Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method

Kyoji Matsushima[1,*] and Sumio Nakahara[2]

[1]Department of Electrical and Electronic Engineering, Kansai University, Yamate-cho 3-3-35, Suita, Osaka 564-8680, Japan

[2]Department of Mechanical Engineering, Kansai University, Yamate-cho 3-3-35, Suita, Osaka 564-8680, Japan

*Corresponding author: matsu@kansai-u.ac.jp

A large-scale full-parallax computer-generated hologram (CGH) with four billion ($2^{16} \times 2^{16}$) pixels is created to reconstruct a fine true 3D image of a scene, with occlusions. The polygon-based method numerically generates the object field of a surface object, whose shape is provided by a set of vertex data of polygonal facets, while the silhouette method makes it possible to reconstruct the occluded scene. A novel technique using the segmented frame buffer is presented for handling and propagating large wave fields even in the case where the whole wave field cannot be stored in memory. We demonstrate that the full-parallax CGH, calculated by the proposed method and fabricated by a laser lithography system, reconstructs a fine 3D image accompanied by a strong sensation of depth.   © 2009 Optical Society of America

OCIS codes: 090.1760, 090.2870.

1. Introduction

The technology of computer-generated holograms (CGHs) is the counterpart of computer graphics in holography. The technology has a long history and is sometimes referred to as the final 3D technology, because CGHs not only produce a sensation of depth but also generate light from the objects themselves. However, currently available CGHs cannot yet produce fine true 3D images accompanied by a strong sensation of depth. Such fine CGHs commonly require the following two conditions: the CGHs must have a large viewing zone to acquire the autostereoscopic property, i.e., motion parallax and, second, the dimensions of the CGHs must be large enough to reconstruct a 3D object that can be observed by two naked eyes. Both of these conditions lead to an extremely large number of pixels for a CGH, because the large viewing-zone requires high spatial resolution and the large dimensions require a large number of pixels for high resolution. In addition, scenes with occlusions should be reconstructed to give a CGH a strong sensation of depth, because the ability to handle occlusions is one of the most important mechanisms in the perception of 3D scenes.

The reason why fine 3D images could not be produced by CGH technology was that there was no practical technique to compute the object fields for such high-definition CGHs that reconstruct 3D scenes with occlusions. Computation of the object fields should be performed with an extremely large number of sampling points within a short span of time (typically a few hours or at least a few days), but this was very difficult to achieve even for a state-of-the-art computer. Research during the past few decades evolved from ray-oriented point-based methods, which are widely used for calculating the wave field. However, point-based methods are very time consuming, especially when full-parallax CGHs of surface objects are being computed. Although many techniques to accelerate computation of point-based

methods have been proposed and developed in the past decade [1–7], the computation time or CGH size is still insufficient to produce fine full-parallax CGHs. Furthermore, there is no practical algorithm for full-parallax handling of occlusions in 3D scenes. This is an unresolved and fundamentally difficult problem of the point-based methods.

A field-oriented polygon-based method has been proposed [8] to overcome the limits of the point-based methods. In this new technique, an object is composed of polygonal facets. Each polygon is regarded as a surface source of light. The field emitted by the polygon is referred to as the polygon field. The field of an object is computed from the sum of the polygon fields. Numerical operations necessary for computing the polygon field are a double fast Fourier transform (FFT) and an interpolation for each polygon. Consequently, computation of a polygon field is slower than that of a spherical wave used in point-based methods. However, the number of polygons necessary for forming object surfaces is much less than that of point sources of light. As a result, the total computation time of the object field by the polygon-based method is shorter than the computation time of point-based methods. Several researchers have already proposed improvements on our method in order to compute faster but sacrifice texture mapping and uniform diffusiveness [9,10].

In calculating object fields, light-shielding by obstacles should be realized to add an effect of mutual occlusion to the 3D scene reconstructed by the CGH. We have also proposed a practical solution to this problem. It is based on wave-optical consideration of field propagation [11–13]. This technique is referred to as silhouette masking or simply the silhouette method. Silhouette masking is applicable not only to polygon-based methods but also to point-based methods. However, silhouette masking best matches the polygon-based method because both methods use field propagation operations.

Though these techniques have advantages, so far the polygon-based methods cannot produce fine 3D images due to the inherent difficulty of segmented processing. Since the polygon-based methods use numerical field propagation, the size of the main memory needed for storing complex-valued wave fields limits the pixel size of CGHs. In this study, we propose a technique to overcome the limits on the pixel size in polygon-based methods. The large object field is divided into small segments in the proposed technique. Only a few segments of the field are simultaneously stored in the main memory and processed by multiple processors. Field propagation is also performed in segmented fields by using the shifted-Fresnel method [14].

We demonstrate the implementation of the large-scale CGH using a laser lithography system. This investigation is intended to produce a CGH by combining these techniques. The CGH reconstructs fine true 3D images that can be appreciated as works of art.

2. 3D Scene and 3D Object

The CGH created in this research is named "The Venus" because the 3D object is similar to the famous statue of the Venus de Milo. The Venus statue is 5.8 cm in height and composed of 1396 polygons. The shape of the Venus is provided as a set of vertex coordinates of the polygons. The scene including the Venus statue and the hologram coordinates used in the computation are shown in Fig. 1. The hologram coordinate system $(\hat{x}, \hat{y}, \hat{z})$ is the world coordinates of the 3D scene, in which the hologram is placed in the $(\hat{x}, \hat{y}, 0)$ plane. The Venus statue is placed at $\hat{z} = -15$ cm behind the hologram. In addition, wallpaper with a simple planar image featuring a binary check pattern, is placed at $\hat{z} = -30$ cm behind the Venus statue to provide an occlusion and enhance the viewer's sensation of depth. Both the wallpaper and hologram have the same dimensions, approximately $6.5 \text{ cm} \times 6.5 \text{ cm}$. Some parameters used for creating the Venus CGH are summarized in Table 1.

3. Polygon-Based Method for Computing 3D Objects

In the polygon-based method, a wave field, given in the tilted plane that includes the polygon, represents the polygonal surface source of light. The polygon field in a plane parallel to the hologram is computed from the wave field in the tilted plane by using rotational transformation of the wave fields [15,16]. This section comprehensively summarizes the details of the polygon-based method proposed in [8].

