

High-speed full analytical holographic computations for true-life scenes

Yuan-Zhi Liu^{1,2}, Jian-Wen Dong^{1,2}, Yi-Ying Pu¹, Bing-Chu Chen¹, He-Xiang He¹, and He-Zhou Wang^{1*}

¹State Key Laboratory of Optoelectronic Materials and Technologies,
Zhongshan (Sun Yat-sen) University, Guangzhou 510275, China

²These authors contributed equally to this work.

*stswzhz@mail.sysu.edu.cn

Abstract: We develop a novel method to generate hologram of three-dimensional (3D) textured triangle-mesh-model that is reconstructed from ordinary digital photos. This method allows *analytically* encoding the 3D model consisting of triangles. In contrast to other polygon based holographic computations, our full analytical method will free oneself from the numerical error that is in the angular spectrum due to the Whittaker-Shannon sampling. In order to saving the computation time, we employ the GPU platform that is remarkably superior to the CPU's. We have rendered a true-life scene with colored textures as the first demo by our homemade software. The holographic reconstructed scene possesses high performances in many aspects such as depth cues, surface textures, shadings, and occlusions, etc. The GPU's algorithm performs hundreds of times faster than those of CPU.

©2010 Optical Society of America

OCIS codes: (090.0090) Holography; (090.1760) Computer holography; (090.2870) Holographic display.

References and links

1. K. Iizuka, "Welcome to the wonderful world of 3D: introduction, principles and history," *Opt. Photon. News* **17**(7), 42–51 (2006).
2. See for example, Stanford 3D Scanning Repository, and XYZ RGB Inc.
3. J. L. Zhao, H. Z. Jiang, and J. L. Di, "Recording and reconstruction of a color holographic image by using digital lensless Fourier transform holography," *Opt. Express* **16**(4), 2514–2519 (2008).
4. C. Slinger, C. Cameron, and M. Stanley, "Computer-generated holography as a generic display technology," *Computer* **38**(8), 46–53 (2005).
5. J. W. Goodman, *Introduction to Fourier optics* (Roberts & Company Publishers, 2004).
6. M. Lucente, "Interactive Computation of Holograms Using a Look-Up Table," *J. Electron. Imaging* **2**(1), 28–34 (1993).
7. 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).
8. D. Leseberg, and C. Frère, "Computer-generated holograms of 3-D objects composed of tilted planar segments," *Appl. Opt.* **27**(14), 3020–3024 (1988).
9. N. Delen, and B. Hooker, "Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: a fast Fourier transform approach," *J. Opt. Soc. Am. A* **15**(4), 857–867 (1998).
10. K. Matsushima, H. Schimmel, and F. Wyrowski, "Fast calculation method for optical diffraction on tilted planes by use of the angular spectrum of plane waves," *J. Opt. Soc. Am. A* **20**(9), 1755–1762 (2003).
11. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holograms from three dimensional meshes using an analytic light transport model," *Appl. Opt.* **47**(10), 1567–1574 (2008).
12. H. Kim, J. Hahn, and B. Lee, "Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography," *Appl. Opt.* **47**(19), D117–D127 (2008).
13. E. Lalor, "Conditions for the Validity of the Angular Spectrum of Plane Waves," *J. Opt. Soc. Am. A* **58**(9), 1235–1237 (1968).
14. J. García, D. Mas, and R. G. Dorsch, "Fractional-Fourier-transform calculation through the fast-Fourier-transform algorithm," *Appl. Opt.* **35**(35), 7013–7018 (1996).
15. D. Mendlovic, Z. Zalevsky, and N. Konforti, "Computation considerations and fast algorithms for calculation the diffraction integral," *J. Mod. Opt.* **44**, 407–414 (1997).
16. D. Mas, J. Garcia, C. Ferreira, L. M. Bernardo, and F. Marinho, "Fast algorithms for free-space diffraction patterns calculation," *Opt. Commun.* **164**(4-6), 233–245 (1999).

