Computer-generated-hologram-accelerated computing method based on mixed programming

Yaping Zhang (张亚萍)*, **Wang Peng (王 鹏)**, **Hong Chen (陈 虹)**, **Yinhua Xu (许寅华)**, **Wei Chen (陈 伟)**, and **Wei Xu (许 蔚)**

School of Science, Kunming University of Science and Technology, Kunming 650500, China

**Corresponding author: yaping.zhang@gmail.com*

Received November 1, 2013; accepted January 15, 2014; posted online February 28, 2014

Component object model technology is used to solve problems encountered when using three-dimentional (3D) objects to conduct computer-generated hologram (CGH) fast coding. MATLAB and C/C++ are combined for relevant programming under experimental conditions. The proposed method effectively reduces the time required for holographic encoding of large amounts of 3D object data. The CGH-accelerated computing method based on mixed programming is proven to be highly reliable and practical by testing the 3D data of different data volumes. According to the test results, the proposed method improves the efficiency of holographic encoding. The higher the data volume is, the more significantly the computation speed is improved.

OCIS codes: 090.1760, 090.1995.

doi: 10.3788/COL201412.030902.

With the development of computer technology and electronic display technology in recent years, computer-generated holography has attracted attention from relevant industrial sectors[1−3]. Three-dimentional (3D) display technology based on computer-generated holography is expected to lead to a display revolution in the industry because of its unique advantages[4−6]. A computer-generated hologram (CGH) can comprehensively record the amplitude and phase of light waves and has more obvious advantages than traditional optical holography, including low noise, zero environmental impact, and high repeatability. CGH can also record the hologram of any existent or nonexistent object and allow easy manual control. However, creating a CGH of a 3D object involves two main difficulties: the large data volume of an object and macrooperation of the hologram. Large data volumes require high-speed hardware and high display resolution. For this technology to be practical, the problem of transmission bandwidth also needs to be solved. Therefore, under the current conditions, one method of expediting the calculation speed of 3D holograms is improving calculation speed by using algorithms (called "soft processing"). The typical soft processing method involves dividing the 3D object into a polygonal mesh bin, and then using a geometric method for conversion, rotation, and processing to improve the computational efficiency of the hologram [7,8]. Another method is using hardware to improve calculation speed (called "hard processing"). Hard processing usually involves data calculation and image transmission of large data volumes through the design of the hardware circuit[9−11]. However, these two kinds of processing modes cannot fundamentally solve the problem of low hologram computation speed. Thus, this paper proposes a method of accelerating computing and also provides a reference for future studies on hologram calculation acceleration. Many other methods can be used to improve computational efficiency, such as fast Fourier transform, simplifying the hologram encoding process, compressing the bandwidth of the hologram, and using the "difference method" to rapidly solve a Fresnel hologram.

This study achieves accelerated holographic encoding of large volumes of 3D object data through mixed programming, which takes advantage of the high efficiency of C/C++ and makes up for the slow execution speed of this method when displaying the resource-intensive computing functions of MATLAB. This study tests different data sizes, compares the time spent when only MATLAB is used for computing, and reaches the conclusion that the method based on mixed programming can significantly improve computational efficiency. Furthermore, the larger the data volume, the greater the improvement in computation speed is.

A 3D shape measurement system is combined with CGH technology to achieve a 3D holographic display. A 3D scanner manufactured by Hangzhou Conformal and Digital Technology Limited Company is used to collect the 3D data of static objects and create a CGH. This system can obtain 3D information about actual objects and is more convincing than a virtual 3D object. The 3D information obtained by the scanner is considered as the holographic data used in this study.

Figure 1 presents a schematic diagram and physical map of the fringe projection 3D measurement system. The projector first projects the stripe onto the surface of the measured object. The depth information of the object surface modulates the amplitude and phase of the stripe. Then, a charge coupled device camera is used to acquire images of the deformed grating pattern after the object surface structure is modulated. The image capture card is then used to transmit these images to a computer for processing. The deformed grating pattern images carry the height information for the measured objects. Digital image processing technology allows certain algorithms to demodulate the phase of the object depth information contained in the images and unwrap the phase after obtaining the decoding information and phase value. The spatial value of the measured object can be obtained based on the phase value that corresponds to each pixel point of the image, to obtain the 3D

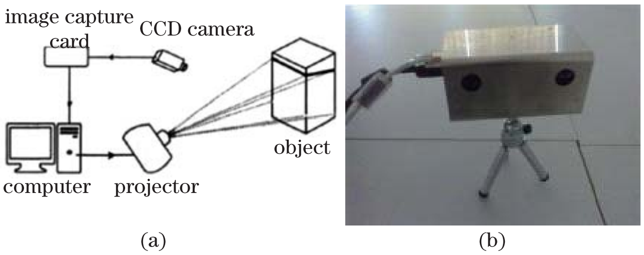


Fig. 1. (a) Composition sketch of 3D measurement system and (b) 3D camera.

information of the object.