A. Local Coordinate System and Rotation Matrix

The procedure for computing the object field of a triangular prism is shown in Fig. 2 as an example. There are three polygons, P_1, P_2, and P_3, on the front face of the object.

A set of tilted local coordinate systems are defined as $\mathbf{r}_n = (x_n, y_n, z_n)$ for the polygon P_n, as shown in Fig. 2(a). One more set of local coordinate systems, referred to as parallel local coordinates, is defined as $\hat{\mathbf{r}}_n = (\hat{x}_n, \hat{y}_n, \hat{z}_n)$. All axes of the parallel local coordinate systems are parallel to that of the hologram coordinates, but the origins are shared with the corresponding tilted coordinate systems. The parallel
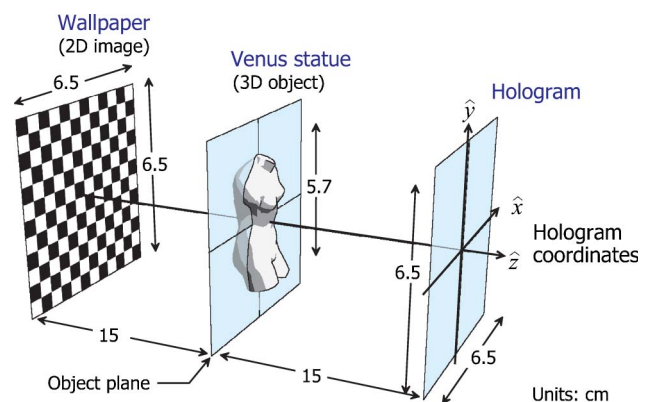

Fig. 1. (Color online) 3D scene of "The Venus" and the definition of the hologram coordinate system.

Table 1. Parameters for Computing and Fabricating "The Venus" CGH

Parameter	Value
Hologram	
No. of pixels ($N_x \times N_y$)	4.3×10^9 ($65{,}536 \times 65{,}536$)
Pixel pitches	$1\,\mu m \times 1\,\mu m$
Hologram dimensions	$65.5\,mm \times 65.5\,mm$
No. of segments ($M_x \times M_y$)	8×4
Hologram type	Binary amplitude
Reconstruction wavelength	$632.8\,nm$
3D object and scene	
No. of polygons of Venus (front face only)	718
Dimension of Venus ($W \times H \times D$)	$26.7\,mm \times 57.8\,mm \times$ $21.8\,mm$
Dimension of wallpaper ($W \times H$)	$65.5\,mm \times 65.5\,mm$
Distance of Venus from hologram	$150\,mm$
Distance of wallpaper from hologram	$300\,mm$

local coordinates are transformed from or to the tilted local coordinates by

$$\hat{\mathbf{r}}_n = \mathbf{T}_n \mathbf{r}_n, \qquad \mathbf{r}_n = \mathbf{T}_n^{-1} \hat{\mathbf{r}}_n. \qquad (1)$$

where $\mathbf{T}_n$ is the rotation matrix for the polygon P_n.

There is an arbitrariness in determining the matrix $\mathbf{T}_n$. We use the following Rodrigues rotation formula:

$$\mathbf{T}_n = \cos\theta_n \mathbf{I} + (1 - \cos\theta_n)\mathbf{s}_n^t \cdot \mathbf{s}_n + \sin\theta_n [\mathbf{s}_n]_\times, \qquad (2)$$

where $\mathbf{I}$ and $\mathbf{s}_n$ are the 3×3 identity matrix and the unit vector of the rotation axis. Symbols $[\mathbf{s}_n]_\times$ and $\mathbf{s}_n^t$ denote the cross-product matrix and transposed matrix for $\mathbf{s}_n$. Here, the vector $\mathbf{s}_n$ is given by using the normal vector $\mathbf{n}_n$ of the polygon P_n and the unit vector $\hat{\mathbf{z}} = (0, 0, 1)$ in the hologram coordinates as follows:

$$\mathbf{s}_n = \hat{\mathbf{z}} \times \mathbf{n}_n. \qquad (3)$$

The rotation angle θ_n in Eq. (2) is also given by $\theta_n = \cos^{-1}(\hat{\mathbf{z}} \cdot \mathbf{n}_n)$.

B. Surface Functions

The distribution of complex amplitudes, referred to as a surface function, is defined in the $(x_n, y_n, 0)$ plane for the tilted local coordinates of the polygon P_n. This function plays an important role to give the appearance of the polygon in optical reconstruction. Examples of the surface functions of the triangular prism are shown in Fig. 2(b).

The surface function $h_n(x_n, y_n)$ for a polygon P_n is generally given in the form

$$h_n(x_n, y_n) = a_n(x_n, y_n) \exp[i\phi(x_n, y_n)], \qquad (4)$$

where the real-valued amplitude $a_n(x_n, y_n)$ gives the shape of the polygon as follows:
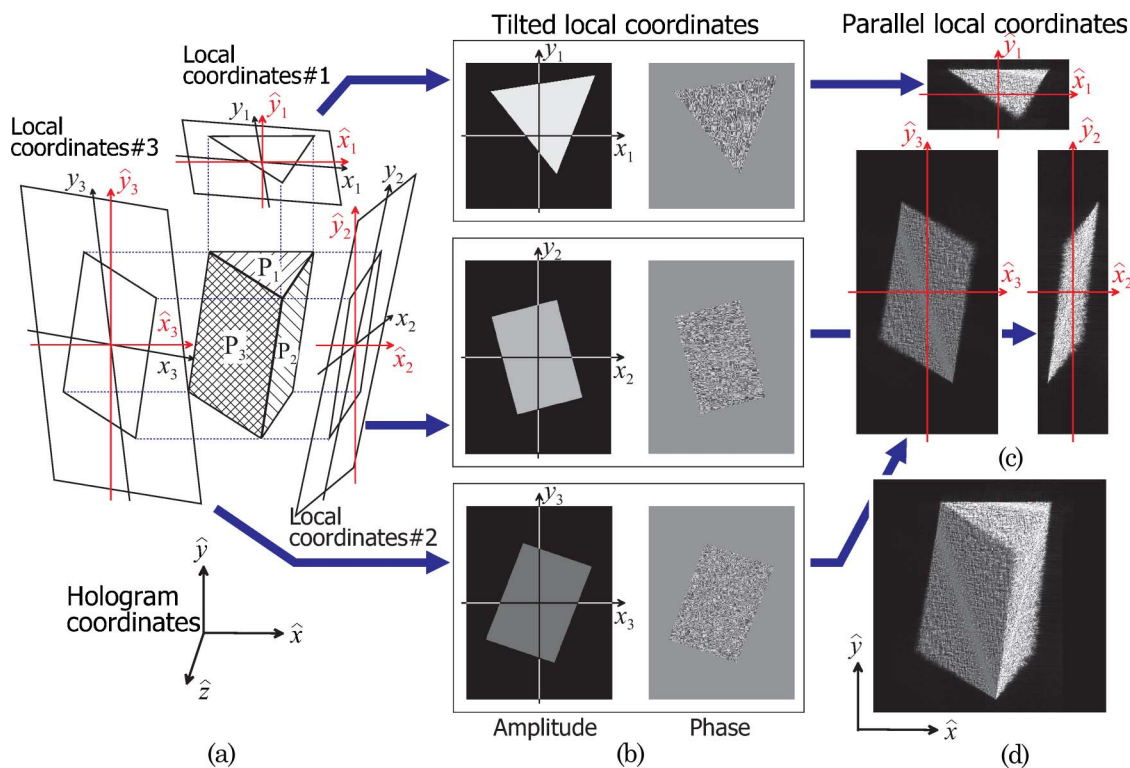