17. K. Matsushima, and T. Shimobaba, "Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields," *Opt. Express* **17**(22), 19662–19673 (2009).
18. M. Zhang, T. S. Yeo, L. W. Li, and Y. B. Gan, "Parallel FFT Based Fast Algorithms for Solving Large Scale Electromagnetic Problems," in *IEEE Antennas and Propagation Society International Symposium*, pp. 3995–3998 (2006).
19. J. Watlington, M. Lucente, C. Sparrell, V. M. Bove, and J. Tamitani, "A Hardware Architecture for Rapid Generation of Electro-Holographic Fringe Patterns," *Proc. SPIE* **2406**, 172–183 (1995).
20. C. Petz, and M. Magnor, "Fast Hologram Synthesis for 3D Geometry Models using Graphics Hardware," *Proc. SPIE* **5005**, 266–275 (2003).
21. V. M. Bove, Jr., W. J. Plesniak, T. Quentmeyer, and J. Barabas, "Real-Time Holographic Video Images with Commodity PC Hardware," *Proc. SPIE Stereoscopic Displays and Applications* **5664**, 255–262 (2005).
22. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holography using parallel commodity graphics hardware," *Opt. Express* **14**(17), 7636–7641 (2006).
23. NVIDIA, "NVIDIA CUDA Compute Unified Device Architecture Programming Guide Version 2.1," (2008).
24. R. N. Bracewell, K.-Y. Chang, A. K. Jha, and Y.-H. Wang, "Affine theorem for two-dimensional Fourier transform," *Electron. Lett.* **29**(3), 304–306 (1993).
25. "Autodesk® ImageModeler™," URL <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=11390028>
26. J. Hahn, H. Kim, Y. Lim, G. Park, and B. Lee, "Wide viewing angle dynamic holographic stereogram with a curved array of spatial light modulators," *Opt. Express* **16**(16), 12372–12386 (2008).

1. Introduction

3D imaging attracts increasing attention nowadays [1]. 3D data for real existent or nonexistent objects can be obtained more conveniently since the information technology made great progress during the past decades [2,3]. They have been widely used in games, movies, virtual reality, reverse engineering, etc. However, the existing two-dimensional display limits the performance. One of the promising technologies for true 3D display is computer generated holograms (CGHs) [4]. It can use the prescribed 3D object data stored in computer to produce wavefronts with amplitude and phase distribution for creating the most accurate depth cues of 3D objects [5].

However, the holographic computation is time consuming. Researchers have developed fast algorithms to accelerate the calculation [6,7]. These methods treat 3D models as a combination of individual self-luminous points. The object wave distribution on the hologram is calculated for each point and then superimposed. It requires a large number of points to achieve the solid effect, and enormous memory size for sampling [7]. Alternatively, 3D objects may be constructed of planar segments whose elements number is much less than that of point sets. In this way, the time will decrease significantly. In these methods, the numerical algorithm- fast Fourier transform (FFT), with the extra interpolated should be employed for each plane [8–10]. Recently, a triangle-mesh-based algorithm is proposed to further reduce the computational time [11,12]. The analytical angular spectrum of tilted triangle is calculated in frequency domain and a single FFT is required for the whole wave field. However, the uses of FFT will introduce numerical errors when the object goes beyond a certain distance. This limitation comes from the Whittaker-Shannon sampling theorem [5,13–17]. As a result, the diffraction field of the triangular aperture becomes noisy when the object moves away from the hologram [17]. Such noises will mainly affect the surface performances (e.g. introducing additional artifacts) in the holographic reconstruction. The additional artifacts will bring a blur defocus feeling and weaken the quality of the reconstruction scene. In addition, if the hologram becomes large, the huge number of sampling points for FFT will bring some calculation problems [18]. Using analytical Fourier CGH may save the last FFT, but it will also suffer from the short depth-of-field because of the limitation of aperture and focus depth of lens [12].

On the other hand, people use high-class hardware to accelerate the computation speed. For example, a MIT group has developed a special purpose computational board for holographic rendering [19]. They achieved a record that the implementation was 50 times faster than workstation of the time. Most recently, some researchers presented parallel algorithms based on commodity graphic processing units (GPUs) [20–22]. These approaches, using GPGPU (General-Purpose computation on GPU) techniques, can accelerate the CGH calculation in an inexpensive way.

In this paper, we derive the analytical formulas to describe the diffraction field of 3D true-life scenes. Since there no need for FFT, we can get rid of the numerical errors from the discrete algorithm. In addition, we propose a simple and useful phase adjustment technique to remove the visible mesh edges that appear in previous analytical polygon-based methods. Therefore, the smooth reconstructed surface, as well as the high quality 3D display, can be accomplished. By using CUDA [23], another GPU technique, the algorithm performs hundreds of times faster than those of CPUs.