Data acquired during scanning include spatial coordinate information and gray information. The required data are extracted. Simulated object light distribution on the CGH plane can be determined according to the propagation law of light waves. The digital images of 3D objects are formed mainly by triangular meshes consisting of acquired 3D data.

To realize CGH, discrete data of 3D objects must be obtained firstly. To realize the holographic representation of real objects, a cartoon face model is taken as a real object in the test. In the test, a 3D static scanner is used to obtain the 3D object data and create an obj file. The information on all of the effective object points constituting the object are then read from this obj file. The images of the 3D object acquired using the 3D scanner are shown in Fig. 2(b). Figure 2(c) presents grid information about the acquired images.

CGH can be formed after 3D information is obtained. An object can be considered as a set consisting of a limited number of points. The computer simulates the optical propagation of the point light source in limited directions and determines the complex amplitude distribution of this point on a holographic plane. Then, the complex amplitude of all object points and the information data of the wave surface of object O are superimposed on the holographic plane. Figure 3 shows the CGH creation model.

If we suppose that the total number of object points is L and the distance between the ith point Pi and holographic plane is z_i, then the complex amplitude distribution of P_i is $O_i(x_i, y_i)$, multiple random phase matrix $\varphi_r(x_i, y_i)$ by the complex amplitude of each point, to avoid the loss of object wave frequency spectrum information, which can result in distortion of the reproduced image. The Fresnel diffraction distribution of the holographic plane can be represented as

$$E_i(\xi,\eta) = \frac{\exp(\mathrm{j}kz_i)}{\mathrm{j}\lambda z_i} \iint O_i(x_i,y_i)\exp[\mathrm{j}\phi_r(x_i,y_i)] \cdot \exp\left\{\frac{\mathrm{j}k}{2\mathrm{z}_i}\left[(\zeta - x_i)^2 + (\eta - y_i)^2\right]\right\}\mathrm{d}x_i\mathrm{d}y_i. \quad (1)$$

Then, the result of superposing the diffraction distribution of each object point on the holographic plane is

$$E(\xi,\eta) = \sum_{i=1}^{L} E_i(\xi,\eta) = A_0(\xi,\eta)\exp\left[\mathrm{j}\phi(\xi,\eta)\right], \quad (2)$$

where $A_0(\xi,\eta)$ and $\phi(\xi,\eta)$ are the amplitude and phase, respectively, of the holographic plane complex amplitude after superposition. Provided that the reference light is

$$R(\xi,\eta) = R_0(\xi,\eta)\exp(\mathrm{j}2\pi(f_\xi\xi + f_\eta\eta)), \quad (3)$$

where f_ξ and f_η are the inclination factors in the directions of ξ and η, respectively, and $R_0(\xi,\eta)$ is the amplitude of the reference light. After $A_0(\xi,\eta)$ is normalized, the hologram obtained through Burch coding can be represented as

$$h(\xi,\eta) = 0.5\{1 + A_0(\xi,\eta)\cos[2\pi(f_\xi\xi + f_\eta\eta) - \varphi(\xi,\eta)]\}. \quad (4)$$

MATLAB is matrix language designed for vector and matrix operations. Generally, if an operation can be conducted with vector or matrix, its speed can be significantly high. However, if the operation involves a large number of cyclic iterations, the speed of MATLAB becomes unbearable. In MATLAB, the speed at which the M file calculates cyclic iteration is considerably lower than that of C/C++. Therefore, the parts that require a large number of cyclic iterations can be written into an MEX file using C/C++. In this manner, MATLAB does not have to interpret the statements whenever the statements in the loop are being run, and the file does not have to be called every time to improve computation speed. Designing algorithms using C/C++ and the fully functional programming environment of MATLAB is necessary to ensure the interaction between data and programs[12].

MEX file, a dynamic linking subprogram written in C/C++, is called automatically by the MATLAB interpreter and executes dynamic linking functions. Using MATLAB to compile an M file into a DLL and offer it to C/C++ for calling provides a method that enables rapid mixed programming of C/C++ and MATLAB.

When employing 3D data obtained using a 3D scanner to generate CGH, this study performs diffraction calculation and coding for each object point, and then superposes the hologram of each object point to create a complete hologram of the 3D object. When the program is running, a large number of cyclic iterations is required. For practicality and usability, we use mixed programming of C/C++ and MATLAB in accelerated hologram generation.

