Realistic expression for full-parallax computer-generated holograms with the ray-tracing method

Tsubasa Ichikawa,* Kazuhiro Yamaguchi, and Yuji Sakamoto

Graduate School of Information Science and Technology, Hokkaido University,
North 14 West 9, Sapporo 060-0814, Japan

*Corresponding author: ichikawa@ist.hokudai.ac.jp

Received 13 August 2012; revised 18 October 2012; accepted 19 October 2012;
posted 19 October 2012 (Doc. ID 174090); published 16 November 2012

This paper presents a calculation method of computer-generated holograms that involves removing the hidden surface and provides realistic rendering. The method was based on the ray-tracing method that simulates rays traveling paths. Rays are cast from every elementary hologram into virtual objects and then the traveling paths of the rays are determined. Since the method is considering intersection with objects, absorption, reflection, and refraction, the method is capable of rendering realistic images. Multiple reflections and refraction are expressed by casting additional rays into the reflection direction and the transmission direction and calculating the length of the light path. To express the quality of materials, the Phong reflection model and Cook–Torrance reflection model were used. Results of optical reconstructions show that the hidden surface removal was conducted. Moreover, the texture of material appeared as well as other effects by the proposed method. © 2012 Optical Society of America

OCIS codes: 090.0090, 090.1760.

1. Introduction

We have recently seen highly realistic images in movies and computer games produced by using computer graphics (CG), and images created by three-dimensional (3D) display technologies are currently attracting rising attention too. Various 3D display systems such as stereography and integral photography have been proposed. However, these systems cause fatigue when viewed for a long time because of a mismatch between vergence and accommodation. Holography is an ideal 3D display technology. Holography fills a human's visual features to perceive 3D images because a hologram is recorded to the information not only on the amplitude of light but also on a phase that is unlike a photograph. Holography is currently used in diverse fields such as 3D displays, security, holographic memory, and medicine. To generate holograms, a wide range of devices including lasers and a dark room are needed. Recording holograms requires great care.

Computer-generated holograms (CGHs) calculate an interference pattern by simulating the definition of virtual objects, propagation, and diffraction of light waves with a computer [1]. The interference patterns are displayed on a spatial light modulator (SLM) or photographic plates. A key feature of CGHs is that virtual objects are used for recording holograms. The configuration and location of objects are freely defined, and imaginary images can be generated. However, rendering techniques for displaying realistic images such as those used in CG are inadequate. In general, when 3D images are rendered, the hidden surface needs to be removed. In addition removing, the hidden surface removal in accordance with viewpoint motion is important because CGHs have parallax. Several methods for doing this have been proposed, including the wave optics ways and the geometric optics ways. In wave optics ways, light

1559-128X/13/01A201-09$15.00/0

waves of objects in the background are cut off by the silhouette mask of objects in the foreground. These methods treat virtual objects as polygon models, so a complicated program needs to be constructed [2,3]. Approaches using geometric optics involve methods that apply the Z buffer method in the field of CG to CGH [4,5]. However, this method is not suitable for expressing realistic images including multiple reflection, refraction, and shadowing. Another method for removing the hidden surface uses a holographic stereogram [6,7]. A hologram is made in the holographic stereogram by combining multiview images with only intensity information. These methods cannot display the scene with deep depth.

In addition, various rendering techniques to express realistic CGH images have been proposed, such as shadowing and reflectance distribution [8]–[10], refraction [11], and multiple reflection [12]. However, each method is suitable only for a specific rendering technique, and combining these methods is problematic.

We thus propose a way to calculate CGHs that is implemented in the hidden surface removal and provides realistic rendering by using the ray-tracing method. In the field of CG, the ray-tracing method is used to express realistic images including hidden surface removal, as well as refraction and multireflection [13,14]. The ray-tracing method is not used in CGHs with a very high resolution because of the enormous calculation cost. We then accelerated the ray-tracing process using a graphics processing unit (GPU) to calculate light waves.

We report on how to apply the ray-tracing method to CGH having free viewpoints. In Section 2, we explain how to create a full-parallax CGH with the ray-tracing method, and we also explain how this method is used to create CGHs with free viewpoints using the ray-tracing method. In Section 3, we introduce a rendering technique to express realistic images for CGH by using the proposed method. Here, expression methods, i.e., shading, shadowing, multiple reflection, and refraction, are explained. Our calculation systems are described in Section 4. In addition, we introduce the algorithm of our method used in GPU computing. To determine whether our method is adequate, we conducted the optical reconstructions and computer simulations. Results of the experiments are shown in Section 5, and it was found that the images were realistic because they had various renderings and that the depths of virtual objects were correctly set. Moreover, the computation time for generating holograms is considered, and the improvement of the algorithm is considered. The paper is summarized in Section 6.

2. Full-Parallax CGH

We introduce how to implement the hidden surface removal by using the ray-tracing method. A flow chart of the proposed method is shown in Fig. 1. The proposed method generates a full-parallax CGH by calculating light waves from the point light source groups observable from each element hologram. Formats of virtual objects are not restricted, so not only a polygon model but also various objects expressed by clouds of points, a voxel model, and function representations are usable. A detailed flow of the proposed method is as follows.

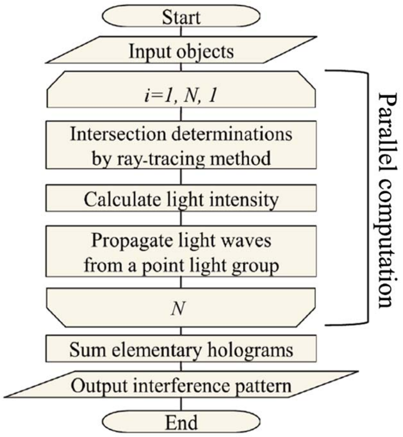

Start
Input objects
i=1, N, 1
Intersection determinations by ray-tracing method
Calculate light intensity
Propagate light waves from a point light group
N
Sum elementary holograms
Output interference pattern
End
Parallel computation

Fig. 1. (Color online) Flow chart of proposed method.

