Acceleration method for computer generated spherical hologram calculation of real objects using graphics processing unit

(Invited Paper)

Gang Li[1], Keehoon Hong[1], Jiwoon Yeom[1], Ni Chen[1], Jae-Hyeung Park[2], Nam Kim[3], and Byoungho Lee[1*]

[1] *School of Electrical Engineering, Seoul National University, Gwanak-Gu Gwanakro 1, Seoul 151-744, South Korea*
[2] *School of Information and Communication Engineering, Inha University,*
100 Inha-ro, Nam-gu, Incheon, 402-751, South Korea
[3] *School of Information and Communication Engineering, Chungbuk National University,*
52 Naesudong-ro, Heungduk-gu, Cheongju, 361-763, South Korea
[*] *Corresponding author: byoungho@snu.ac.kr*

Received February 27, 2014; accepted April 4, 2014; posted online May 28, 2014

Graphics processing unit (GPU) based fast calculation method for computer generated spherical hologram (CGSH) of a real-existing object is proposed. Three-dimensional (3D) point cloud is constructed by capturing a real-existing object from multiple directions using a depth camera. The GPU based calculation is used in both hologram generation part and numerical reconstruction part of the CGSH. The improved calculation efficiency is verified by comparing the computation speed between central processing unit (CPU) based and GPU based implementation.

OCIS codes: 090.0090, 090.1760.
doi: 10.3788/COL201412.060016.

Holography has been considered as a perfect three-dimensional (3D) display technique because of its many advantages. In comparison to other 3D display techniques like stereoscopy and autostereoscopy, a special feature of the holography is that it can shape not only the intensity but also the phase distribution of objects' wavefronts[1−4]. In traditional holograms, seemingly random patterns which contain phase and intensity information of objects are recorded on special photosensitive materials. The size and shape of the hologram recording surface determine the field of view and information capacity that can be extracted from objects. Wider diffraction angles can be obtained by using curved holograms[5]. However, optical synthesis of the curved holograms has several obstacles, because it is difficult to construct a coherent system for spherical or cylindrical holograms. Computer generated hologram (CGH), which is made by computer simulation, can be an ideal alternative. The interference between a reference wave and an object wave can be computed using wave propagation theory in a computer. Therefore, CGH enables us to obtain optical field of hologram with any shape and any size[6−12].

Recently, some of us reported an initial idea for computer generated spherical hologram (CGSH) of a real-existing object[7]. One of the advantages using the CGSH is that it can provide same quality reconstruction images in significantly lower bandwidth comparing with a planar hologram. Thus the required number of sampling points for a given field of view is much smaller than a conventional planar CGH. Another one is that it is able to provide omnidirectional perspective view images of 3D objects. Though the number of samples to calculate omnidirectional optical field of 3D objects in the CGSH is much smaller than in the case of using many planar recording surfaces, it still requires a large number of samples in the CGSH. Thus, huge amount of computation load is one of the main issues in curved CGH calculation. In order to solve this problem, several literatures associated with speed enhancement in curved hologram generation have been reported[8−12]. Sando *et al.*[9] proposed a typical acceleration method for calculating cylindrical CGH. In their method, the core idea is the use of fast Fourier transform (FFT) algorithm for cylindrical hologram, in which the diffraction integral for a cylindrical surface is calculated through the convolution method. Later, Tachiki *et al.*[10] proposed a similar idea for CGSH. In these methods, however, selection of objects has some limitations. In order to obtain the convolution form, the shape of objects must be equal to the shape of hologram. Moreover they should be concentric to each other. Jackin and Yatagai[11,12] proposed the wave propagation method in the spectral domain and defining 3D objects in the cylindrical and spherical coordinates. Although these methods, which are able to use FFT algorithm for obtaining high-speed computation, have significant advantages in non-planar CGH synthesis, they still suffer from several drawbacks, such as the use of real-existing objects and the error caused by approximation calculation. Since most of all electro-holographic displays are manufactured as planar type so far, precise optical reconstruction of curved CGHs is extremely difficult to carry out in the actual practice. Thus, curved CGHs have to use numerical experiments to verify the reconstruction results. It is, however, hard to devise a fast calculation that calculates optical field propagation between a curved and a planar surface, and there are few reports which deal with this

