Published in SPIE Proceeding #1667 “Practical Holography VI” (SPIE, Bellingham, WA, February 1992) paper #04 (in press).

Optimization of Hologram Computation for Real-Time Display

Mark Lucente

MIT Media Laboratory
Spatial Imaging Group
20 Ames St.
Cambridge, MA 02139
USA

ABSTRACT

Several methods of increasing the speed and simplicity of the computation of off-axis transmission holograms are presented, with applications to the real-time display of holographic images. A bipolar intensity approach enables a linear summation of interference fringes, a factor of two speed increase, and the elimination of image noise caused by object self-interference. An order of magnitude speed increase is obtained through the use of precomputed look-up tables containing a large array of elemental interference patterns corresponding to point source contributions from each of the possible locations in image space. Results achieved using a data-parallel supercomputer to compute horizontal-parallax-only holographic patterns containing 6 megasamples indicate that an image comprised of 10,000 points with arbitrary brightness (grayscale) can be computed in under one second.

INTRODUCTION

The real-time display of holographic images has recently become a reality. The MIT Spatial Imaging Group has reported the successful generation of small three-dimensional (3D) computer-generated holographic images reconstructed in real time using a display system based on acousto-optic modulation of light[1, 2, 3]. In any real-time display system, a computer-generated hologram (CGH) must be computed as quickly as possible in order to provide for dynamic and interactive images. However, numerical synthesis of a holographic interference pattern demands an enormous amount of computation, making rapid ($\sim$1 second) generation of holograms of even limited size impossible with conventional computers.

A holographic fringe pattern is computed by numerically simulating the physical phenomena of light diffraction and interference. In general, light diffracts from a three-dimensional object to the hologram plane. Since the analytical expressions that model diffractive propagation through free space resemble the Fourier transform integral, computation of holographic interference patterns often utilizes the Fast Fourier Transform (FFT) algorithm[4]. Though relatively fast, this approach is useful only for images possessing discrete depth surfaces[5, 6], and becomes slow when applied to images that extend throughout an image volume.

A more general approach is a ray-tracing method in which the contribution from each object point source is computed at each point in the hologram plane. This method can produce arbitrary three-dimensional (3D) images, but is slow, since it requires one calculation per image point per hologram sample. As presented in this paper, the application of several methods of reducing computation complexity leads to computation times as short as one second on a data-parallel-processing supercomputer. First, a “bipolar intensity” representation of the holographic interference pattern is developed and shown to eliminate unwanted image artifacts and simplify calculations without loss of image quality or generality. Second, a look-up table approach is described and shown to provide further speed increase, though image resolution and quantization noise become issues. Finally, exemplary computation times are presented.

HOLOGRAPHIC IMAGING SPECIFICS

This paper focuses on the computation of off-axis transmission holograms possessing horizontal parallax only (HPO), a quality of the “rainbow” or Benton hologram. It is possible to represent an HPO hologram with a vertically stacked array of one-dimensional holographic lines[6, 7]. Consider an HPO hologram made optically using a reference beam with a horizontal angle of incidence. Spatial frequencies are large in the horizontal direction ($\sim$ 1000 lp/mm) and increase with the reference beam angle. However, by limiting the view zone to only a single vertical view, vertical spatial frequencies are low ($\sim$ 10 lp/mm). It is evident that elimination of vertical parallax provides a factor of 100 (or greater) reduction of CGH size. During reconstruction of this hologram, diffraction occurs predominantly in the horizontal direction. It is appropriate to represent this holographic pattern with a relatively low vertical sample spacing (or “pitch”), roughly that used in a two-dimensional (2D) imaging system. In the horizontal dimension, however, the sampling pitch must be very high in order to accurately represent the holographic information. For each horizontal plane (“scan-plane”), the associated horizontal line of the hologram diffracts light to form image points in that plane only. Therefore, the 2D holographic pattern representing an HPO 3D image can be thought of as a stack of one-dimensional (1D) holograms or “holo-lines”. The goal of this paper, then, is to compute these 1D holographic lines as quickly as possible.

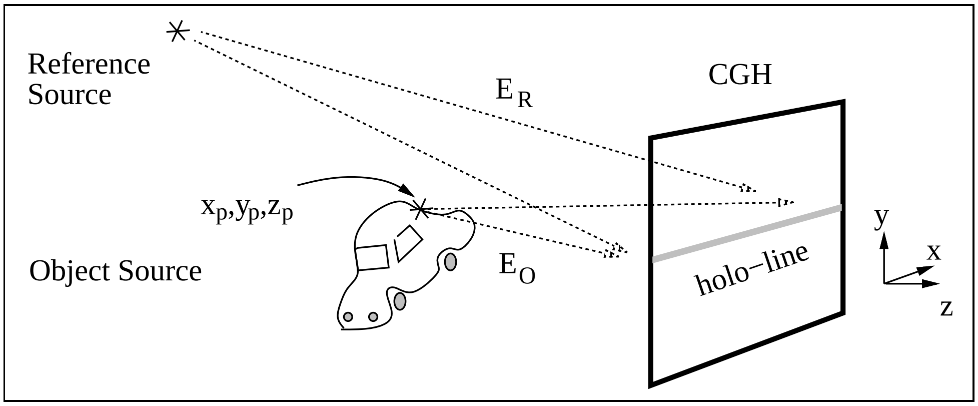

Figure 1: General geometry for HPO CGH.