First, a hologram plane is divided into elementary holograms, and the center of each of these is assigned a ray-tracing viewpoint. For every elementary hologram, the intersection is determined as shown in Fig. 2. So each elementary hologram has a peculiar point light source group. Then CGHs generated by the proposed method have discrete viewpoints. In this method, the continuity of parallax depends on the width of the elementary holograms. Here, the continuity of the parallax is the smoothness of a motion of the parallax at the time the viewpoint is moved. The parallax between elementary holograms increases with the size of an elementary hologram. The sampling interval of rays and size of elementary holograms for the ray tracing are shown in Fig. 3. The size of elementary hologram M is defined as 1.216 mm (128 pixels). Results of research on CGH generated by multiview images [15] suggested that if the size is about 1 mm, observers do not feel a jerky motion of reconstructed images. Here, θ denotes the angle at which frequency is limited, and depends on the pitch of the output device. By casting rays into

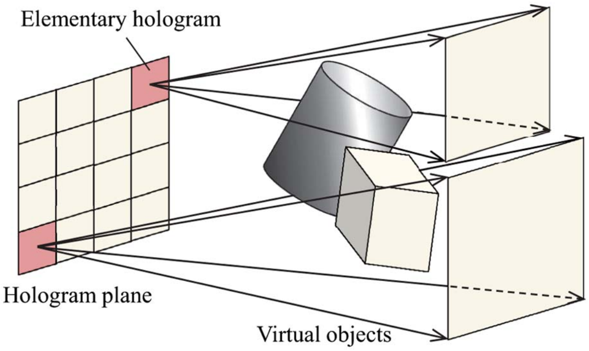


Fig. 2. (Color online) Ray tracing from each elementary hologram.

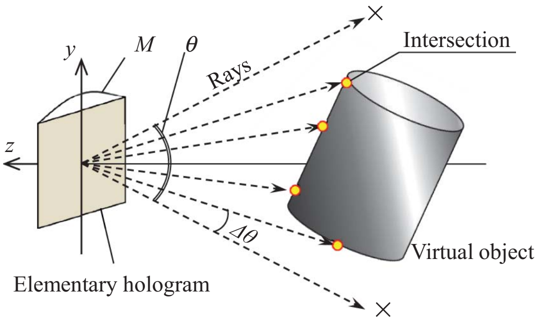


Fig. 3. (Color online) Intersection determination.

the area limited by θ, ghost images that appear by the aliasing of the interference pattern are not shown. This also leads to a lower calculation cost of the ray-tracing process. The interval between rays $\Delta\theta$ is significant for seeing a smoothly reconstructed image. If the interval is too large, point light sources separate and do not appear as a plane. Conversely, if it is too small, the calculation cost becomes enormous because of the increase in the number of point light sources. This is because the resolution of the human eye is about $1/60$ deg, and the angle between rays is the same. Therefore, the gap between the intersections cannot be distinguished by the human eye. This enables us to express a plane with the minimum number of point light sources.

The intersections are determined by using the ray-tracing method. The ray-tracing method obtains intersection by tracing a ray from the viewpoint to the light sources. By reflecting or refracting the ray when a ray hits the object surface, the ray-tracing method can express the reflection, refraction, and so on. In the ray-tracing process, a ray obtains information of the brightness and the coordinates of the intersection nearest the hologram plane. The calculation of light brightness of intersections is described in Section 3. The detailed computational algorithm of the ray tracing is described elsewhere [13,14].

After the intersection determination, light waves on an elementary hologram from a group of point light sources are calculated (Fig. 4). We use the point-wise methods, which assume virtual objects are an assemblage of point light sources [16]. The calculation uses the intensities and coordinate information of each point light source. In the holographic stereogram, a hologram is generated by combining multi-view images that have information only on the intensity. While expressing the scene with sufficient depth in one viewpoint is usually difficult, the proposed method can be used to express depth perception by using the depths of the virtual objects in addition to the intensities of intersections. Complex amplitude distributions on a hologram plane $h_m(x,y)$ are calculated by using

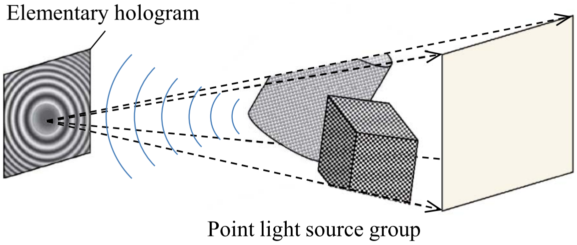


Fig. 4. (Color online) Light propagation from a point light source group.

$$h_m(x,y) = \sum_{i=1}^{N_m} \frac{A_i}{r_i(x,y)} \cdot \exp\left(j\frac{2\pi}{\lambda} r_i(x,y)\right) \cdot \exp(j\phi_i), \tag{1}$$

$$r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + z_i^2}, \tag{2}$$

where parameters x, y, and z represent the horizontal, vertical, and depth components while indices m and i respectively indicate each elementary hologram and point light source. Here, λ is the wavelength, and ϕ_i is the initial phase of each point light source. The number of point light sources that compose 3D objects in each elementary hologram is given by N_m. The amplitude of the point light source A_i is calculated using a shading process (Section 3). Finally, whole light waves on a hologram plane are obtained by lining up elementary hologram light waves.

