Optical Engineering

SPIEDigitalLibrary.org/oe

One-eighth look-up table method for effectively generating computer-generated hologram patterns

Sungjin Cho
Byeong-Kwon Ju
Nam-Young Kim
Min-Chul Park

SPIE

One-eighth look-up table method for effectively generating computer-generated hologram patterns

Sungjin Cho,[a,b] Byeong-Kwon Ju,[a] Nam-Young Kim,[c] and Min-Chul Park[b,*]
[a]Korea University, Department of Electrical Engineering, Anam-Dong 5-ga, Seongbuk-Gu, Seoul 136-713, Republic of Korea
[b]Korea Institute of Science and Technology, Sensor System Research Center, Hwarang-ro-14-gil, Seongbuk-Gu, Seoul 139-791, Republic of Korea
[c]Kwangwoon University, Department of Electronic Engineering, Wolgye 1-dong, Nowon-Gu, Seoul 139-701, Republic of Korea

Abstract. To generate ideal digital holograms, a computer-generated hologram (CGH) has been regarded as a solution. However, it has an unavoidable problem in that the computational burden for generating CGH is very large. Recently, many studies have been conducted to investigate different solutions in order to reduce the computational complexity of CGH by using particular methods such as look-up tables (LUTs) and parallel processing. Each method has a positive effectiveness about reducing computational time for generating CGH. However, it appears to be difficult to apply both methods simultaneously because of heavy memory consumption of the LUT technique. Therefore, we proposed a one-eighth LUT method where the memory usage of the LUT is reduced, making it possible to simultaneously apply both of the fast computing methods for the computation of CGH. With the one-eighth LUT method, only one-eighth of the zone plates were stored in the LUT. All of the zone plates were accessed by indexing method. Through this method, we significantly reduced memory usage of LUT. Also, we confirmed the feasibility of reducing the computational time of the CGH by using general-purpose graphic processing units while reducing the memory usage. [DOI: 10.1117/1.OE.53.5.054108]

Keywords: computer-generated hologram; holograms; parallel processing.

Paper 131876 received Dec. 12, 2013; revised manuscript received Apr. 24, 2014; accepted for publication Apr. 25, 2014; published online May 21, 2014.

1 Introduction

Recently, three-dimensional (3-D) display technology has received a lot of attention throughout the world,[1] and many works have been conducted and have contributed to the development of different new technologies. Among them, stereoscopic display is a technique that can provide a 3-D experience in real space. However, stereoscopic displays have uninvited problems, such as crosstalk, a limited viewing zone, and visual fatigue. For these reasons, a different way of 3-D representation is required. Digital holography, which does not cause any unwanted biomedical effect, has been considered as a powerful method for reconstructing real objects and displaying complete 3-D information. Indeed, holographic technology can supply very high-quality images and accurate depth cues viewed by human eyes without any special observation devices such as 3-D glasses. However, in order to record holograms of real 3-D objects, it is necessary to produce wave interferences between two intense laser beams with a high degree of coherence in a dark room. Moreover, this system must be established in a very stable environment because the interference fringes, in which intensity and phase information of the real objects are contained, can be easily destroyed by very slight surrounding stimulation.[2] Because of these requirements, there are limitations to the use of optical holographic systems for conventional hologram recording methods in a common outdoor surrounding. The method of computer simulation, called computer-generated hologram (CGH), enables the generation of interference fringe patterns in an ideal situation and has been widely used as an alternative to the direct recording of holograms under coherent illumination.

Several approaches have been proposed so far to compute CGH.[3–5] Among them, the ray-tracing method has been known as a rigorous approach, but it also presents a huge computational complexity because it requires calculating the contributions of all the point light sources composing the object to every pixel of the recording plane. Because of the huge computational complexity, many studies have made efforts to reduce the computational burden by applying particular methods such as look-up tables (LUTs) and parallel processing by using the general purpose graphic processing unit (GPGPU).[6]