The images to be generated are approximated as a collection of self-luminous points of light located in x, y, and z locations. Each point possesses a magnitude and a phase. The square of the magnitude is proportional to the desired brightness of an image point, and the phase is relative to the reference beam. Each point radiates a fan-shaped wavefront that is a horizontal slice of an anisotropic spherical wave. It is important to be able to assign a range of propagation angles for each point of image light in order to limit the spatial frequencies contained in the holographic fringe pattern. At one extreme, “light” used to compute the CGH must have an angle of incidence that is greater than that of the reference beam to prevent overlapping real and virtual reconstructed images (image aliasing). At the other extreme, the total angle subtended by the incident reference and object beams cannot be so large as to give rise to spatial frequencies that cannot be adequately represented by the discretized numerical representation. If the horizontal sampling pitch is physical distance d, then the maximum spatial frequency (f_{max}) that can be represented is $1/2d$, according to the Nyquist Sampling Theorem. Higher spatial frequencies cause aliasing, thus destroying image quality. Anti-aliasing is therefore accomplished by limiting the minimum and maximum angles of incidence of object light. In addition to anti-aliasing, the range of direction of object light propagation is used for the purposes of image occlusion and advanced image lighting models[8], resulting in a more realistic looking image.

The information content of a CGH must be reduced to a size and format that can be manipulated by existing computers. Consider a typical CGH pattern: it is composed of a large but finite 2D array of numbers representing the intensity of the computed total wavefront. The horizontal sampling pitch puts an upper limit on the maximum angle of diffraction of the CGH and consequently the maximum range of image viewing angles. Reducing the range of viewing angles reduces the required horizontal sampling rate and therefore the amount of data (space-bandwidth product) in the CGH. Furthermore, by reducing the size of the hologram (and therefore of the image), the data content of the CGH is as low as a few megabytes. These size reductions are an undesirable trade-off used only when no further information reduction is possible.

Quantization is also an important consideration. Each sample must represent an arbitrary physical value, but digital electronics commonly limit the number of quantization levels available when manipulating data. For example, the output device used in the current MIT system is a framebuffer capable of storing 6 megasamples, each represented by one byte (8 bits), giving the output data 256 possible quantized values. Therefore, a computed holographic interference pattern must be normalized to fit within this range. It is then quantized, increasing image noise due to the loss of accuracy. Quantization is also important when considering computation speeds, since less accurate representations of values (fewer bits) can be used to increase speed, but also sacrifice image quality.

COMPUTATION USING POINT SOURCE SUMMATION

In general, the physics of optical holography are as follows. The object light and the reference light are incident at the plane of the hologram. Each beam is represented with a complex time-harmonic electric field vector, E_O and E_R. It is assumed that both are mutually coherent sources of monochromatic light. For this analysis, the units of an electric field amplitude are normalized so that the square of its magnitude corresponds to optical intensity; the polarizations are assumed identical and for simplicity are not specified. The object beam E_O is generally a superposition of light scattered from locations throughout the object volume. The total time-harmonic electric field incident upon the hologram is $E_O + E_R$, which represents the interference of the total object light and the reference light. The resulting intensity pattern is

$$I_{TOTAL} = |E_O|^2 + |E_R|^2 + 2\Re e\{E_O E_R^*\} \tag{1}$$

and is a real physical light distribution comprised of three components. The second term is the reference beam intensity and represents an essentially constant or “DC” bias which increases the value of the intensity uniformly over the hologram. In computational holography, it can be left out, since normalization will subtract any DC bias present in the total holographic pattern. The first term is the object self-interference: a spatially varying pattern that is generated when interference occurs between light scattered from two or more object locations. During image reconstruction, this component of the holographic pattern is unnecessary and often produces unwanted image artifacts. In optical holography, a common solution is to spatially separate the self-interference artifacts from the desired image by increasing the reference beam angle to at least three times the angle subtended by the object. However, in computational holography, a large reference beam angle is a luxury that one does not have. Therefore, the obvious solution is to exclude this object self-interference term during computation. Finally, it is the third term that contains all of the necessary and useful holographic information, and is referred to as I_F.