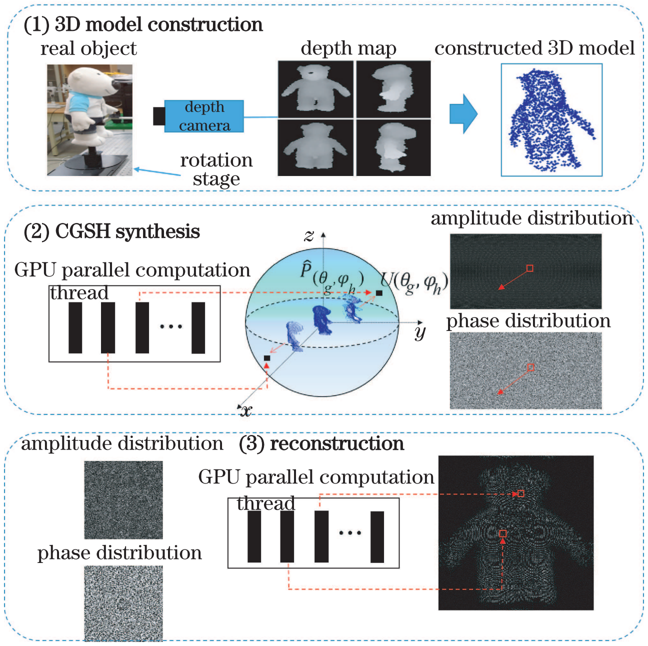


Fig. 1. (Color online) Overall procedure of the proposed method.

problem so far.

In this letter, we propose a fast calculation method for the CGSH in both hologram generation and numerical reconstruction processes on the graphics processing unit (GPU) platform. The GPU technique allows us to use a set of parallel calculations for every sampling point on the spherical hologram. Furthermore, low cost of GPU board and flexibility in both software modification and hardware extension can also be advantages of using the accelerating calculation.

The proposed method consists of three stages, and the entire procedure is shown in Fig. 1. Firstly, the 360 degree depth information of a real object is extracted by using a depth camera to capture several perspective images of the object[13]. Multiple point cloud sets from the corresponding depth maps are then transformed to be merged into a single 3D model in a common Cartesian coordinate system. Secondly, the CGSH for the object is obtained by proposed GPU based calculation. Finally, several partial regions of the synthesized CGSH are selected and numerically reconstructed to the corresponding observation plane using the GPU computation.

In the CGSH synthesis, occlusion effect should be considered correctly. In the proposed method, all of the hologram points play the role of viewpoints. Thus, only the visible object points which are extracted by using hidden point removal (HPR) method[14] are used to calculate the complex amplitude of the corresponding pixel in the CGSH. We denoted the visible points set and the corresponding hologram point as $\hat{P}(\theta_g,\ \phi_h)$ and $U(\theta_g,\ \phi_h)$, respectively. Calculation for the corresponding hologram point using the HPR is explained in detail in the previous work[7]. θ and ϕ indicate the latitudinal and azimuthal axis as shown in second part of Fig. 1. g and h indicate index of the corresponding variables. The optical field induced by $\hat{P}(\theta_g,\ \phi_h)$ is calculated using a ray-tracing method between the object points and the spherical hologram as

$$U(\theta_g,\phi_h)=\sum_{j}^{N_\mathrm{v}}\frac{A_j}{d_j}\exp\left(\frac{j2\pi d_j}{\lambda}\right), \qquad (1)$$

where

$$\begin{aligned} d_j &= \sqrt{X^2+Y^2+Z^2},\\ X &= r\sin\theta_g\cos\phi_h - x_j,\\ Y &= r\sin\theta_g\sin\phi_h - y_j,\\ Z &= r\cos\theta_g - z_j, \end{aligned}$$

is the distance between the object point $P_j(x_j,\ y_j,\ z_j)$ and the hologram point $U(\theta_g,\ \phi_h)$, λ is the wavelength,A_j is the intensity of j-th object point, and N_v is the number of the object points in the visible point set $\hat{P}(\theta_g,\ \phi_h)$.

