PHOTONICS Research

High-speed rendering pipeline for polygon-based holograms

Fan Wang,* Tomoyoshi Ito, and Tomoyoshi Shimobaba

Graduate School of Engineering, Chiba University, Chiba 263-8522, Japan
**Corresponding author: wangfan@chiba-u.jp*

Received 29 August 2022; revised 11 November 2022; accepted 19 December 2022; posted 20 December 2022 (Doc. ID 474158); published 1 February 2023

As an important three-dimensional (3D) display technology, computer-generated holograms (CGHs) have been facing challenges of computational efficiency and realism. The polygon-based method, as the mainstream CGH algorithm, has been widely studied and improved over the past 20 years. However, few comprehensive and high-speed methods have been proposed. In this study, we propose an analytical spectrum method based on the principle of spectral energy concentration, which can achieve a speedup of nearly 30 times and generate high-resolution (8K) holograms with low memory requirements. Based on the Phong illumination model and the sub-triangles method, we propose a shading rendering algorithm to achieve a very smooth and realistic reconstruction with only a small increase in computational effort. Benefiting from the idea of triangular subdivision and octree structures, the proposed original occlusion culling scheme can closely crop the overlapping areas with almost no additional overhead, thus rendering a 3D parallax sense. With this, we built a comprehensive high-speed rendering pipeline of polygon-based holograms capable of computing any complex 3D object. Numerical and optical reconstructions confirmed the generalizability of the pipeline.

https://doi.org/10.1364/PRJ.474158

1. INTRODUCTION

Holography is a very promising technology in the field of real three-dimensional (3D) displays, as it is capable of reconstructing all depth information of 3D objects. Computer-generated holograms (CGHs) can simulate virtual objects, thereby providing opportunities for dynamic 3D displays and augmented reality interactions [1]. In CGHs, a single unit (point, polygon, or any other unit) affects all pixels in the hologram, which introduces huge computations. Therefore, efficient computing of holograms is essential for video-rate 3D displays. Many studies have reported efficient methods that fall into three main routes: point-based [2–4], layer-based [5–7], and polygon-based methods [8–14]. Notably, the line-based method reported in recent years employs a new computational idea that calculates only the contour lines of the object [15–17].

Point-based methods have rapidly developed owing to their computational simplicity. Lookup table methods [18], wavefront recording plane methods [19], parallel computation in graphics processing units (GPUs) [20], and field-programmable gate arrays [21] significantly improve the efficiency of point-based holograms. However, for a smooth and continuous complex 3D object, the computational effort versus the reconstructed image quality is still a trade-off [22]. Deep learning-generated holograms using layer-based RGBD data open exciting avenues for fewer calculations and higher quality [23–27]. Nevertheless, CGHs using deep learning do not simulate optical diffraction and interference processes, which may cause unwanted artifacts in scenes with a large range depth and hinder the generation of holograms in large fields of view [22]. The polygon-based method can render more realistic 3D scenes by drawing on computer graphics techniques, but it is computationally intensive because of the difficulty of simplifying calculations in the frequency domain [28].

The polygon-based method treats objects as a collection of triangular faces and has undergone a progression from spectral interpolation [8,9] to analytical spectral expression [10–14]. The former finds spectra by solving the fast Fourier transform (FFT) for the polygon and then interpolates the spectrum to obtain the spectrum distribution in the hologram plane. This easily renders the shading or texture information for individual polygons; however, each loop must perform one FFT and one interpolation, which is highly costly [28–30]. The spectral interpolation-based approach admittedly presents a highly realistic 3D viewing experience, especially by adding random phases to enable multi-viewing with full parallax [28,31], whereas this is more applicable to printed holograms using laser lithography rather than electronic display holograms. The analytic polygon-based method generally employs rotational transformation [10] or affine transformation [11–14] to derive an analytical spectral expression for an arbitrary

triangle. The analytical expression significantly improves the computational performance of the polygon-based method because it does not require an FFT operation or interpolation. Nevertheless, individual triangles should be treated as having uniform transmittance, so detailed renderings such as shading require further subdivision surfaces, thus increasing the computational effort. Although researchers have reported some feasible solutions for presenting continuous shading without subdividing the objects [32,33], they either do not render specular reflection or add considerable additional overhead, as discussed in detail in Section 4.

In this study, we developed a comprehensive polygon-based hologram generation pipeline based on compact spectral regions and sub-triangulation, which essentially improves the computational efficiency of polygon-based holograms by rendering realistic reconstructed images. From the analysis of energy spectral density of the polygon [34], we observed that most spectral energy is concentrated in a small region with low frequency. Calculating only the low-frequency region can significantly accelerate the spectrum calculation, with little loss of quality in the hologram. For smooth objects, a shading model based on the surface normal is the fundamental path of shading. We propose a novel and simple shading method using the vertex-normal of sub-triangles, which can render very smooth shading without increasing the number of triangles.

In addition, the visibility culling technique is an important but challenging issue in CGH to obtain realistic 3D senses with motion parallax. Backface culling removes all backward triangles only [11,14], but not those facing forward, which may be occluded by groups of other objects in front of them. Occlusion culling is the most complex culling technique because it involves computing how objects (triangles) affect each other [35]. One candidate is the z-buffer technique. However, this approach works in the point-based method [36,37], but not in the polygon-based method where each triangular surface is treated as a whole. For a polygon-based hologram, a few effective occlusion culling methods have been proposed: Matsushima *et al.* [38–40] proposed a silhouette method used for the interpolation-based algorithm, and Askari *et al.* [41] improved the silhouette method to enable its use in the analytical expression-based algorithm. Both methods require considerable additional computational effort.