2. Theory

2.1 Analytical diffraction field of 3D triangle-mesh model

In the common computer graphics field, 3D surface models are represented by polygons, for instance, triangles, as shown in Fig. 1(a). If the diffraction field emitted from each triangle of the 3D model can be calculated explicitly on the hologram plane, the whole objects field distribution will be obtained through their linear superposition.



Fig. 1. (a) Local coordinate system of triangle mesh object and the global coordinate system. (b) Diffraction by one of the triangles

For the yellow triangle of the object [Fig. 1(a)], the local coordinate system \$(x_l, y_l, z_l)\$ and the global coordinate system \$(x_g, y_g, z_g)\$ can be related by the rotation matrix:

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_g - x_c \\ y_g - y_c \\ z_g - z_c \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (1)$$

where \$(x_c, y_c, z_c)\$ is the center of mass of the triangle in global coordinate system, and \$z_l\$ on the triangle facet is 0. Let three vertices of the triangle be \$(x_g^i, y_g^i, z_g^i)\$, \$i = 1, 2, 3\$ in global coordinate system, and \$(x_l^i, y_l^i, 0)\$, \$i = 1, 2, 3\$ in the local coordinate system.

Consider the diffraction of monochromatic plane wave by the finite triangular-shape aperture in an infinite opaque screen, as shown in Fig. 1(b). The wave propagates along the axis \$z = z_g\$. The hologram is perpendicular to \$z = z_g\$ and located at \$z_g = 0\$. In local coordinate system, the complex amplitude of the plane wave is \$\exp[i2\pi(r_{31}x_l + r_{32}y_l + z_c)/\lambda]\$. By using the Rayleigh-Sommerfeld diffraction integral [5], the diffraction field of the yellow triangle on the hologram [see Fig. 1(b)] can be expressed,

$$O_H(x_H, y_H) = (1/i\lambda) \iint O_O(x_l, y_l) \exp[i2\pi(r_{31}x_l + r_{32}y_l + z_c)/\lambda] [\exp(ikr)/r] dx_l dy_l \quad (2)$$

where \$k = 2\pi/\lambda\$ and \$\lambda\$ is the wavelength; \$r = \sqrt{(x_H - x_g)^2 + (y_H - y_g)^2 + (z_H - z_g)^2}\$ is the distance between the points within triangle and the hologram pixels. \$(x_H, y_H)\$ denotes the hologram coordinate system. Combined by Eq. (1), it can be found that the quantity \$r\$ is a square root function with the local coordinate variables \$(x_l, y_l)\$. Therefore, if the assumption

$$(x_H - x_c)^2 + (y_H - y_c)^2 + z_c^2 \gg (x_g - x_c)^2 \quad \text{and} \quad (y_g - y_c)^2 \quad (3)$$

is satisfied, then r can be expanded into:

$$r = r_0 + \frac{x_l^2 + y_l^2}{2r_0} - \frac{[r_{11}'(x_H - x_c) + r_{21}'(y_H - y_c) - r_{31}'z_c]x_l + [r_{12}'(x_H - x_c) + r_{22}'(y_H - y_c) - r_{32}'z_c]y_l}{r_0} \quad (4)$$

where $r_0 = \sqrt{(x_H - x_c)^2 + (y_H - y_c)^2 + z_c^2}$ is the distance between the triangular mass center and the hologram pixel. In fact, Eq. (3) can be regarded as the modified paraxial approximation. It indicates that the linear scale of each triangle should be much less than the distance between the mass center of triangle and the hologram plane. This restriction can be easily satisfied as the scene usually contains numbers of triangles and each triangle is small. In this way, Eq. (2) becomes,

$$O_H(x_H, y_H) = \frac{\exp[ik(z_c + r_0)]}{i\lambda r_0} \iint O_o(x_l, y_l) Q(x_l, y_l) \exp\left[-i\frac{2\pi}{\lambda r_0}(x_H'x_l + y_H'y_l)\right] dx_l dy_l \quad (5)$$

