Fast generation of Fresnel holograms based on multirate filtering

Peter Tsang,[1,*] Jung-Ping Liu,[2] Wai-Keung Cheung,[1] and Ting-Chung Poon[3]

[1]Department of Electronic Engineering, City University of Hong Kong,
83 Tat Chee Avenue, Kowloon, Hong Kong

[2]Department of Photonics, Feng Chia University, 100 Wenhwa Road,
Seatwen, Taichung 40724, Taiwan

[3]Bradley Department of Electrical and Computer Engineering,
Virginia Tech, Blacksburg, Virginia 24061, USA

*Corresponding author: eewmtsan@cityu.edu.hk

Received 13 July 2009; revised 31 August 2009; accepted 1 September 2009;
posted 2 September 2009 (Doc. ID 114226); published 17 September 2009

One of the major problems in computer-generated holography is the high computation cost involved for the calculation of fringe patterns. Recently, the problem has been addressed by imposing a horizontal parallax only constraint whereby the process can be simplified to the computation of one-dimensional sublines, each representing a scan plane of the object scene. Subsequently the sublines can be expanded to a two-dimensional hologram through multiplication with a reference signal. Furthermore, economical hardware is available with which sublines can be generated in a computationally free manner with high throughput of approximately 100 M pixels/second. Apart from decreasing the computation loading, the sublines can be treated as intermediate data that can be compressed by simply downsampling the number of sublines. Despite these favorable features, the method is suitable only for the generation of white light (rainbow) holograms, and the resolution of the reconstructed image is inferior to the classical Fresnel hologram. We propose to generate holograms from one-dimensional sublines so that the above-mentioned problems can be alleviated. However, such an approach also leads to a substantial increase in computation loading. To overcome this problem we encapsulated the conversion of sublines to holograms as a multirate filtering process and implemented the latter by use of a fast Fourier transform. Evaluation reveals that, for holograms of moderate size, our method is capable of operating 40,000 times faster than the calculation of Fresnel holograms based on the precomputed table lookup method. Although there is no relative vertical parallax between object points at different distance planes, a global vertical parallax is preserved for the object scene as a whole and the reconstructed image can be observed easily. © 2009 Optical Society of America

OCIS codes: 090.0090, 090.1995.

1. Introduction

The advancement of computing technology in the past two decades has enabled holograms to be generated numerically from three-dimensional (3-D) models that do not actually exist in the real world [1]. The process, commonly known as computer-generated holography, provides an alternative means to the traditional and often complicated way of capturing holograms of physical objects by optical means, which requires the use of laser beams and precision setup of optical equipment. Given a set of 3-D object points $O = [o_0(x_0, y_0, z_0), o_1(x_1, y_1, z_1), \ldots \ldots, o_{N-1}(x_{N-1}, y_{N-1}, z_{N-1})]$, the diffraction pattern $D(x, y)$ can be generated numerically by the following equation:

0003-6935/09/340H23-08$15.00/0

$$D(x,y) = \sum_{j=0}^{N_P-1} \frac{a_j}{r_j} \exp(ikr_j) = \sum_{j=0}^{N_P-1} [a_j \cos(kr_j) + ia_j \sin(kr_j)], \quad (1)$$

where a_j and r_j represent the intensity of the jth point in O and its distance to the position (x, y) on the diffraction plane, $k = 2\pi/\lambda$ is the wavenumber, and λ is the wavelength of the reference beam. Subsequently, a hologram $D(x, y)$ can be generated by adding a reference beam to the diffraction pattern. In addition, the availability of an electronically accessed spatial light modulator (SLM) and liquid crystal on silicon (LCOS) technology also allows computer-generated holograms (CGHs) to be displayed in real time without the need to produce hardcopies with photographic techniques or an expensive fringe writer [2,3].

There are two major issues associated with CGHs. First, computational loading in the generation of a hologram is extremely high even with the standard of modern computers. For a hologram with horizontal and vertical resolution of $X \times Y$ pixels, straightforward computation based on Eq. (1) involves $2 \times X \times Y \times N_P$ multiplication operations with the assumption that all the sinusoidal terms have been computed in advance and stored in lookup tables. Second, because of the high data rate of holographic signals, it is difficult to distribute or record them by means of existing video chains [4].