The LUT method had been originally proposed by Lucente[7], and different versions of this method have also been proposed afterward.[8–10] In this method, 3-D objects are represented as a cloud of point light source. The fringe patterns, generated on the pixels of the photoplate from each of the possible locations in the point cloud of object, are precalculated and stored in the LUT. The fringe patterns of the 3-D objects can then be quickly generated with this LUT. However, the LUT methods have a critical drawback in the fact that they require a very large memory size for storing the fringe pattern data of all the possible self-luminous points of the real objects. Because of this drawback, it is hard to apply the LUT method and the parallel processing method simultaneously for generating CGH. To apply both of them, it is needed to reduce the LUT memory usage. In an effort to reduce the memory usage, methods that calculated a zone plate by using only line information were proposed.[11,12]

*Address all correspondence to: Min-Chul Park, E-mail: minchul@kist.re.kr

0091-3286/2014/$25.00 © 2014 SPIE

However, these methods introduce normalized errors. Therefore, in order to generate CGHs without the quality degradation of the hologram, LUT should contain sufficient information about the zone plate.

In this article, we proposed a one-eighth LUT method that is able to reduce memory usage of LUT in order to apply both LUT and parallel processing by GPGPU methods for generating CGH. Also, to confirm the feasibility of the proposed method, we compared computational time and memory usage of LUT by each method respectively.

2 Methods

2.1 *Ray-Tracing Method for Computer-Generated Hologram*

In this article, in order to generate the fringe patterns for holograms, we have calculated the Fresnel hologram directly.[13] The geometry and the notation used to compute the Fresnel hologram are shown in Fig. 1. The horizontal, vertical, and depth locations of each self-luminous point of light are specified as O_x, O_y, O_z, and the photoplate pixels are denoted as P_x and P_y. The pth self-luminous point has an associate real-valued magnitude a_p and a phase φ_p. The brightness of self-luminous points determines a_p, and φ_p is the phase relative to the reference beam. The photoplate is located at the plane $P_z = 0$.

In the Fresnel hologram, the object light and the reference light propagated to the photoplate are represented with spatially varying complex time-harmonic electronic field vectors, E_O and E_R. E_O and E_R are described, respectively, by Eqs. (1) and (2):

$$E_O = \sum_{p=1}^{N_p} \frac{a_p}{r_{ijp}} \exp[j(kr_{ijp} + \varphi_p)], \tag{1}$$

$$E_R = a_R \exp[j(kx \sin \theta_R)], \tag{2}$$

where N represents the number of object points. λ is the wavelength of light and k is the wave number defined as $k = 2\pi/\lambda$. The amplitude and phase of the object light and the reference light are defined as a_p, φ_p, a_R, and φ_R, respectively. r_{ijp} is the oblique distance between the p'th object point $O_p(O_{x_p}, O_{y_p}, O_{z_p})$ and the pixel $P_{ij}(P_{x_i}, P_{y_j})$. It is defined by Eq. (3),

$$r_{ijp} = \sqrt{(O_{x_p} - P_{x_i})^2 + (O_{y_p} - P_{y_j})^2 + O_{z_p}^2}. \tag{3}$$

The total time-harmonic electric field on the hologram is the interference between the light from the object and the reference light. Therefore, the total intensity of the photoplate is defined as shown in Eq. (4),

$$I_{\text{Total}} = |E_O|^2 + |E_R|^2 + 2\Re e\{E_O \cdot E_R^*\}. \tag{4}$$

In Eq. (4), the first and second terms represent the reference light and the object light intensity, respectively. It is the third term that contains the information about the interference pattern induced by the reference light and the object light. Therefore in Eq. (4), I_{Total} can be substituted by $I(P_x, P_y)$ as defined in Eq. (5),

$$I(p_x, p_y) = \sum_{p=1}^{N_p} \frac{a_p}{r_{P_x P_y O_p}} \cos(kr_{P_x P_y O_p} + kx \sin\theta_R + \varphi_p). \tag{5}$$

According to Eq. (5), the fringe patterns for all the object points must be directly computed for all the photoplate pixels. Therefore, if the number of object points or the resolution of the photoplate increase, the computation time for generating fringe patterns increases as well with the Fresnel approach. As a result, real-time implementation with this method is limited.