Fig. 2. (Color online) Procedure for producing the object field of a triangular prism by the polygon-based method. (a) Object model. (b) Surface functions in tilted local coordinates. (c) Amplitude images of polygon fields in parallel local coordinates. (d) Amplitude image of the object field.

$$a_n(x_n, y_n) = \begin{cases} A_n & \text{inside polygon} \\ 0 & \text{outside polygon} \end{cases}. \quad (5)$$

To shade the object as intended by the designer [8], the amplitude A_n has a different value dependent on the polygon. The texture of the polygon can be given as a modulation of the amplitude. In this case, A_n is not constant but a function of x_n and y_n.

If the surface function is a simple polygon-shaped distribution of real-valued amplitude, the surface function behaves like an aperture that shapes the polygon. In this case, light from the polygon does not spread over the hologram. The amplitude of the surface function should be multiplied by a phase distribution $\phi(x_n, y_n)$ to give diffusiveness to the polygon field. Random functions are candidates for the diffusive phase. However, full random functions are not appropriate, because the random phases are discontinuous and have a large spatial frequency. As a result, full random phases usually cause speckles in the reconstruction. In this paper, speckle-free quasi-random phase distribution proposed as a digital diffuser in Fourier holograms [17] is used for the phase distribution.

The wave field of a polygon is diffused by the diffuser phase. However, if the carrier frequency of the surface function is zero on the surface of the polygon, the emitted wave field propagates mainly in the direction normal to the polygon, and thus the polygon field cannot reach the hologram effectively. The center of the spectrum of the surface function should be shifted so that light travels along the optical axis of $\hat{z}$ and almost perpendicularly intersects the plane parallel to the hologram. This shift operation is a part of rotational transformation described in the following section.

C. Rotational Transformation and Spectrum Remapping

The polygons forming the object surfaces are usually not parallel to the hologram, and thus wave fields in the hologram plane cannot be obtained simply by using ordinary formulas for the propagation of wave fields between parallel planes, such as the Fresnel diffraction formula. Before translational propagation, the surface function must be rotationally transformed into the wave field given in a plane parallel to the hologram by using the formula for rotational transformation [16].

Surface functions are Fourier transformed as follows:

$$H_n(u_n, v_n) = \mathbf{F}\{h_n(x_n, y_n)\}, \quad (6)$$

where $\mathbf{F}$ and $\mathbf{F}^{-1}$ stand for the Fourier and the inverse Fourier transform. The Fourier spectrum $H_n(u_n, v_n)$ of polygon n is obtained numerically by using the FFT. Here, u_n and v_n are Fourier frequencies.

Supposing that the transformation matrix for changing the parallel local coordinates into the tilted local coordinates is given as

$$\mathbf{T}_n^{-1} = \begin{bmatrix} t_{1,n} & t_{2,n} & t_{3,n} \\ t_{4,n} & t_{5,n} & t_{6,n} \\ t_{7,n} & t_{8,n} & t_{9,n} \end{bmatrix}, \quad (7)$$

the Fourier space $(\hat{u}, \hat{v})$ of the parallel coordinates is mapped onto the Fourier space (u_n, v_n) of the tilted coordinates by

$$u_n = \alpha_n(\hat{u}_n, \hat{v}_n) = t_{1,n}\hat{u}_n + t_{2,n}\hat{v}_n + t_{3,n}\hat{w}(\hat{u}_n, \hat{v}_n),$$
$$v_n = \beta_n(\hat{u}_n, \hat{v}_n) = t_{4,n}\hat{u}_n + t_{5,n}\hat{v}_n + t_{6,n}\hat{w}(\hat{u}_n, \hat{v}_n), \quad (8)$$

where $\hat{w}(\hat{u}_n, \hat{v}_n) = (\lambda^{-2} - \hat{u}_n^2 - \hat{v}_n^2)^{1/2}$ is the Fourier frequency on the $\hat{z}_n$ axis. Hence, the spectrum in the tilted coordinates is mapped onto the Fourier space of the parallel coordinates as

$$\hat{H}'_n(\hat{u}_n, \hat{v}_n) = H_n(\alpha_n(\hat{u}_n, \hat{v}_n), \beta_n(\hat{u}_n, \hat{v}_n)). \quad (9)$$

This mapping moves the center of the spectrum far from the origin of the parallel Fourier space. This is interpreted as the fact that the polygon field does not travel perpendicular to the hologram plane as described in the former section. The spectrum $\hat{H}(\hat{u}_n, \hat{v}_n)$ for the parallel coordinates should be shifted to change the direction of the polygon field as follows:

$$\hat{H}_n(\hat{u}_n, \hat{v}_n) \cong H_n(\alpha_n(\hat{u}_n, \hat{v}_n) - u_n^{(0)}, \beta_n(\hat{u}_n, \hat{v}_n) - v_n^{(0)}), \quad (10)$$

where $u_n^{(0)}$ and $v_n^{(0)}$ are shift values for moving the center of spectrum to a certain position around the origin of the Fourier space $(\hat{u}_n, \hat{v}_n)$. In most polygons, the appropriate shift values are

$$u_n^{(0)} = \alpha_n(0, 0) = t_{3,n}/\lambda, \quad (11)$$

$$v_n^{(0)} = \beta_n(0, 0) = t_{6,n}/\lambda, \quad (12)$$

because the carrier frequencies are changed to zero by using these values, and thus the polygon fields are forced to travel precisely to the hologram [8]. However, there are a few exceptions. In some polygons for which normal vectors are almost parallel to the hologram, fine adjustments need to be made to the shift values. Note that the nearly equal sign in Eq. (10) implies that an interpolation is required because coordinate rotation not only moves the origin but also distorts the sampling grid [16].