For a small hologram size and moderate number of object points, the first problem can be handled directly by the use of commodity hardware such as a graphic processing unit (GPU) [5,6] and a field programmable gate array (FPGA) [7,8]. However it is likely that the complexity and cost of the hardware will increase with the size and number of object points. To address the problem from a more fundamental and analytical perspective, Lucente [9] proposed the use of a precomputed lookup table at the expense of using an inordinate amount of memory to store the fringe patterns. To reduce the memory requirement, a novel lookup table was suggested in [10,11], which results in an approximately 30% reduction in computational load. Yamaguchi *et al.* suggested two methods for fast hologram generation. In the first, image holograms are generated in real time with the constraints that the scene objects must be positioned in close proximity to the hologram [12]. The second method generates horizontal parallax only (HPO) rainbow holograms [13–15] that overcome both of the aforementioned problems. In this approach a 3-D scene is first partitioned into a vertical stack of regularly spaced scan planes. Each scan plane contains a certain number of object points. These object points generate diffraction patterns on the hologram plane, and only the line with the same vertical height of the scan plane is taken into account. The line that contains the diffraction pattern of the object is called a subline and a fast algorithm is also available in [16]. Recently, Tsang *et al.* reported a low-cost solution based on a single, standalone, FPGA device to generate subline pixels with a throughput of 100 M pixels/second [17]. The speed is generally high enough for holographic video applications. The difference in FPGA solutions in [7,8] from those in [17] is that the latter were designed to generate sublines instead of complete holograms. As a result its complexity is substantially lower and can be easily extended to handle longer sublines simply by cascading multiple units. With each subline, a one-dimensional (1-D) hologram, referred to as a hololine, is then generated for each scan plane by adding an off-axis reference beam. A collection of these hololines becomes a full two-dimensional (2-D) hologram. The method is much faster than the direct generation of a Fresnel hologram, and the computation time can be further reduced by downsampling the number of sublines (i.e., increasing the separation between adjacent scan planes). The technology makes possible a holographic video system (for example, holographic television), in which the low data rate sublines can be transmitted to the receiver through ordinary communication channels and converted to holograms for reconstructing the original 3-D scene through special devices such as a SLM. Regardless of such successful technology, there are three major limitations. First, the method is designed to generate a rainbow or white light holograms that could cause color changes as the viewpoint is varied. Second, if the hologram is illuminated with a monochromatic beam, the viewing angle is narrow along the vertical direction. Third, the scan planes are separated by a specified distance so that each hololine includes an adequate number of rows in the hologram to support diffraction along the vertical direction. As a result, the resolution of the scene image will be lowered. In addition, it might be necessary to further decrease the number of scan planes to lower the data rate (if the sublines are distributed through certain channels) and computation load, resulting in even more degradation in the visual quality of the reconstructed images.

We propose a method based on the principles outlined in [15] for the generation of holograms from 1-D sublines that provide enhancement of the visual quality of the reconstructed image. In essence, a 3-D scene is partitioned into a horizontal stack of regularly spaced vertical subplanes, each contributed to a subline. Subsequently the sublines are expanded into a collection of diffraction patterns that are summed up and interfered with a reference beam to construct a complete hologram. To speed up the process, the conversion of sublines to diffraction patterns is encapsulated as a multirate filtering process.

In Section 2 we give a brief outline of the work in [15] as an introduction to the proposed method, which is presented in Section 3. We describe the formulation of a multirate filter to expand the sublines to the hologram and illustrate how the interpolation process can be incorporated into the architecture.

Evaluation of the computational loading and the enhancement attained with the interpolation of the sublines are described in Section 4, which is followed by a conclusion to summarize the essential findings.

2. Fast Generation of a White Light Hologram

A detailed description of the fast generation of white light holograms has been reported in [15]; only a brief outline is presented here. Figure 1 shows a block diagram of the idea.

A 3-D scene comprising a set of self-illuminated object points is first partitioned into a stack of L regularly spaced horizontal scan planes as shown in Fig. 2(a). A hologram comprising X columns and Y rows of pixels is to be generated from the object points. Since the hologram is assumed to be HPO, object points on a scan plane at vertical position $y = 0$ will contribute to a 1-D horizontal subline of object beam $O(x)$ given by

$$O(x) = \sum_{j=0}^{N-1} \frac{a_j}{r_j} \exp(ikr_j), \tag{2}$$

where $i = \sqrt{-1}$ and $k = 2\pi/\lambda$ is the wavenumber of the light with λ being the wavelength of the laser. a_j and $r_j = \sqrt{(x - x_j)^2 + z_j^2}$ represent the amplitude of the jth object point and its distance to the horizontal position x of the hologram, and N is the total number of object points.

On the hologram plane, i.e., the $x - y$ plane shown in Fig. 2(a), the subline interferes with reference beam $R(y)$ to form a hololine given by

$$h(x,y) = \mathrm{Re}[O(x)R*(y)] \quad \text{for} - \Delta y/2 \leq y \leq \Delta y/2, \tag{3}$$

where Δy is the separation of the scan plane as well as the height of a hololine. A complete 2-D HPO hologram is then computed by stacking up the array of hololines from each scan plane positioned at the corresponding vertical level of the latter. Depending on the distribution media, the spacing of the scan planes can be decreased to suit the channel bandwidth and resolution of the reconstructed images. A straightforward approach is to downsample the scan planes M times (a process often referred to as decimation within the context of multirate processing), resulting in a reduction of sublines as well as computation loading by equal proportion. An example with $M = 2$ is shown in Fig. 3.

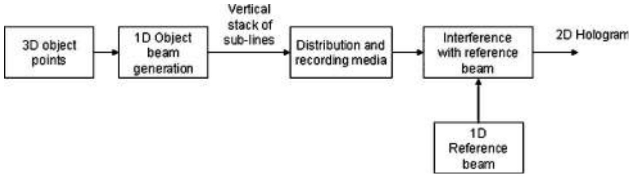
3D object points
1D Object beam generation
Vertical stack of sub-lines
Distribution and recording media
Interference with reference beam
2D Hologram
1D Reference beam

Fig. 1. Generation of HPO hologram based on [15].

