

# Holographic video at 40 frames per second for 4-million object points

Peter Tsang,<sup>1,\*</sup> W.-K. Cheung,<sup>1</sup> T.-C. Poon,<sup>2,3</sup> and C. Zhou<sup>3</sup>

<sup>1</sup>*Department of Electronic Engineering, City University of Hong Kong, Hong Kong*

<sup>2</sup>*Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA*

<sup>3</sup>*Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, P.O. Box 800-211, Shanghai 201800, China*

\*[ewmts@cityu.edu.hk](mailto:ewmts@cityu.edu.hk)

**Abstract:** We propose a fast method for generating digital Fresnel holograms based on an interpolated wavefront-recording plane (IWRP) approach. Our method can be divided into two stages. First, a small, virtual IWRP is derived in a computational-free manner. Second, the IWRP is expanded into a Fresnel hologram with a pair of fast Fourier transform processes, which are realized with the graphic processing unit (GPU). We demonstrate state-of-the-art experimental results, capable of generating a 2048x2048 Fresnel hologram of around  $4 \times 10^6$  object points at a rate of over 40 frames per second.

©2011 Optical Society of America

**OCIS codes:** (090.0090) Holography; (090.1995) Digital holography; (090.1760) Computer holography.

---

## References and links

1. T.-C. Poon, ed., "Digital holography and three-dimensional display: Principles and Applications," Springer (2006).
2. S. C. Kim and E. S. Kim, "Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods," *Appl. Opt.* **48**(6), 1030–1041 (2009).
3. S.-C. Kim and E.-S. Kim, "Effective generation of digital holograms of three-dimensional objects using a novel look-up table method," *Appl. Opt.* **47**(19), D55–D62 (2008).
4. S.-C. Kim, J.-H. Yoon, and E.-S. Kim, "Fast generation of three-dimensional video holograms by combined use of data compression and lookup table techniques," *Appl. Opt.* **47**(32), 5986–5995 (2008).
5. H. Sakata and Y. Sakamoto, "Fast computation method for a Fresnel hologram using three-dimensional affine transformations in real space," *Appl. Opt.* **48**(34 Issue 34), H212–H221 (2009).
6. T. Yamaguchi, G. Okabe, and H. Yoshikawa, "Real-time image plane full-color and full-parallax holographic video display system," *Opt. Eng.* **46**(12), 125801 (2007).
7. H. Yoshikawa, "Fast computation of Fresnel holograms employing difference," *Opt. Rev.* **8**(5), 331–335 (2001).
8. T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie, "Special-purpose computer HORN-5 for a real-time electroholography," *Opt. Express* **13**(6), 1923–1932 (2005).
9. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holography using parallel commodity graphics hardware," *Opt. Express* **14**(17), 7636–7641 (2006).
10. H. Kang, F. Yaraş, and L. Onural, "Graphics processing unit accelerated computation of digital holograms," *Appl. Opt.* **48**(34), H137–H143 (2009).
11. Y. Seo, H. Cho, and D. Kim, "High-performance CGH processor for real-time digital holography," *Laser App. Chem., Sec. and Env. Ana., OSA Tech. Digest (CD) (OSA, 2008)*, paper JMA9.
12. P. W. M. Tsang, J.-P. Liu, W. K. Cheung, and T.-C. Poon, "Fast generation of Fresnel holograms based on multirate filtering," *Appl. Opt.* **48**(34), H23–H30 (2009).
13. T. Shimobaba, H. Nakayama, N. Masuda, and T. Ito, "Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display," *Opt. Express* **18**(19), 19504–19509 (2010).