The numerical computation of a holographic pattern is now examined, beginning with the simple physics of point-source light propagation. The hologram is positioned at the $z = 0$ plane, and has horizontal and vertical axes of x and y respectively. Each object point emits light from position (x_p, y_p, z_p). The fan-shaped object sources expose a limited width of a particular holographic line (“holo-line”). For the HPO CGH considered henceforth, an object point contributes only on the holo-line that is at the same vertical position ($y_p = y$). (To be more accurate, one must account for vertical foreshortening, absent due to the elimination of vertical parallax. Depending on specific display geometries, a more general image-point selection criterion is to include on holo-line y each point with $y_p + \mu(z_p - z_{view}) = y$, where z_{view} is the intended view distance from the hologram, and $\mu \equiv y/z_{view}$ is the slope of the path of light from the holo-line to the viewer.) Throughout the remainder of this discussion, the computation of a single holo-line is analyzed, and a full 2D CGH is computed simply by generating an array of holo-lines for each value of y. Only the x-dependence of E_O and E_R and other physical quantities need to be considered in computing a single holo-line.

For the purposes of computation, each object point is treated as an angularly truncated two-dimensional point source. Each has a complex amplitude of $A_p = a_p \exp(i\phi_p)$, where the real-valued magnitude is a_p and the real-valued relative phase of point source number p is ϕ_p. Within the region of contribution, the phase of the object wavefront, $\Phi_p(x)$, is approximated as a spherical wave[9] centered at the point source location:

$$\Phi_p(x) = k\, r_p(x) + \phi_p \quad \text{where} \quad r_p(x) = [(x - x_p)^2 + z_p^2]^{\frac{1}{2}}$$

where $r_p(x)$ is the oblique distance to a location on the holo-line and is a function of x. The wavenumber is $k = 2\pi/\lambda$, where λ is the free-space wavelength of the light. The time-harmonic representation of the total object field for a single holo-line is

$$E_O(x) = \sum_{p=1}^{N_{POINTS}} a_p(x)\, r_p^{-1}(x)\ \exp[i\Phi_p(x)] \tag{2}$$

where N_{POINTS} is the number of object points contributing to this particular y-valued holo-line. The added dependence of a_p on x facilitates anti-aliasing and occlusion simply by not including contributions outside of specific ranges of x. Finally, to avoid singularities, it is assumed that the magnitude of z_p is never less than some small amount, e.g., 10λ.

The reference beam E_R is a point source at some specific location ($x_R, y_R = y, z_R$) with a horizontal angle of incidence $\theta_R = \arctan(x_R/z_R)$ and curvature in the x dimension only, i.e., collimated in the y dimension. The time-harmonic representation of the reference beam field at any holo-line is

$$E_R(x) = a_R\ \exp[i\Phi_R(x)] \tag{3}$$

where a_R is the magnitude (assumed constant versus x) of the reference wave at the hologram plane and $\Phi_R(x) = k[(x - x_R)^2 + z_R^2]^{\frac{1}{2}}$. Note that all magnitudes and phases are real quantities.

BIPOLAR INTENSITY

The third term of Equation 1, called $I_F(x)$, contains all of the information needed to reconstruct the image in a given horizontal plane. Note that it is real-valued; it represents the combined intensity variations (“fringes”) resulting from each object point interfering with the reference beam. Since it contains negative values as well as non-negative values, it is a “bipolar intensity” which exists physically only when superimposed on the first two bias terms in Equation 1. Computationally, however, $I_F(x)$ can range both positive and negative since it is represented numerically, and is later offset during normalization.

The bipolar interference pattern $I_F(x)$ has the advantage of containing no object self-interference or bias components, and is numerically simpler to compute. $I_F(x)$ is further simplified:

$$\begin{aligned}I_F(x) &= 2\Re e\{\ [\ \sum_{p=1}^{N_{POINTS}} a_p(x)\, r_p^{-1}(x) \exp\{i\Phi_p(x)\}\ ]\ \ [\ a_R\ \exp\{i\Phi_R(x)\}\ ]^*\ \ \} \\ &= 2a_R \sum_{p=1}^{N_{POINTS}} \Re e\{\ a_p(x)\, r_p^{-1}(x) \exp[i\Phi_p(x) - i\Phi_R(x)]\ \} \\ &= 2a_R \sum_{p=1}^{N_{POINTS}} a_p(x)\, r_p^{-1}(x) \cos[\Phi_p(x) - \Phi_R(x)] \qquad (4)\end{aligned}$$

The right-hand side of Equation 4 is simply a scaled sum of the real-valued cosinusoidal fringe pattern resulting from the interference of point source p with the reference beam. Each of these constituent fringes is summed to obtain the full bipolar fringe pattern.

The advantages of this approach are readily seen by comparison to computation of the full interference pattern $I_{TOTAL}(x)$ which requires keeping track of both the real and imaginary parts of the object light. Each point requires a function call to both sine and cosine, and complex-value arithmetic must be used. In the bipolar intensity approach, the real-valued cosinusoidal fringes need simply to be summed to achieve the desired interference pattern. Each point requires only a single cosine function call. Therefore, a factor of two speed-up is expected.