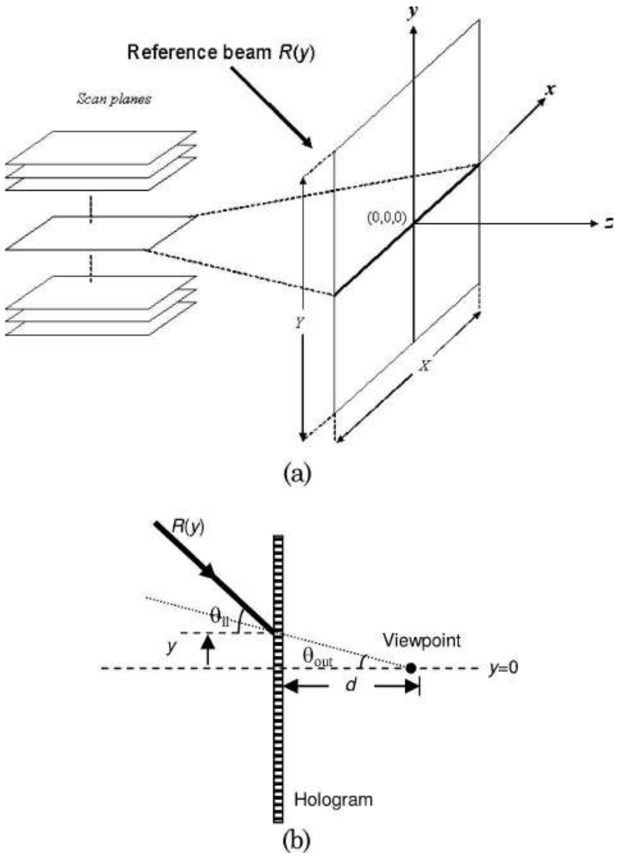

Fig. 2. (a) Generation of a 2-D hologram from scan planes and (b) side view of (a) without the scan planes for incident angle at θ_{ill}; d represents the distance between the hologram and the viewer.

It can be inferred from Fig. 3 as well as by illustration in the latter part of this paper that widening the separation between the scan planes with decimation will result in regularly spaced horizontal blank lines in the reconstructed images. Although the gaps can be filled simply by duplicating the neighboring sublines, the degradation in the visual quality is rather prominent to human perception and also escalates when the value of M increases.

3. Proposed Method for Generating Holograms from Sublines

The proposed method follows the principles in [15] that decompose the hologram formation process into a 1-D pair as shown in Fig. 1. However, the objective is to generate a Fresnel hologram instead of a white light hologram. Our method is outlined as follows. To begin, we expand Eq. (1) to object points located on a

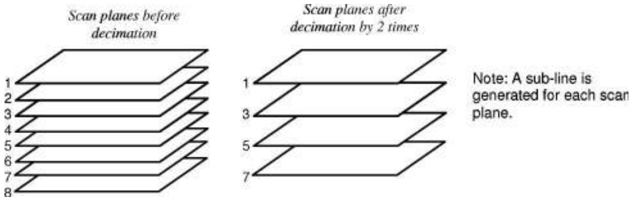
Scan planes before decimation
Scan planes after decimation by 2 times
1
2
3
4
5
6
7
8
1
3
5
7
Note: A sub-line is generated for each scan plane.

Fig. 3. Downsampling of scan planes by a factor of 2.

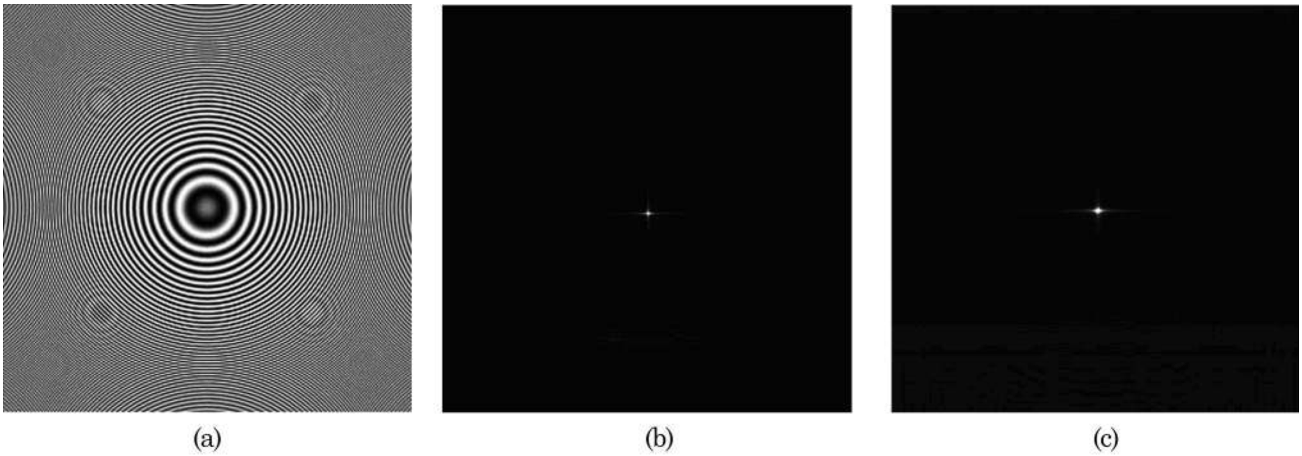

Fig. 4. (a) Diffraction pattern of a single point at $z_j = d/2 = -0.5\,\mathrm{m}$, (b reconstructed image with $z_j = d/2 = -0.5\,\mathrm{m}$, (c) reconstructed image with $z_j = -0.496\,\mathrm{m}$.