---

## 1. Introduction

Past research has demonstrated that the Fresnel hologram of a three-dimensional scene can be generated numerically by computing the fringe patterns emerged from each object point to the hologram plane. In brief, given a scene is of self-illuminating object

points  $O = [o_0(x_0, y_0, z_0), o_1(x_1, y_1, z_1), \dots, o_{N-1}(x_{N-1}, y_{N-1}, z_{N-1})]$ , the diffraction pattern  $D(x, y)$  on the hologram plane can be derived as

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

where  $a_j$  and  $r_j$  represents 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  $\lambda$  is the wavelength of the light. Although the method is effective, the computation involved in generating a hologram is extremely high. In the past lots of research attempts have been conducted to overcome the above problems, such as the works developed in [2- 12]. Recently, a fast method has been reported by Shimobaba et al. in [13]. In their approach, Eq. (1) is first applied to compute the fringe pattern of each object point within a small window on a virtual wavefront recording plane (WRP) which is placed very close to the scene. Subsequently, the hologram is generated from the WRP with Fresnel diffraction. However, as the number of object points increases, the time taken to derive the WRP will be lengthened in a linear manner and real-time generation of holographic video sequence is not possible. In this paper, a method to overcome the limitation in [13], is proposed. Essentially, we have formulated a novel, computation-free algorithm for generating what we call an interpolated WRP (IWRP). We then expand the IWRP into a Fresnel hologram. Experimental evaluation demonstrates that our proposed method is capable of generating a 2048x2048 hologram for an object scene with around  $4 \times 10^6$  object points in less than 25ms.

## 2. Background of the wavefront-recording plane (WRP) method

For clarity of explanation, a brief outline of the method in [13] is summarized in this section. To begin with, the following terminology is adopted. The hologram  $u(x, y)$  is a vertical, 2D image that is positioned at the origin. A virtual WRP,  $u_w(x, y)$ , is placed at a depth  $z_w$  from  $u(x, y)$ . The object scene is composed of a set of self-illuminating pixels, each having an intensity value of  $a_j$  and located at a perpendicular distance of  $d_j$  from the WRP. Without loss of generality we assume that both the hologram, the WRP, and the object scene have the same horizontal and vertical extents of  $X$  and  $Y$  pixels, respectively, as well as identical sampling pitch  $p$ . The hologram generation process can be divided into two stages. In the first stage, the complex wavefront contributed by the object points is computed as

$$u_w(x, y) = \sum_{j=0}^{N-1} \left( A_j / R_{wj} (x - x_j, y - y_j) \right) \exp \left( i \frac{2\pi}{\lambda} R_{wj} (x - x_j, y - y_j) \right), \quad (2)$$

where  $0 \leq x_j < X$  and  $0 \leq y_j < Y$  are the horizontal and vertical positions of the  $j$ th object point, and  $R_{wj}(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2 + d_j^2}$  is the distance of the point from the WRP. As the object scene is very close to the WRP, the diffracted beam of each object point is assumed to cover a small square window of size  $W \times W$  (hereafter refer as the virtual window). As such, Eq. (2) can be rewritten as

$$u_w(x, y) = \sum_{j=0}^{N-1} f_j, \quad (3)$$

$$\text{where } f_j = \begin{cases} \frac{A_j}{R_{w_j}(x, y)} \exp\left(i \frac{2\pi}{\lambda} R_{w_j}(x, y)\right) & \text{if } |x - x_j| \text{ and } |y - y_j| < \frac{1}{2}W \\ 0 & \text{otherwise} \end{cases}$$

In Eq. (3), the computation of the WRP for each object point is only confined to the region of the virtual window on the WRP. As  $W$  is much smaller than  $X$  and  $Y$ , the computation load is significantly reduced as compared with Eq. (2). In [13], the calculation is further simplified by pre-computing the exponential terms for all combinations of  $(x_j, y_j, d_j)$ , and the estimated computational amount is  $2\alpha N\bar{L}^2$ .  $\bar{L}$  is the mean perpendicular distance of the object points to the WRP, and  $\alpha$  is the arithmetic operations involved in computing the wavefront contributed by each object point. In the second stage, the WRP is expanded to the hologram as