where $Q(x_l, y_l) = \exp[ik(x_l^2 + y_l^2)/2r_0]$ is the quadratic phase factor; and $x_H' = r_{11}'(x_H - x_c) + r_{21}'(y_H - y_c) - r_{31}'z_c - r_{31}'r_0$, $y_H' = r_{12}'(x_H - x_c) + r_{22}'(y_H - y_c) - r_{32}'z_c - r_{32}'r_0$. Note that Eq. (5) is similar in form to the well known Fresnel diffraction between two parallel planes [5]. As we know the Fresnel diffraction formula can be simplified to Fraunhofer form when an infinity aperture lens set immediately in front of the object [5]. Numerous experiment results of Fraunhofer holography reveal that the quadratic phase factor within Fresnel integral does not affect the reconstruction performance [5]. So in the CGH, the quadratic phase factor $Q(x_l, y_l)$ in Eq. (5) can also be discarded but would not affect the reconstruction result. Thus, Eq. (5) will be simplified as

$$O_H(x_H, y_H) = \frac{\exp[ik(z_c + r_0)]}{i\lambda r_0} \mathcal{F}[O_o(x_l, y_l)] \quad (6)$$

where $\mathcal{F}[O_o(x_l, y_l)]$ is the Fourier transform of the yellow triangle, evaluated at frequencies $(x_H'/\lambda r_0, y_H'/\lambda r_0)$. Although the yellow triangle is arbitrary, we can still derive the analytical form with the help of the affine transformation for two-dimensional Fourier transform [11,12,24]. After the cumbersome algebras, it yields

$$\mathcal{F}[O_o(x_l, y_l)] = (a_{22}a_{11} - a_{12}a_{21}) \exp\left(-i2\pi \frac{a_{13}x_H' + a_{23}y_H'}{\lambda r_0}\right) \mathcal{F}_\triangle\left(\frac{a_{11}x_H' + a_{21}y_H'}{\lambda r_0}, \frac{a_{12}x_H' + a_{22}y_H'}{\lambda r_0}\right) \quad (7)$$

where $a_{11} = x_l^2 - x_l^1$, $a_{12} = x_l^3 - x_l^2$, $a_{13} = x_l^1$, $a_{21} = y_l^2 - y_l^1$, $a_{22} = y_l^3 - y_l^2$, $a_{23} = y_l^1$. The last term \mathcal{F}_\triangle is Fourier transform of a right triangle with vertices in the points (0,0), (1,0), (1,1), and it can be expressed analytically as Ref. 11. Using Eqs. (6) and (7), we can compute the diffraction field of an arbitrary tilted triangle analytically. The reconstruction results of a slanted triangle are presented in Fig. 2 for verification purpose. The reconstructions at different distances reveal that the different part of a triangle is focused while other parts are defocused. For example, in Fig. 2(b), point A is focus but point B and C are blur, when we focus on plane I where point A locates; in Fig. 2(c), point B is focus but point A and C are blur, when we focus on plane II where point B locates. This is a very important result, as it allows us to generate the hologram directly pixel by pixel without FFT, and easily to perform

the theory on the GPU's parallel platform. Finally, the whole objects field can be obtained through the linear superposition of the diffraction field per triangle.

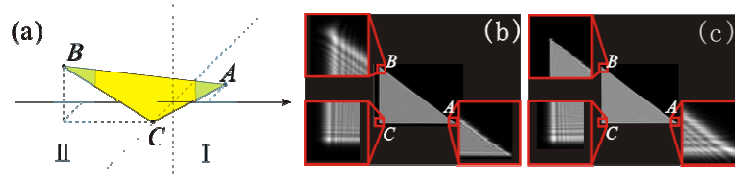


Fig. 2. (a) A schematic illustration of a slanted triangle reconstruction. (b) and (c) are reconstruction results at the plane I and II where point A and B locate.

2.2 Texture mapping, occlusion, and phase adjustment

To achieve a better performance, some specifics should be taken into account. For example, we adopt a modified Lambert brightness to avoid the unexpected shading [10]. Another important feature is texture of the whole scene. 3D model with texture will bring us a more realistic sense than those without texture. In our method, the amplitude of each triangle is a constant that is proportional to the color of the triangular mass center. It requests the texture of each triangle is as color-similar as possible. Otherwise, one should subdivide the triangle into small ones before extracting the texture. Hidden-surface removal is a necessary and troublesome issue since an object may occlude or partially overlapping by another one. Before coding, we let all the triangles be projected onto the hologram plane, and then sort them according to their distance. The shielded rear triangles are discarded. For nowadays the bandwidth of SLM is narrow, this method is quite well for our situation.