scan plane at $y = \tau$ [refer to Fig. 2(a) for definition of the coordinate system], yielding

$$D_\tau(x,y) = \sum_{j=0}^{N(\tau)} \frac{a_j}{r_j} \exp\left(ik\sqrt{(x-x_j)^2 + (y-\tau)^2 + z_j^2}\right), \tag{4}$$

where $N(\tau)$ is the number of object points on the scan plane at $y = \tau$. If z is sufficiently far away, a Fresnel approximation is derived and Eq. (4) can be rewritten as

$$\begin{aligned} D_\tau(x,y) &\approx \sum_{j=0}^{N(\tau)} \frac{a_j}{r_j} \exp\left(ik\frac{(x-x_j)^2 + (y-\tau)^2}{2z_j}\right) \\ &= \sum_{j=0}^{N(\tau)} \frac{a_j}{r_j} \exp\left(ik\frac{(x-x_j)^2}{2z_j}\right) \exp\left(ik\frac{(y-\tau)^2}{2z_j}\right). \end{aligned} \tag{5}$$

We further assume that the range of depth of the object points is small and centered at $z = z_o$, a condition that is common practice. Equation (5) can be simplified as

$$\begin{aligned} D_\tau(x,y) &\approx \sum_{j=0}^{N(\tau)} \frac{a_j}{r_j} \exp\left(ik\frac{(x-x_j)^2}{2z_j}\right) \exp\left(ik\frac{(y-\tau)^2}{2z_o}\right) \\ &= \exp\left(ik\frac{(y-\tau)^2}{2z_o}\right) \sum_{i=0}^{N(\tau)} \frac{a_j}{r_j} \exp\left(ik\frac{(x-x_j)^2}{2z_j}\right) \\ &= B(y-\tau)O(x). \end{aligned} \tag{6}$$

The approximation of Eq. (6) results in a change in aspect ratio of the reconstructed image of the scene object, but the deviation is not noticeable for the small range of z. The proposed algorithm in Eq. (6) is quite distinct from that used in Eq. (3). Each scan plane produces a diffraction pattern along x, and, at the same time, the hololine covers the entire y direction. In other words, along y there are overlapping patterns for the scan planes. This formalism, as we describe in Section 4, will save a tremendous amount of computation in comparison with that in the classical straightforward calculation of a Fresnel hologram, especially when the size of the hologram is large.

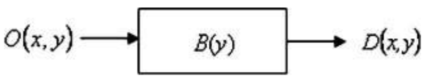

Fig. 5. Diffraction pattern generation encapsulated as a filtering process.

To illustrate the above, the diffraction pattern recorded at 632 nm wavelength and an image resolution of 20 μm of a single point source positioned at (0,0,–0.5 m) is shown in Fig. 4(a). The diffraction pattern interferes with a reference beam at an angle of $\theta = 1°$, and the simulated reconstructed image obtained with the Holovision [18] is shown in Fig. 4(b). The result with the point source moved to –0.496 m is shown in Fig, 4(c). It can be seen that the point is only slightly blurred. In practice the blurring is mainly along the vertical direction and not prominent for small differences between z_o and z_j.

Based on the above evaluation our method is presented as follows. To begin, object beam $O(x)$ in Eq. (6) is generalized to scan planes at other vertical positions such as

$$O(x,\tau) = \sum_{j=0}^{N(\tau)} \frac{a_j}{r_j} \exp\left(ik\frac{(x-x_j)^2}{2z_j}\right)\Bigg|_{y=\tau}. \tag{7}$$

For the time being we assume that there is no downsampling on the scan planes so that the scene image and the hologram have an identical vertical extent. In other words, the number of scan planes L is equal

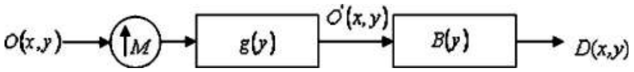

Fig. 6. Multirate filtering: interpolation of the missing gaps between sublines to eliminate the horizontal blank lines generated after upsampling.

Table 1. Breakdown of Number of Complex Multiplications for Each Step in Eq. (16)

Operation	Number of Complex Multiplications
$O(x,y) \to \tilde{O}(x,e^{iM\omega})$	$X \times Y/M\log_2(Y/M)$
$\tilde{O}(x,e^{iM\omega})E(e^{i\omega})$	$X \times Y$
$\tilde{D}(x,e^{i\omega}) \to D(x,y)$	$X \times Y\log_2(Y)$

to the number of hologram pixels Y along the vertical direction. Later we show that this assumption can be relaxed.