$$u(x, y) = KF^{-1}\left[F\left[u_w(x, y)\right] \cdot F\left[h(x, y)\right]\right], \quad (4)$$

where  $F[\cdot]$  and  $F^{-1}[\cdot]$  denote the forward and inverse Fourier transform, respectively.  $K = -i/(\lambda z_w) \exp(i2\pi z_w/\lambda)$  is a constant and  $h(x, y) = \exp(i\pi(x^2 + y^2)/(\lambda z_w))$  is an impulse function which is fixed for a given separation  $z_w$  between the WRP and the hologram. In Eq. (4), the term  $F[h(x, y)]$  can be pre-computed in advance, and hence it is only necessary to compute the forward and an inverse Fourier transform operations. As reported in the article, these two processes can be conducted swiftly with GPU.

### 3. Proposed computational-free interpolated wavefront-recording plane (IWRP) method

Our proposed method is described as follows. First, we note that the resolution of the scene image is generally smaller than that of the hologram. Hence, it is unnecessary to convert every object point of the scene to its wavefront on the WRP. On this basis, we propose to sub-sample the scene image evenly by  $M$  times (where  $M > 0$ ) along the horizontal and the vertical directions. Let  $I(m, n)$  and  $d(m, n)$  represent the intensity and distance from the WRP of the sample object point located at the  $n$ th row and the  $m$ th column of the object scene. With the sample pitch set to  $Mp$ , the physical horizontal and vertical positions of the sample point are located at  $x_m = mMp + Mp/2$  and  $y_n = nMp + Mp/2$ , respectively, where  $m$  and  $n$  are integer values. A square support, as shown in Fig. 1a, is defined for each sample point, with the left side  $l_m$ , and the right side  $r_m$  given by  $l_m = x_m - Mp/2$ , and  $r_m = x_m + (M - 1)p/2$ . Similarly, the bottom side  $t_n$  and top side  $b_n$  of the square support are given by  $b_n = y_n - Mp/2$ , and  $t_n = y_n + (M - 1)p/2$ . We point out that the square supports of adjacent sample points are non-overlapping and just touching each other at their boundaries. Next, we assume the contribution of each sample point is contributing to a square virtual window in the WRP with side length equals to  $Mp$  as shown in Fig. 1b.

The virtual window is aligned with the square support of the object point, and the wavefront within the virtual window is only contributed by the object point in the square support. Under this approximation, Eq. (2) therefore can be re-written as

$$u_w(x, y) \Big|_{l_m \leq x < r_m, b_n \leq y < t_n} = I(m, n) \exp\left(i2\pi R_{d(m, n)}(x - x_m, y - y_n)/\lambda\right), \quad (5)$$

where  $R_{d(m,n)}(x, y) = \sqrt{x^2 + y^2 + d(m,n)^2}$  is the distance of the sample object point to the location  $(x, y)$  on the WRP. As such, the right hand side of Eq. (4) can be further encapsulated into a single function of two independent variables ( $d(m,n)$  and  $I(m,n)$ ) as given by

$$u_w(x, y) = I(m, n) \exp(i2\pi R_{d(m,n)}(x - x_m, y - y_n) / \lambda) = G(x - x_m, y - y_n, I(m, n), d(m, n)) \quad (6)$$