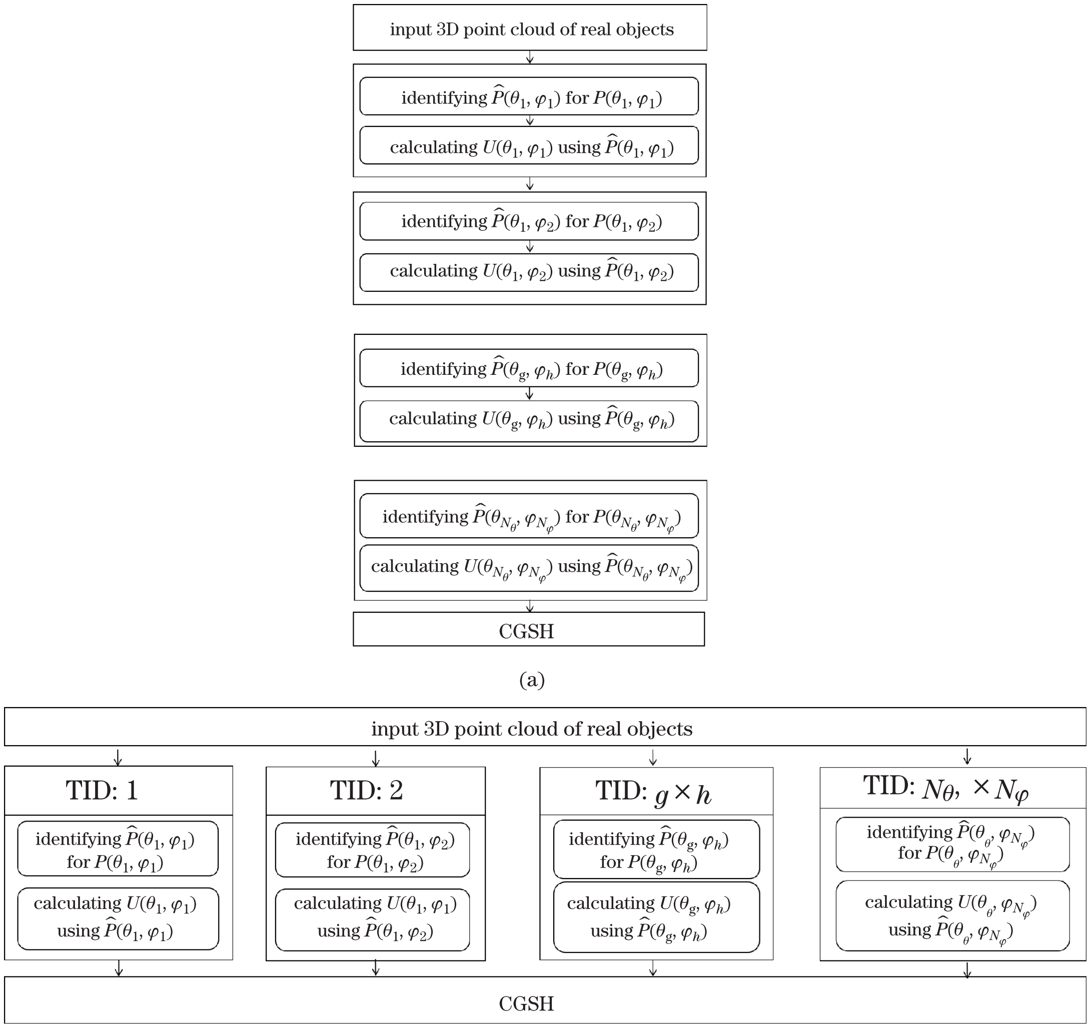
Figures 2(a) and (b) show flowcharts of the CGSH synthesis on central processing unit (CPU) and GPU, respectively. According to the big O notation, the total computational order of the conventional CPU based calculation method is $O = N_\mathrm{v} \times N_\theta \times N_\phi$, where N_θ and N_ϕ are the minimum number of necessary samples in θ and ϕ directions, respectively. N_θ and N_ϕ have been defined as