In Eq. (6), which describes the diffraction pattern on the hologram plane from a single scan plane, the contributions from all the scan planes can then be generalized to

$$D(x,y) = \sum_{\tau}[O(x,\tau)B(y-\tau)]. \tag{8}$$

If $L = Y$, Eq. (8) is recognized as a convolution process as shown in Fig. 5, which can be represented by

$$D(x,y) = O(x,y) * B(y), \tag{9}$$

where the $*$ denotes the convolution operation along the y direction. Equation (9) can be considered as a filtering process as shown in Fig. 5. Finally, the $D(x,y)$ interferes with an off-axis plane reference beam $R(y)$ to generate the off-axis Fresnel hologram as

$$H(x,y) = D(x,y)R*(y). \tag{10}$$

To produce the hologram on devices that present only amplitude information, such as photographic film or LCOS, only the real part of the hologram is retained as

$$H_R(x,y) = \mathrm{Re}[D(x,y)R*(y)]. \tag{11}$$

As mentioned previously, the number of sublines can be made smaller than the number of pixels along the vertical extent of the hologram to reduce the data rate. This can be accomplished simply by decimating the scan planes, and hence the resulting number of sublines by M times. However the process will lead to a difference in sampling rate between $O(x,y)$ and $B(y)$ so that the convolution process depicted in Eq. (9) is no longer applicable. To rectify the discrepancy, the downsampled sublines must be upsampled M times along the y direction prior to convolution. The upsampling process inserts $M-1$ blank lines between each pair of decimated sublines, resulting in empty gaps between the latter and degradation of the visual quality. However if the computation is posed as a convolution process in Eq. (10), the empty gaps can be suppressed with an interpolation filter prior to interfering with the reference beam as illustrated in Fig. 6, a concept of multirate filtering [19], which provides us with a computational efficiency and another advantage for the algorithm proposed in Eq. (6). In Fig. 6, $g(y)$ is the impulse response of the interpolation filter. Taking this into account, Eq. (9) can be rewritten as

$$D(x,y) = [O(x,y/M) * g(y)] * B(y) = O'(x,y) * B(y), \tag{12}$$

where $O'(x,y) = O(x,y/M) * g(y)$.

4. Fast Generation of Holograms

As mentioned in Section 1, sublines can be generated with throughput of 100 M pixels/second with the economical hardware described in [17]. Hence we can neglect computation loading in this process. However, direct rendering of $D(x,y)$ based on Eq. (10) is still computationally intensive. To overcome this problem the process is conducted in the frequency domain. Since y takes on the values of a sequence of real numbers, which corresponds to the different discretely distributed sublines, we can use the z transform along y. Hence, taking the z transform of Eq. (12) gives

$$\begin{aligned}\tilde{D}(x,\mathrm{z}) &= \tilde{O}'(x,\mathrm{z})\tilde{B}(\mathrm{z}) = \tilde{O}(x,\mathrm{z}^M)G(\mathrm{z})\tilde{B}(\mathrm{z})\\ &= \tilde{O}(x,\mathrm{z}^M)E(\mathrm{z}),\end{aligned} \tag{13}$$

where $\tilde{D}(x,\mathrm{z})$ and $\tilde{O}(x,\mathrm{z}^M)$ are the z transforms of $D(x,y)$ and $O(x,y/M)$, and $E(\mathrm{z}) = G(\mathrm{z})\tilde{R}*(\mathrm{z})$. Substituting $\mathrm{z} = e^{i\omega}$, Eq. (12) becomes

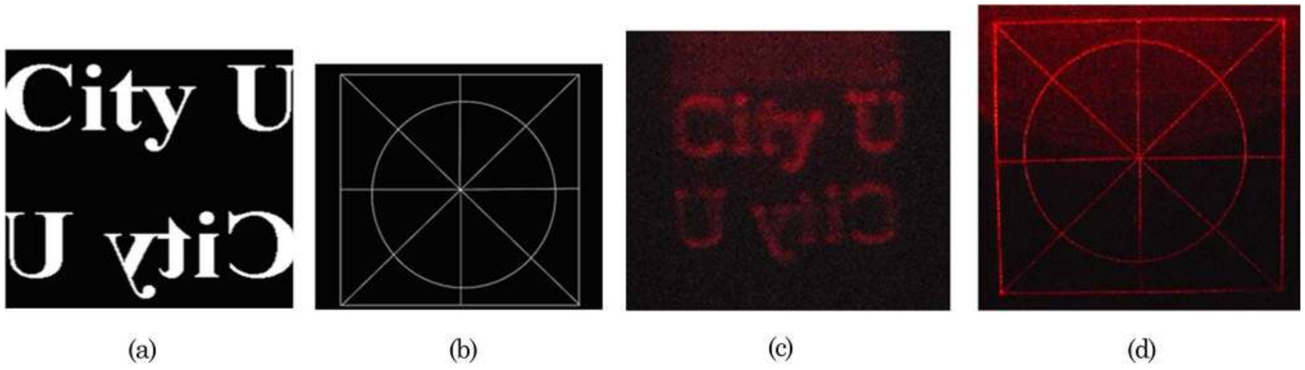

Fig. 7. (Color online) Test image of (a) CityU and (b) Star. Reconstructed image of (c) the CityU hologram and (d) the Star hologram.