One problem of using analytical formulas is that there are some visible mesh-edges in the reconstruction. This is because of the interferences between the diffraction fields of the same edge that belongs to two conjoint triangles. In other words, we record two virtual triangular edges with an additional phase difference (δz) instead of one real edge. Here, we develop a phase adjustment technique to smooth such visible edges. Our method only needs to adjust the z component of mass center of each triangle (z_c) to an integral multiple of wavelength. There are two reasons. It is the fact that, when we reconstruct the scene, the view angle is narrow along the z axis due to the narrow SLM bandwidth nowadays. Thus, only the z component of the phase adjustment is considered. Another is that, the light intensity of the triangular edge is equal to $I = \alpha^2 [1 + \exp(i2\pi\delta z / \lambda)]^2$, where α relates to the response of coherent system to sharp triangular edges, with an approximate value of 0.5 at the edge [5]. When δz is an integer multiple of the wavelength, it yields $I \sim 1$. It means that the intensity of triangular edge is almost the same as that inside the triangle, and thus, the visible edges is smoothed. In practice, we let all the z components of mass center of triangles (z_c) to be an integer multiple of the wavelength, so that δz is also the integer multiple of the wavelength. In addition, this simple phase adjustment can be easily integrated into our analytical algorithm, and make us employ using the GPU platform. Since the adjustment is wavelength order of magnitude, it is too slight to affect the reconstruction positions of the macroscopical triangle. The reconstructed scene shown below will demonstrate how the phase adjustment works well.

3. Parallel algorithm and implementation on CUDA

To accelerate the computation in an inexpensive way, we consider the parallel algorithm based on GPU. The CUDA compiler is employed as a programming environment. It can compile a C-like language source code for parallel application on the GPU [23]. Data communication is always a troublesome problem in the parallel calculation. In our full analytical encoding method, however, the sampling points on the hologram can be calculated individually, leading that the data communication time could be extremely decreased. We use

a CPU server (Dell PE2900, Quad-Core Intel Xeon Pro X5460 2X3.16GHz) and a graphics card with GPU chipset (GeForce GTX 285) for comparison. From the Table 1, it can be seen that while the triangles is 120, we can achieve a video rate. The performance in the case of huge number is more encouraging. When the number of triangles is more than three thousand, the accelerated effect by CUDA is amazing. The CUDA algorithm runs more than 500 times faster than that of serial algorithm in CPU.

Table 1. Holographic computation times for several triangular-mesh models with the resolution of 1024 by 768

No. of triangles	120	277	495	1543	3159	6215	10296	19337
CUDA (ms)	40	60	108	250	460	860	1390	2450
CPU(MPI) /CUDA	17.1	41.8	61.0	87.1	93.7	98.8	100.9	102.7
CPU(Serial) /CUDA	136.6	261.7	348.6	473.1	506.6	526.6	541.4	542.4

4. Results

To calculate the hologram for true life, we should firstly bring the real 3D scene into computer. The image-based modeling and photogrammetry software- ImageModeler [25] is adopted to generate 3D models from ordinary 2D digital photos. This software provides an inexpensive, flexible and fast way to record the real scene and even for a living.

Figures 3(a) and 3(c) are the digital photos we take from two arbitrary different aspects for modeling. Needless to input information about the camera, we use the ImageModeler for calibration and modeling. After that, photorealistic texture maps of the model are automatically extracted from the pixel information of the source images. Till then, we obtain the textured 3D scene model. Finally, we encode them in the full analytical CGH algorithm as proposed above. The 3D scene model here contains 22408 triangles, and the resolution of CGH is 4096×3072 , while pixel size is $2\mu m$. The scene is at a distance of 15cm from the hologram, and the pawns in back are 1.76cm apart from the king in front.