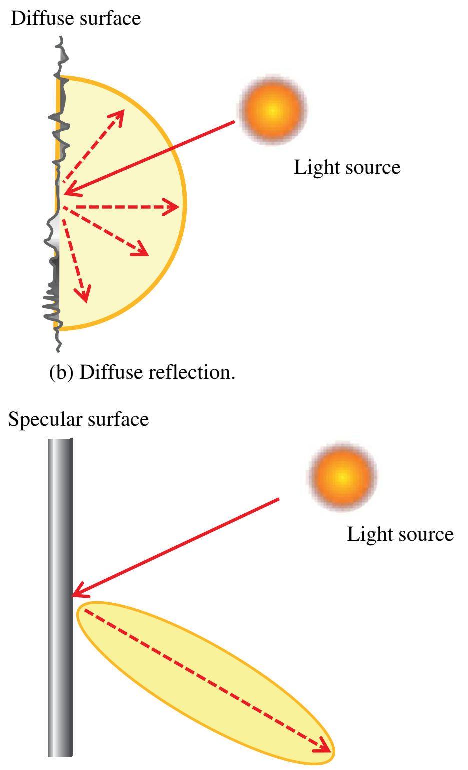
3. Rendering Techniques

This section describes the rendering techniques used to express realistic images. Some rendering techniques have been used with numerous methods for generating CGHs. However, while these methods are independent, they cannot be easily combined. Our method is more flexible, and it enables us to use a wide variety of rendering techniques. The algorithm for each rendering technique used in this study is introduced below.

A. Shading

Shading methods that give a spatial effect by adding contrast and reflectance distributions of light are determined by the material quality and express the colors and the texture of the objects in the scene. Reflection on an object surface is divided into three components: diffuse, specular, and ambient. The brightness on the object surface is determined by overlapping these components.

A diffuse reflection and a specular reflection are shown in Figs. 5(a) and 5(b). A diffuse reflection appears on object surfaces without gloss. The intensity distributions of the reflected light form a semicircle with a constant strength that centers on the incident point [Fig. 5(a)]. A specular reflection is strongly reflected in the specular direction of the light incident to the normal like a mirror surface. Therefore, the reflectance distributions of specular reflection tend to increase in a specular direction [Fig. 5(b)]. Shading is done by changing the proportion of specular reflection to the diffuse reflection or by forming the shape of reflectance distributions of the specular reflection.



(b) Diffuse reflection.

(a) Specular reflection.

Fig. 5. (Color online) Reflectance distributions.

The light intensity of intersection I_r is determined by using the normal vector of the surface, the viewpoint, and so on. Reflectance distributions in CGH are expressed by setting the intensity of reflected light I_r to the intensity of the point light source in Eq. (1). When a diffuse reflection is defined as a Lambert reflection, the intensity of the reflected light is calculated by using the following equation:

$$I_r = I_a\rho_a + (\mathbf{N}\cdot\mathbf{L})k_d\rho_d + k_s\rho_s, \tag{3}$$

where I_a is the intensity of ambient light, k_d and k_s are respectively the ratio of Lambert light and specular light ($k_d + k_s = 1$), and values ρ_a, ρ_d, and ρ_s are the reflectance of ambient, Lambert, and specular light, respectively. Vectors **N** and **L** are a normal unit vector and a unit vector to the light source. Material quality is mostly expressed by changing the reflectance of the specular light ρ_s. We used the Phong model [17] to express plastic objects and the Cook–Torrance model [18] to express metal surfaces.

The Phong reflection model calculates the brightness using the direction of the ray and position of the light source that illuminates the objects (Fig. 6). In the Phong model, the reflectance of a specular light is

$$\rho_s = (\mathbf{R}\cdot\mathbf{V})^{\alpha} i_s, \tag{4}$$

where i_s is the chroma of the specular reflection in each light source. Here, α signifies the specular gloss and determines how much light is reflected at the point. The **R** is the vector of specular direction from the illumination light to the object surface. The **V** is the vector from the incident point on the surface to the viewpoint. These vectors are illustrated in Fig. 6. When α is large, the object is expressed as if a mirror reflects a light.

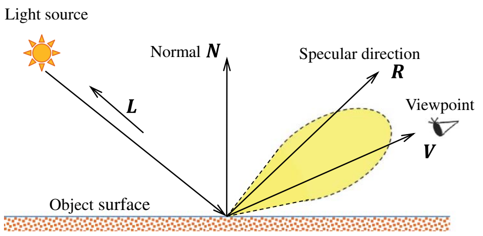


Fig. 6. (Color online) Phong reflection model.

In addition, we used the Cook–Torrance reflection model, which is suitable for expressing metallic objects. The Cook–Torrance model is based on physics and is formulated using thermal emission logic. The Cook–Torrance model assumes that an object surface is an aggregate of minute surfaces and that the direction of the minute surface is distributed centering on the normal direction of a surface. The Cook–Torrance model takes into account that reflectance distributions depend on the wavelength and incident angle of light, and the reflectance is calculated by using the Fresnel equation. The reflectance by the Cook–Torrance model ρ_s is given by

$$\rho_s = \frac{F}{\pi}\frac{DG}{(\mathbf{N}\cdot\mathbf{V})}. \tag{5}$$

The Fresnel term F describes how light is reflected from each smooth microfacet and is given by

$$F = \frac{1}{2}\left[\frac{g-c}{g+c}\right]^2 \left\{1 + \left[\frac{c(g+c)-1}{c(g-c)+1}\right]^2\right\}, \tag{6}$$

where $c = \sin\theta_1$ and $g^2 = n^2 + c^2 - 1$. Here θ_1 is the incident angle from the light source to the object surface, and n is the relative index from refraction of the object surface. In Eq. (5), D stands for a distribution function, and is defined by the Beckmann distribution with

$$D = \frac{1}{4m^2\cos^4\xi}\exp\left(\frac{\tan\xi}{m}\right)^2. \tag{7}$$

Here, m is the coefficient that is used to adjust the roughness of the surface. As m becomes smaller, the surface becomes smoother and more specular. The ξ is the angle between the normal vector of the object surface and a half vector (Fig. 7). Light intensities of intersections on a specular surface are obtained by the above equations. The geometrical attenuation factor G accounts for the shadowing and masking of one face by another.