Table 2. Setting for Generating Sublines and Holograms

Wavelength	650 nm	Image/hologram dimension of CityU	$X = Y = 1024$
Pixel size of hologram	$10.58\,\mu m \times 10.58\,\mu m$	Image/hologram dimension of Star	$X = Y = 2048$
Distance of object plane from hologram	40 cm	Illumination angle of reference beam (θ_{ill})	1.2 deg

$$\tilde{D}(x, e^{i\omega}) = \tilde{O}(x, e^{iM\omega})G(e^{i\omega})\tilde{R} * (e^{i\omega}) = \tilde{O}(x, e^{i\omega})E(e^{i\omega}). \tag{14}$$

$G(e^{i\omega})$ is basically a low-pass response to interpolate the empty space between consecutive sublines. Here we have adopted the Hamming filter with unit impulse response given by

$$g(y) = 0.54 + 0.46 \cos\left(\frac{2\pi y}{\text{len}}\right), \tag{15}$$

where len is the tap length of the filter that can be varied as a trade-off between the continuity and the resolution of the reconstructed image along the y direction.

In Eq. (14), $\tilde{D}(x, e^{i\omega})$, $\tilde{O}(x, e^{iM\omega})$, and $E(e^{i\omega})$ represent the Fourier transform (FT{.}) of $D(x, y)$, $O(x, y/M)$, and $E(y)$. Since $O(x, y/M)$ is simply the interpolation of $O(x, y)$, their Fourier transforms are similar and differ only by scaling in the frequency axis. In other words, $\tilde{O}(x, e^{iM\omega})$ can be deduced directly from $\tilde{O}(x, e^{i\omega})$. The diffraction pattern can be obtained from the inverse of $\tilde{D}(x, e^{i\omega})$ as

$$D(x, y) = \text{FT}^{-1}[\tilde{D}(x, e^{i\omega})] = \text{FT}^{-1}[\tilde{O}(x, e^{iM\omega})E(e^{i\omega})], \tag{16}$$

where $\text{FT}^{-1}\{.\}$ denotes the inverse Fourier transform operation. Since $E(e^{i\omega})$ can be computed in advance, to derive $D(x, y)$ it is only necessary to determine the Fourier transform of $O(x, y)$, the product of $\tilde{O}(x, e^{iM\omega})$ and $E(e^{i\omega})$, and the inverse Fourier transforms of $\tilde{D}(x, e^{i\omega})$. By use of the fast Fourier transform (FFT), the number of complex multiplications involved in each of the above steps for a diffraction pattern of size $X \times Y$ is listed in Table 1. For example, the transform of $O(x, y)$ to $\tilde{O}(x, y)$, denoted by $O(x, y) \to \tilde{O}(x, e^{iM\omega})$ in the left-hand column in Table 1, requires $X \times Y/M \log_2(Y/M)$ number of complex multiplications. We excluded the operations taken to generate sublines since the process could be realized with the solution given in [17].

Note that $\tilde{O}(x, e^{iM\omega})$ can be obtained by scaling the frequency axis of the Fourier transform of $O(x, y)$. As the latter is $1/M$ of the length of $O'(x, y)$, the number of arithmetic operations is reduced in comparison with the direct application of the FFT on $O'(x, y)$.

To complete the hologram an off-axis reference beam, $R(y)$, is added to the diffraction pattern [Eq. (10)], which is merely a point-by-point operation that involves additional $(X \times Y)$ operations. The total number of complex multiplications to render hologram $H(x, y)$, according to Eq. (16) and Table 1, is therefore

$$X \times Y\left[1 + \log_2 Y + \frac{1}{M}\log_2(Y/M)\right] + X \times Y = X \times Y\left[2 + \log_2 Y + \frac{1}{M}\log_2(Y/M)\right]. \tag{17}$$

Note that a complex multiplication involves four multiplication operations. In comparison with the straightforward generation of a Fresnel hologram based on the lookup table approach and assuming no decimation on the scan planes (i.e., $M = 1$) in our method, the computation time is reduced by a factor of

$$\frac{2 \times X \times Y \times N_P}{4\{X \times Y[2 + \log_2 Y + \log_2(Y)]\}} = \frac{N_P}{4[1 + \log_2(Y)]} \tag{18}$$

in comparison with the classical straightforward calculation of a Fresnel hologram.