To solve this problem, we propose a novel occlusion culling method for polygon-based holograms using the idea of sub-triangles. The proposed occlusion culling method adopts the concept of collision detection to determine whether the ray from the viewpoint to the detection point intersects with other triangles in the view frustum [35]. If it intersects, it is invisible; otherwise, it is visible. Because traversing all triangles for the intersection test is costly and redundant, we used an octree structure to quickly search for possible collision objects.

As an outline, in Section 2, the analytical spectral expression is briefly derived. In Section 3, the analytical spectrum is accelerated using the controllable energy angular spectrum method (CE-ASM) [42]. Based on the subdivision of triangles into up-and-down congruent triangles, we propose a fast shading method in Section 4 and an occlusion culling method in Section 5. In Section 6, we discuss the contributions and limitations of the proposed rendering pipeline. Finally, we present our conclusions in Section 7.

2. POLYGON-BASED HOLOGRAM

Full-analytical spectral methods for polygon-based holograms have been derived using several approaches [10–14]. In this study, we use the full-analytic method based on 3D affine transformation originating from Pan *et al.* [13,43]

An arbitrary triangle $\Gamma(x, y, z)$ in the global coordinate system is mapped onto a primitive triangle $\Delta(x_0, y_0, z_0)$ in the local coordinate system, as shown in Fig. 1(a). The affine transformation theory states that the mapping relationship between them is

$$[x, y, z]^{\intercal} = \mathbf{T}[x_0, y_0, z_0, 1]^{\intercal}, \tag{1}$$

where $\mathbf{T} = [T_{ij}]_{3\times4}$ represents the 3D affine transform and $^{\intercal}$ denotes the transpose. $\mathbf{T}$ can be solved simply using

$$\mathbf{T} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^{\intercal}([\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0, \mathbf{1}]^{\intercal}), \tag{2}$$

where † is a pseudo-inverse operation [42,44]. $\mathbf{x}, \mathbf{y}, \mathbf{z}$ and $\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0$ are the vertex coordinate vectors of triangle Γ and triangle Δ, respectively. Let the vertices of triangle Γ be (x_i, y_i, z_i) and the vertices of primitive triangle Δ be (x_{0i}, y_{0i}, z_{0i}) with (0, 0, 0), (0, 1, 0), and (1, 1, 0), where $i = 1, 2, 3$.

A plane light wave at angles α and β to the x and y axes, respectively, passes through the triangular polygon with a uniform transmittance, as shown in Fig. 1(b). The complex amplitude on the triangular surface is

$$E_0 = \exp[j2\pi(x\cos\alpha + y\cos\beta + z\cos\gamma)/\lambda], \tag{3}$$

where λ is the wavelength, $j = \sqrt{-1}$, and $\cos\gamma = (1 - \cos^2\alpha - \cos^2\beta)^{1/2}$, where γ is the angle between the light source and z axis. In general, we define a light source parallel to the z axis, implying $\alpha = \beta = 0$, and then $E_0 = \exp(j2\pi z/\lambda)$.

In the frequency domain (f_x, f_y, f_z), the angular spectrum of E_0 is propagated to the hologram plane using the transfer function $H = \exp(-j2\pi f_z z)$, where $f_z = (1/\lambda^2 - f_x^2 - f_y^2)^{1/2}$. Therefore, the spectra of the triangle Γ in the hologram plane are

$$\begin{aligned} F(f_x, f_y) &= \iint_{\Gamma} E_0 \cdot \exp[-j2\pi \cdot (xf_x + yf_y)]\mathrm{d}x\mathrm{d}y \cdot H \\ &= \iint_{\Gamma} \exp[-j2\pi(\mathbf{f}\cdot\mathbf{v})]\mathrm{d}x\mathrm{d}y, \end{aligned} \tag{4}$$

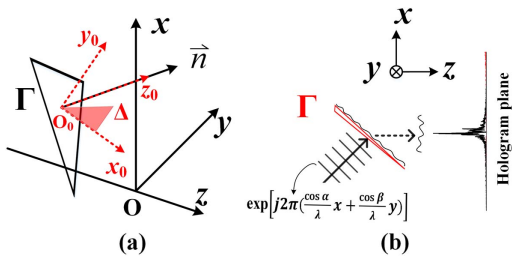

Fig. 1. Schematic of (a) affine transformation of two triangles and (b) triangular mesh diffraction with carrier wave.

where the vectors $\mathbf{f} = [f_x, f_y, f_z - 1/\lambda]$ and $\mathbf{v} = [x, y, z]$. Based on the mapping relationship given in Eq. (1), the above equation can be further derived as

$$F(f_x, f_y) = J \iint_{\Delta} \exp[-j2\pi(\mathbf{f}' \cdot \mathbf{v_0})]\mathrm{d}x_0\mathrm{d}y_0, \tag{5}$$

where J is the Jacobian factor, and the vectors $\mathbf{v_0} = [x_0, y_0, z_0, 1]$ and $\mathbf{f}' = [f'_x, f'_y, 0, f'_z]$ follow $\mathbf{f}' = \mathbf{fT}$.

By integrating Eq. (5), we can obtain a fully analytic spectral expression as follows:

$$F(f_x, f_y) = J \exp(-j2\pi f'_z) \cdot \left\{ \frac{\exp(-j2\pi f'_x) - 1}{(2\pi)^2 f'_x f'_y} + \frac{1 - \exp[-j2\pi(f'_x + f'_y)]}{(2\pi)^2 (f'_x + f'_y) f'_y} \right\}, \tag{6}$$

where $f'_x + f'_y \neq 0 \wedge f'_x \neq 0 \wedge f'_y \neq 0$. Equation (6) is the diffractive spectrum of a single triangle on the hologram plane, indicating that it relates only to the affine transform matrix $\mathbf{T}$ given in Eq. (2) and the frequency coordinates (f_x, f_y) defined in the global system. It is noteworthy that Eq. (6) does not apply to cases such as $f'_x = 0, f'_y = 0$ or $f'_x + f'_y = 0$. Therefore, we suggest a tip to avoid conditional judgement in the program, plus a tiny value to the matrix $\mathbf{T}$, that is, $\mathbf{T} = \mathbf{T} + \epsilon$, where $0 < \epsilon \ll 1$.

