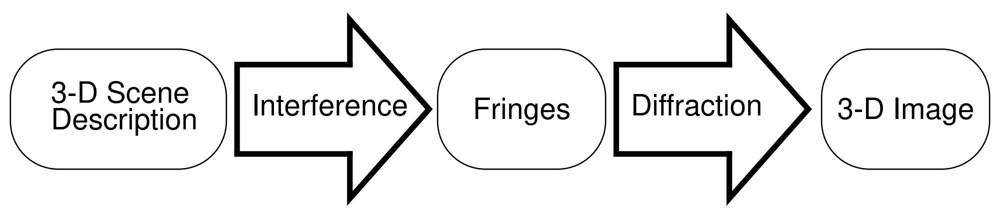
Chapter 4

Diffraction-Specific Computation

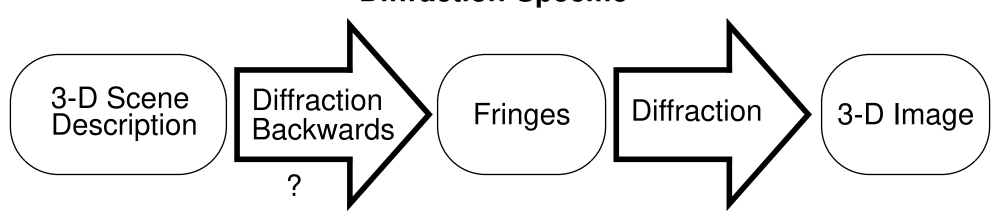
The purpose of this thesis is to design, implement, and analyze a new “diffraction-specific” fringe computation method. It solves the four main problems inherent to traditional (interference-based) holographic computation outlined in the previous chapter. Stately simply, the diffraction-specific approach 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



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

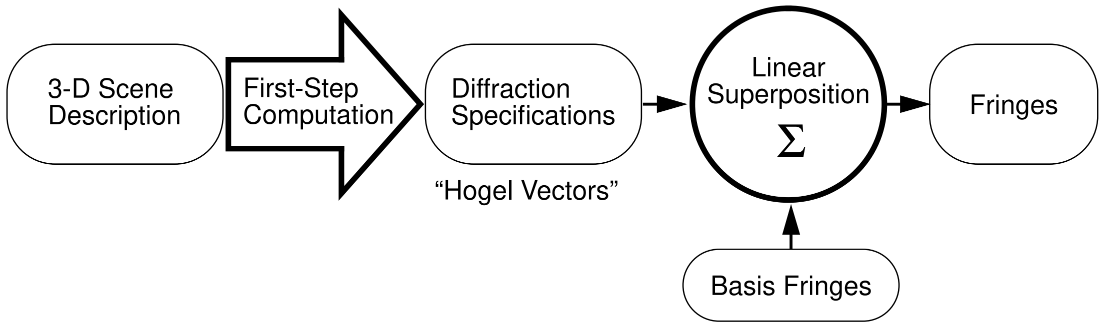
Diffraction-Specific



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 holography: 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.

Schematic of Diffraction-Specific Computing



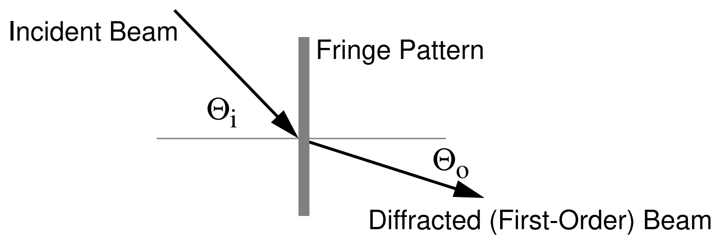
4.1 Recipe for Diffraction-Specific Computation

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:

$$f\,\lambda = \sin\Theta_o - \sin\Theta_i \tag{7}$$



where Θ_i is the angle of the incident light and Θ_o is the angle of a diffracted beam, λ 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.