The object field is a superposition of the polygon fields in a plane that is parallel to the hologram. This parallel plane is referred to as the object plane as shown in Fig. 1. Since the $\hat{z}$ positions of the polygons are not same, the polygon fields should be translationally propagated back or forth along the $\hat{z}$ axis before superposition. We use the angular spectrum method [18] for translational propagation. Supposing that the object plane is placed at $\hat{z}_{\text{obj}}$ and the origin of the local coordinates is positioned at

$(\hat{x}_n^{(0)}, \hat{y}_n^{(0)}, \hat{z}_n^{(0)})$, translational propagation in the Fourier space is given as

$$\hat{H}_n(\hat{u}_n, \hat{v}_n; \hat{z}_{\text{obj}}) = \hat{H}_n(\hat{u}_n, \hat{v}_n) \exp[i2\pi \hat{w}(\hat{u}, \hat{v}) \\ \times (\hat{z}_{\text{obj}} - \hat{z}_n^{(0)})]. \tag{13}$$

Polygon fields in the object plane are calculated by the inverse Fourier transform:

$$\hat{h}_n(\hat{x}_n, \hat{y}_n; \hat{z}_{\text{obj}}) = \mathbf{F}^{-1}\{\hat{H}_n(\hat{u}_n, \hat{v}_n; \hat{z}_{\text{obj}})\}. \tag{14}$$

The amplitude images of the polygon fields of the triangular prism are shown in Fig. 2(c).

Finally, the object field is given by superposition of all polygon fields in the object plane as follows:

$$h(\hat{x}, \hat{y}; \hat{z}_{\text{obj}}) = \sum_{n=1}^{N_p} \hat{h}_n(\hat{x}_n - \hat{x}_n^{(0)}, \hat{y}_n - \hat{y}_n^{(0)}; \hat{z}_{\text{obj}}), \tag{15}$$

where N_p is the number of polygons. The object field $h(\hat{x}, \hat{y}; \hat{z}_{\text{obj}})$ of the triangular prism is shown as the amplitude image in Fig. 2(d). Note that the object plane is not the same as the hologram plane. The object plane should be placed so that it slices the object at the center, as shown in Fig. 1. The object field in the hologram plane $h(\hat{x}, \hat{y}; 0)$ is computed by translational propagation of the object field in the object plane.

The reason why the object field is not directly calculated in the hologram plane and the translational propagation is used is that calculating the object field in the object plane is faster than doing it in the hologram plane. Since the polygons are closer to the object plane than to the hologram plane, polygon fields are not diffracted as much in the object plane. In other words, the area for calculating the polygon field is much smaller than it is in the hologram plane. As a result, the object field can be computed much faster in the object plane than in the hologram plane.

Furthermore, there is another significant reason: light shielding of the wallpaper field is necessary to avoid a phantom image of the 3D object, as explained in the Section 5.

4. Segmentalized Computation of Object Fields by the Polygon-Based Method

A. Segmented Frame Buffers

The wave field of Eq. (15) is a distribution of complex-valued amplitudes and is generally treated as a sampled 2D array of two floating-point variables in the numerical implementation. This 2D array is referred to as the frame buffer. The memory size required for the frame buffer for the surface function $h_n(x_n, y_n)$ and the parallel local field $\hat{h}_n(\hat{u}_n, \hat{v}_n)$ directly depends on the dimensions of the polygon. These are generally much smaller than the memory size for the object field if the object is constructed of curved surfaces, because curved surfaces are com-

posed of many small polygons. In contrast, the memory size for the object field $h(\hat{x}, \hat{y}; \hat{z}_{\text{obj}})$ is excessively large.

Since two single-precision floating points represent a complex value in our implementation, the memory size necessary for the frame buffer is $8N$ for an array of N sampling points. Therefore, the memory size required for storing the whole object field is 32 Gbytes in the object plane, because our CGH has $2^{16} \times 2^{16}$ pixels. We note that this estimation is only for a single frame buffer. Other memory spaces required for running the program, such as the operating system area, code area, temporal frame buffers, I/O buffers, and so on are not included. As a result, segmentation of the frame buffer is essential for producing the Venus CGH by using ordinary personal computers (PCs). The segmented structure of the frame buffer for the object field is shown in Fig. 3. The frame buffer is divided into the same rectangular segments. Only a few segments are loaded in memory, and others are saved as files in a secondary storage medium such as a hard disk drive.

B. Computation of Object Field by Using the Segmented Frame Buffer

The polygon field stored in the small frame buffers is numerically propagated in the object plane, and then all the polygon fields are gathered and summed up as described in Eqs. (13)–(15). The segmented frame buffer is used for summation in the object plane. Polygon fields are summed for a few segments loaded in memory. When the summation process is finished in a segment, the segment is saved in the file. After the segment is purged from memory, the next segment is loaded and processed in memory. This process is parallelized; i.e., a certain processor handles a certain segment, and multiple segments are processed simultaneously by multiple processors, which share the memory. Therefore, the number of
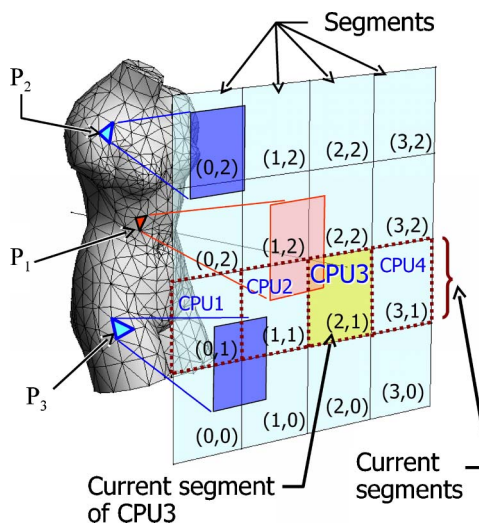

Fig. 3. (Color online) Segmentation of the frame buffer storing the object field. Note that the object plane is illustrated in this figure as a plane that does not slice the object for easy understanding, but in practical computation, it slices the object.

segments simultaneously processed in memory is the same as the number of processors used for computation.