Consequently, the hologram generated by an object with N triangles is

$$H = \mathcal{F}^{-1}[F_{\mathrm{all}}(f_x, f_y)] = \sum_{i=1}^{N} a_i \cdot F_i(f_x, f_y), \tag{7}$$

where $\mathcal{F}^{-1}$ denotes inverse Fourier transform. $F_{\mathrm{all}}(f_x, f_y)$ is the aggregate spectrum of all triangles, where $F_i(f_x, f_y)$ is the spectrum of the ith triangle obtained using Eq. (6). a_i is the amplitude constant of the ith triangle using which the shading in different illumination models can be controlled, as described in Section 4.

3. ACCELERATION: SPECTRAL ENERGY FOCUS METHOD

Although an accurate and fast hologram calculation from triangles is given in Section 2, this section proposes a method called the spectral energy focus method for faster calculation of accelerated holograms by skipping unwanted spectrum calculations.

A. Theorem

The diffraction based on the angular spectrum method (ASM) shown in Eq. (4) is a circular convolution process with a transfer function. To avoid edge errors caused by circular convolution, Matsushima and Shimobaba pointed out that the computational window should be calculated by double sampling [45]. However, at a position farther than the critical distance $z_c = 2Np^2/\lambda$, where N is the number of samples and p is the sampling pitch, the high-frequency region cannot satisfy the Nyquist theorem because of the considerable oscillation of the transfer function of H. The band-limited ASM (BL-ASM) proposed by Matsushima and Shimobaba [45] suggests a boundary frequency $f_{\mathrm{BL}} = Np/z\lambda$, such that the frequency of sampling in the region beyond f_{BL} is replaced by 0, as shown in Fig. 2(a). The blue area is the effective bandwidth within f_{BL}, and the external region of f_{BL} is zero-padded and is marked by white circles. The number of pixels in the calculation was $2N \times 2N$ whereas the effective number of samples was $N_{\mathrm{BL}} = 4(Np)^2/z\lambda$. The effective bandwidth shrinks rapidly with increasing diffraction distance, leading to a decrease in accuracy.

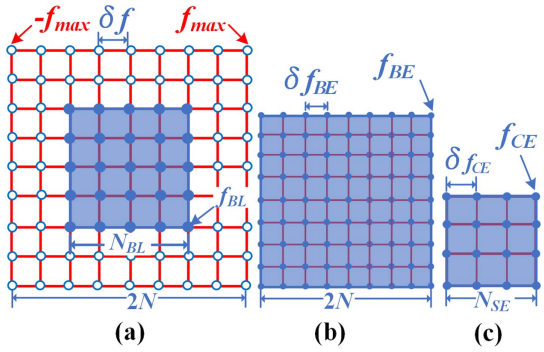

Fig. 2. Schematic diagram of the sampling strategy for (a) the band-limited, (b) the band-extended, and (c) the controllable energy angular spectrum methods. The blue areas are the effective bandwidth. The blue dots are the effective sampling points. The white circles are zero-padded. δf, δf_{BE}, and δf_{CE} are the sampling intervals of the three methods.

Zhang *et al.* proposed a band-extended ASM (BE-ASM) with a boundary frequency $f_{\mathrm{BE}} = \sqrt{N/2\lambda z}$ [46]. As shown in Fig. 2(b), the BE-ASM provides a wider bandwidth and is more robust at a longer distance than the BL-ASM, which implies the highest accuracy of results. However, this requires substantial computational effort because of the full use of $2N \times 2N$ sampling.

Herein, the CE-ASM, we previously proposed, is used to accelerate the calculation of the polygon-based hologram [34], which involves a lighter load compared to BL-ASM and BE-ASM. The CE-ASM considers spectral energy in addition to the sampling theorem. For the triangular frequency spectrum $F(f_x, f_y)$, the energy spectral density is defined as $|F(f_x, f_y)|^2$. According to Parseval's theorem [47], the energy of the frequency spectrum is

$$E(\tilde{f}_x, \tilde{f}_y) = 2 \int_0^{\tilde{f}_y} \int_0^{\tilde{f}_x} |F(f_x, f_y)|^2 \mathrm{d}f_x \mathrm{d}f_y, \tag{8}$$

where $\tilde{f}_x, \tilde{f}_y$ are the frequency boundaries. The full spectral energy of the triangle is $E(f_{x\max}, f_{y\max})$, where $f_{x\max} = f_{y\max} = 1/2p$ is the maximum sampling frequency of the hologram. The spectral energies contained in BL-ASM and BE-ASM are $E(f_{x\mathrm{BL}}, f_{y\mathrm{BL}})$ and $E(f_{x\mathrm{BE}}, f_{y\mathrm{BE}})$, respectively.