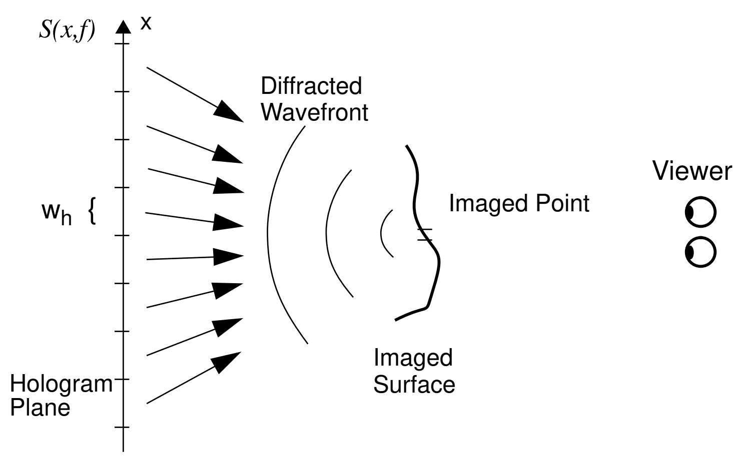
4.2.1 Sampling: Concepts

Sampling and subsequent reconstruction of a signal is common in communication systems[70,71] involving continuous physical properties (e.g. acoustic signals, images[66,67], 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 hololine. This may seem odd: how can a single sample possess a spatial frequency content? Nevertheless, sampling theory[72] 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 (Δ_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:



Sampling theory[72] 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 Δ_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 $2/w_h$:

$$S(x_i, f) = \sum_j S_{ij} \operatorname{sinc}(j - f w_h) \quad (8)$$

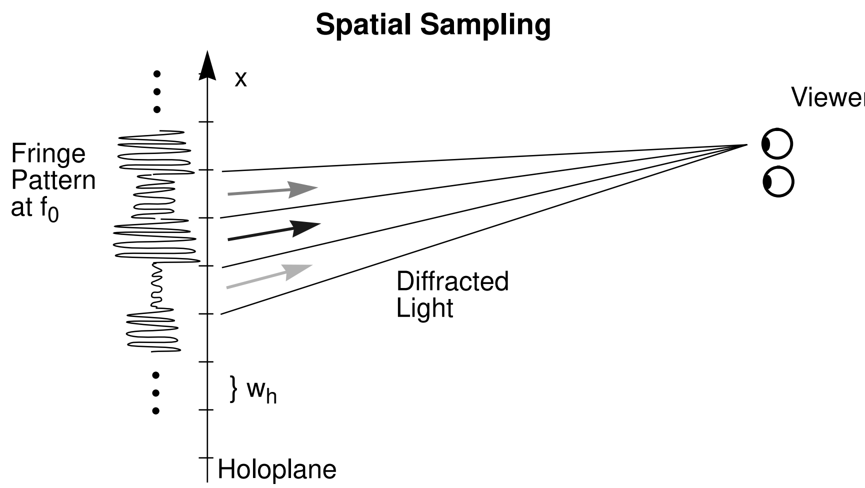
Furthermore, *S(x,f)* can be fully recovered through a convolution with a sinc function of width $2/\Delta_f$:

$$S(x, f) = \sum_i \sum_j S_{ij} \operatorname{sinc}(j - f w_h) \operatorname{sinc}(i - x\Delta_f) \quad (9)$$

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[12] 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

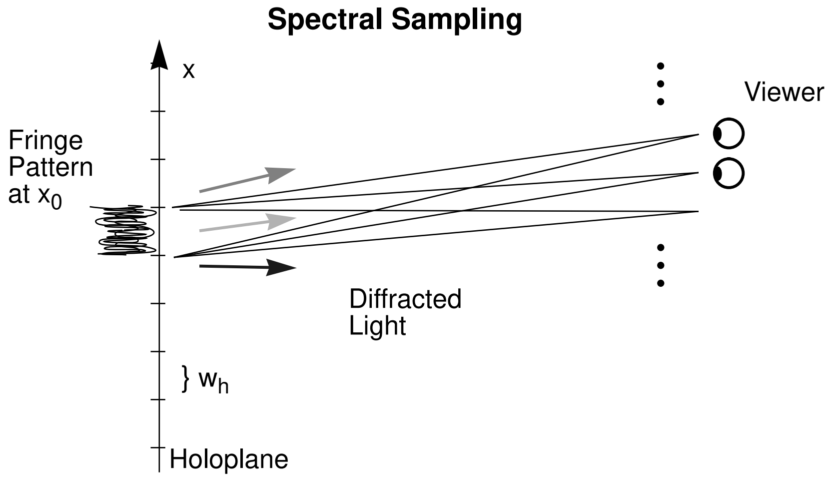
To sufficiently sample *S(x,f)*, the most rapid variations in *S(x,f)* must be determined as separate functions of x and of f. These are determined by the imaging requirements of a holographic imaging system. Given one spatial frequency f_0, the spatial sample spacing w_h must be small enough so that 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)x600 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 Δ_f must be sufficiently small to reconstruct the most rapid variations in $S(x_0,f)$ 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-dependant 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 Δ_f. The grating equation (Equation 7) relates the diffracted angle Θ_o to the incident angle Θ_i of the illuminating light. Taking the derivative of both sides with respect to Θ_o:

