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.

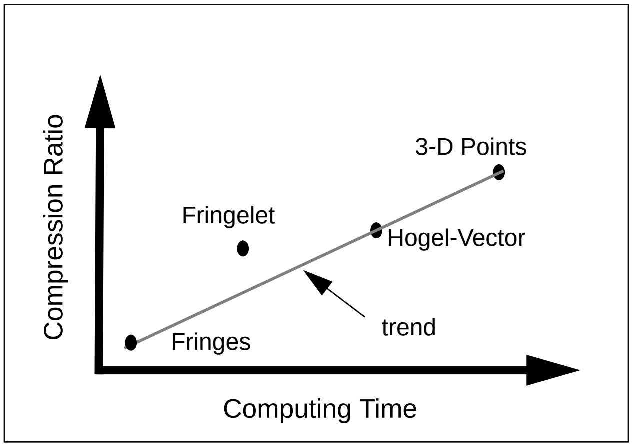
Different encoding schemes provide different degrees of compressibility and speed. There is a general trend in compressibility: encoded fringes that resemble the 3-D description are more compressed (have a higher CR), whereas encoded fringes that more closely resemble the final fringes are less compressed (have a lower CR). Computing speed is also a function of the “look” of an encoded format, i.e., its semblance 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.

Computation Method	MACs/sample	Improvement
Unencoded	N_h	--
Hogel-Vector	N_h/CR	CR
Fringelet	N_h/CR^2	CR^2

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

Computation Method	Computation Time	Improvement	Improvement vs. Traditional
Unencoded	330 s	--	2.4 x
Hogel-Vector	21 s	16 x	38 x
Fringelet	7 s	47 x	114 x

- 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[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*[68], is found in subband coding and pyramid encoding schemes used for image bandwidth compression[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*[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*[68] - 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[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 king 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.

	Fully Computed Traditional	Fully Computed Diff.-Spec.	Hogel-Vector Encoded	Fringelet Encoded
Total Speed		*	**	****
Decoding Speed	- - - -	- - - -	**	****
Compressibility			****	***
Interoperability			***	
Extensibility			****	****
Scalability		*	****	**
Manipulability	*	**	****	***
2nd-ord. encodability		*	****	*
2-D compatibility			****	***

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 hogel width w_h is chosen to balance the contributions from aperture blur and spectral-sampling blur, which is (Equation 27) equivalent to choosing

$$w_h = \delta\sqrt{BW} \quad . \tag{28}$$

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→2BW·CR.) Therefore, the first step to designing a bandwidth-efficient holovideo system is to chose the hogel width to be

$$w_h = \frac{\delta}{\sqrt{2}}. \tag{29}$$

Next, one of the fundamental system parameters can be calculated given the other system parameters. For example, if the image resolution δ and maximum image depth Z are fixed, the minimum required bandwidth (recalling Equation 26) is

Bandwidth: N in symbols/hogel → $$N \geq \frac{Z}{\delta}\left(2\sqrt{2}\sin\frac{\Theta}{2}\right) \tag{30}$$

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

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

Resolution: δ = point spread (mm) → $$\delta \geq \frac{Z}{N}\left(2\sqrt{2}\sin\frac{\Theta}{2}\right) \tag{32}$$

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 bandwidth compressibility of an existing system can be determined. Recalling that $N=N_h/CR$ and Equation 30 gives the expression

$$CR = \frac{\delta^2}{\lambda Z} \ . \tag{33}$$

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 system, 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 Θ=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 $\delta = 0.424\ (\mathrm{mm})$ 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 Θ=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 Θ=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 $\Theta \approx 20$ degrees gives $N \approx 1$.) 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.