In the summation process, discrimination of polygons is important for avoiding unnecessary computations of the polygon fields. Polygons whose fields do not touch the current segment should be culled in computing the object field of the segment. For example, the fields of polygon P_1 in Fig. 3 should be computed and totaled in the segment $(2, 1)$ that is processed by CPU3, but polygons P_2 and P_3 should not be processed by CPU3. This discrimination is achieved by determining the maximum diffraction area of the polygon.

C. Estimation of Maximum Diffraction Area of a Polygon

The Venus object is composed of 718 polygons. Since the processors scan all the polygons for each segment every time, discrimination should be made as fast as possible. It is not easy to calculate the exact diffraction area of a polygon in the object plane, and it is unnecessary for discrimination. We use a simple approach to estimate the maximum diffraction area as shown in Fig. 4. The maximum diffraction area of the vertex of a polygon can be obtained from the maximum diffraction angle given by

$$\theta_x = \sin^{-1}\frac{\lambda}{2\Delta x}, \qquad \theta_y = \sin^{-1}\frac{\lambda}{2\Delta y}, \qquad (16)$$

where Δx and Δy are the sampling pitch of the hologram. The field emitted from the vertex is spread in the form of a rectangular pyramid with apex angles of θ_x and θ_y and forms a rectangle in the object plane. We define the maximum diffraction area of the polygon as the bounding box of the diffraction rectangles in the object plane. This is not same as the exact diffraction area of the polygon but meets the purpose for discrimination of the polygon.

D. Numerical Propagation by Using Segmented Frame Buffers

Wave fields are propagated numerically using the shifted Fresnel method [14]. The method is suited for propagation by using segmented frame buffers, because the sampling area in the destination plane can be shifted from the sampling area in the source plane, as shown in Fig. 5.

Suppose that the wave field $h_{p,q}(\hat{x}, \hat{y}; \hat{z}_{\text{dest}})$ in a segment (p, q) of the destination plane is computed by numerical propagation of the wave field in the source plane at $\hat{z} = \hat{z}_{\text{source}}$. If the destination plane at $\hat{z} = \hat{z}_{\text{dest}}$ is sufficiently far from the source plane, the wave fields in all segments of the source plane contribute to the wave field in the segment (p, q) of the destination plane. Therefore, the wave field in the segment (p, q) is computed by

$$h_{p,q}(\hat{x}, \hat{y}; \hat{z}_{\text{dest}}) = \sum_{s,t}\mathcal{P}_{\hat{z}_{\text{dest}}-\hat{z}_{\text{source}}}\{h_{s,t}(\hat{x}, \hat{y}; \hat{z}_{\text{source}})\},$$

$$(17)$$

where the symbol $\mathcal{P}_d\{\cdot\}$ represents the field propagation for the distance of d using the shifted Fresnel method.

5. Light-Shielding by Silhouette Method

A. Generation of Wallpaper Field

The wallpaper is a planar object and is given by a 256×256 pixel texture. To give the texture image the appearance of wallpaper that is perpendicular to the optical axis and has the same dimensions as the hologram, the texture image is oversampled with the same sampling pitches as the hologram. The enlarged texture image is regarded as an amplitude distribution of the wave field and handled by the same segmented frame buffer as the object field. We multiply the same diffusive phase as in Eq. (4) into the oversampled texture image $a_{\text{wp}}(\hat{x}, \hat{y})$ to give a diffusiveness to the wallpaper. Hence, the wallpaper field is given by
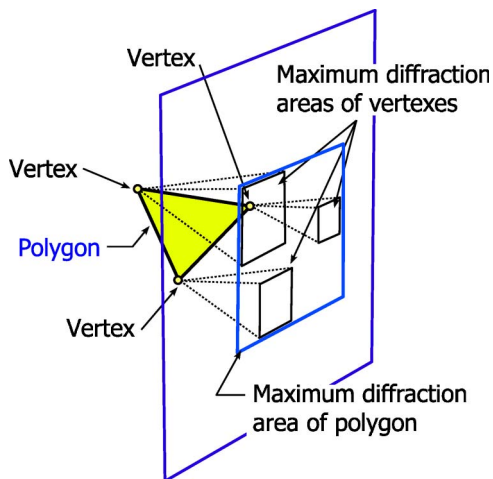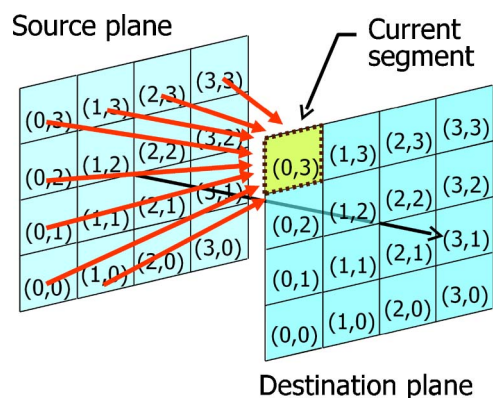

Fig. 4. (Color online) Estimation of the maximum diffraction area of a polygon as a bounding box of diffraction rectangles formed by each vertex of the polygon.

Fig. 5. (Color online) Numerical propagation in segmented frame buffers by the shifted Fresnel method.

$$h_{\mathrm{wp}}(\hat{x},\hat{y};\hat{z}_{\mathrm{wp}}) = a_{\mathrm{wp}}(\hat{x},\hat{y})\exp[i\phi(\hat{x},\hat{y})], \qquad (18)$$

where $\hat{z}_{\mathrm{wp}}$ is the $\hat{z}$ position of the wallpaper.