$$N_\theta = 2\pi\,|f_{\mathrm{L}\theta}|_{\max} = kr_{\max}, \qquad (2)$$

$$N_\phi = 4\pi\,|f_{\mathrm{L}\phi}|_{\max} = 2kr_{\max}, \qquad (3)$$

where ($|f_{\mathrm{L}\theta}|_{\max}$, $|f_{\mathrm{L}\phi}|_{\max}$) is a pair of the maximum local spatial frequency for phase of the optical field, k=$2\pi/\lambda$ is the wave number, and $r_{\max}$ is the radial distance of an object point which has longest distance from the coordinate origin. GPU parallel computation is very appropriate for the CGSH because computation for complex value of each pixel does not affect those on other pixels. Thus, the optical field induced from the 3D objects can be simultaneously calculated to all of the pixels in the spherical surface by using the proposed method. In the GPU based CGSH synthesis, it firstly needs to provide a reasonable amount of infrastructure to create the threads to execute a kernel function. The NVIDIA compute unified device architecture (CUDA)[15] technology can process more than thousands of threads simultaneously, and enables massively parallel execution of instructions. The kernel function plays the role of a connector that transmits all required parameters from host (CPU) to device (GPU). Required memory of the threads should be pre-assigned in the GPU. In the implementation, the amount of threads, which is equal to the total sample number $N_\theta \times N_\phi$, is allocated to guarantee completely parallel processing. As shown in Fig. 2(b), each thread has an identification thread ID (TID) in charge of computing the value of CGSH pixels. The large number of threads are fed to obtain the complex amplitude of all pixels in the CGSH in parallel. Therefore, the computational load is significantly decreased as $O = N_\mathrm{v}$. After obtaining the hologram field, the kernel function returns the whole information from the device to host.

In the same manner, we enhanced the calculation speed of the numerical reconstruction process using the GPU parallel computation. In this process, several small regions of the synthesized CGSH are arbitrarily selected



(a)

(b)

Fig. 2. Comparison of the operation flow between CPU and GPU based CGSH synthesis. (a) Conventional operation process on CPU and (b) proposed parallel operation process on GPU.

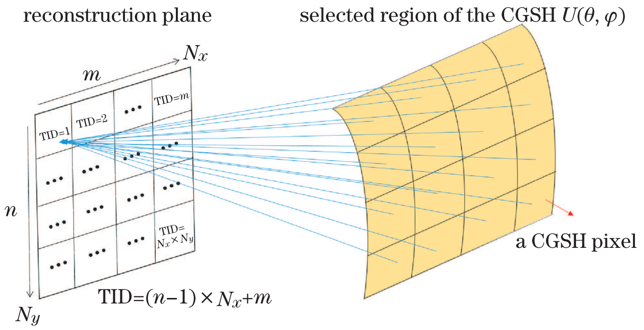


Fig. 3. (Color online) Configuration of the GPU based curved hologram reconstruction.

and back propagated to corresponding observation planes in order to verify accuracy of the CGSH. Generally, when hologram field is specified over a curved surface, optical field of the reconstruction plane requires one by one calculation, resulting in a cumbersome computation load. Hence, GPU parallel processing is also very effective for numerical reconstruction of curved holograms. Figure 3 shows the configuration of the reconstruction process using the proposed method. The reconstruction process between the curved plane and flat one is computed by assuming each element in the curved hologram is a spherical wave source. Note that, it is not an exact solution for calculating diffraction field between curved plane and flat one. But this is a reasonable paraxial approximation, in the sense that the extent of the reconstruction plane and the spherical hologram patch is very small in comparison with the radius of the spherical hologram. Hence, the reconstruction of the proposed method can be considered as a reverse operation of the hologram generation process. Thus, the complex value of a pixel in the reconstruction plane is calculated by accumulating back propagated spherical wave from pixels in the patch of the CGSH. All the threads with corresponding TID handle the complex value calculation of the corresponding pixel in the reconstruction plane simultaneously. In the end, the allocated device memories and the GPU resources should be completely released.

The proposed method is verified by some numerical experiments. The CGSH of a single object point is used for confirming the accuracy of the CGSH which is calculated on the GPU. We generated the spherical hologram with radius 1 cm from a single point when it is located at $P_1(x_1, y_1, z_1)$=(50 μm, 0, 0), $P_2(x_2, y_2, z_2)$=(0, 50 μm, 0) and $P_3(x_3, y_3, z_3)$=(0, 0, 50 μm), respectively, as shown in Fig. 4(a). Amplitude and phase distributions of the CGSH for the three cases are shown in Figs. 4(b) and (c), respectively.

Without loss of generality, the comparison of the elapsed time for different resolution CGSH, which is calculated from a single object point using the CPU and the GPU, is shown in Fig. 5. In the GPU computation, it needs to transmit all required parameters from the host to the device, and then return the obtained hologram data from device to host, after completing the calculation. Actually, the data transmission between CPU and GPU also spends time. As the hologram size increases, because the consuming time on data transmission occupies relatively small portion of total elapsed time, the performance of the GPU based implementation shows that it is remarkably faster than that of CPU based implementation. Hence the GPU parallel computation is very appropriate for accelerating a large size CGSH synthesis.