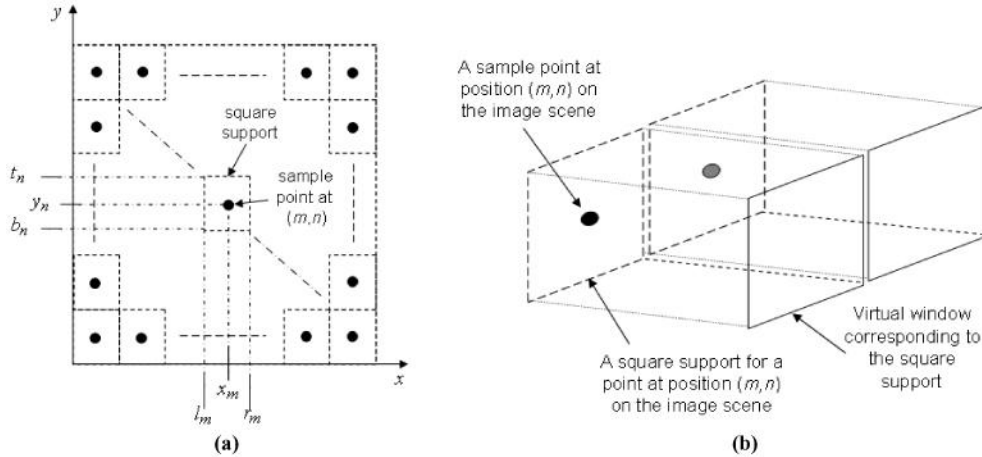


Fig. 1a. Sampling lattice and square support Fig. 1b. A pair of sample points, each associate with a square support and a virtual window on the scene image and the WRP, respectively. Note that the depth (i.e.,  $z$  position) of the object points are not shown in the diagram.

It can be inferred from Eq. (6) that the function  $G(x - x_m, y - y_n, I(m, n), d(m, n))$  represents a Fresnel zone plate  $G(x, y, I(m, n), d(m, n))$ , within the virtual window, which is shifted to the position  $(x_m, y_n)$ . The Fresnel zone plate is contributed by an object point of intensity  $I(m, n)$  and at distance  $d(m, n)$  from the wavefront plane. Hence for a finite variations of  $d(m, n)$  and  $I(m, n)$ , all the possible combinations of  $G(x, y, I(m, n), d(m, n))$  can be pre-computed in advance, and store in a look up table (LUT). For example, if the depth  $d(m, n)$  and  $I(m, n)$  are quantized into  $N_d$  and  $N_I$  levels, respectively, there will be a total of  $N_d \times N_I$  combinations. As a result, in the generation of  $u_w(x, y)$ , each of its constituting virtual window can be retrieved from the corresponding entries in the LUT. In another words, the process is computational-free.

Although the decimated has effectively reduced the computation time, as will be shown later, the reconstructed images obtained with the WRP derived from Eq. (6) are weak, noisy, and difficult to observe. This is caused by the sparse distribution of the object points caused by the sub-sampling of the scene image. To overcome this problem, we propose the interpolated WRP (IWRP) to interpolate the associated support of each object point with padding, i.e., the object point is duplicated to all the pixels within each square support. After the interpolation, the wavefront of a virtual window will be contributed by all the object points (which are identical in intensity and depth) within the support as given by

$$\begin{aligned} u_w(x, y) \Big|_{l_m \leq x < r_m, t_n \leq y < b_n} &= I(m, n) \sum_{\tau_x = -\frac{M}{2}}^{\frac{M}{2}-1} \sum_{\tau_y = -\frac{M}{2}}^{\frac{M}{2}-1} \exp(i2\pi R_{d(m,n)}(x - x_m + \tau_x, y - y_n + \tau_y) / \lambda) \quad (7) \\ &= G_A(x - x_m, y - y_n, I(m, n), d(m, n)) \quad (8) \end{aligned}$$

Similar to Eq. (6), the wavefront function  $= G_A(x - x_m, y - y_n, I(m, n), d(m, n))$  in Eq. (8) is simply a shifted version of  $G_A(x, y, I(m, n), d(m, n))$  which can be pre-computed and stored in a LUT for different combinations of  $I(m, n)$  and  $d(m, n)$ . Consequently, each virtual window in the IWRP can be generated in a computation-free manner by retrieving, from the LUT, the wavefront corresponding to the intensity and depth of the corresponding object point. Comparing Eq. (6) and Eq. (8), it can also be inferred that the number of combination on the values of  $G(x, y, I(m, n), d(m, n))$  and  $G_A(x, y, I(m, n), d(m, n))$ , and hence the size of the corresponding LUTs, are identical. After  $u_w(x, y)|_{t_m \leq x < r_m, t_n \leq y < b_n}$  is generated within the IWRP, Eq. (4) is applied to generate the hologram  $u(x, y)$ .