A subtler advantage is revealed by considering numerical precision. The integer multiples of 2π spanned by $\Phi_p(x)$ must be calculated but are discarded when computing the cosine or the sine of the object light phase. Typically, $\Phi_p(x)$ is represented by a floating-point number composed of four 8-bit bytes possessing a precision of roughly 7 decimal digits. Values for $\Phi_p(x)$ often exceed 10^7 and must therefore make use of double-precision floating point representation, decreasing computation speed. In computing only the bipolar intensity component $I_F(x)$, $\Phi_R(x)$ (and any arbitrary integer) is first subtracted from $\Phi_p(x)$ before applying the cosine function, reducing the number of required significant digits; thus, the important fractional phase information is adequately represented with a single-precision floating-point expression.

After an intensity pattern has been computed, it must be normalized in order to satisfy the output device requirements of the CGH display system. Since normalizing generally scales the entire pattern, the leading factor of $2\, a_R$ on the right-hand side of Equation 4 is hereafter excluded. The reference beam intensity (the square of a_R) is no longer meaningful. This makes physical sense when considering that in optical holography, the purpose of choosing the ideal reference beam intensity ratio (relative to the object light) is to provide a sufficient DC offset and scaling to the interference fringes in order to keep them within the range of sensitivity of the recording medium. Computationally, offset and scaling are provided automatically during normalization. With the factor of $2\, a_R$ set arbitrarily to unity, (and substituting the definition of $\Phi_p(x)$) Equation 4 becomes

$$I_F(x) = \sum_{p=1}^{N_{POINTS}} a_p(x)\, r_p^{-1}(x)\ \ \cos[k\, r_p(x) - \Phi_R(x) + \phi_p] \qquad (5)$$

which is hereafter called the bipolar fringe method of CGH computation. No reference beam ratio needs to be specified during computation, and bias buildup is not an issue. Compare this bipolar intensity method to the physical process occurring in some photorefractive crystals[10] (e.g. lithium niobate), in which uniform ("DC") intensity is not recorded due to the material's negligible response to intensity patterns with low spatial frequencies. Researchers exploit this absence of bias build-up in order to sequentially expose multiple holographic intensity patterns.

PRECOMPUTED ELEMENTAL FRINGES: THE LOOK-UP TABLE APPROACH

Continuing with the bipolar intensity summation approach, further improvements in computation speed are gained through the use of precomputed look-up tables containing all possible elemental fringes. Consider a two-dimensional display which requires no computation (other than normalization and perhaps logarithmic correction) in order to display a two-dimensional image. This simple fact is due to the one-to-one correspondence between each image element and each display element, both often referred to ambiguously as a “pixel”. To illuminate a particular *image* pixel, simply display some non-zero value in the corresponding *display* pixel. In a three-dimensional holographic display, this simple correspondence between each image element and each display element does not exist. In this case, a 3D image element is a point of light in some (x, y, z) location with a brightness and relative phase. A display element corresponds to a numerical sample of a line of a holographic pattern modulating a beam of light. By determining how each possible image element relates to the display elements (holographic pattern), computation is reduced to a minimum.

Specifically, it is possible to precompute the contributions to $I_F(x)$ of an image point of unity magnitude for each possible value of (x_p, z_p). Since each holo-line is computed using the same $E_R(x)$, the precomputed tables are used in the computation of each holo-line. Rather than having to compute the cosinusoidal fringe each time it are needed, a large precomputed lookup table maps each (x_p, z_p) to the appropriate elemental fringe pattern contribution. To define these tables, Equation 5 is expanded.

$$I_F(x) = \sum_{p=1}^{N_{POINTS}} \{\, a_p(x)\, \cos\phi_p\; r_p^{-1}(x)\, \cos[k\, r_p(x) - \Phi_R(x)] \;+\; a_p(x)\, \sin\phi_p\; r_p^{-1}(x)\, \sin[\Phi_R(x) - k\, r_p(x)] \,\} \quad (6)$$

Other than the dependence of a_p on x, all spatial dependence of Equation 6 is in the following two expressions, used to define the two look-up tables:

$$\begin{aligned} \mathrm{TABLE_C}[x, X_i, Z_i] &= r_i^{-1}(x)\, \cos[k\, r_i(x) - \Phi_R(x)] \\ \mathrm{TABLE_S}[x, X_i, Z_i] &= r_i^{-1}(x)\, \sin[\Phi_R(x) - k\, r_i(x)] \end{aligned}$$

where $r_i(x) = [(x - X_i)^2 + Z_i^2]^{\frac{1}{2}}$. For both of these tables, the first index is x, which is already discretized due the the sampled representation of the CGH. However, image point positions (x_p, z_p) are not explicitly discretized, and must be rounded off to generate the X_i and Z_i source location indices. Before the tables are generated, the X_i and Z_i resolutions must be chosen in order to discretize the image volume. Since the acuity of the human visual system is limited, it is possible to chose resolutions that do not visibly degrade the image.