B. Shadowing

Shadowing expresses the shadow made by masking objects between light sources. Shadowing is

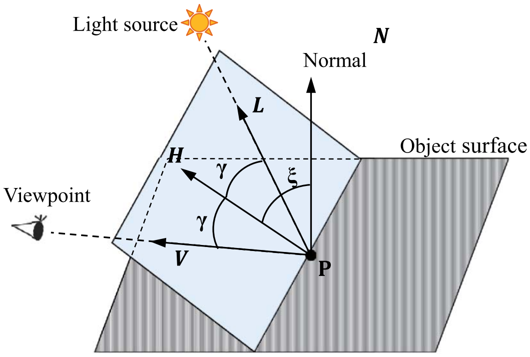


Fig. 7. (Color online) Cook–Torrance reflection model.

implemented by additionally casting the ray in the direction of the light source from the intersection, as shown in Fig. 8, using the ray-tracing method. When a ray does not hit an object, like shadow ray$_1$, Eq. (3) is calculated with both the diffusion and specular components. When there are masking objects between the light source, like shadow ray$_2$, the shadow made by the brightness is calculated with only the ambient component.

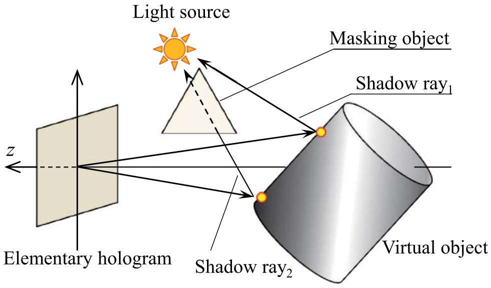


Fig. 8. (Color online) Casting a shadow ray.

C. Multireflection

Our method for representing specular surfaces with reflected images in CGH is explained here. Reflected images are mostly expressed by mapping the texture on the object surface as intensities of point light sources. However, the depth of reflected images is not correctly reconstructed. Thus, we propose a method that takes the length of the light path of reflected images into consideration.

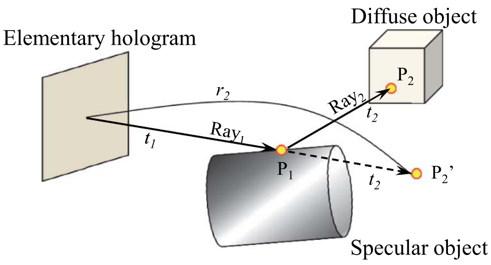


Fig. 9. (Color online) Expression of multiple reflections.

A simple graphic explanation of the proposed method is shown in Fig. 9. First, a ray (Ray$_1$) is cast from the center of an elementary hologram in the intersection determination process. When a ray hits the specular object, the length $r_1 = t_1$ of the intersection P_1 is calculated, and the intensity and the coordinate of P_1 are stored in a buffer. A ray (Ray$_2$) is additionally cast in the specular direction from the intersection. If the ray hits other diffuse objects or background, the length of light path r_2 from the viewpoint to the diffuse object P_2 is obtained. Here, if the ray hits the specular surface again, the same process is repeated until an additional ray hits the diffuse surface. The length of light path is calculated by using

$$r_I = \sum_{i=1}^{I} t_i, \tag{8}$$

where I is the number of reflections. Here, since a reflected ray advances in the vacuum, the refractive index around a ray is 1. The refractive index n is thus disregarded in Eq. (8). The coordinate of P_2, a reflected figure, is stored as P_2' at a distant r_2 in the direction of ray$_1$. Furthermore, the reflectance is set up on a specular surface, and the intensity of intersection P_2' is obtained by multiplying the reflectance.

D. Refraction

The ray-tracing method can be used to obtain an intersection through transparent objects by bending the ray on a surface using Snell's law. However, the refraction is not expressed by only putting other intersections to point light sources. To express the correct depth of intersections through transparent objects, it is necessary to calculate the length of the light path exactly as in the case of multiple reflections. In the case shown in Fig. 10, the length of light path r_t to the opaque object is calculated by using

$$r_t = n_1 \cdot t_1 + n_2 \cdot t_2 + n_3 \cdot t_3, \tag{9}$$

where n_1, n_2, and n_3 are the refractive indices of each medium through which the light passes. Refractive phenomena are captured by calculating a light wave from a point light source located at P_t'.

Furthermore, attenuation of light in the medium should be considered. Attenuation is calculated by using the Beer–Lambert law, which formulates absorption of the light by a medium. The implementation of the Beer–Lambert law is shown in Fig. 11. Here, I_0 is the intensity of light incident to the medium, and I_1 is the intensity of light when light passes inside the medium length, L. The Beer–Lambert law then provides an equation of the relation between I_0 and I_1:

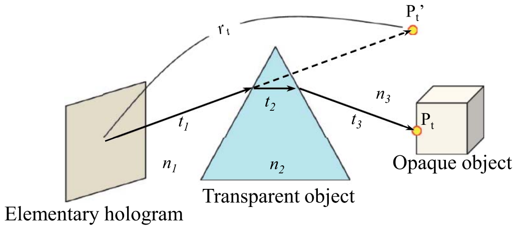


Fig. 10. (Color online) Refraction of the light ray.