In MATLAB, C/C++ can be used to implement the MEX file, the functions of which can then be called through MATLAB. However, for the specific implementation, the name of the entry program must be mexFunction and contain four parameters, which are defined void mexFunction (int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]), where the two parameters, nlhs and nrhs, stand for the number of output and input variables, respectively; plhs and prhs stand for the pointer vectors of output and input variables, respectively, the lengths of which are the two pointer arrays nlhs and nrhs, respectively.

Through C/C++, the concrete functions of mexFunction can be implemented. These functions are then compiled into a MEX file. In MATLAB, this file is referred to as the function name to finish the use of MATLAB function accessed through C/C++.

This study uses two CGH methods. Only MATLAB is used for computation in method 1, whereas mixed programming is used for computation in method 2. This paper lists the computation time required for different numbers of object points to generate CGH to compare the computation speeds of the two methods. The data on 3D objects are collected using a dynamic 3D scanner manufactured by Hangzhou Conformal & Digital Technology Ltd. Co. The specifications of the computer used in the test is as follows: Intel(R) Core(IM) i7 CPU with 3.40-GHZ, 4-GB memory. Its soft environment is Microsoft Windows 7 operating system and MATLAB 7.11 language. When the volume of the data on the 3D object used for the CGH is extremely small, the holographic reconstructed images become distorted. Therefore, to obtain accurate holographic representation results, we calculate the holograms of five different numbers of object points. The numbers of object points are denoted by x and y. The resolution of the hologram is denoted by xx. In the calculations used to generate the hologram, the number of object points is the same for the two methods discussed in this paper. The CGH results obtained using method 2 and the corresponding reproduction results are presented in Fig. 4.

Given that the quantity of object points of each group is different, the superposition results formed by the diffraction distribution of the object-light field in the holographic plane are not identical and are affected by random phases. Therefore, the stripe distribution of the hologram of each group also varies, as indicated in Figs. 4 (a1), (b1), (c1), (d1), and (e1). The upper right corner of each figure shows a partially enlarged view.

According to the experimental results, although the data computation of each group can reflect the basic characteristics of the original 3D object, the details of the object are missing, as shown in Figs. 4(a2), (b2), (c2), (d2), and (e2). As the number of object points increases, the details of the reproductive image also increase and the reproduction results become more distinct. Thus, the reproduction results are improved. In the opposite case, the result is worse. As the object points increase, the time to compute the hologram increases. Therefore, the sampling points of the object and the accuracy of the algorithm should be improved to significantly affect the computation speed. Table 1 shows the time required to compute the hologram with different objects using methods 1 and 2.

(a) (b) (c)

Fig. 2. (a) Original 3D object, (b) image of collected 3D object, and (c) grid information.

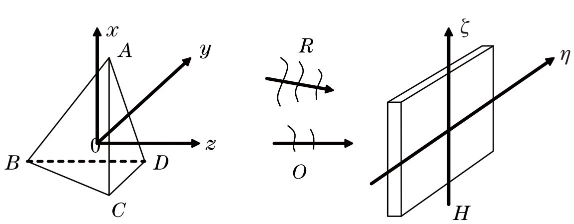
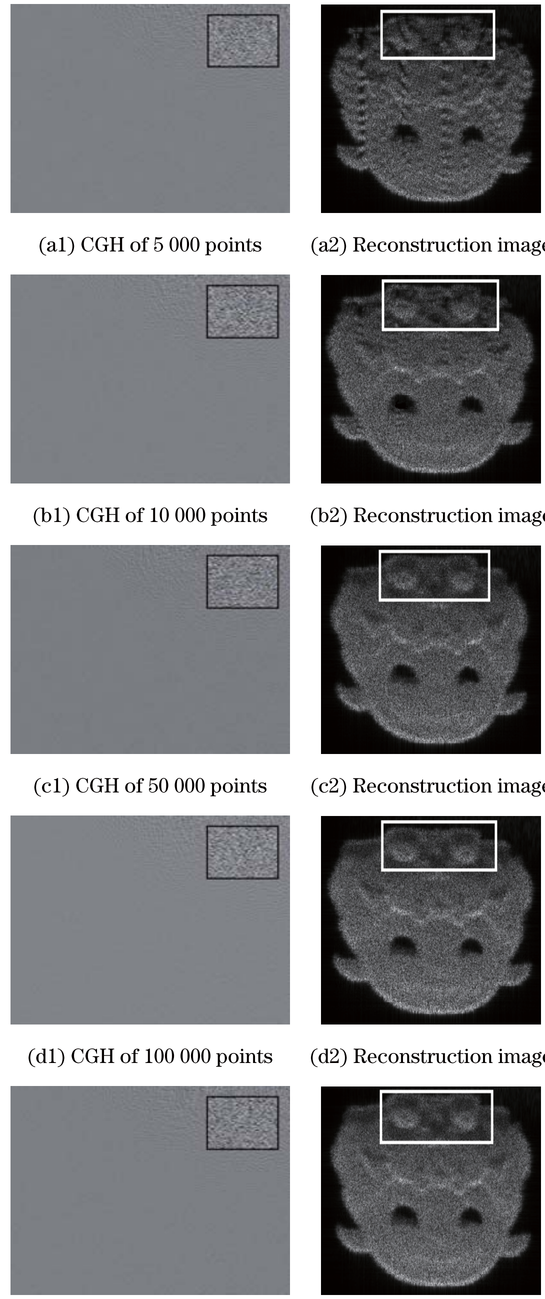