#### 4. Experimental results

Our proposed method is evaluated with the test image Fig. 2a. The horizontal and vertical extents of the hologram, the IWRP, and the test image are identical, comprising of 2048 by 2048 square pixels each with a size of 9 by 9  $\mu\text{m}$  and quantized with 8bits. The test image is divided into a left and a right part located at distances of  $z_1 = 0.005\text{m}$  and  $z_s = 0.01\text{m}$  from the IWRP. Each pixel in the image is taken to generate the hologram, constituting to a total of around  $4 \times 10^6$  object points.  $\lambda$  and the distance  $z_w$  between the WRP/IWRP and the hologram are set to  $650\text{nm}$  and  $0.4\text{m}$ , respectively.  $N_d$  and  $N_d$  are both set to 256, and  $M = 8$ , resulting in a LUT of around 4.1Mb. We decimated the source image by 8 times in the horizontal and vertical direction (i.e.,  $M = 8$  and size of virtual window =  $8 \times 8$ ), and applied Eqs. (5) and 6 to derive a WRP. The latter is then expand it into a hologram with Eq. (4). A real, off-axis hologram  $H(x, y)$  is generated by adding a planar reference wave  $R(y)$  (illuminating at an inclined angle  $1.2^\circ$  on the hologram) to  $u(x, y)$ , and taking the real part of the result as given by

$$H(x, y) = RE[u(x, y) \cdot R(y)], \quad (9)$$

where  $RE[\cdot]$  denotes the real part of a complex variable. The hologram is displayed on a liquid crystal on silicon (LCOS) modified from the Sony VPL-HW15 Bravia projector. The projector has a horizontal and vertical resolution of 1920 and 1080, respectively. Due to the limited size and resolution of the LCOS only part of the hologram (and hence the reconstructed image) can be displayed. The reconstructed images corresponding to the upper half and the lower half of the hologram are shown in Figs. 2b and 2c, respectively. We observe that the images are extremely weak and noisy. Next we repeat the above process by generating the IWRP with Eqs. (7) and 8. The reconstructed images are shown in Figs. 2d and 2e. Evidently, the reconstructed image is much clearer in appearance. To further illustrate our proposed method, we have generated a sequence of holograms of a rotating globe which is rendered with the texture of the earth image. The radius of the globe is around  $0.005\text{m}$ , and the front tip of the globe is located at  $0.01\text{m}$  from the IWRP. The latter is at a distance of  $0.3\text{m}$  from the hologram. A single frame excerpt of the optical reconstructed animation clip ([Media 1](#)) is shown in Fig. 2f. It can be seen from the excerpt, as well as in the animation clip that, despite the complexity of the texture, the earth image on the globe is clearly reconstructed in every views. Next, we evaluate the computation efficiency of our proposed method. The IWRP, and its subsequent expansion to a Fresnel hologram are conducted with the PC (intel i7-950 @ 3.06GHz) and the GPU (Nvidia Geforce GTX580), respectively. The total hologram generation time and equivalent frame-rate (measure in *fps* representing the number of hologram frames per second), versus the number of object points, are shown in Table 1.

We have assumed that the number of object points and hologram pixels are identical. From the result, it can be seen that the hologram generation time is very short as it only involves table lookup and data transfer between memory arrays. For a hologram (as well as image size) of 2048x2048 pixels, our proposed method is capable of attaining a generation speed of over 40 frames per second.