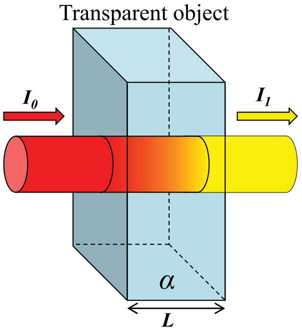


Fig. 11. (Color online) Attenuation of the light.

$$\log_{10}\left(\frac{I_1}{I_0}\right) = -\alpha L, \tag{10}$$

where α represents the absorption coefficient. From Eq. (10), we can see that the intensity of the intersection is attenuated by length t_2 until the light reaches the viewpoint. When light waves on a hologram plane are calculated from intersection P'_t, the intensity is attenuated beforehand. Intensity I_t is thus defined by

$$I_t = I_r \cdot 10^{-\alpha \cdot t_2}, \tag{11}$$

and here I_r is the intensity of the light of the opaque object calculated by Eq. (3).

4. Calculation System

A. GPU Programming

Our calculation system uses a CPU and a GPU. The chief cause of fast calculation is parallelization by using the GPU. There are two main types of GPU modules: the ray-tracing kernel for the ray tracing and the CGH kernel for calculating hologram data. To parallelize CGH calculation with the GPU calculations, we used the CUDA programming environment (NVIDIA). The CUDA SDK version is 4.0. To accelerate calculation of the ray-tracing process with the GPU, we used the OptiX Application Acceleration Engine. OptiX is a real-time ray-tracing engine for CUDA-based video cards. Optix supports intersection determinations in the ray-tracing process with the GPU. Using OptiX, we created the ray-tracing kernel applied to CGH. The GPU we used is GeForce GTX580, and its main specifications are shown in Table 1. The calculation flow for CGH with GPU is shown in Fig. 12.

B. Implementation of CGH Algorithm

First, the CPU invokes an Optix initialization function to prepare ray tracing. In a GPU, virtual object data are stored in global memory, and contexts are created such as illumination light sources, casting range of rays, ray type, number of rays, and setup of the material quality. While the ray-tracing process in the GPU is being initialized, the CPU prepares point light buffers that store information on the point light sources for each elementary hologram.

Table 1. Specifications of the GPU

GPU	NVIDIA GTX580
CUDA cores	512
Processor clock	1544 MHz
Memory clock	2004 MHz
Video memory interface width	384 bit
Memory type	3 GB GDDR5

The ray-tracing kernel is then launched. The ray-tracing kernels are iteratively invoked as many times as there are elementary holograms by the CPU. In the ray-tracing kernel, rays are cast in parallel with each core. Each ray determines an intersection provided by the OptiX. Next, the intensity and the length of the light path is calculated at an intersection. As explained in Sections 3.C and 3.D, when the multireflection or the refraction occurs, a ray is additionally cast. The ray-tracing kernel stores the ray data in the output buffer and sends the result of intersection determination to the point light buffer of the host.

Finally, the light wave is calculated by using the pointwise method. The CPU invokes the CGH kernel and sends the point light source data to the kernel. In the CGH kernel, light waves on elementary holograms are calculated by using Eq. (1). Pixels on the hologram plane are parallelized by CUDA cores. Light waves on a hologram plane are obtained by summing up the elementary holograms. Reference lights are added, and a hologram by the proposed method is generated.

5. Experiment

We conducted experiments to determine whether the proposed method is adequate. The experimental parameters are shown in Table 2. Interference patterns of holograms were displayed on an SLM. As a reference light, we used an LED, and the intense light of a reference light appeared as the zeroth-order light at the top of all reconstructed images. In this study, a device that extinguishes zeroth-order light was not implemented in the reconstruction optical system.

A. Full-Parallax CGH

An experiment to confirm the motion parallax and shadowing demonstrated that full-parallax CGH was displayed by locating the cube in front and a checkerboard in the back. The checkerboard and the cube were made of diffuse surfaces. These objects were illuminated by a point light source.

Reconstructed images taken from three viewpoints (left, center, and right) using a digital camera are shown in Fig. 13. It is difficult to observe the motion parallax with only a front cube because the viewing angle is narrow. However, the position of the cube in relation to the checkerboard changed with the viewpoint. In Fig. 13(a), the shadow of the cube was generated on the checkerboard. Moreover, even if the viewpoint changed, the reconstructed images of multiple objects did not overlap or were not missing.