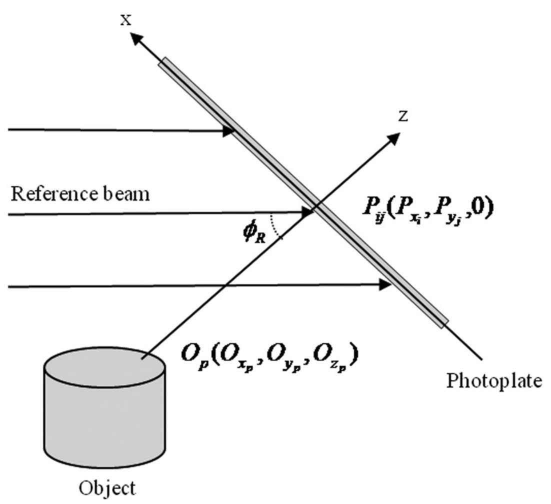


Fig. 1 Description of the generation of a computer-generated hologram (CGH) by using ray-tracing method.

2.2 *Look-Up Table Method for Computer-Generated Hologram*

The LUT method was proposed to reduce the computational complexity of the traditional ray-tracing method for generating the fringe patterns. In this approach, in order to generate the fringe patterns, all of the possible points are precalculated as shown in Eq. (6), and the calculated results are stored in the memory

$$T(P_x, P_y; O_p) \equiv \frac{1}{r_{P_x P_y O_p}} \cos(kr_{P_x P_y O_p}). \tag{6}$$

In order to perform this computation, each point light source of the object needs to be discretized. The discretization step should be sufficiently small so that the object points appear to be continuous when observing the reconstructed holograms. An efficient LUT method should contain the fringe patterns about all of the object points. The contribution of one point for one pixel is calculated by multiplying the appropriate fringe patterns stored in the LUT by the intensity, a_p, of the point. The value of the pixel is determined by adding the contribution coming from every object point, as described in Eq. (7)

$$I(P_x, P_y) = \sum_{p=1}^{N_p} a_p T(P_x, P_y; O_p). \tag{7}$$

In the LUT method, by just fetching the corresponding fringe pattern of each object point from the LUT and adding them together, one can generate the CGH patterns. Therefore, this LUT method can exceedingly increase the speed of computation for the generation of CGH patterns, because it needs only two operations of multiplication and addition. However, as mentioned earlier, this method requires a massive memory space for storing the fringe patterns from all the possible object points.

A novel LUT (N-LUT) method was proposed by Kim and Kim[8] to reduce the memory usage. They used the new concept of principal fringe pattern (PFP) for reducing the memory usage. PFPs are the zone plates coming from the light point source located in $O(0, 0, O_z)$. Through the translation along the horizontal and vertical dimensions, PFPs can cover all of the regions in possible locations of the point light source. Therefore, they could efficiently reduce the memory usage for storing only the PFPs coming from different position along the z-axis in the LUT.

2.3 *Proposed Look-Up Table Method*

As mentioned earlier, the traditional LUT method needed a very large amount of memory because all the zone plates coming from all the possible object points must be stored. In order to reduce the memory usage, N-LUT uses only PFPs as zone plate. Through the PFPs, they can dramatically reduce the memory usage. However, N-LUT method has memory usage that is still too large to apply parallel processing method. So, we proposed a modified LUT method that used only one-eighth information of PFPs.

Zone plates have the shape of a sinusoidal wave propagated from the center to all around. Because of their symmetry, one line data taken from a longitudinal radius are enough in order to express all pixels of the fringe patterns. However, computer cannot store the line data because it is an analog data. Therefore, we proposed the one-eighth LUT method for reducing the memory usage of LUT by storing only one-eighth of zone plates as shown in Fig. 2.

In the proposed method, data stored in the LUT have a shape of a triangle. The data stored in the LUT could cover all pixels about PFPs by mirroring and rotating itself. Method of indexing the LUT is shown in Fig. 3.