For a small triangle shown in Fig. 3(a), Fig. 3(b) presents its spectral energy density, a distribution with a central bulge and a flat surrounding. Different frequency boundaries enclose different regions, and the spectral energy within these regions can be calculated using Eq. (8) as shown in Fig. 3(c). The trend of Fig. 3(c) indicates that the spectral energy of the triangle is

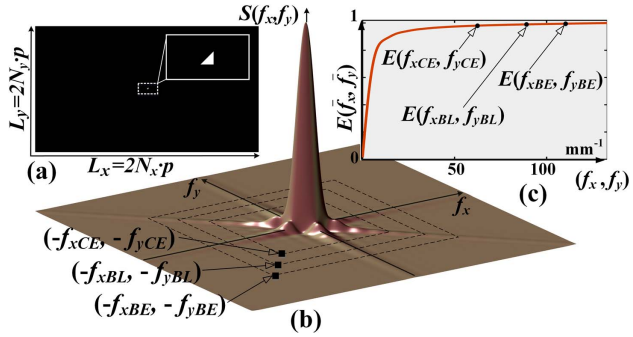

Fig. 3. (a) Rasterized triangle in the canvas under the parameters in Table 1. (b) Spectral energy density distributions at different frequency boundaries. (c) Normalized spectral energy distributions for different regions.

mainly focused on the low-frequency components, while the high-frequency component is negligible since the triangle is assumed to be uniform in amplitude. Therefore, we can obtain the vast majority of information by controlling a certain percentage of spectral energy.

In the CE-ASM mentioned in Ref. [34], the frequency boundaries $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$ were determined based on the widest bandwidth $(f_{x\mathrm{BE}}, f_{y\mathrm{BE}})$ because BE-ASM delivers an adequate reconstruction quality for considerably long propagation cases, such as tens of times the critical distance z_c. In general, however, the CGH displayed in a spatial light modulator (SLM) is not propagated to that extent; in this work, considering the optical setup, a distance of 250 mm was used, which is approximately twice as long as z_c under the parameters in Table 1. Therefore, in this study, we adopted $(f_{x\mathrm{BL}}, f_{y\mathrm{BL}})$ as the basis for spectral energy; that is, the proposed new frequency boundaries $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$ satisfy

$$E(f_{x\mathrm{CE}}, f_{y\mathrm{CE}}) \geq \eta E(f_{x\mathrm{BL}}, f_{y\mathrm{BL}}), \quad 0 < \eta \leq 100\%, \tag{9}$$

where η denotes a user-defined value. Above, $\eta \leq 100\%$ implies that $f_{x\mathrm{CE}} \leq f_{x\mathrm{BL}}$ and $f_{y\mathrm{CE}} \leq f_{y\mathrm{BL}}$ always exist, which further indicates that the number of samples in the CE-ASM follows

$$\begin{cases} N_{x\mathrm{CE}} = 4\lambda z f_{x\mathrm{CE}}^2 \leq N_{x\mathrm{BL}} \\ N_{y\mathrm{CE}} = 4\lambda z f_{y\mathrm{CE}}^2 \leq N_{y\mathrm{BL}} \end{cases}. \tag{10}$$

Then the sampling interval is

$$\begin{cases} \delta f_{x\mathrm{CE}} = 2f_{x\mathrm{CE}}/N_{x\mathrm{CE}} \\ \delta f_{y\mathrm{CE}} = 2f_{y\mathrm{CE}}/N_{y\mathrm{CE}} \end{cases}. \tag{11}$$

A sampling schematic of the CE-ASM is shown in Fig. 2(c). The number of samples and the spectral region of the CE-ASM are smaller than those of the BL-ASM, which indicates that the computational effort of the proposed method can be reduced. The extent of the reduction depends on the value of η, which will be discussed in detail in Section 3.B.

From the above, we can use $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$ to find the spectrum of the object $F_{\mathrm{all}}(f_x, f_y)$ using Eq. (6). However, the hologram cannot be obtained by executing an inverse FFT in Eq. (7) because $F_{\mathrm{all}}(f_x, f_y)$ is sampled at intervals of $\delta f_{x\mathrm{CE}}, \delta f_{y\mathrm{CE}}$, which is inconsistent with the sampling pitch of the hologram $\delta f_x = 1/2N_x p, \delta f_y = 1/2N_y p$. Therefore, a technique called non-uniform FFT (NUFFT) has been suggested for performing Eq. (7). NUFFT has been widely utilized to overcome the limitations of mismatched sampling intervals in the frequency and space domains [34,46,48]. Herein, we used the inverse type-3 NUFFT proposed by Lee and Greengard [49], where the hologram of the object is

$$H = \mathrm{NUFFT}_3^{-1}[F_{\mathrm{all}}(f_x, f_y)]. \tag{12}$$

Table 1. Frequency Parameters of Different Sampling Methods

Parameters	Values
Hologram pixels	$N_x = 1920, N_y = 1080$
Pixel pitch	$p = 3.74$ μm
Physical size	$L_x = 7.18$ mm, $L_y = 4.03$ mm
Wavelength	$\lambda = 532$ nm
Diffraction distance	$d = 250$ mm
Reference triangle	$l = 0.058$ mm
Vertices of the triangle	$(0,0,d), (0,l,d), (l,l,d)$
$(f_{x\max}, f_{y\max})$	$(133.7, 133.7)$ mm^{-1}
$2N_x \times 2N_y$	3840 × 2160
$(f_{x\mathrm{BE}}, f_{y\mathrm{BE}})$	$(95.0, 71.2)$ mm^{-1}
$N_{x\mathrm{BE}} \times N_{y\mathrm{BE}}$	3840 × 2160
$(f_{x\mathrm{BL}}, f_{y\mathrm{BL}})$	$(67.5, 40.0)$ mm^{-1}
$N_{x\mathrm{BL}} \times N_{y\mathrm{BL}}$	1938 × 613
$(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$	$(40.7, 22.1)$ mm^{-1}
$N_{x\mathrm{CE}} \times N_{y\mathrm{CE}}$	706 × 208

B. Solving for the Frequency Sampling Boundary of CE-ASM

In Section 3.A, a compact spectral region, $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$ is deduced from Eq. (9), which is an approximate region that includes most of the spectral energy. However, an appropriate value of η affects the energy within $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$, which also implies that the quality of the hologram is lost because of the approximation.