The first table looks like an array of cosinusoid-like fringes that have an approximately linear chirp in spatial frequency with respect to x. The rate of chirp is a function of the point source depth z_p, and the horizontal position of the fringe is a function of x_p. The second table is essentially the same but in quadrature to the first, i.e., with a $\pi/2$ phase difference, needed in order to represent any arbitrary point source phase. The dependence of a_p on x due to anti-aliasing is conveniently included in the two tables simply by leaving zeroes in all table locations in which there is no contribution. It is then assumed that any further variation in a_p will be dealt with during computation time, leaving a_p independent of x. Thus, $I_F(x)$, expressed in terms of precomputed tables, is

$$I_F(x) = \sum_{p=1}^{N_{POINTS}} \{\, (a_p \cos\phi_p)\, \mathrm{TABLE_C}[x, X_p, Z_p] \;+\; (a_p \sin\phi_p)\, \mathrm{TABLE_S}[x, X_p, Z_p] \,\} \quad (7)$$

Computation for a given holo-line at vertical position y is as follows:

- For every point with $y_p = y$ (i.e. on given scan-plane):
  - Round off (x_p, z_p) to $[X_p, Z_p]$ to index the desired elemental fringe.
  - For each x sample in $I_F(x)$:
    - Scale TABLE$_{\mathrm{C}}[x, X_p, Z_p]$ by $a_p \cos\phi_p$.
    - Scale TABLE$_{\mathrm{S}}[x, X_p, Z_p]$ by $a_p \sin\phi_p$.
    - Accumulate these scaled values in $I_F(x)$.

After each holo-line is computed (at each value of y), then normalizing and output is performed depending on the specific display system.

Computational Complexity

Computational complexity is dramatically reduced through the use of the two precomputed look-up tables. For a single object point contributing to a particular hologram point, the amount of computation required is two multiplications and two additions. In comparison, without the tables, computation involves a minimum of five additions, five multiplications, one square-root, and one cosine function call. (That is, using the bipolar fringe summation method and a precomputed $\Phi_R(x)$ known at each x.) Full complex computation of $I_{TOTAL}(x)$ would require still more computational steps, at least twice as many. Therefore, an order of magnitude of speedup is expected through use of the precomputed tables. Notice that the factors used to scale the table values (see Equation 7) are simply the real and imaginary parts of each point amplitude.

For further simplification, consider the case where all object points are to have the same relative phase. Only one table and half of the computation are needed, and an additional factor of two speed-up is obtained per point per holo-point. (One table can actually provide two object phases that are differing by π.) In practical holographic displays, arbitrary image point phase is often unnecessary since the individual image points may be non-overlapping, even at densities that give the appearance of continuous curves or surfaces.

Look-up Table Compaction

The greatest drawback to the precomputed table method is the enormous size of the tables. Essentially, memory requirements have been traded off against computational complexity. Consider the rather minimal dimensions of the first generation MIT real-time display system. The image volume of roughly 40 mm on each side is viewed from a distance of 600 mm. Using commonly accepted values for human visual acuity, as well as empirical tests performed using the display itself, the image volume should be discretized into approximately 250 horizontal positions and 50 depth positions. To provide a range of viewing angles of about 16 degrees, a holo-line contains 32 kilosamples (horizontal pitch of about one micron, wavelength of 632.8 nm). Assuming a standard four-byte representation, the two tables require over 3.2 gigabytes of memory! (Note that use of the bipolar intensity approach eliminates the additional necessity for the tables to contain both real and imaginary values and therefore reduces the table size by one half.)

Since the tables need only to be computed once for a given image volume and resolution, the use of cheaper less dynamic storage methods (e.g. PROM or EEPROM) may solve the size problem. However, hardware alterations may not be practical in most cases. Since many of the table entries are zeroes due to anti-aliasing, table size can be reduced by keeping track of the extent of the non-zero entries. However, in a data-parallel machine, this does not help, since memory allocation is generally identical for all processors. Given a limited memory capacity, how can the size of the two tables be reduced without reducing the speed of accessing the stored elemental fringe data? The best solution is to reduce the number of bits used to store each value. A common 4-byte representation contains 32 bits or 2^{32} quantization levels. This is clearly a waste of numerical precision since each value will be normalized and quantized to fit into an output device possessing far fewer quantization levels. Let us assume that the output device has 8-bit representation, as is commonly the case in high-resolution computer graphic display frame-buffers. In this case, a precomputed fringe need not be stored using more than 8 bits of memory.

For many practical applications, two bits can sufficiently represent the precomputed fringes in each of the two look-up tables. Consider the summation of many elemental fringes, summed in a manner that depends on specific object information. On the average, many points contribute to a sample of $I_F(x)$, which must be scaled down during normalization, effectively reducing the precision requirements of the look-up tables. For example, if a 32-bit integer (a sample of an elemental fringe) is ultimately scaled down by 2^{30} then its magnitude will span up to only two bits in the output device. The optimum case is where the tables contain values that are quantized to the same number of levels that they will occupy after being scaled down, re-quantized and written to the output device. Consider again the MIT real-time display system. Typically, the number of object points contributing to a particular sample of the hologram is at least $64 = 2^6$. If two-bit tables are used and 64 fringe elements are summed, then 256 is the maximum number of different values that this sample may contain, assuming an average point source amplitude of unity. As long as the image is sufficiently complex, table entries need only a four-level (two-bit) quantized version of the chirped cosinusoidal fringes.