B. Silhouette Masking

The wallpaper field generated according to Eq. (18) should be numerically propagated to the hologram plane and superimposed on the object field that is also propagated to the hologram plane. However, if the wallpaper field is simply propagated and added to the object field, the Venus object would be reconstructed as a phantom image, because simple superposition involves light transmission of the Venus object. To shield the wallpaper behind the Venus object and prevent the Venus from being a phantom image, the propagation process should be divided into two stages, as shown in Fig 6. The wallpaper field is first propagated to the object plane and then masked with the silhouette of the Venus object, as shown in Fig 6(a). The silhouette mask $a_{\mathrm{sil}}(\hat{x},\hat{y})$ is simply obtained by orthogonal projection of the polygons forming the object on the object plane as follows:

$$a_{\mathrm{sil}}(\hat{x},\hat{y}) = \begin{cases} 0 & \text{inside any projected polygon} \\ 1 & \text{outside any projected polygon} \end{cases}.$$
$$(19)$$

The object field $h(\hat{x},\hat{y};\hat{z}_{\mathrm{obj}})$ shown in Fig 6(b) is superimposed on the masked wallpaper field. Thus, the object field combined with the wallpaper field is represented as

$$h'(\hat{x},\hat{y};\hat{z}_{\mathrm{obj}}) = a_{\mathrm{sil}}(\hat{x},\hat{y})h_{\mathrm{wp}}(\hat{x},\hat{y};\hat{z}_{\mathrm{obj}}) + h(\hat{x},\hat{y};\hat{z}_{\mathrm{obj}}),$$
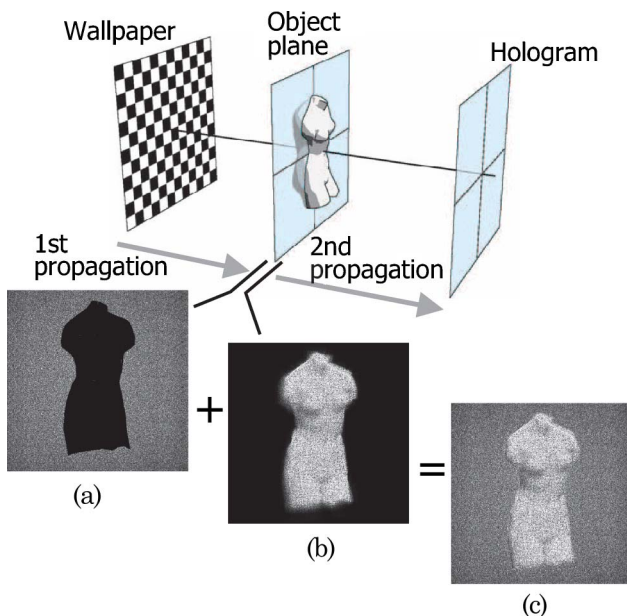$$(20)$$

Fig. 6. (Color online) Two-stage propagation for light shielding by the silhouette-masking technique. (a) Silhouette-masked wave field of the wallpaper. (b) Object field. (c) Combined field. All images are of amplitude.

where $h_{\mathrm{wp}}(\hat{x},\hat{y};\hat{z}_{\mathrm{obj}})$ is the wallpaper field propagated onto the object plane. The combined field shown in Fig 6(c) is finally propagated to the hologram plane, and the field for the 3D scene $h(\hat{x},\hat{y};0)$ is computed.

6. Creation of "The Venus"

A. Producing Fringe Pattern

The fringe intensity $I(\hat{x},\hat{y})$ of the CGH is generated by numerical interference of the combined field $h(\hat{x},\hat{y};0)$ with a reference field $r(\hat{x},\hat{y})$ in the hologram plane as follows:

$$I(\hat{x},\hat{y}) = |h(\hat{x},\hat{y};0) + r(\hat{x},\hat{y})|^2 \approx h(\hat{x},\hat{y};0)r(\hat{x},\hat{y})^* + B,$$
$$(21)$$

where the constant B is the offset to keep the fringe intensity positive. The plane wave with an incident angle of 9.1° is used for the reference field in the Venus CGH. The fringe actually printed is the binary pattern because of the fabrication method used in this investigation. Thus, the offset B is set to zero in practice and $I(\hat{x},\hat{y})$ is quantized by the zero threshold.

The DWL 66 laser lithography system made by Heidelberg Instruments GmbH was used for printing the fringe pattern onto the photoresist that is coated on a chromium thin film on a substrate of fused silica. After development of the photoresist, the chromium thin film was etched and formed the binary transmittance pattern. As a result, the fabricated CGH has a fringe of binary amplitude.

B. Optical Reconstruction

Photographs of the optical reconstruction of the fabricated hologram are shown in Fig. 7. A He–Ne laser is used for the transmission light source. The photographs in Fig. 7(a)–7(d) captured from various viewpoints verify that the CGH reconstructs the 3D scene because the occlusion of the Venus and wallpaper is changed when the view point is changed. This motion parallax creates a strong sensation of depth in the viewer's perception.

The fabricated CGH can be reconstructed as a reflection hologram as well as a transmission hologram because of the high reflectance of the chromium thin film. Photographs of optical reconstruction in the reflection mode are shown in Fig. 8. An ordinary red LED is used for the reflection light source. The CGH also gives good reconstruction. However, the dimensions of the reconstructed objects are somewhat reduced in this case, because the LED works as a point source of light, whereas a plane wave is assumed as the reference wave.