First, we discuss reference spectral energy $E(f_{x\mathrm{BL}}, f_{y\mathrm{BL}})$. Equation (8) indicates that the size of the reference triangle shown in Fig. 3(a), determines the effective spectral energy within $(f_{x\mathrm{BL}}, f_{y\mathrm{BL}})$. If the reference triangle is too small, a larger spectral region is involved; thus, the computational work is not significantly reduced. If it is large, the calculation can be accelerated, but the detailed representation of the 3D objects is lost. Therefore, for this trade-off, we analyzed the reference triangle in terms of visual acuity, which is a measurement of the eye's ability to perceive details. The limit of the human eye for resolving two distinct points is 1′ of arc [50]. We assume an isosceles right triangle with a side length of l as the reference triangle, which follows

$$l = d \tan(1'), \tag{13}$$

where d denotes the distance from the viewer, as shown in Fig. 4. Therefore, the reference spectral energy $E(f_{x\mathrm{BL}}, f_{y\mathrm{BL}})$ can be obtained using this small reference right triangle.

Second, we analyzed the weight factor η of the reference spectral energy in Eq. (9). η is a custom value that depends

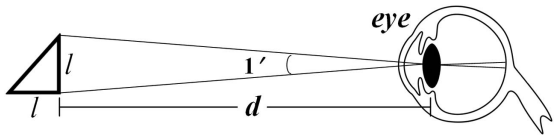

Fig. 4. Small triangle is defined as a reference triangle at the limiting angular resolution of the human eye.

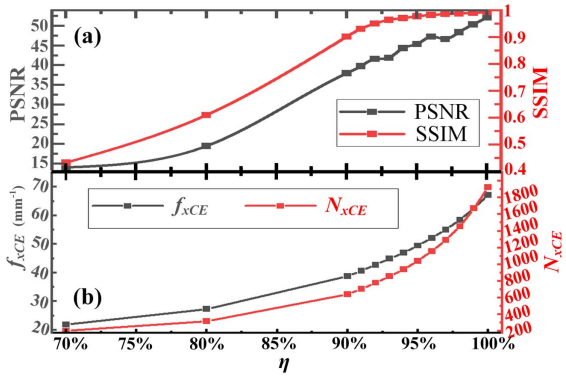

Fig. 5. (a) Quality evaluation of the holograms obtained by the CE-ASM by PSNR (left) and SSIM (right) at different η values. The reference holograms are obtained by the BE-ASM with the expanded spectra. (b) Spectral range (left) and the number of sampling (right) of the CE-ASM on the x axis at different η values. The hologram has 1920×1080 pixels.

on the quality of the hologram required by the user. Figure 5(a) shows the dependence of the hologram quality on the η value, where the peak signal-to-noise ratio (PSNR) and structure similarity index (SSIM) are used to measure the hologram quality. Holograms were calculated using the parameters listed in Table 1. The reference hologram used as a metric was obtained by the BE-ASM under double sampling because the BE-ASM has the highest accuracy.

From Fig. 5(a), the PSNR exceeds 35 dB and the SSIM stays above 0.9 when $\eta > 90\%$, which indicates that the value of η is directly proportional to the quality. The boundary frequency and number of samples of the CE-ASM corresponding to different values of η are shown in Fig. 5(b). From this, we can observe that the spectral region and number of samples increase with η. When $\eta = 100\%$, $f_{x\mathrm{CE}} = f_{x\mathrm{BL}}$ and $N_{x\mathrm{CE}} = N_{x\mathrm{BL}}$.

In general, images with a PSNR greater than 30 dB are considered acceptable because the human eye has a limited ability to recognize noise [51], and the highest quality of holographic displays in current experiments [24,27] is difficult to reach 30 dB. Therefore, to obtain higher quality holograms, we select $\eta = 91\%$ as an energy threshold in all calculations below, with a PSNR of 37.5 dB and SSIM of 0.93. In this case, the hologram samples the spectral region of $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}}) = (40.7, 22.1)\,\mathrm{mm}^{-1}$ with the number of samples $N_{x\mathrm{CE}} \times N_{y\mathrm{CE}} = 706 \times 208$, reducing the number of samples by approximately 56 times at a high fidelity compared with the BE-ASM with the number of samples $N_{x\mathrm{BE}} \times N_{y\mathrm{BE}} = 3840 \times 2160$. Note that for higher-resolution holograms (such as 8K pixels), the BE-ASM requires a large amount of memory, which is difficult to achieve on a normal PC, whereas the proposed method requires very little memory owing to the compressed sampling.

The spectral region $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$ deduced by the CE-ASM is referred to as the compact spectra, whereas the spectral region $(f_{x\mathrm{BE}}, f_{y\mathrm{BE}})$ deduced by the BE-ASM is referred to as the extended spectra.

C. Confirmed in Results

In this subsection, we confirm the principle and efficiency of the proposed compact spectral method by simulating concrete 3D objects.