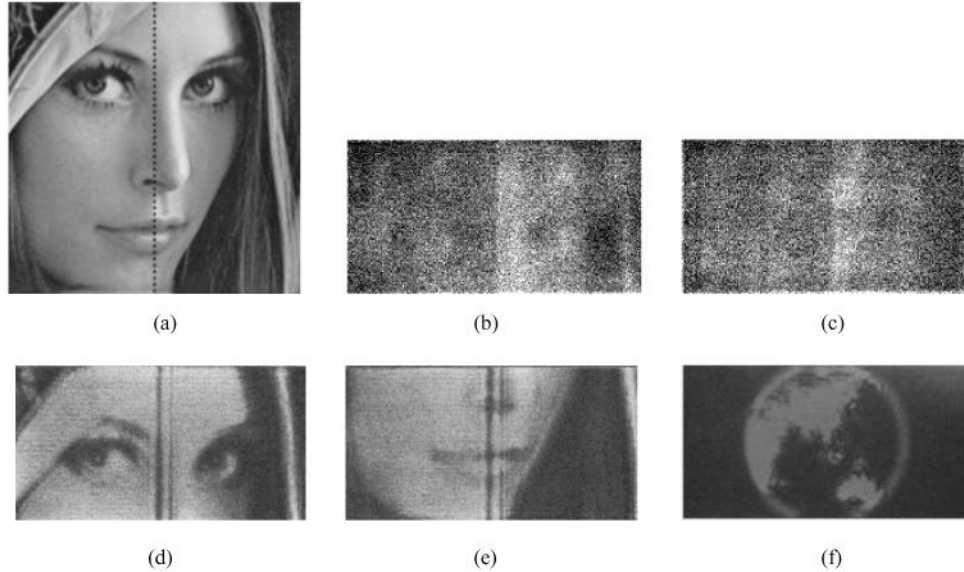


Fig. 2a. An image divided (dotted line) into a left and right parts, positioned at  $z_1 = 0.005\text{m}$  (left) and  $z_2 = 0.01\text{m}$  (right) from the IWRP, respectively. The IWRP is positioned at  $z_w = 0.4\text{m}$  from the hologram. Fig. 2b. Optical reconstructed image of the upper part of the hologram corresponding to the WRP derived from Eqs. (5) and 6. Fig. 2c. Optical reconstructed image of the lower part of the hologram corresponding to the WRP derived from Eqs. (5) and 6. Fig. 2d. Optical reconstructed image of the upper part of the hologram corresponding to the proposed IWRP derived from Eqs. (7) and 8. Fig. 2e. Optical reconstructed image of the lower part of the hologram corresponding to the proposed IWRP derived from Eqs. (7) and 8. Fig. 2f. Single-frame excerpts from the optical reconstructed animation clip of the hologram sequence representing a rotating earth globe located at 0.3m from the hologram. The hologram sequence is generated from the IWRP derived from Eqs. (7) and (8) (Media 1).

**Table 1. Computation Time and Equivalent Frame Rate in Generating the IWRP, and Expanding the Latter to a Fresnel Hologram**

Hologram size	No. of object points	IWRP generation (ms)	IWRP to hologram (ms)	Computation time (ms)	Frame-rate (fps)
1024x1024	$10^6$	2.2	4	6.2	161.2
2048x2048	$4.2 \times 10^6$	8.6	16	24.5	40.8
4096x4096	$1.7 \times 10^7$	30.4	47	77.4	12.9

## 5. Conclusion

In this paper, we propose a method for real-time generation of Fresnel holograms. A proposed interpolated wavefront recording plane (IWRP) is first constructed with a computation-free process. Subsequently, the IWRP is expanded into a Fresnel hologram via a pair of fast Fourier transform operations that are realized with the GPU. Based on our method a hologram size of 2048x2048, representing an image scene comprising of over  $4 \times 10^6$  points, can be generated in less than 25ms, equivalent to 40 frames per second. These results correspond to state-of-the-art speed in the calculation of CGH.

**Acknowledgments**

The work is partly supported by the Chinese Academy of Sciences Visiting Professorships for Senior International scientists. Grant Number: 2010T2G17