$$\frac{\partial f}{\partial \Theta_o} = \frac{\cos\Theta_o}{\lambda}. \tag{10}$$

For a viewing distance of 600 mm and a λ=0.633 μm (and allowing the cos term to be its maximum of 1), the calculation of Δ_f becomes

$$\Delta_f = \frac{(3mm)/(600mm)}{0.633\mu m} = 8 \quad mm^{-1} \quad . \tag{11}$$

In a small holographic fringe of width 1024 samples, where each sample is spaced by $1/p$=0.6 μm, 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 Δ_f. Under these typical 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 Δ_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.

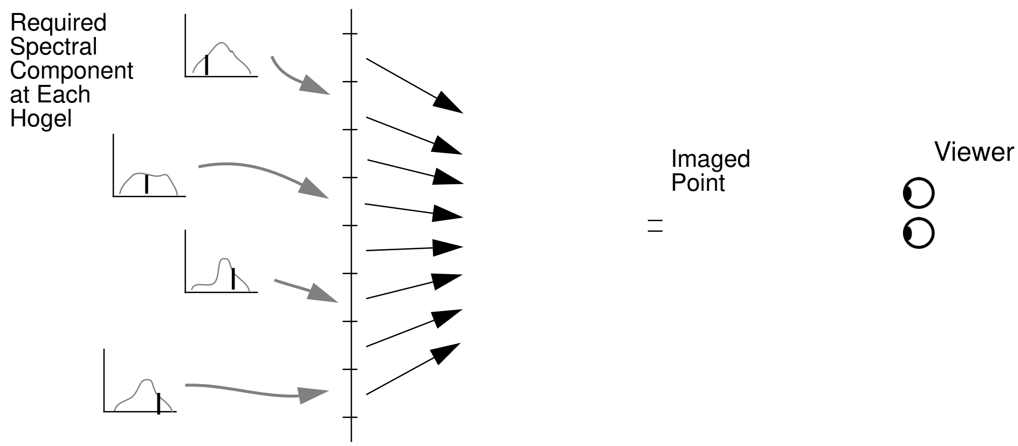
4.3 Generation of Hogel Vectors

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 computer-graphics 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.

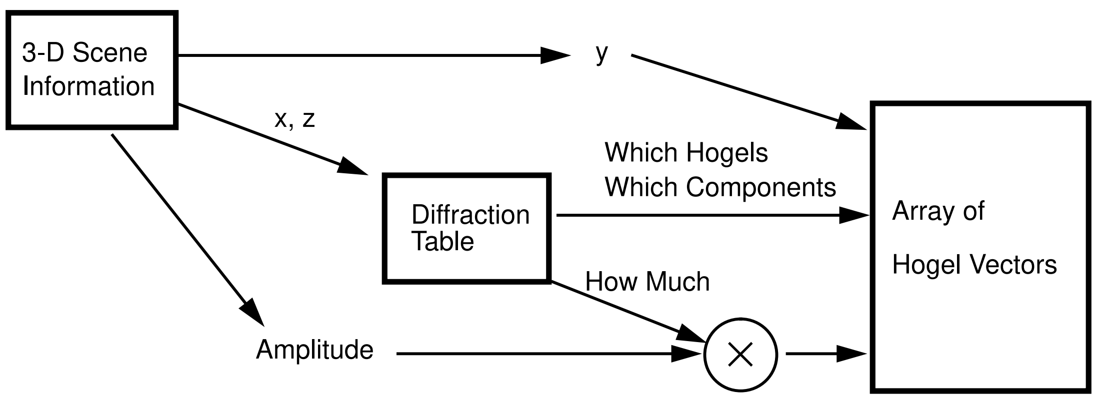
Next, consider a point source located at a distance in front of the hologram, i.e., between the holoplane and the viewer. To diffract light to form this point, light must be diffracted in slightly different directions from each adjacent hogel in a finite linear region of a hololine. One or more components 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.