Based on the parameters defined in Table 1 and the deduced $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$, a hologram of a teapot model consisting of 1560 triangles located at $z = 250$ mm with a depth of field of 5 mm was generated. Figures 6(a) and 6(b) show the holograms calculated using the extended spectra (BE-ASM) and compact spectra (CE-ASM), respectively, and Figs. 6(c) and 6(d) show

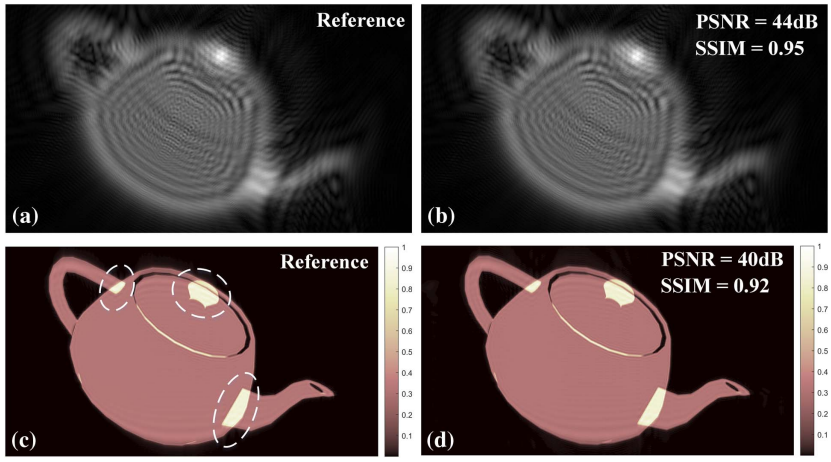

Fig. 6. (a) and (b) are holograms generated with the extended spectral region (BE-ASM) and the proposed compact spectral region (CE-ASM), respectively. (c) and (d) are the numerical reconstructions of (a) and (b), respectively. With the results of the extended spectral region as references, (b) and (d) show image quality by PSNR and SSIM.

Table 2. Calculated Results and Comparison for Objects Composed of Different Numbers of Triangles

Objects	Teapot	Rings	Hand	Soccer	Wang	Bunny	Angel
Number of triangles	1560	5760	8420	12,282	18,228	32,030	64,363
Extended spectra (s)	5.2	32.5	57.1	78.85	119.9	203.1	417.7
Compact spectra (s) (proposed)	0.4	1.2	2.0	2.70	4.0	6.6	13.7
Acceleration	**14.1**	**26.0**	**28.6**	**29.2**	**29.8**	**30.8**	**30.5**
PSNR[a]	44	45.1	42.4	46.2	51	46.3	47.4
SSIM[a]	0.95	0.97	0.96	0.97	0.99	0.96	0.97

[a]The PSNR and SSIM listed here are only for holograms and not for the reconstructed images.

the corresponding reconstructed images. The reconstructed image in Fig. 6(d) shows a negligible background apparition. Using the results of Figs. 6(a) and 6(c) as the references, PSNRs of the hologram and the reconstructed image are 44 dB and 40 dB, and their SSIMs are 0.95 and 0.92, respectively. This high fidelity demonstrated the effectiveness of the proposed method.

Notably, the actual number of triangles used for spectrum calculation is almost half of the total number because the backward triangles are easily recognized and removed by the surface normal. The very bright parts of the hologram in Fig. 6, such as the area indicated by dotted circles, are representations of the triangles overlapping each other, which requires occlusion culling, as discussed in Section 5. In addition, the amplitude of the reconstructed images was uniform because of the absence of illumination. The complex shading model is described in the following section.

In the computational environment of MATLAB 2021a, using a personal computer configured with CPU: AMD Ryzen 5-3600 at 3.59 GHz and GPU: Nvidia GeForce RTX 3070, we further calculated 3D objects with more triangles using the extended spectral region and the proposed compact spectral region. The computational efficiency values are listed in Table 2. Because the number of pixels used for the calculation in the compact spectral region is greatly reduced, the computational efficiency of the proposed method can be improved by a factor of approximately 30 as the number of polygons increases. Moreover, the PSNR and SSIM values of the holograms remained above 40 dB and 0.95, respectively. Thus, we conclude that the proposed compact spectral method is both efficient and accurate.

4. SMOOTH SHADING

A. Shading Model in Computer Graphics

Shading renders a photorealistic scene of 3D objects. The commonly used rendering model in computer graphics is the Phong shading model [52], which linearly interpolates the normal across the surface of the polygon. As shown in Fig. 7(a), based on known vertex normals (solid line arrows), the normal vectors of each pixel inside the surface (dashed arrows) are obtained by interpolation and then normalized. Figure 7(b) shows an interpolation schematic of a triangular surface based on the known normal vectors of the three vertices.

Under the illumination of incident light $-\hat{\mathbf{L}}$, as shown in Fig. 7(c), the amplitude of pixels on a polygon is obtained from a reflection model when viewed along the direction vector $\hat{\mathbf{V}}$. In this study, we use the Blinn–Phong reflection model [53],

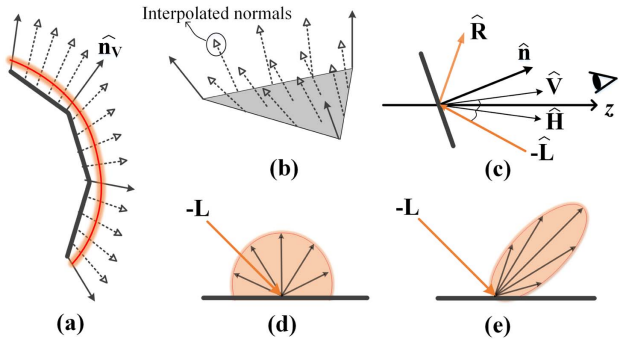

Fig. 7. (a) Phong shading model: obtaining the normal of each pixel using linear interpolation based on three known vertex normals. (b) Schematic of interpolation of pixel normals within a triangle. (c) Blinn–Phong reflection model. $\hat{\mathbf{L}}$ points along the direction of the light source. (d) Diffuse reflection diagram. (e) Specular reflection diagram.