A disadvantage to reducing the numerical precision of the elemental fringes is an increase in image noise. Quantization causes light power to be diffracted into the undesirable higher diffraction orders. Some fraction of the range of spatial frequencies that are intentionally computed diffract higher-order noise into the image volume. For example, the third harmonics of the spatial frequencies ranging from zero to $f_{max}/3$ diffract light into the image volume. Higher odd orders contribute diminishingly smaller amounts. If all of these higher orders are taken into account, then a signal-to-noise ratio (SNR) due to quantization can be calculated. Numerical analysis shows that by simply rounding the cosinusoidal fringes in a standard fashion to four evenly spaced levels gives a SNR of 148:1. However, by tailoring the levels coded by each bit and altering the thresholds during conversion from floating-point representation, the SNR can be made as high as 243:1. Using only a binary one-bit representation yields an unacceptably low SNR of 20:1.

Another obstacle is the inability of standard computers to deal with two-bit numbers during rapid computation of a CGH. The solution is to use the bits stored in the two tables not as numerical values but instead as Boolean representations of the particular table entry. Consider the one-table approach where $\phi_p = 0$. During computation, the two bits are indexed from $\mathrm{TABLE}_{\mathrm{C}}[x, X_p, Z_p]$ and then, if the low-order bit is one, 1/4 of the scale factor a_p is accumulated into $I_F(x)$. If the high-order bit is one, then 1/2 of the scale factor is accumulated into $I_F(x)$. Using this conditional approach, no time is wasted converting the two-bit value into standard integer or floating-point formats.

The real advantage of the two-bit method is that now the two tables can occupy as little as 1.6 gigabits of memory, or the equivalent of 200 megabytes, rather than 3.2 gigabytes. In addition, the use of the tables as boolean conditionals substitutes two additions for the two time-consuming multiplications. The computational complexity is now only four additions per object point per holo-point, and only two if the one-table method is used. Additional speed is expected.

Notice that the $r_p^{-1}(x)$ term cannot be included in the two-bit look-up tables since it requires a more continuous representation. However, this term can be approximated by z_p^{-1} and used to prescale a_p, resulting in a negligible decrease in speed.

Reduction of Table Rank from Three to Two

The precomputed tables are data arrays of rank three, indexed by x (on the hologram) and by the discretized values of x_p and z_p (in the image scan-plane). One way to reduce the size of the tables is to precompute values for each $\Delta_x \equiv (x - x_p)$ rather than for each x and each x_p. In this way, the tables are reduced to rank two. A simple restriction on the discretization step of x_p enables Equation 5 to be expressed as a function of Δ_x and z_p, exclusively.

In Equation 5, $r_p(x)$ is a function of $(x - x_p) = \Delta_x$ and z_p, leaving only $\Phi_R(x)$ as an explicit function of x. This must be manipulated into a function of Δ_x . First, by restricting the reference beam to be a plane-wave, the reference phase is simply $\Phi_R(x) = k_R\, x$, where $k_R = k\, \sin\theta_R$. The second restriction is that x_p be discretized by $2\pi/k_R$, making every possible value of $k_R\, \Delta_x$ differ by exactly $m2\pi$ for a given value of x , where m is some unimportant integer value. Consider that when computing the cosine or sine of total phase, any integer multiple of 2π is ignored. Therefore, $\Phi_R(x)$ can be expressed as $\Phi_R(x) = k_R\, x + m2\pi = k_R\, \Delta_x = \Phi_R(\Delta_x)$ for all discretized values of x_p. Finally, Equation 5 can be expressed as a function of only two variables Δ_x and z_p,

$$I_F(x) = \sum_{p=1}^{N_{POINTS}} a_p(x)\, r_p^{-1}(\Delta_x)\ \ \cos[k\,({\Delta_x}^2 + {z_p}^2)^{\frac{1}{2}} - k_R\Delta_x + \phi_p]$$

and corresponding look-up tables of rank two can be used. The indices are Δ_x and a discretized z_p.

Essentially, the elemental fringe patterns can now be moved in the x-direction. That is, they are shift-invariant in x for any X_p equal to an integer multiple of $k_R\, \Delta_x$. Clearly, the first advantage is a reduction in size equal to at least the number of discrete values of x_p used in the rank-three tables ($\sim$ 100 or more). The second advantage is that x_p is discretized in very small steps without increasing the size of the rank-two table. The horizontal image discretization step $2\pi/k_R$ is on the order of ten microns. For further flexibility, this step can be any integer multiple of $2\pi/k_R$. The only drawback to the rank-two tables approach is the requirement that the reference beam be a plane wave. However, this is a common case, and is applicable to many display architectures. In general, this contraction of rank is especially useful when implemented on a standard serial-processing computer.