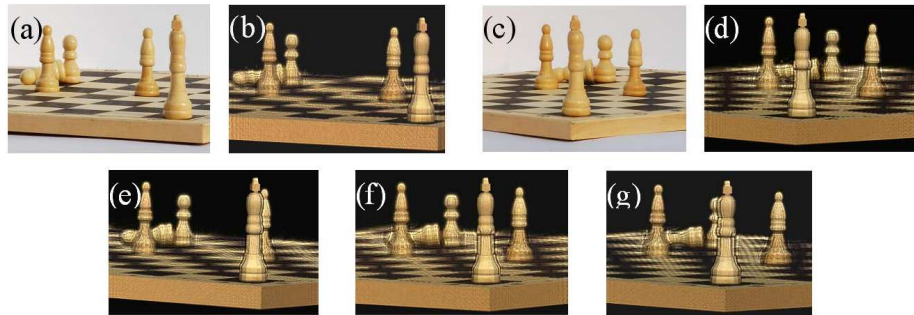


Fig. 3. Images for different aspects of the scene: (a), (c) original photos of the true-life scene taken by digital camera. (b), (d) and (e)-(g) numerical reconstructions of the scene from different aspects.

To verify the texture technique, numerical colored reconstruction is employed. The textured scene model is separated into three primary color components; i.e., red, green, and blue. Each monochromatic target is encoded and reconstructed independently. The final colored result is merged from these three monochromatic reconstructions. Figure 3 shows five aspects of the reconstructed scene to demonstrate the enhancement on the realistic effect. The reconstructed views in Figs. 3(b) and 3(d) are consistent with Figs. 3(a) and 3(c), respectively. Moreover, since we obtained the 3D scene model, we can reconstruct not only the same aspects as those of photos but also other aspects, as shown in Figs. 3(e)–3(g). The occlusion effects between chesses cause a strong depth sensation. For instance, the king in front shields a part of the right bishop in Fig. 3(e); when the scene rotates a slight clockwise angle, the king

does not shield any chesses [Fig. 3(f)]; when it rotates more, the king shields the back pawn. It demonstrates our occlusion methods works quite well.

Thanks to our phase adjustment technique, the artificial edges now disappear in our reconstruction, so that the reconstructed scenes perform a well solid effect. Meanwhile, the objects retain their positions and are not affected by the phase adjustment. The shading technique used here also adds to the impression of surface curvature.

In Fig. 4, simulation results of two cases focused on the king [Fig. 4(a)] and pawn [Fig. 4(b)] are presented. It is clear that the pawn becomes defocus blur when focusing on the king, and vice versa. This defocus effect further enhances the depth sensation, and further demonstrates that the 3D true-life scene is faithfully reconstructed.

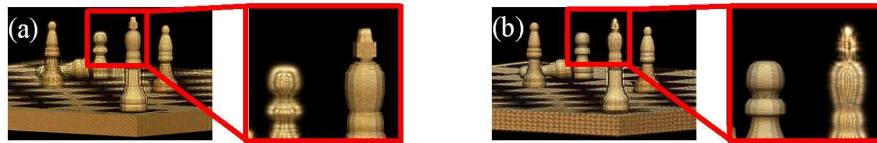


Fig. 4. Images of numerical reconstruction: (a) focus on king, (b) focus on pawn

5. Conclusion

We proposed a novel method for high-speed hologram generation of 3D real scenes. A full analytic theory is derived to directly describe the object field on the hologram plane. FFT is no need, and thus, we can get rid of some limitation from it. To remove the artificial visible edges in analytical Fourier transforms for polygon based model, we develop a simple but useful phase adjustment technique. Finally we employ the GPU to accelerate the calculation, and obtain a five hundred times faster than the serial CPU algorithm. One limitation of our theory is the modified paraxial approximation. But it seems that this limitation can be further overcome by using a curved array of spatial light modulators [26].

Since we just use a single GPU and achieve such an amazing speed, we believe that our full analytical algorithm based on Multi-GPU will make the real time display come true in the future. Our method described above is suitable for 3D data from not only image-based modelings, but also any other ways such as CT, NMR, laser scanner, 3D design software, and so on. Furthermore, if we have two different aspects synchronized videos, we can record and display a motional scene.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (10874250, 10674183, 10804131), Ph.D Degrees Foundation of Education Ministry of China (20060558068), and SYSU Young Teacher Funding (2009300003161450). J.W.D.'s email: dongjwen@mail.sysu.edu.cn .