Basically, the indexing formula is described in Eq. (8). In this method, it is essential to compare the size of the horizontal and vertical dimensions because the vertical dimension is always longer than the horizontal one. Assumed that the longer axis and the shorter axis are y and x, respectively, all of the PFP pixels can be found by figuring out the following:

$$\text{Indexing} = \begin{cases} x(x-1)/2 + y, & (x > y) \\ y(y-1)/2 + x, & (y > x) \end{cases}. \quad (8)$$

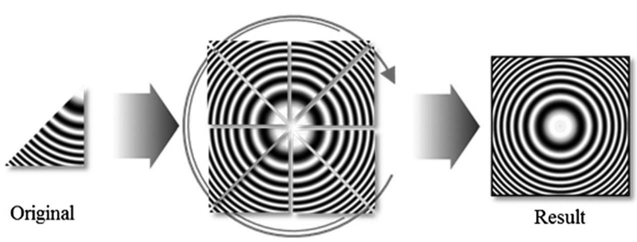


Fig. 2 Proposed look-up table (LUT) method.

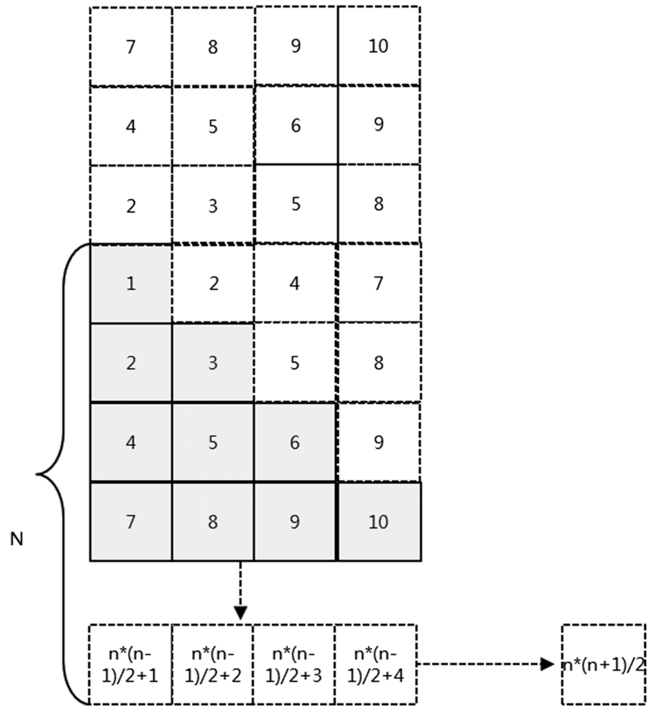


Fig. 3 Indexing method for proposed LUT method.

3 Experiment and Result

In our experiment, the computer system consisted of an Intel Core™ i7-3820 CPU @ 3.60 GHz, a main memory of 16 GB, a graphic card NVIDIA GeForce GTX 560 Ti, and a Microsoft Windows 7 operating system as well as Visual Studio 2010. The computing parameters used and the block diagram of experimental procedure are shown in Table 1 and Fig. 4, respectively.

First of all, the 3-D information is extracted from the 3-D object. Zone plates are then obtained from the LUT. Then, by shifting and adding all of the zone plates, CGH pattern can be computed.

Figures 5(a) and 5(b) show the result of the reconstructed CGH patterns made by the ray-tracing method and the proposed method, respectively. When reconstruction of CGHs was performed numerically with a 800×800-pixels space, the peak signal-to-noise ratio between the ray-tracing method and the proposed method was estimated at 36.64 dB, which is an acceptable value.[14] This noise is caused by the quantization of the LUT so that it can be controlled by modifying the quantization step. However, because quantization step is one of the main factors determining memory consumption,

Table 1 Parameter for computer-generated hologram (CGH).

Parameter	Value
Photoplate width	1280 pixel
Photoplate height	720 pixel
Object distance	25 cm
Pixel size	0.007637 mm