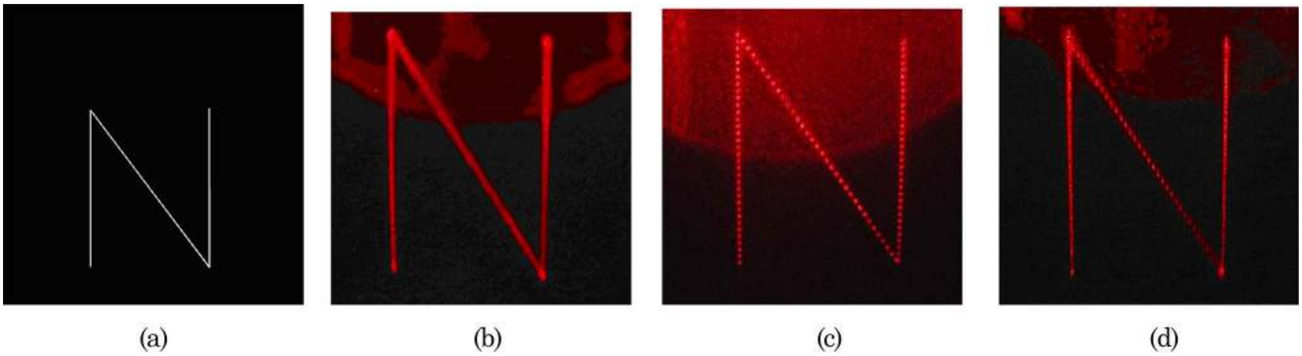

Fig. 8. (Color online) (a) Original object image and (b) reconstructed image. Reconstructed image for $M = 16$ (c) before and (d) after interpolation.

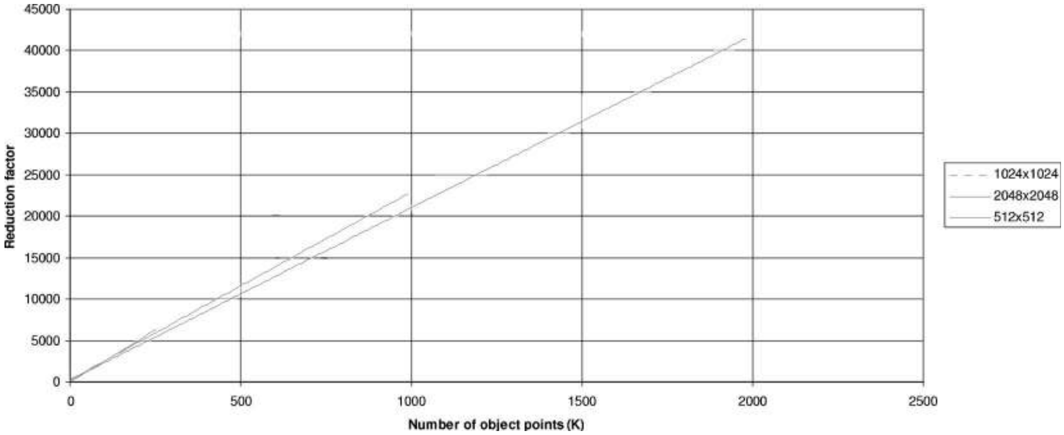

Fig. 9. (Color online) Reduction factor.

5. Evaluation of the Proposed Method

The test images, CityU and Star, shown in Figs. 7(a) and 7(b) are used to demonstrate the proposed method. For each sample image, the diffraction pattern is calculated with Eq. (16) and combined with a reference beam to obtain the hologram. The size of the image and its corresponding diffraction pattern and hologram are identical. Because of the constraint of the printing device, the real parts of the holograms are binarized and printed with the CreoScitex Dolev 800 V2 printer in the form of black-and-white fringe patterns on Agfa Red Sensitive film. The essential parameters involved in the hologram generation process are listed in Table 2. The reconstructed images are shown in Figs. 7(c) and 7(d). It can be seen that, apart from a partial overlap with the zero-order beam, the results are close to those obtained for the original images.

To illustrate the effect of downsampling and interpolation on sublines, the sample image in Fig. 8(a) is first converted into sublines and expanded to a 2-D diffraction pattern and hologram based on Eqs. (16) and (11), respectively. The reconstructed images before and after applying a downsampling factor of 16 on the number of sublines are shown in Figs. 8(b) and 8(c), respectively. It can be seen that, as a result of downsampling, the continuity of all the lines in the image in Fig. 8(c) is disrupted. Because of the limited resolution of the printing device and the camera that imposes a low-pass effect on the reconstructed images, the discontinuity appears to be less severe. After interpolation, the discontinuity of the vertical lines is removed as shown in Fig. 8(d). However, the interpolation is not applicable to the slant line since there is no adjacent pixel along the vertical direction for those sampled points.

Table 3. Reduction Factors for the Test Images in Figs. 7(a), 7(b), and 8(a)

Test Image	Figure	Object Points	Reduction Factor
CityU	7(a)	40220	914
Star	7(b)	30597	637
N	8(a)	26934	561

The reduction factor versus number of object points is computed based on Eq. (18) and illustrated in Fig. 9 for 512 × 512, 1024 × 1024, and 2048 × 2048 holograms. In all three cases the image and hologram size are identical, and the number of object points is less than or equal to that of the hologram. It can be seen that the reduction factor is almost linearly proportional to the number of object points. For a moderate 2048 × 2048 hologram, the reduction factor can be as high as 40,000 times. For the three test images shown in Figs. 7(a), 7(b), and 8(a), the number of object points and reduction factors are listed in Table 3.