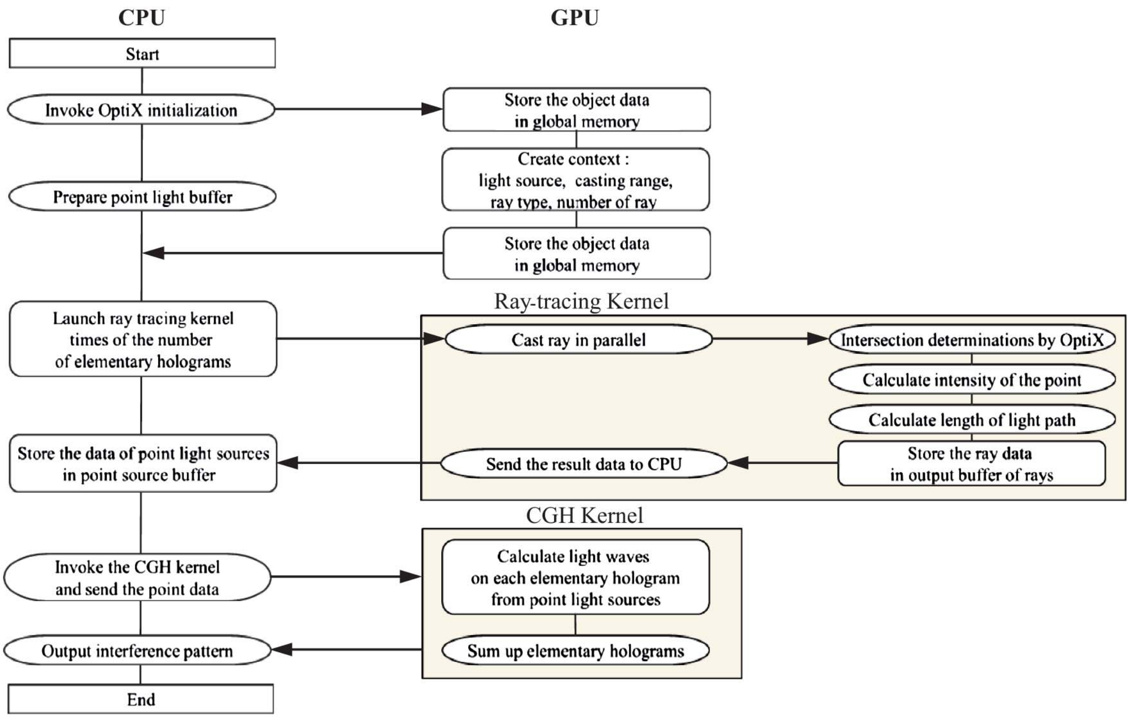

CPU
GPU
Start
Invoke OptiX initialization
Store the object data in global memory
Create context : light source, casting range, ray type, number of ray
Prepare point light buffer
Store the object data in global memory
Ray-tracing Kernel
Launch ray tracing kernel times of the number of elementary holograms
Cast ray in parallel
Intersection determinations by OptiX
Calculate intensity of the point
Calculate length of light path
Store the data of point light sources in point source buffer
Send the result data to CPU
Store the ray data in output buffer of rays
CGH Kernel
Invoke the CGH kernel and send the point data
Calculate light waves on each elementary hologram from point light sources
Output interference pattern
Sum up elementary holograms
End

Fig. 12. (Color online) Block diagram of calculation for CGH.

Table 2. Setup Parameters for Experiment

Number of elementary holograms	$32(H) \times 16(V)$
Size of elementary holograms	1.216 × 1.216 mm
Number of rays of each elementary hologram	161 × 161
Number of pixels	4096 × 2048 pixels
Sampling pitch	9.5 × 9.5 μm
Wavelength	632 nm

When we observed a CGH while moving the viewpoint, there was a satisfactory continuity of the parallax. Therefore, the proposed method enabled the hidden surface to be removed.

B. Reflection

We checked whether the image reflected by a mirror would be reconstructed at the correct position. The arrangement of virtual objects is shown in Fig. 14. There were two checkerboards, one on a hologram plane and the other one 300 mm away from the hologram plane, and the mirror was placed 200 mm from the hologram plane. The two checkerboards were made of diffuse surfaces by using the Phong reflection model, and a mirror had a specular surface.

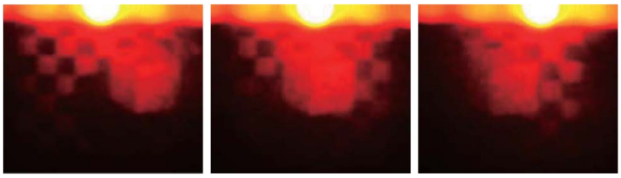

(a) Left (b) Center (c) Right

Fig. 13. (Color online) Reconstructed images of a box and check.

Reconstructed images taken by each distance are shown in Fig. 15. The f in the caption of Fig. 15 stands for the focal length from a hologram plane. Since the image shown in Fig. 15(a) was focused on the virtual mirror, the edge of the mirror was clear. On the other hand, the checkerboard behind the mirror and the reflected image of the checkerboard on a hologram plane were blurred. As shown in Fig. 15(b), the focal length was 30 cm from a hologram plane. The checkerboard behind the mirror was thus focused on, and the form of the mirror was blurred and not identified. As shown in Fig. 15(c), as the checker pattern reflected on the mirror was focused on, a different checker offset 45 deg from the checkerboard behind the mirror was clear. These results show that the image reflected in the mirror is reconstructed at the correct position. Therefore, the

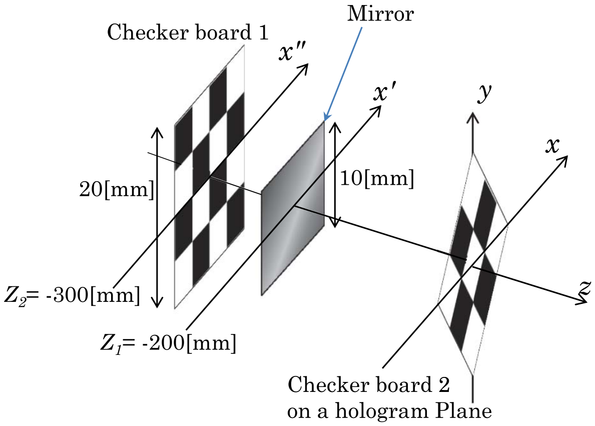


Fig. 14. (Color online) Experimental geometry of reflection.