Fig. 3. Computational model of a CGH.



(a1) CGH of 5 000 points (a2) Reconstruction image
(b1) CGH of 10 000 points (b2) Reconstruction image
(c1) CGH of 50 000 points (c2) Reconstruction image
(d1) CGH of 100 000 points (d2) Reconstruction image
(e1) CGH of 150 000 points (e2) Reconstruction image

Fig. 4. CGH of different points and reconstruction images.

Table 1. Computation Time for the Two Methods (s)

Object Point	Method 1	Method 2
5000	112.24 s	58.64 s
10000	233.75 s	123.38 s
50000	1119.03 s	618.56 s
100000	2348.68 s	1256.84 s
150000	3526.54 s	1895.97 s

Compared with method 2, method 1 can improve the speed of the hologram computation, an advantage that becomes more obvious as the object points increase.

In Ref. [11], the author uses the GPU for the hologram computation of over 10000 object points at 1920×1200 resolution, which took approximately 6 s. Operation time is closely related to the algorithm. If the algorithm is applied properly, the operation time is decreased. Regardless of the method used, the computation should be in real time, that is, the operation time must be close to zero, which is difficult to achieve. Therefore, although the proposed method may not be the most optimal method[13], it can accelerate the 3D computation of a hologram. The experimental results show that the proposed hybrid programming method can effectively shorten the hologram computation time.

In conclusion, we obtain the data of 3D objects by using a 3D static scanner and the hologram-accelerated computing method based on mixed programming, as well as achieve the accelerated computation of holographic encoding. The experimental results show that the proposed method can effectively reduce hologram computation time. This method is significant to both CGH and the photoelectric representation of a hologram. However, the computation speed of this method requires improvement. We believe that with technological advancement and the coming era of parallel computing, the processing speed of computer hardware will be continuously improved and the speed of the mixed programming of C/C++ and MATLAB in calculating large data volumes of holographic encoding will increase substantially. Therefore, the proposed method is significant to further research and development of holographic 3D display.

This work was supported by the National Natural Science Foundation of China (Nos. 61007061 and 11172116) and the Scientific Research Foundation for the Returned Overseas Chinese Scholars of the State Education Ministry (No. KKQA201307001).

References

1. D. P. Kelly, D. S. Monaghan, N. Pandey, T. Kozacki, A. Michalkiewizc, G. Finke, B. M. Hennelly, and M. Kujawinska, International Journal of Digital Multimedia Broadcasting **13,** 627 (2010).
2. X. Z. Jian, H. Zhang, J. P. Fan, and Z. G. Zhou, Laser Technol. (in Chinese) **36,** 26 (2012).
3. T. Kozacki, Opt. Express **18,** 27118 (2010).
4. H. Choi, S. A. Shestak, S. K. Kim, J. Y. Son, and J. W. Wu, Proc. SPIE **4659,** 76 (2002).
5. F. Yaras, H. Kang, and L. Onural, J. Display Technol. **6,** 443 (2010).
6. P. Yeh and C. Gu, Chin. Opt. Lett. **11,** 010901 (2013).
7. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, Appl. Opt. **47,** 1567 (2008).
8. K. Matsushima and S. Nakahara, Appl. Opt. **48,** H54 (2009).
9. F. Yaras, H. J. Kang, and L. Onural, Proc. SPIE **6,** 283 (2009).
10. F. Yaras, H. J. Kang, and L. Onural, in *Proceedings of Digital Holography and Three Dimensional Imaging, OSA Technical Digest* DWA4 (2009).
11. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, Opt. Express **14,** 7636 (2006).
12. W. G. Li, Computer Knowledge and Technology (in Chinese) **5,** 7002 (2009).
13. Y. K. Lam, W. C. Situ, and P. W. M. Tsang, Chin. Opt. Lett. **11,** 050901 (2013).