which provides an equation for computing the amplitude of each surface point:

$$\begin{aligned} I &= \text{Blinn-Phong}(\hat{\mathbf{n}}, \hat{\mathbf{L}}, \hat{\mathbf{V}}) \\ &= K_a + K_d(\hat{\mathbf{n}} \cdot \hat{\mathbf{R}}) + K_s(\hat{\mathbf{n}} \cdot \hat{\mathbf{H}})^{N_s}, \end{aligned} \tag{14}$$

where $\hat{\mathbf{n}}$ is the interpolated normal and $\hat{}$ denotes the unit vector. The vector $\hat{\mathbf{R}}$ denotes the reflected ray of $-\hat{\mathbf{L}}$ at the point, which can be calculated using

$$\hat{\mathbf{R}} = 2(\hat{\mathbf{L}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \hat{\mathbf{L}}. \tag{15}$$

$\hat{\mathbf{H}}$ is defined as the halfway vector between the viewer and light-source vectors, calculated using

$$\hat{\mathbf{H}} = \frac{\hat{\mathbf{V}} + \hat{\mathbf{L}}}{||\hat{\mathbf{V}} + \hat{\mathbf{L}}||}, \tag{16}$$

which is a modification of the Phong reflection model. The constants K_a, K_d, and K_s in Eq. (14) represent the ambient, diffuse, and specular reflection factors, respectively. Diffuse reflection depicts a rough surface that scatters the incident ray at many angles, as shown in Fig. 7(d). Specular reflection is the mirror-like reflection of light, which results in a significant effect on the surface along a specific angle, as shown in Fig. 7(e). The constant N_s is the shininess factor of the material that determines the size of the specular highlight.

In CGH, a common method is to first calculate the amplitude distribution within the surface based on the shading model above and then perform diffraction calculations. For example, Yamaguchi *et al.* rendered rough planes using the

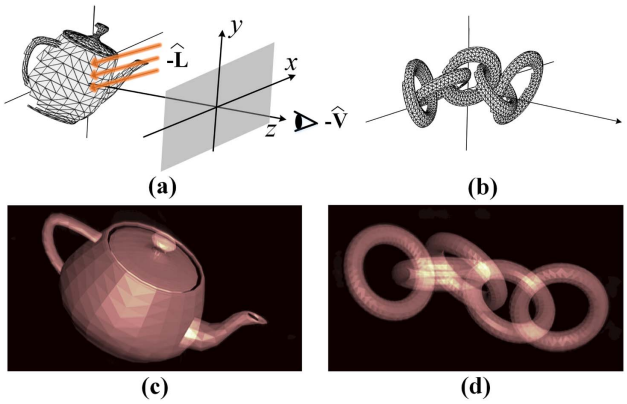

Fig. 8. (a) Teapot with 1560 triangles and (b) rings with 5760 triangles located in the 3D space are illuminated by the light ray $-\hat{\mathbf{L}}$ and observed with the vector $-\hat{\mathbf{V}}$. (c) and (d) are reconstructed images of the teapot and rings rendered with the flat shading model, respectively.

Table 3. Parameters Used in the Shading Model[a]

Illumination	L	V	K_a	K_d	K_s	N_s
$\phi_x = 70°$ $\phi_y = 85°$	$(\cos\phi_x, \cos\phi_y, \cos\phi_z)$	(0,0,1)	0.2	0.8	0.8	25

[a]We define $\cos\phi_z = (1 - \cos^2\phi_x - \cos^2\phi_y)^{1/2}$.

Cook–Torrance reflection model [54], expecting to present a vision-effect of different materials [30,55]; Nishi *et al.* added smooth shading to triangular surfaces using the Phong reflection model and analyzed in detail the effect of each parameter on the results [29]. However, they all used an interpolation-based method to calculate the spectrum, which is time-consuming. An efficient method is to introduce the shading model into the analytical spectra method discussed in Section 2. However, from Eq. (6), the analytical spectrum considers the amplitude of each triangle to be uniform, making the flat shading model easy to implement, as shown in Fig. 8. The two 3D objects, shown in Figs. 8(a) and 8(b), teapot and rings, are illuminated by a parallel beam of light, which is at the angle ϕ_x to the x axis and ϕ_y to the y axis. The light ray is reflected in the hologram plane following the Blinn–Phong reflection model in Eq. (14) and is observed along the z axis. The reflection parameters are listed in Table 3. From the reconstructions in Figs. 8(c) and 8(d), the exaggerated step of adjacent triangles produces a distinct ridge, called the Chevreul illusion [50].

To address the Chevreul illusion, Park *et al.* developed an attractive continuous shading model that is essentially an extension of the Gouraud shading model, from interpolation of vertex amplitude to analytical spectra [32]. Park *et al.*'s shading method uses the reflection model

$$I^{\text{Park}} = K_a + K_d(\hat{\mathbf{n}} \cdot \hat{\mathbf{R}}), \tag{17}$$

which includes only the ambient and diffuse reflections, while the Blinn–Phong reflection model in Eq. (14) also includes the specular reflection. The triangle Γ with known vertex normals $\hat{\mathbf{n}}_i$, $i = 1, 2, 3$, as shown in Fig. 9, is illuminated by the light $-\hat{\mathbf{L}}$. The reflection amplitude of the three vertices can be

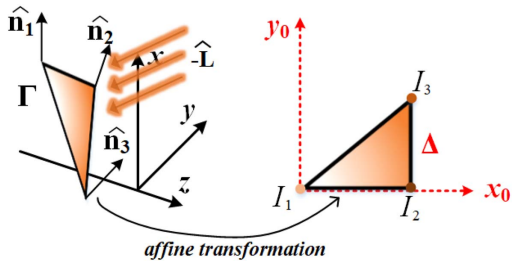

Fig. 9. Schematic of continuous shading method without specular reflection proposed by Park *et al.* [32]. On the left is the spatial triangle in global coordinate system and on the right is the original triangle in local coordinate system. An affine transformation exists between them.