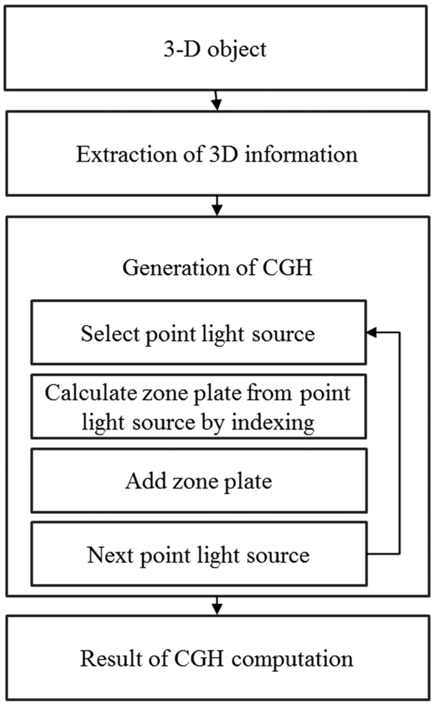


Fig. 4 Procedure for generating CGH in experiment.

Table 2 Memory usage of each method for generating CGH.

Method	Look-up table (LUT) memory usage
Ray-tracing method	0
Traditional LUT method	1280 ∗ 720 ∗ 512 ∗ 655 ∗ 655 ∗ 4 byte≒ 809 Tbyte
N-LUT method	512 ∗ 1295 ∗ 1015 ∗ 4 byte≒2.7 Gbyte
Proposed LUT method	512 ∗ 374 ∗ 649 ∗ 4 byte≒497 Mbyte

Table 3 Computational speed of each method for generating CGH.

Method	Computational time (s)
Ray-tracing method	0.1032
Ray-tracing method + GPGPU	0.0120
Traditional LUT method	0.0076
N-LUT method	0.0150
Proposed LUT method	0.0158
Proposed LUT method + GPGPU	0.0013

the quantization step should be adjusted according to the system requirements.[4]

3.1 *Analysis of Memory Usage about Proposed Method*

A comparison of the memory usage for each method is shown in Table 2.

Memory usage of the traditional ray-tracing method is zero because it does not use a LUT. We can see that the traditional LUT method uses a large amount of memory. In contrast, our proposed method uses 82% less memory space than the N-LUT method.

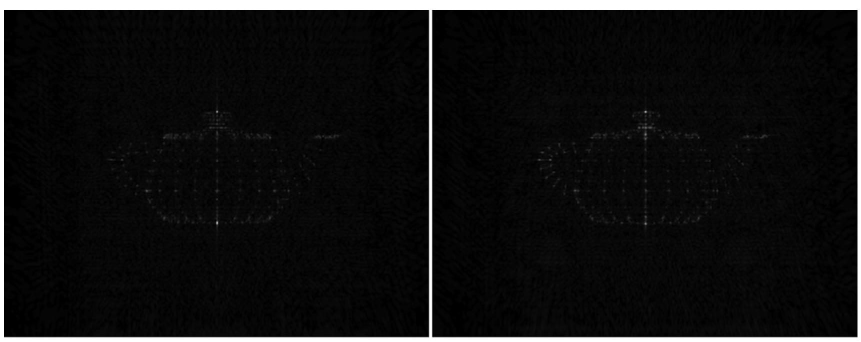
3.2 *Analysis of Speed about Proposed Method*

The computational speed for each method is shown in Table 3.

In order to compare the computational speed for the generation of CGH, we performed the calculation with each method for only one point light source. The ray-tracing method is, as expected, the method presenting the heaviest computational burden. Among LUT methods, traditional LUT method has the best computational time because it does not require any operation for calculating the zone plates. Finally, we see that although our proposed method reduces the memory usage comparing with N-LUT, there is no significant difference in computational speed between N-LUT and our proposed LUT method.

4 Conclusion

In order to reduce memory consumption of LUT, we proposed a quantized one-eighth LUT method. Also, through reduced memory usage, the computational speed could be improved by applying LUT as well as parallel processing methods. In the proposed method, the memory usage is reduced by



(a) (b)

Fig. 5 Numerical reconstruction result for CGHs about a teapot: (a) reconstruction result for method of direct computation and (b) proposed method.

storing only one-eighth of zone plates. Nonstored pixel of zone plates can then be accessed by using an indexing method. As a result, our proposed LUT method needs only 18% memory usage compared with N-LUT method. Also, thanks to the reduction of memory usage, it is possible to apply both LUT and parallel processing methods, and we could obtain improvement of computational speed by 580% compared with traditional LUT method.