Calculating Diffraction Specifications



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 $z \rightarrow \infty$; 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[19].

4.3.3 Additional Techniques

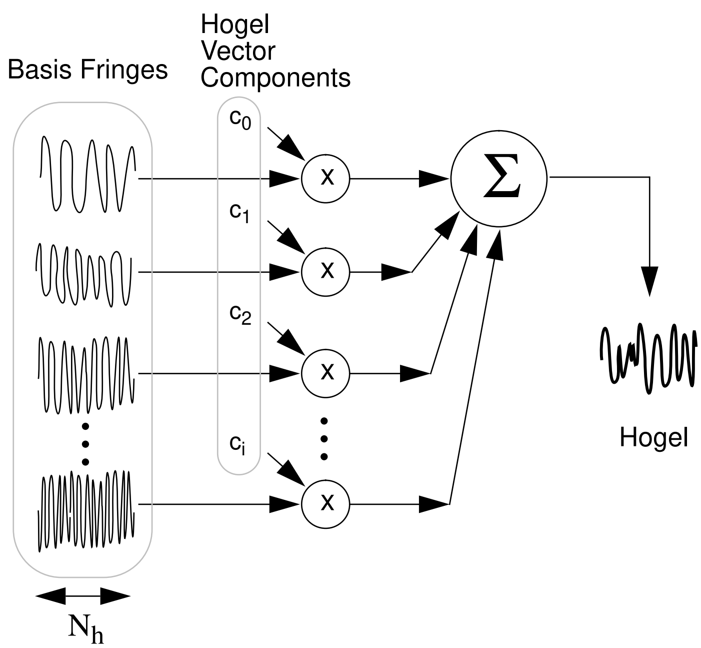
Diffraction-specific computation provides an easy solution to the problem of analytically constrained image elements (discussed in the previous chapter, Section 3.1.) To employ higher-level image or scene elements, multiple diffraction tables are used. For example, if line segments or polygons of various 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[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:

$$f = \frac{\sin\Theta_O}{\lambda} - \frac{k_x^i}{2\pi} \tag{12}$$

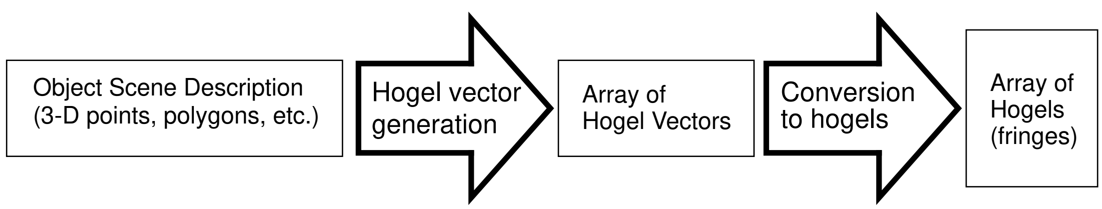
$$\text{where} \quad k_x^i = \frac{2\pi}{\lambda}\sin\Theta_i \tag{13}$$

is the x-component of the effective direction of light incident on the modulator of the display system, and Θ_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