7. Discussion

A. Total Computation Time and the Bottleneck

The software for computing the CGH was implemented by using Intel C++ Compiler 10 and Visual Studio 2008. The Intel Math Kernel Library was also
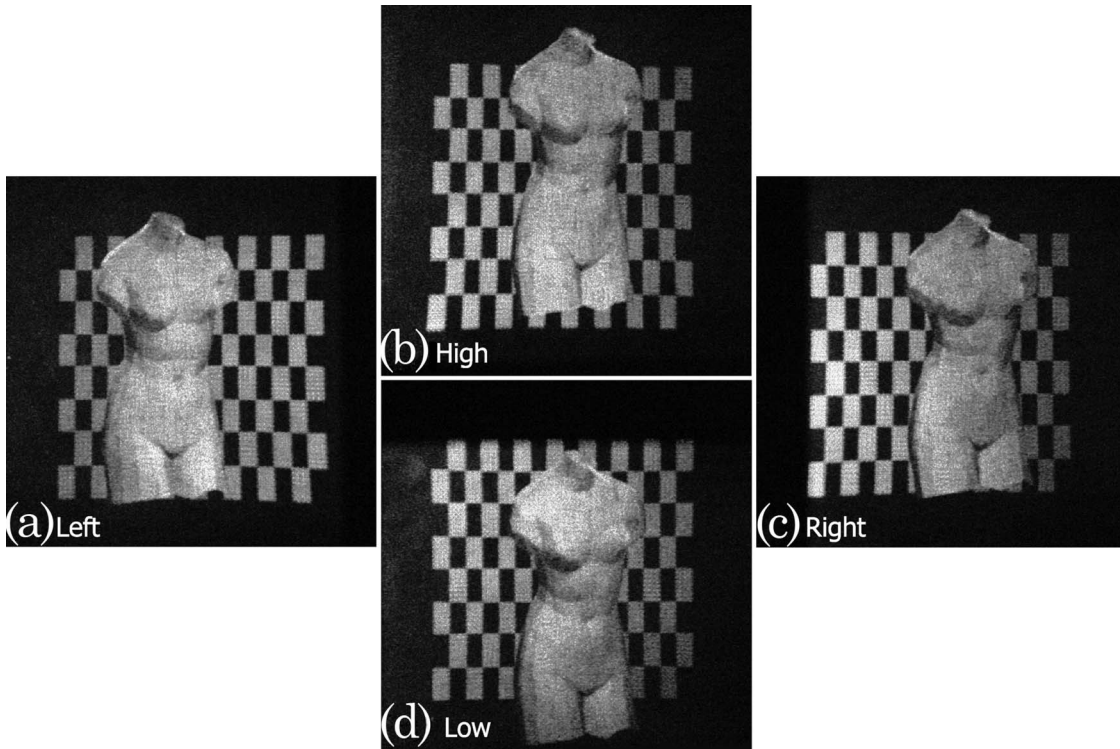
Fig. 7. Photographs of the optical reconstruction of the fabricated CGH by using transmitted illumination of a He–Ne laser. Photographs (a) – (d) are taken from different viewpoints.

used as the FFT package. Computation of the Venus CGH was executed by using a PC with the four CPUs of the AMD Opteron 852 (2.6 GHz) processor and a shared memory of 32 Gbytes. The total computation time was approximately 45 h, and the itemized computation times are shown in Fig. 9. The longest time was consumed by the numerical propagation using the shifted Fresnel method and the segmented frame buffer. The computation time for this was 37 h and accounted for 82% of the total computation time.

Since the computation time of a 2D FFT is proportional to $2N \log_2 N$ for N sampling points, the computation time of the shifted Fresnel method for a wave field sampled in $N (= N_x \times N_y)$ points without segmentation is estimated as

$$T(N) \propto 24N(2 + \log_2 N), \qquad (22)$$

because the method requires the FFT operation three times for $4N$ points. When the frame buffer is divided into $M (= M_x \times M_y)$ segments, the computation time required for propagation of all segments from source to destination is proportional to $T(N/M)M^2$. In this case, however, we can omit one FFT for each segment by using a precomputed N-point table [19]. Therefore, the computation time with segmented frame buffers is given approximately by

$$T'\left(\frac{N}{M}\right)M^2 \propto 16NM(2 + \log_2 N - \log_2 M), \qquad (23)$$
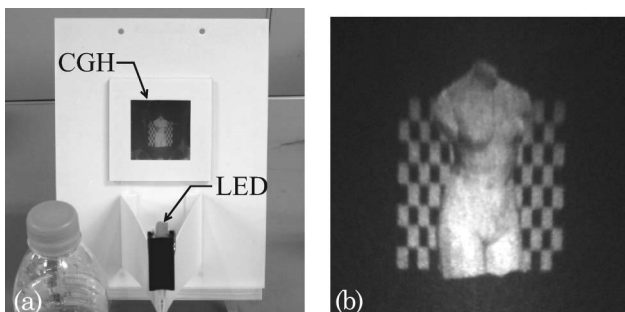

Fig. 8. Photographs of the optical reconstruction of the fabricated CGH by using reflected illumination of an ordinary red LED. (a) Medium shot (Media 1, Media 2). (b) Close-up (Media 3, Media 4).
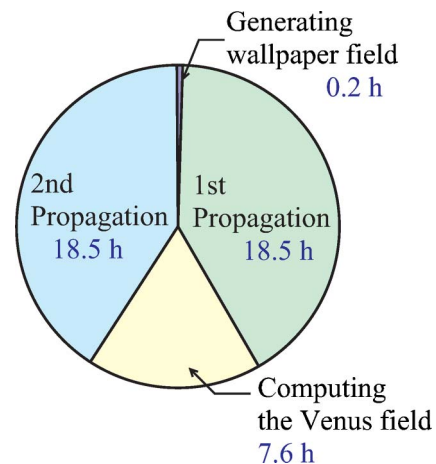

Fig. 9. (Color online) Itemized computation time for computing the Venus CGH.

where $T'(N) \propto 16N(2 + \log_2 N)$ is the time required for propagation of an N-point wave field by using the precomputed table. As a rough estimation, computation time is approximately $T'(N/M)M^2 \propto M$, because $N = 2^{32}$, whereas $M = 2^5$ in the Venus CGH. Therefore, the computation time is directly dependent on the number of segments. Decreasing the number of segments by increasing the size of memory installed into the PC is the most effective way of reducing computation time.

B. Computation Time of Object Fields

The computation time of the object field is short compared with that of numerical propagation in creating the Venus CGH. However, when the number of polygons increases, as is needed to create more complex objects, or the number of segments decreases as a result of increasing the memory size installed in the PC, the computation time of the object field cannot be ignored. The computation time is given by a summation of the computation times of the polygon fields. The computation time of a polygon field is generally given by

$$T_{\text{polygon}} = T_{\text{FFT1}} + T_{\text{interpol}} + T_{\text{prop}} + T_{\text{FFT2}}, \quad (24)$$

where the terms on the right-hand side are the computation time of the FFT of the surface function, interpolation, translational propagation by the angular spectrum method, and the inverse FFT of the spectrum of the polygon field in the object plane, respectively.

