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)

$$I_T = \underset{\substack{\text{Object}\\ \text{Self-}\\ \text{Inter-}\\ \text{ference}}}{|E_O|^2} + \underset{\substack{\text{Reference}\\ \text{Bias}}}{|E_R|^2} + \underbrace{2Re\{E_O \cdot E_R^{*}\}}_{\substack{\text{Useful}\\ \text{Fringes}}} \tag{6}$$

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[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 Msamples would require a minimum of 5 trillion (i.e., $5 \mathrm{x} 10^{12}$) 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

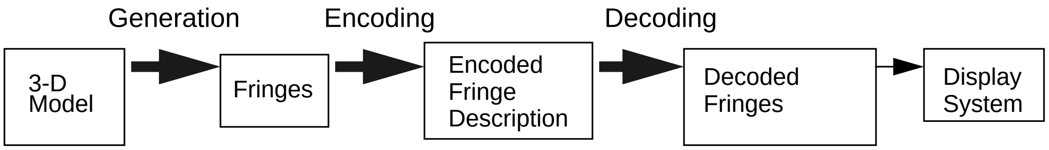
Fringe computation begins with a 3-D description of the object or scene to be imaged. Usually this a point-by-point description. Often, a higher level description is useful to represent lines and patches and curved surfaces. Higher-level image elements can enable higher quality and more directly computed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

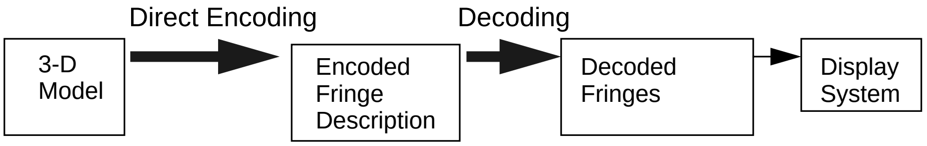
3.1.4 Need for Encoding

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[67]. 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[67]. 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:



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:



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[18,19], 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[19].

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[18,19].

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[19]. 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[18,19]. 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 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.