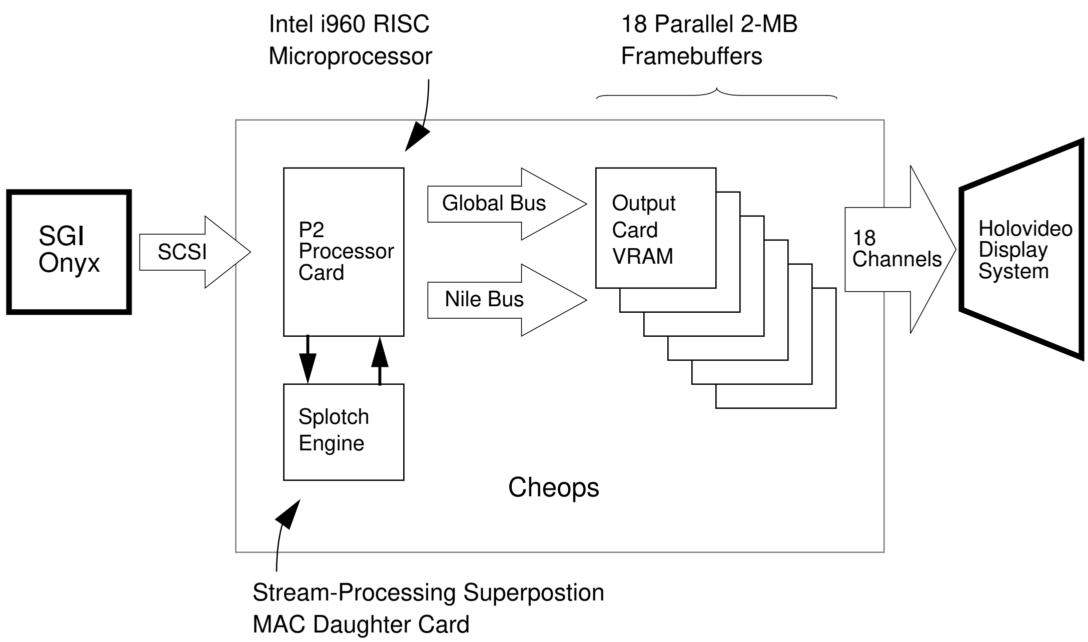
Diffraction-specific computing was implemented and used to generate holographic fringes. The general data flow is diagramed below. There are two computation steps

Diffraction-Specific Fringe Computation



(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.

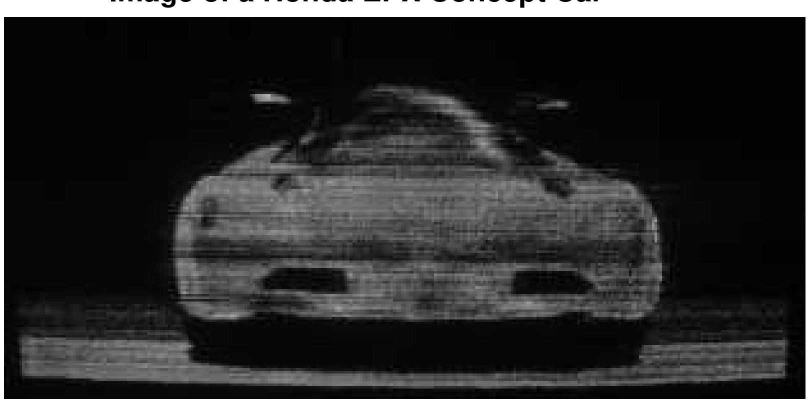
4.6 Image Generation

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



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.