In the experiment, the constructed 3D model is composed of 14531 points while the size of the object is limited in the range from −0.14 to 0.14 cm in each x, y, and z axis. Radius of the spherical hologram is 9.57 cm. According to Eqs. (2) and (3), the number of entire samples in the CGSH is calculated as $21765(\theta)\times43530(\phi)$, thus angular pitches in θ and ϕ directions are both 1.4434×10^{-4} rad. By using the proposed method, the whole hologram computation takes 238 seconds, which is more than 1000 times faster than that of CPU based calculation for same resolution CGSH of same objects.

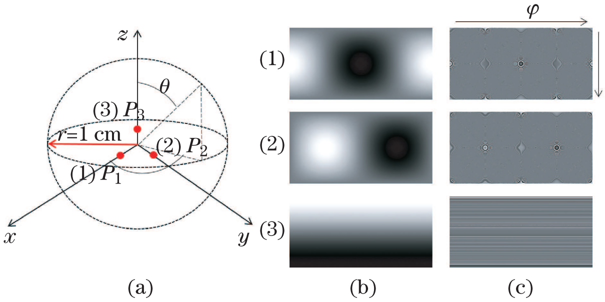


Fig. 4. (Color online) (a) Position of the object point and hologram surface, (b) amplitude distribution and (c) phase distribution of the CGSH.

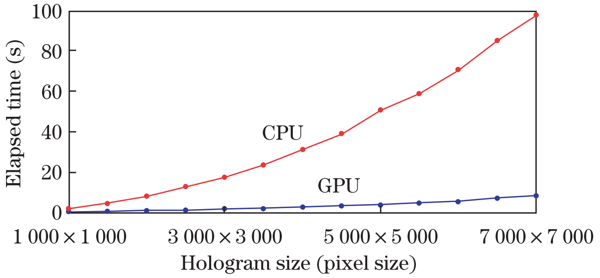


Fig. 5. (Color online) Comparison of the calculation speed between CPU based and GPU based CGSH of a single object point.

Table 1. Computing Environment and Parameters

Item	Model	Details
CPU	Intel Core i7-4770	1 CPU, 8 cores, 3240 GHz
GPU	NVIDIA GTX 650	1 GPU 384 CUDA cores and 1 GB memory
OS	Windows 7	64 bits
IDE	Visual Studio 2010	64 bits
GPU API	NVIDIA CUDA	Ver. 5.5
Depth-camera	XBOX 360 KINECT Sensor	C# Program to run KINECT SDK

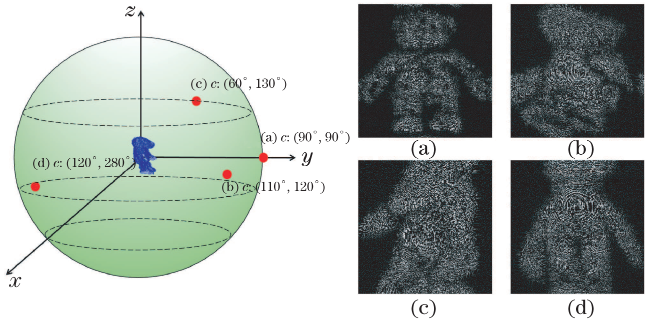


Fig. 6. (Color online) Reconstruction images of partial regions of the CGSH using GPU parallel computation. Center position of selected curved holograms: (a)c:(90°, 90°), (b) c:(110°, 120°), (c) c:(60°, 130°) and c:(120°, 280°).

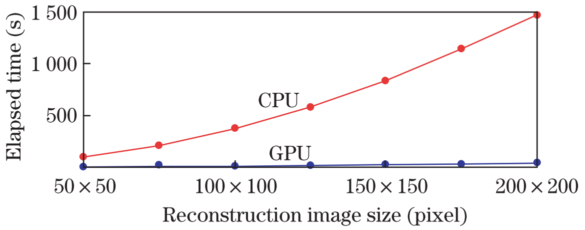


Fig. 7. (Color online) Comparison of calculation speed of CPU based and GPU based numerical reconstruction.

In the reconstruction process, size of the selected regions is $150(\theta)\times150(\phi)$, the size of the corresponding reconstruction plane is 256×256 and the pixel pitch is 13.5 μm, thus the width and height of the reconstruction plane are both 3.45 mm. Because the area of the reconstruction plane is bigger than that of the curved hologram, the higher order reconstruction images also appear in the result. For this reason, we crop some higher order parts that overlap with the image we want to obtain in Figs. 6(c) and (d). The specifications of the environment for the computer simulation are listed in Table 1.