Application to the Computation of Stereograms

For some applications, a stereogram[11] approach may be desirable when presenting three-dimensional images. In general, a stereogram consists of a series of two-dimensional object views differing in horizontal point-of-view. These views are presented to the viewer in the correct horizontal location, resulting in the depth cues of stereopsis and (horizontal) motion parallax. The 2D perspective views are generally imaged at a particular depth position, and are multiplexed by horizontal angle of view. A given holo-line in this case must contain a holographic pattern that diffracts light to each of the horizontal locations on the image plane. The intensity from a particular horizontal viewing angle should be the image intensity for the correct perspective view. This is accomplished simply by making the amplitude of the fringe contribution a step-wise x-function of the intensity of each image point from each of the views. To facilitate rapid computation of stereogram-type CGHs, the precomputed tables are indexed by image x-position and view-angle (rather than by x_p and z_p). Summation is performed as each of the perspective views is read into the computer. Furthermore, the tables can be indexed by Δ_x as described above, making the tables much smaller. Many stereogram CGHs have been computed and displayed on the MIT real-time holographic display system, producing realistic images computed utilizing sophisticated lighting and shading models and exhibiting occlusion and specular reflections.

RESULTS

Using the methods of bipolar intensity summation and precomputed elemental fringe patterns, hologram computation has been implemented for use in the MIT real-time display system. A Connection Machine Model 2 employs a data-parallel approach in order to perform real-time CGH computation. This means that each x location on the hologram is assigned to one of 32k virtual processors. (The 16k physical processors are internally programmed to imitate 32k "virtual" processors.) A Sun 4 workstation is used as a front-end for the CM2, and the parallel data programming language C Paris is used to implement holographic computation.

The following table contains the computation times (time per image point) using different approaches. "Full (complex) I_{TOTAL}" is the common general case where Equations 1, 2 and 3 are used to compute fully I_{TOTAL}. By eliminating the unnecessary interference terms and simplifying to obtain Equation 5, the calculation of the fringe pattern I_F is performed using the "Bipolar intensity" approach. The "Look-up tables" method employs two precomputed elemental fringe tables, represented in the memory of the CM2 as two bits per sample. As each object point is read from an input file, the position (x_p, z_p) is used to index the two tables in each processor. A conditional is performed on each of the four bits, and the appropriate fractions of either the real or imaginary parts of the object point amplitude are accumulated into the register representing $I_F(x)$. Since this is performed in parallel for all 32 kilosamples of $I_F(x)$, rapid computation of images is possible. A "Single look-up table" is used when object phase is not important.

Computation method	CM2	Sun4
Full (complex) I_{TOTAL}	2.180 ms	943.4 ms
Bipolar intensity	1.135 ms	486.2 ms
Look-up tables	0.174 ms	39.0 ms
Single look-up table	0.084 ms	22.1 ms

The time per point listed here is the amount of computation time required (on average) to accumulate the fringe pattern contribution of a single object point source. These numbers were obtained by computing holograms of several different test images of varying complexity. The computation time per point is simply the total execution time divided by the number of points processed. Despite the different image complexities (from 100 to 50,000 points), the time per point quotient remained within a 2% range; computation time scaled linearly with image complexity. For practical purposes, additional procedures that are independent of object complexity must be performed, including normalizing the computed holographic pattern and moving it into the frame-buffer. Therefore, a generally fixed overhead time must be added when expressing the total time to compute and output the holographic pattern. For a six megabyte holographic pattern, this time is approximately 0.4 seconds or less. For example, the actual time to compute a ten-thousand-point image using the single look-up table approach is $10,000 \times 0.084(ms) + 0.4(sec) = 1.24$ seconds, or less.

For comparison, the different computational approaches were also implemented on a serial computer, a Sun 4 workstation. Though computation times are much longer, the relative speed-up afforded by the "Look-up tables" approach is evident. While preserving full object phase generality, the look-up table approach is over 20 times faster than the full complex approach.

As expected, the bipolar approach is roughly twice as fast as the traditional complex method of computation. This is evident on both the parallel-data and serial machines. Moving to the look-up tables, the CM2 improves by about a factor of 7, and the serial machine improved by a larger factor of 12. The CM2 actually has an array of floating-point math accelerator chips which are no longer utilized in per point look-up table calculations. The serial machine, lacking the math co-processing capabilities which would speed up the non-look-up table approaches gains more from the use of the look-up tables. Finally, as expected, the use of a single table results in a speed-up of approximately two on both machines.

It must also be noted that the holographic patterns computed by these three approaches are equivalent, with the following exceptions. Use of the bipolar intensity eliminates object self-interference and DC terms, making the CGH brighter and less noisy than when using the full complex method. The look-up table approach results in a pattern that is identical to the bipolar intensity approach, with the addition of some quantization noise if two-bit tables are used. However, for objects of sufficient complexity, this quantization noise is comparable to that of the more straight-forward approaches, given the quantization of the output frame-buffer device used in the system.