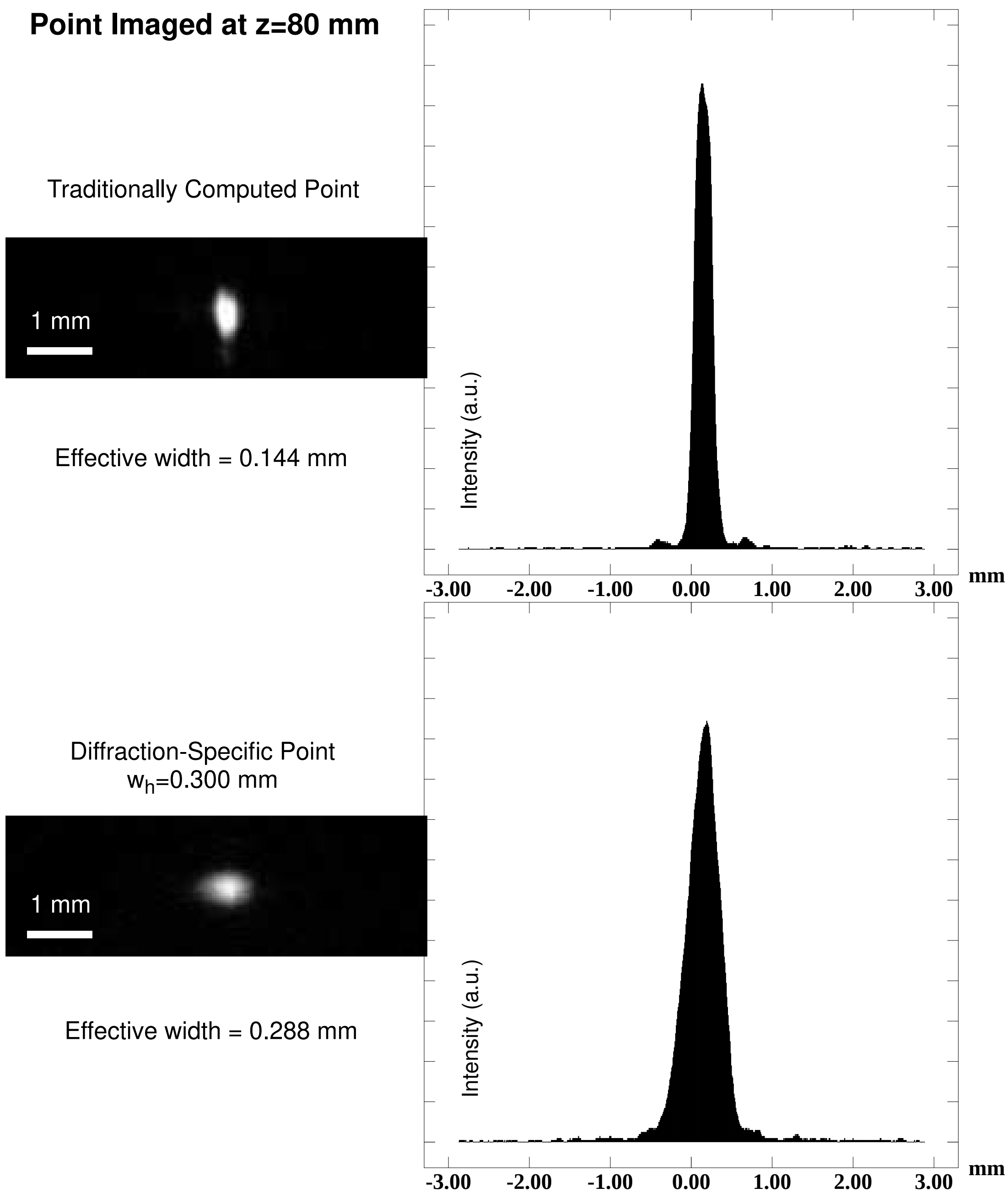
- Digital photographs of close-ups of images were acquired using a Sony CCD array placed directly at the plane of the image. The CCD array measured 768x494 tricolor pixels, with approximately 8-μm spacing horizontally. Each frame was grabbed and digitized using an SGI Sirius Video system. Frame integration 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.



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.