6. Conclusion

Past research has demonstrated that the generation of a white light HPO hologram can be decomposed into two stages. Apart from speeding up the process, the method also enables hologram information to be converted to intermediate data in the form of 1-D sublines of the object wave. The latter, which can be decimated to reduce the data size, are recorded or distributed to the receiving ends by certain transmission channels and subsequently rendered to the full hologram by interfering with the reference beam. Conceptually, such an approach bears resemblance to a contemporary video broadcasting infrastructure in which pictorial data are first compressed to reduce the bit rate and reconstructed at the receivers. In addition, sublines can be generated in a computationally free manner by the use of a compact and economical hardware solution. The method, however, is not applicable to the generation of Fresnel holograms and is also relatively low in resolution. We have reported our investigation of generating an approximation of a Fresnel hologram by first converting sublines into diffraction patterns, followed by interference with a reference beam. Our method has three major aspects of improvement over the classical, straightforward generation of holograms. First, we propose that the conversion of object waves to diffraction patterns can be encapsulated as a multirate filtering process. Instead of applying convolution in a straightforward manner, we adopted the fast Fourier transform to deduce the output of the filter (i.e., the full hologram). This results in a saving in computational loading, and the amount of reduction increases significantly with the size of the hologram and the number of object points. Evaluation revealed that, for small holograms, our method is capable of calculating as much as 40,000 times faster than calculation with a Fresnel hologram based on the precomputed table lookup method. Second, the size of the intermediate data can be reduced by simply downsampling the number of sublines. However, if the decimation factor is too large, blank gaps will appear between the sampled lines. To suppress the artifacts, an interpolation filter is employed to reconstruct the missing information in the blank regions. Third, the transfer function of the interpolation filter and the frequency spectrum of the reference wave can be integrated into a single entity that can be precomputed in advance. In other words, the interpolation process does not impose any additional computation in the generation of the full hologram. On the downside, holograms generated with our method do not have relative vertical parallax for object points at different depths along the optical axis. However a global vertical parallax is preserved for the entire scene, which allows the reconstructed images to be observed easily along the vertical direction.

References

1. T.-C. Poon, ed., *Digital Holography and Three-Dimensional Display* (Springer-Verlag, 2006).
2. H. Yoshikawa and K. Takei, “Development of a compact direct fringe printer for computer-generated holograms,” Proc. SPIE **5290**, 114–121 (2004).
3. H. Yoshikawa and M. Tachinami, “Development of direct fringe printer for computer-generated holograms,” Proc. SPIE **5742**, 259–266 (2005).
4. T.-C. Poon, “Three-dimensional television using optical scanning holography,” Inf. Disp. **3**, 12–16 (2002).
5. T. Shimobaba, Y. Sato, J. Miura, M. Takenouchi, and T. Ito, “Real-time digital holographic microscopy using the graphic processing unit,” Opt. Express **16**, 11776–11781 (2008).
6. L. Ahrenberg, “Computer generated holography using parallel commodity graphics hardware,” Opt. Express **14**, 7636–7641 (2006).
7. Y. Seo, H. Cho, and D. Kim, “High-performance CGH processor for real-time digital holography,” in *Laser Applications to Chemical, Security, and Environmental Analysis*, OSA Technical Digest (CD) (Optical Society, 2008), paper JMA9.
8. T. Shimobaba, A. Shiraki, Y. Ichihashi, N. Masuda, and T. Ito, “Interactive color electroholography using the FPGA technology and time division switching method,” IEICE Electron. Express **5**, 271–277 (2008).
9. M. Lucente, “Interactive computation of holograms using a look-up table,” J. Electron Imaging **2**, 28–34 (1993).
10. S. C. Kim and E. S. Kim, “Effective generation of digital holograms of three-dimensional objects using a novel lookup table method,” Appl. Opt. **47**, D55–D62 (2008).
11. S. C. Kim and E. S. Kim, “Fast computation of hologram patterns of a 3D object using run-length encoding and novel lookup table methods,” Appl. Opt. **48**, 1030–1041 (2009).
12. T. Yamaguchi, G. Okabe, and H. Yoshikawa, “Real-time image plane full-color and full-parallax holographic video display system,” Opt. Eng. **46**, 125801 (2007).
13. H. Yoshikawa and H. Taniguchi, “Computer henerated rainbow hologram,” Opt. Rev. **6**, 118–123 (1999).
14. H. Yoshikawa and A. Kagotani, “Full color computer-generated rainbow hologram with enlarged viewing angle,” Opt. Rev. **9**, 251–254 (2002).
15. H. Yoshikawa, “Computer-generated holograms for white light reconstruction,” in *Digital Holography and Three-Dimensional Display: Principles and Applications*, T.-C. Poon, ed. (Springer-Verlag, 2006).
16. H. Yoshikawa, “Fast computation of Fresnel holograms employing difference,” Opt. Rev. **8**, 331–335 (2001).
17. P. Tsang, J. Liu, T.-C. Poon, and W. K. Cheung, “Fast generation of hologram sublines based on field programmable gate array,” in *Holography and Three-Dimensional Imaging*,OSA Technical Digest Series (Optical Society of America, 2009), paper Dwc2.
18. V. L. Tuft and S. Øystein , “Holovision 2.2 user’s manual,” http://www2.edge.no/projects/holovision/doc/holovision_221_manual.pdf, 2001.
19. P. P. Vaidyanathan, “*Multirate Systems and Filter Banks* (Prentice-Hall, 1993).