In the current MIT display system, simple images generated from 3D computer graphics data-bases contain only a few thousand points. Using the fastest look-up table computation method, images are computed at a rate of over one frame per second. To demonstrate interactivity, the viewer can turn any of several dials (interfaced to the computer) in order to translate the image in horizontal, vertical, and depth locations, to change its size, and to spin it along different axes of rotation. In addition, a simple drawing program has been written in which the user can move a 3D cursor to draw a 3D image that can also be manipulated.

CONCLUSION

Experimental results demonstrate that a horizontal-parallax-only off-axis transmission hologram can be computed in times as low as one second. The overall speedup demonstrated here is remarkable. CGH computation that traditionally would require several minutes or hours on a mainframe computer, is reduced to one second. The look-up table approach, by eliminating the need for all mathematical functions other than simple addition, is especially advantageous when only minimal computing power is available for CGH computation. Given adequate memory space to hold the precomputed elemental fringes, it is possible to design a dedicated CGH computer that requires no floating-point mathematics and uses only integer addition (and perhaps bit-shifting for normalization purposes). Such a simple machine can be implemented in parallel, e.g. one computer per holo-line, in order to achieve real-time CGH computation without the need for an expensive supercomputer.

Analytical simplification of the physical model of light interference made possible this increase in speed. These concepts can be applied to other types of holograms. The bipolar intensity method is applicable to all types, including full-parallax CGHs. The use of the bipolar intensity summation method, whether directly or through look-up tables, eliminates object self-interference noise, eliminates the need to adjust a reference beam ratio, and produces an optimally bright image by eliminating unnecessary DC intensity bias. The look-up table approach can be applied to full-parallax holograms, although by requiring both x and y indices, precomputed tables (data-arrays of rank five, reduced to to rank three using the method shown here) would require enormous amounts of memory space. In the future, as computational power increases, the simplification of computation presented here will continue to provide speed-up for any CGH application.

ACKNOWLEDGMENTS

The author gratefully acknowledges the support of Prof. Stephen A. Benton, and researchers in the MIT Spatial Imaging Group past and present, especially John S. Underkoffler, Prof. Hiroshi Yoshikawa (Nihon University, Japan), Pierre St. Hilaire, and Wendy J. Plesniak. The author wishes also to acknowledge enlightening discussions with H. John Caulfield.

This research on computational electronic holography been supported in part by: the Defense Advanced Research Projects Agency (DARPA) through the Rome Air Development Center (RADC) of the Air Force System Command and the Naval Ordnance station, Indian Head, Maryland, contract #N00174-91-C-0117, and by the TV of Tomorrow consortium at the MIT Media Laboratory.

“Connection Machine” is a trademark of the supercomputer’s manufacturer, Thinking Machines, Inc., Cambridge, MA, USA.

References

[1] S. A. Benton. “Experiments in holographic video imaging”. In P. Greguss, editor, Holography, *Proceedings of the SPIE,* volume IS#08, pages 247–267, Bellingham, WA, 1991.

[2] P. St. Hilaire, S. A. Benton, M. Lucente, J. Underkoffler and H. Yoshikawa. “Real-time holographic display: Improvements using a multichannel acousto-optic modulator and holographic optical elements”. In Practical Holography V, *Proceedings of the SPIE,* volume 1461-37, pages 254–261, Bellingham, WA, 1991.

[3] P. St. Hilaire, S. A. Benton, M. Lucente, M. L. Jepsen, J. Kollin, H. Yoshikawa and J. Underkoffler. “Electronic display system for computational holography”. In Practical Holography IV, *Proceedings of the SPIE,* volume 1212-20, pages 174–182, Bellingham, WA, 1990.

[4] W. J. Dallas. Topics in applied physics. In B. R. Frieden, editor, The Computer in Optical Research, volume Vol. 41, chapter 6: "Computer-Generated Holograms", pages 291–366. Springer-Verlag, New York, 1980.

[5] D. Leseberg and C. Frere. “Computer-generated holograms of 3-D objects composed of tilted planar segments”. Applied Optics, 27(14):3020–3024, July 1988.

[6] D. Leseberg. “Computer-generated holograms: display using one-dimensional transforms”. Journal of the Optical Society of America, 3(5):726–730, May 1986.

[7] D. Leseberg and O. Bryngdahl. “Computer-generated rainbow holograms”. Applied Optics, 23(14):2441–2447, July 1984.

[8] J. S. Underkoffler. “Toward Accurate Computation of Optically Reconstructed Holograms”. Master’s thesis, Massachusetts Institute of Technology, June 1991.

[9] Joseph W. Goodman. Introduction to Fourier Optics. McGraw-Hill Book Company, New York, 1968.

[10] A. M. Glass. “The photorefractive effect”. Opt. Eng., 17(5):470–479, 1978.

[11] S. A. Benton. “Survey of holographic stereograms”. In Processing and Display of Three-Dimensional Data, *Proceedings SPIE,* volume 367, pages 15–19, Bellingham, WA, 1983.