Our proposed method presents a problem about speed optimization because it includes comparison operation, but it can be solved by using quarter LUT. Therefore, we can say that this one-eighth LUT method can have a positive impact on the development of CGH.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No. 2013-067321) and the "Cross-Ministry Giga KOREA Project" of the Ministry of Science, ICT and Future Planning, Republic of Korea (ROK) [GK130100, Development of Interactive and Realistic Massive Giga-Content Technology].

References

1. K. Iizuka, "Welcome to the wonderful world of 3D: introduction, principle and history," *Opt. Photonics News* **17**(7), 42–51 (2006).
2. U. Schnars and W. Jueptner, *Digital Holography – Digital Hologram Recording, Numerical Reconstruction, and Related Techniques*, Springer-Verlag, Berling (2004).
3. D. Leseberg and O. bryngdahl, "Computer-generated rainbow holograms," *Appl. Opt.* **23**(14), 2441–2447 (1984).
4. D. Lesberg and C. Frere, "Computer-generated holograms of 3-D objects composed of tilted planar segments," *Appl. Opt.* **27**(14), 3020–3024 (1988).
5. D. Leseberg, "Computer-generated holograms: display using one-dimensional transforms," *J. Opt. Soc. Am.* **3**(5), 726–730 (1986).
6. P. Yuechao, X. Xu, and X. Liang, "Fast distributed large-pixel-count hologram computation using a GPU cluster," *Appl. Opt.* **52**(26), 6562–6571 (2013).
7. M. Lucente, "Interactive computation of holograms using a look-up table," *J. Electron. Imaging* **2**(1), 28–34 (1993).
8. 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).
9. S. C. Kim, W. Y. Choe, and E. S. Kim, "Accelerated computation of hologram patterns by use of interline redundancy of 3-D object images," *Opt. Eng.* **50**(9), 091305 (2011).
10. T. Shimobaba et al., "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).
11. T. Nishitsuji et al., "Fast calculation of computer-generated hologram using the circular symmetry of zone plates," *Opt. Express* **20**(25), 27496–27502 (2012).
12. S. Lee et al., "Fast hologram pattern generation by radial symmetric interpolation," *Proc. SPIE* **8498**, 849804 (2012).
13. E. Hesham and Y. Takashi, "Computer-generated holograms: a fast ray-tracing approach," *J. King Saud Comput. Inf. Sci.* **8**, 139–151 (1996).
14. N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Optimized transmission of JPEG2000 streams over wireless channels," *IEEE Trans. Image Process.* **15**(1), 54–67 (2006).

Sungjin Cho received his BS degree in computer engineering from Kwangwoon University, Seoul, South Korea, in 2012 and his MS degree in electronics, electrical, and computer engineering from Korea University, Seoul, South Korea, in 2014. He is currently a researcher at the Korea Institute of Science and Technology. His research interests include three-dimensional (3-D) display, human factors, and image processing.

Byeong-Kwon Ju received his PhD in semiconductor engineering from Korea University in 1995. In 1988, he joined the Korea Institute of Science and Technology (KIST), Seoul, where he was engaged in the development of flat panel display and MEMS technology. Since 2009, he has been a professor in Department EE of Korea University with his main interest in flexible/stretchable electronics (OLED, TFT backplane), laser interference lithography (LIL), field emission application, Si-micromachining, and carbon nanotube-based nanosystems.

Nam-Young Kim has received two master's and two PhD degrees from State University of New York at Buffalo and Midwest University: MS and PhD in electronic engineering and MDiv and DCE in theology. Since 1994 he joined the Department of Electronic Engineering of Kwangwoon University as an assistant professor. His main research focus was in RFIC devices, ICs, and wireless application techniques in order to develop high-speed structures in GaAs, Si, and other materials.

Min-Chul Park received his MS and PhD degrees in information and communication engineering from the University of Tokyo in 1997 and 2000, respectively. He was invited to become an associate professor with the Department of Electrical Engineering, Tokyo University of Science in 2005 and a visiting researcher to Qualcomm in 2009. He is currently a principal research scientist at KIST and professor at UST. His research interests include 3-D display, human factors, and image processing.