


Special thanks to Edwin Land, James R. Melcher, Harold "Doc" Edgerton, Muriel Cooper, and Constantine Simonides.

Genial thanks to John, Jim, Steve, Ed, Terry, and (always first) Mom and Dad.
Biographical Note

Mark Lucente worked for five years in the MIT Media Lab Spatial Imaging Group, where he developed the interactive generation of 3-D holographic images. His college degrees (Ph.D., S.M., S.B.) were bestowed upon him by the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. He is a member of the SPIE, Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.

In his work, Dr. Lucente combines knowledge of optics, spatial light modulation, computation, visual perception, and communication systems to develop electro-holography into a practical medium. His earlier work involved the application of lasers to high-bandwidth optical communication systems, 3-D imaging systems, and the study of device physics.