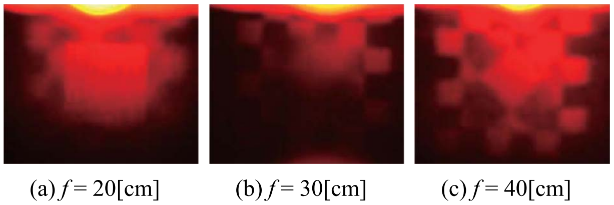

(a) $f = 20$[cm] (b) $f = 30$[cm] (c) $f = 40$[cm]

Fig. 15. (Color online) Reconstructed images of each distance.

proposed method can be used to express reflection and reflected images with the correct depth.

C. Refraction

To see magnified or reduced images through the transparent object, we conducted optical reconstructions of CGHs with refraction. The setup of this experiment is shown in Fig. 16. A forward sphere was made of transparent materials, and the free space was a vacuum.

The differences in the reconstructed images for various refractive indices are shown in Fig. 17. The refractive index of the sphere was the same as that in air (1.000292). The checker pattern through the sphere was also reconstructed for the same magnification because the refractive index of air is nearly 1. In Fig. 17(b), the checker pattern through the transparent object was magnified. Because the refractive index was 1.33, the focal point of the sphere is behind the checkerboard and the magnified virtual image was reconstructed. Moreover, the refractive index shown in Fig. 17(c) was higher, so the image was distorted at the edge of the sphere. A glass sphere with the same refractive index as shown in Fig. 17(c) was photographed in the same situation [Fig. 17(d)]. A comparison with Figs. 17(c) and 17(d) demonstrates that almost the same result was obtained.

To confirm the motion parallax, we took reconstructed images from three viewpoints (left, center, and right) As shown in Fig. 18, the refractive index was 1.06. The portions that expanded through the transparent object differ. Motion parallax of CGH with transparent objects was correctly expressed. These results indicate that refraction by the transparent object was achieved by using our proposed method.

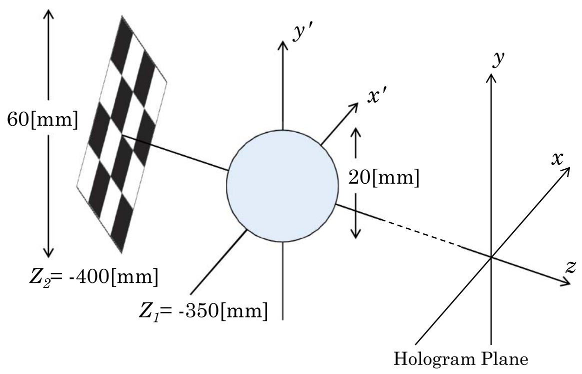
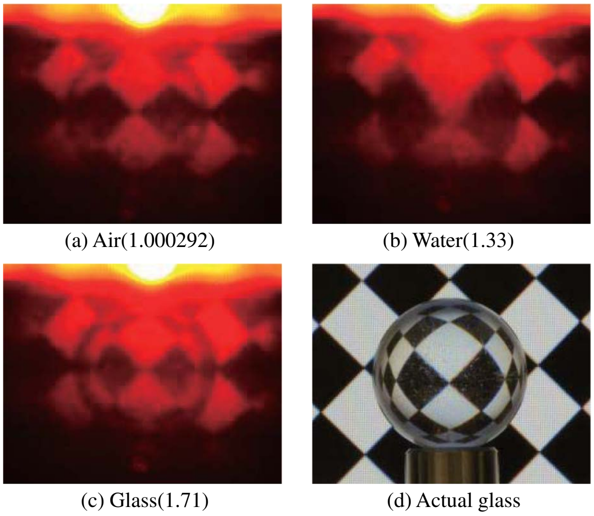


Fig. 16. (Color online) Experimental geometry of refraction.


(a) Air(1.000292) (b) Water(1.33)

(c) Glass(1.71) (d) Actual glass

Fig. 17. (Color online) Reconstructed images of each refractive index.

D. Complex Scene

Finally, we used our method to do an optical reconstruction with a complex scene. The virtual image used in this experiment is shown in Fig. 19(a). There are two spheres that are made of specular surface and a transparent object on the checkerboard. To express this scene, many rendering techniques were needed. The reconstructed image is shown in Fig. 19(b). When the rear sphere was focused on, the sphere has a highlight as a result of using the Cook–Torrance model and reflected images of the circumference. The shadow under the metallic sphere was evident. To the contrary, the shadow under the transparent object was not. Because the attenuation index of the transparent object is low, the portion of the checkerboard under the transparent sphere was illuminated from the light source. On the other hand, the forward sphere correctly had the characteristics of a transparent object. Conventional methods are unable to express a complex scene, which requires the use of several rendering techniques to be used at a time.

E. Computation Time

We discuss the calculation cost of the proposed method here. The measured computation times are shown in Table 3. As shown in Table 3, the times do not widely change as a result of differences of the scene. Because the number of rays that were cast from one elementary hologram was constant, the number of intersection determinations and wave propagations did not increase. However, when the scene did not fill

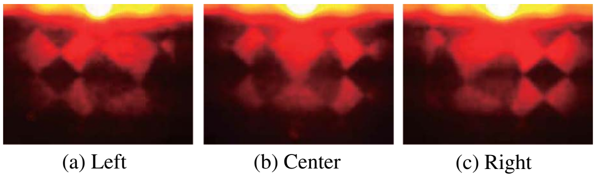
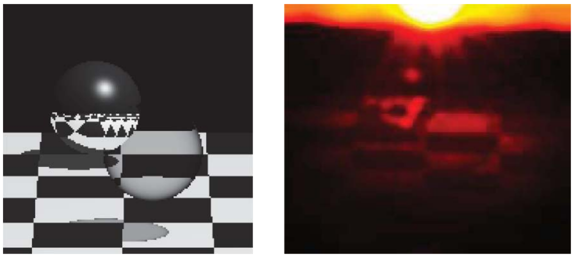

(a) Left (b) Center (c) Right

Fig. 18. (Color online) Reconstructed images from different viewpoints.