The computation times of the FFTs are given by

$$T_{\text{FFT1}}(N_{\text{surf}}) \propto 2N_{\text{surf}} \log_2 N_{\text{surf}}, \quad (25)$$

$$T_{\text{FFT2}}(N_{\text{field}}) \propto 2N_{\text{field}} \log_2 N_{\text{field}}, \quad (26)$$

where N_{surf} and N_{field} are the numbers of sampling points of the surface function and the polygon field in the object plane, respectively. The value of N_{field} is determined by the maximum diffraction area of the polygon in the object plane, shown in Fig. 4. When the object plane is sufficiently close to the polygon, the number of sampling points of the surface function is the same as that of the polygon field, i.e., $N_{\text{surf}} \cong N_{\text{field}}$. Computation times for interpolation and translational propagation are also governed by N_{field} as follows:

$$T_{\text{interpol}}(N_{\text{field}}) \propto N_{\text{field}}, \quad (27)$$

$$T_{\text{prop}}(N_{\text{field}}) \propto N_{\text{field}}. \quad (28)$$

As a result, the computation time of polygon fields is determined by N_{field}, that is, the number of sampling points within the maximum diffraction area of the polygon in the object plane. This is the reason that the object plane should be placed close to the polygons. We note that in cases where the number of polygons of an object increases but the dimension of the object does not change, the total computation time does not increase much. This is because an increase in the number of polygons leads to a reduction in the size of each polygon, and thus a decrease in each value of N_{field} in this case.

C. Limitations of Silhouette Masking in the Venus CGH

The silhouette method adopted for the creation of the Venus CGH is a wave-optical algorithm for hidden-surface removal in CGHs. It is one of the lowest approximations of light shielding by obstacles [13]. This type of silhouette method shields only light that intersects a planar region given as the silhouette of the object in the object plane. Therefore, other light that does not intersect the planar silhouette region, but intersects the 3D region of the Venus object, are not shielded. This means that the silhouette method does not work well if a viewer observes the side of the Venus. However, a viewer cannot take a side view of the Venus in our CGH, because we place the Venus 15 cm behind the hologram. This is a constraint in the configuration of the 3D scene. In addition, this type of silhouette method cannot handle the self-occlusion of objects; i.e., self-occluded objects such as a torus may be reconstructed as a partial phantom image.

Another type of silhouette approximation has been proposed [12]. Light behind the 3D object in the method is not shielded by the full silhouette of the object, but is shielded by individual silhouettes of polygons. This type of silhouette method has the ability to correctly reconstruct self-occluded objects. However, this method requires the same number of numerical propagations as the number of polygons; therefore, it is not practical for high-definition CGHs such as our Venus. More rigorous light shielding by polygon surfaces without silhouette approximation has been proposed [13], but the method is much more costly in computation time and is not applicable to a large-scale CGH.

8. Conclusion

A large-scale full-parallax CGH with 4 G pixels, named "The Venus," is created by using the polygon-based and the silhouette methods. The polygon-based method numerically generates the object fields in a shorter computation time compared with the conventional point-based methods, while the silhouette method makes it possible to reconstruct the occluded scene. These methods had previously suffered from the problem that the pixel size is limited to a value dependent on the memory size installed in the PC, because numerical propagation of wave fields is necessary for these methods. To overcome this limitation, the technique of the segmented frame buffer and shifted Fresnel method for numerical propagation are used in this research. These techniques also contribute to a reduction of computation time by making it easy to introduce high-level parallel

processing. The Venus CGH was fabricated by using a laser lithography system. The optical reconstruction gives a fine true 3D image of the occluded scene, which creates a strong sensation of depth due to the motion parallax.

On the other hand, the shifted Fresnel method adopted in this investigation brings a constraint into the configuration of the 3D scene. A severe aliasing error occurs when the wave field of a large object is numerically propagated for a short distance by using the method. Future work will involve the development of a new method for numerical propagation, which is able to shift the sampling area and is applicable for a wide range of propagation distances.

References

1. M. Lucente, "Interactive computation of holograms using a look-up table," J. Electron. Imaging **2**, 28–34 (1993).
2. A. Ritter, J. Böttger, O. Deussen, M. König, and T. Strothotte, "Hardware-based rendering of full-parallax synthetic holograms," Appl. Opt. **38**, 1364–1369 (1999).
3. K. Matsushima and M. Takai, "Recurrence formulas for fast creation of synthetic three-dimensional holograms," Appl. Opt. **39**, 6587–6594 (2000).
4. H. Yoshikawa, S. Iwase, and T. Oneda, "Fast computation of Fresnel holograms employing difference," Proc. SPIE **3956**, 48–55 (2000).
5. T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie, "Special-purpose computer HORN-5 for a real-time electroholography," Opt. Express **13**, 1923–1932 (2005).
6. N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, "Computer generated holography using a graphics processing unit," Opt. Express **14**, 603–608 (2006).
7. T. Shimobaba, A. Shiraki, Y. Ichihashi, N. Masuda, and T. Ito, "Interactive color electroholography using the FPGA technology and time division switching method," IEICE Electron. Express **5**, 271–277 (2008).
8. K. Matsushima, "Computer-generated holograms for three-dimensional surface objects with shade and texture," Appl. Opt. **44**, 4607–4614 (2005).
9. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holograms from three dimensional meshes using an analytic light transport model," Appl. Opt. **47**, 1567–1574 (2008).
10. 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).
11. K. Matsushima and A. Kondoh, "A wave optical algorithm for hidden-surface removal in digitally synthetic full-parallax holograms for three-dimensional objects," Proc. SPIE Proc. **5290**, 90–97 (2004).
12. A. Kondoh and K. Matsushima, "Hidden surface removal in full-parallax CGHs by silhouette approximation," Syst. Comput. Jpn. **38**(6), 53–61 (2007).
13. K. Matsushima, "Exact hidden-surface removal in digitally synthetic full-parallax holograms," Proc. SPIE **5742**, 25–32 (2005).
14. R. P. Muffoletto, J. M. Tyler, and J. E. Tohline, "Shifted Fresnel diffraction for computational holography," Opt. Express **15**, 5631–5640 (2007).
15. K. Matsushima, H. Schimmel, and F. Wyrowski, "Fast calculation method for optical diffraction on tilted planes by use of the angular spectrum of plane waves," J. Opt. Soc. Am. A **20**, 1755–1762 (2003).
16. K. Matsushima, "Formulation of the rotational transformation of wave fields and their application to digital holography," Appl. Opt. **47**, D110–D116 (2008).
17. R. Bräuer, F. Wyrowski, and O. Bryngdahl, "Diffusers in digital holography," J. Opt. Soc. Am. A **8**, 572–578 (1991).
18. J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed. (McGraw-Hill, 1996), chap. 3.10.
19. D. H. Bailey and P. N. Swarztrauber, "The fractional Fourier transform and applications," SIAM Rev. **33**, 389–404 (1991).