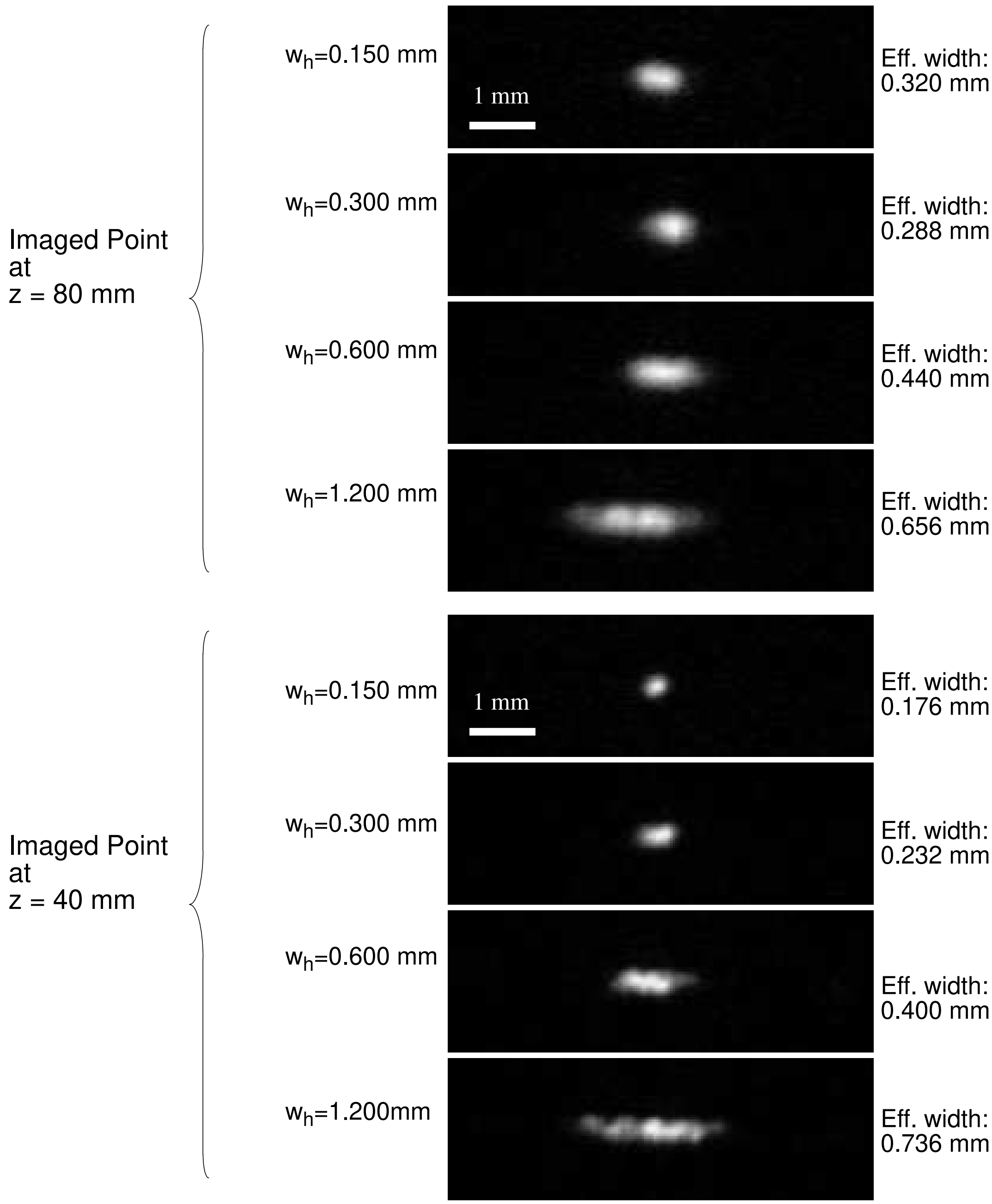
The increase in point spread due to diffraction-specific computation is discussed in the following chapter (Section 5.4). For now, the empirical behavior of point spread is illustrated in the figure on page 70. Selecting w_h=0.3 mm gives the best performance. A larger w_h blurs the imaged point, and 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~0.100 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 ~2.0 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 conversion 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 complexity. 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 demonstrates 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 s + 20 s = 330 s
- Diffraction-specific on CM2: 350 s + 20 s = 370 s
- Traditional interference-based on CM2: 790 s
- SCSI transfer time: add 45 s.

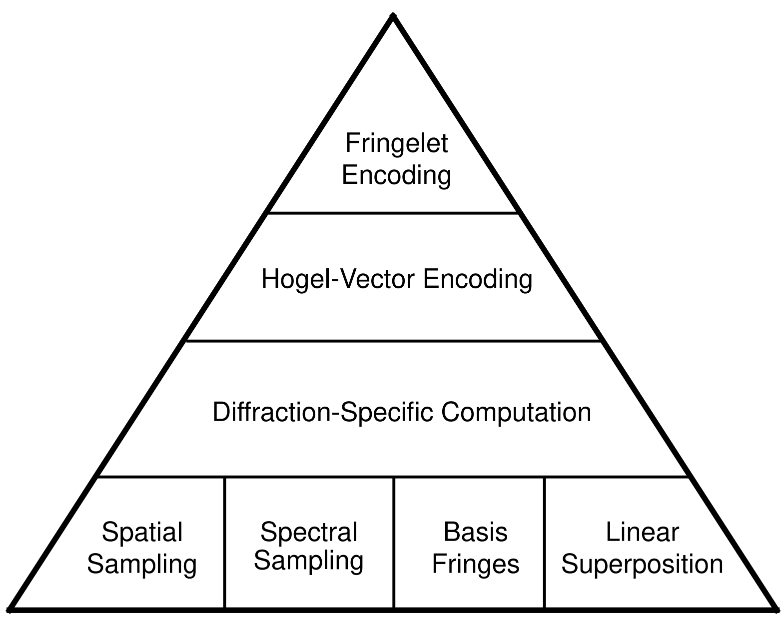
The first two numbers (in the diffraction-specific cases) indicate times for the generation of the hogel-vector array and for their conversion to hogels. These two times add to the total computing time (excluding transfer time).

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.”

Thesis Construction: Progression to Holographic 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.