Figure 6 shows the reconstruction images from the selected curved holograms with center position $c(\theta, \phi)$, where positions of the object and the spherical hologram are as shown in second part of Fig. 1. It demonstrates that the CGSH indeed contains omnidirectional information of the real object and it is able to reconstruct corresponding perspective of the object successfully.

Figure 7 shows the comparison of the reconstruction speed based on CPU and GPU, in the case of reconstructing 150×150 resolution curved hologram to the reconstruction plane with different sizes. In the CPU based reconstruction, the total computation order is $N_{c\theta}\times N_{c\phi}\times N_{rx}\times N_{ry}$. In contrast, using the proposed GPU based method enables us to calculate each pixel of reconstruction plane simultaneously. Thus, the computation order is reduced to $N_{c\theta} \times N_{c\varphi}$. The $N_{c\theta}$ and $N_{c\varphi}$ are the numbers of the samples of the curved hologram in θ and ϕ directions; N_{rx} and N_{ry} are the numbers of the samples of the reconstruction plane. As shown in Fig. 7, for the same resolution curved CGH, by increasing the reconstruction plane resolution, the parallel computation effect is significantly enhanced. Note that, because one GPU is used in the computation process, only the pixels of reconstruction plane are calculated in parallel. However, the operation that accumulates the back propagated spherical wave from points of hologram to one pixel of the reconstruction plane, has to be conducted sequentially. Therefore if multiple GPUs are used in this process, it is able to decrease the computational order of reconstruction process to 1.

In conclusion, GPU based speed enhancement algorithm in the computation of CGSH and the numerical reconstruction process is proposed. It is 1000 times faster than CPU based implementation for synthesizing a $21765(\theta)\times 43530(\phi)$ size CGSH; and 35 times faster for reconstructing $150(\theta)\times 150(\phi)$ size curved hologram which is a patch of the CGSH to a 256×256 size reconstruction plane. The proposed method is able to alleviate the problem of huge amount computation load in the CGSH synthesis which contains 360° field of view of real-existing objects, and enables us to reconstruct any perspective view images of the object more rapidly. Furthermore, if more than one GPU can be used in the reconstruction process, it is able to provide a real-time numerical reconstruction of the CGSH. We believe that it can be an appropriate alternative of providing CGH contents for omnidirectional holographic display.

This work was supported by the IT R&D program of MSIP/KEIT (Fundamental technology development for digital holographic contents).

References

1. B. Lee, Phys. Today **66,** 36 (2013).
2. J. Hong, Y. Kim, H.-J. Choi, J. Hahn, J.-H. Park, H. Kim, S.-W. Min, N. Chen, and B. Lee, Appl. Opt. **50,** H87 (2011).
3. H. Kim, J. Hahn, and B. Lee, Appl. Opt. **47,** D117 (2008).
4. H. Kim, K. Choi, and B. Lee, Jpn. J. Appl. Phys. **45,** 6555 (2006).
5. O. D. D. Soares and J. C. A. Fernandes, Appl. Opt. **21,** 3194 (1982).
6. M. Lucente, "Interactive computation of holograms using a look-up table," J. Electron. Imaging **2,** 28 (1993).
7. G. Li. A.-H. Phan, N. Kim, and J.-H. Park, Appl. Opt. **52,** 3567 (2013).
8. G. Li, K. Hong, N. Chen, J.-H. Park, N. Kim, and B. Lee, in *Proceedings of Internal Conference Optics and Photonics Taiwan 2013 (OPTIC 2013)* 2013-FRI-S0402-O003 (2013).
9. Y. Sando, M. Itoh, and T. Yatagai, Opt. Express **13,** 1418 (2005).
10. M. Tachiki, Y. Sando, M. Itoh, and T. Yatagai, Appl. Opt. **45,** 3527 (2006).
11. B. J. Jackin and T. Yatagai, Opt. Express **18,** 25546 (2010).
12. B. J. Jackin and T. Yatagai, Opt. Express **21,** 935 (2013).
13. G. Li, K.-C. Kwon, G.-H. Shin, J.-S Jeong, K.-H Yoo, and N. Kim, J. Opt. Soc. Korea **16,** 381 (2012).
14. S. Katz, A. Tal, and R. Basri, ACM Trans. Graph. **26,** 24 (2007).
15. http://www.nvidia.com.