(a) Virtual image. (b) Reconstructed image.

Fig. 19. (Color online) Experiment of complex scene.

Table 3. Computation Times (s)

	Ray Tracing	Wave Propagation
Ex. 5.5.A	1.6	7.6
Ex. 5.5.B	2.5	13.8
Ex. 5.5.C	1.7	8.2
Ex. 5.5.D	1.7	11.9

in the visual field, the calculation cost was reduced. If there are specular objects like a mirror in the scene, the number of point light sources increases by the number of reflections. Furthermore, as shown in Table 3, the computation time of wave propagation with the pointwise method is far longer than that of the ray-tracing process apparent from Table 3. To calculate more quickly, we intend to calculate light waves more quickly, e.g., using a look-up table [19]. In addition, once intersection information obtained by the ray-tracing process is sent to the host (Fig. 12), a long computation time is needed for data communication. By performing all calculations from the ray tracing to output the hologram in GPU, the further improvement of the calculation time might be expected.

6. Conclusion

We proposed a calculation method for displaying realistic full-parallax CGHs with the ray-tracing method. By dividing the hologram plane into elementary holograms and conducting ray tracing from each elementary hologram, a full-parallax CGH is generated. Furthermore, we implemented various rendering techniques to express a realistic 3D scene.

Optical reconstructions demonstrated that the hidden surface removal was conducted correctly. Moreover, shading, shadowing, refraction, and multiple reflection with the correct depth appeared in the reconstructed image. The conventional method has not been used to mount two or more renderings in a CGH at once. Until now it has not been possible to use a single technique to mount more than one rendering of a CGH.

The calculations were parallelized by using a GPU. In the future, we hope to provide real-time calculations. The resolution of our output device is so low that the size of reconstructed images is restricted, and the viewing zone and visual field are very narrow. Therefore, demonstrating the advantages of our method is difficult. We need to improve the optical system for the reconstruction. Since the proposed method has a wide range of applications, it is possible to apply it in various optical systems. We expect that the reality of 3D images will be improved by expanding the visual field and colorizing the hologram.

This work was supported by Grant-in-Aid for Scientific Research (B) KAKENHI(23300032).

References

1. J. P. Waters, "Holographic image synthesis utilizing theoretical methods," Appl. Phys. Lett. **9**, 405–407 (1966).
2. Y. Sakamoto, "An algorithm of hidden surface removal using shadow-propagation method for computer-generated hologram," IEICE Trans. Inf. Syst. **J85-D-2**, 1832–1839 (2002), in Japanese.
3. K. Matsushima and S. Nakahara, "Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method," Appl. Opt. **48**, H54–H63 (2009).
4. T. Yamaguchi, T. Fujii, and H. Yoshikawa, "Fast calculation method for computer-generated cylindrical holograms," Appl. Opt. **47**, D63–70 (2008).
5. R. H.-Y. Chen and T. D. Wilkinson, "Computer generated hologram with geometric occlusion using GPU-accelerated depth buffer rasterization for three-dimensional display," Appl. Opt. **48**, 4246–4255 (2009).
6. H. Kang, T. Yamaguychi, and H. Yoshikawa, "Accurate phase-added stereogram to improve the coherent stereogram," Appl. Opt. **47**, D44–D54 (2008).
7. K. Wakunami and M. Yamaguchi, "Calculation for computer generated hologram using ray-sampling plane," Opt. Express **19**, 9086–9101 (2011).
8. K. Matsushima, "Computer-generated holograms for three-dimensional surface objects with shade and texture," Appl. Opt. **44**, 4607–4614 (2005).
9. H. Kim, J. Hahn, and B. Lee, "Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography," Appl. Opt. **47**, D117–D127 (2008).
10. K. Yamaguchi, T. Ichikawa, and Y. Sakamoto, "Calculation method for CGH considering smooth shading with polygon models," Proc. SPIE **7957**, 795706 (2011).
11. Y. Kanazawa, Y. Sakamoto, and Y. Aoki, "Calculation method of computer generation hologram considering refraction," IEICE Trans. Inf. Syst. **J88-D-2**, 88–94 (2005), in Japanese.
12. K. Yamaguchi and Y. Sakamoto, "Computer generated hologram with characteristics of reflection: reflectance distributions and reflected images," Appl. Opt. **48**, H203–H211 (2009).
13. A. Appel, "Some techniques for shading machine rendering of solids," in *Proceedings of AFIPS Joint Computer Conference* (ACM, 1968), pp. 37–45.
14. T. Whitted, "An improved illumination model for shaded display," Commun. ACM **23**, 343–349 (1980).
15. N. Hayashi, Y. Sakamoto, and Y. Honda, "Improvement of camera arrangement in computer-generated holograms synthesized from multi-view images," Proc. SPIE **7957**, 795711 (2011).
16. J. P. Waters, "Holographic image synthesis utilizing theoretical," Appl. Phys. Lett. **9**, 405–407 (1966).
17. B. T. Phong, "Illumination for computer generated pictures," Commun. ACM **18**, 311–317 (2011).
18. R. Cook and K. Torrance, "A reflectance model for computer graphics," ACM Trans. Graph. **1**, 7–24 (2011).
19. H. Yoshikawa, T. Yamaguchi, and R. Kitayama, "Real-time generation of full color image hologram with compact distance look-up table," in *Digital Holography and Three-Dimensional Imaging*, OSA Technical Digest (CD) (Optical Society of America, 2009), paper DWC4.