obtained using Eq. (17), defined as I_1, I_2, and I_3. The amplitude of pixel (x_0, y_0) within triangle Δ satisfies the following interpolation expression:

$$I^{\text{Park}}(x_0, y_0) = I_1 + (I_2 - I_1)x_0 + (I_3 - I_2)y_0. \tag{18}$$

Because triangle Γ is affine transformed into the right triangle Δ in the local system described in Section 2 (see Fig. 1), the amplitude of pixel (x, y, z) within the triangle Γ defined in Eq. (7) is

$$a^{\text{Park}}(x, y, z) = I^{\text{Park}}(x_0, y_0). \tag{19}$$

By substituting the above equation into Eqs. (4) and (5), the spectrum of triangle Γ using integration operations is

$$F^{\text{Park}}(f_x, f_y) = I_1 D_1 + (I_2 - I_1)D_2 + (I_3 - I_2)D_3, \tag{20}$$

where D_1 is the same as in Eq. (6), and

$$D_2 = Je^{-j2\pi f'_z} \cdot \left\{ \frac{je^{-j2\pi f'_x}(-j2\pi f'_x + e^{j2\pi f'_x} - 1)}{8\pi^3 f'^2_x f'_y} + \frac{je^{-j2\pi(f'_x + f'_y)}[1 + j2\pi(f'_x + f'_y)] - j}{8\pi^3 (f'_x + f'_y)^2 f'_y} \right\}, \tag{21}$$

$$D_3 = Je^{-j2\pi f'_z} \cdot je^{-j2\pi(f'_x + f'_y)} \left[\frac{f'_x(j2\pi f'_y - e^{j2\pi f'_y} + 1)}{8\pi^3 (f'_x + f'_y)^2 f'^2_y} + \frac{(-1 + e^{j2\pi f'_x})e^{j2\pi f'_y}}{8\pi^3 (f'_x + f'_y)^2 f'_x} + \frac{2(j\pi f'_y - e^{j2\pi f'_y} + 1)}{8\pi^3 (f'_x + f'_y)^2 f'_y} \right]. \tag{22}$$

Equation (20) provides a fully analytical expression with continuous shading at the cost of adding additional computational overhead to Eqs. (21) and (22). Although this cost is not significant, the weakness of Park *et al.*'s shading method is the absence of specular reflections, as shown in the first column of Fig. 10. As a comparison, Fig. 10 also shows the reconstructed results of the proposed sub-triangle-based shading method, which will be introduced below. From Fig. 10, Park *et al.*'s shading model lacks vividness due to the absence of highlights generated by specular reflection.

B. Proposed Shading Model with Specular Reflection

Benefitting from the sub-triangle concept of Kim *et al.* [10], we propose an economical smooth shading method with specular

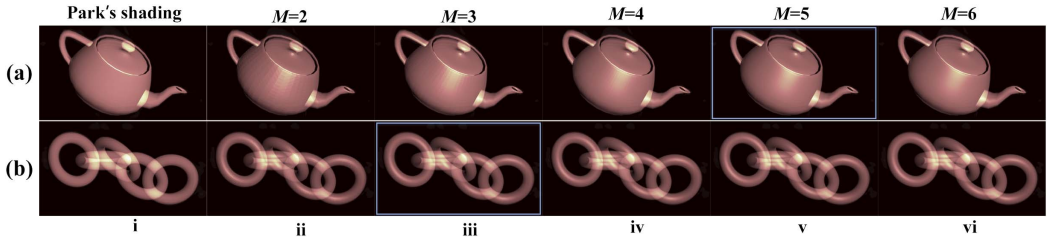

Fig. 10. Reconstructed results of Park *et al.*'s shading method (first column) and of the proposed sub-triangle-based shading method (columns 2 to 6). Teapot (a) and rings (b) are subdivided at $M = 2$ to 6 in the proposed method. All results follow the Blinn–Phong reflection model and the parameters given in Table 3.

reflection based on vertex normal interpolation of sub-triangles, unlike Park *et al.*'s shading method of interpolating pixel amplitude instead of normals [given in Eq. (18)]. Figure 11 shows a schematic of the subdivision of mother triangle ΔABC. The vertices of the mother triangle are $\mathbf{v}_A, \mathbf{v}_B$, and $\mathbf{v}_C$, and the known vertex normals are $\hat{\mathbf{n}}_A, \hat{\mathbf{n}}_B$, and $\hat{\mathbf{n}}_C$. Each side of the mother triangle is equally divided into M segments, yielding M^2 sub-triangles, which include $M(M+1)/2$ upper sub-triangles denoted $\mathbf{v}_\uparrow$ and $M(M-1)/2$ lower sub-triangles denoted $\mathbf{v}_\downarrow$. A non-orthogonal coordinate system is established with $\overrightarrow{AB}$ as m-axis and $\overrightarrow{AC}$ as n-axis. Each sub-triangle is then expressed as $\mathbf{v}_\uparrow(m,n)$, which can be obtained by shifting the vertices of the original sub-triangle as follows:

$$\begin{aligned}\mathbf{v}^s_\uparrow(m,n) &= \mathbf{v}^s_\uparrow(0,0) + m\mathbf{e}_{AB} + n\mathbf{e}_{AC},\\ 0 \le m+n \le M-1 \ &\text{for } m,\ n \in \mathbf{N}_0,\end{aligned} \tag{23}$$

where $s = 1,2,3$ denotes the three vertices of the sub-triangle, and $\mathbf{e}_{AB}$ and $\mathbf{e}_{AC}$ are the unit vectors of m and n axes, respectively. $\mathbf{v}^s_\uparrow(0,0)$ represents the vertex coordinates of the original sub-triangle, as $\Delta AB'C'$ shown in the upper-right corner of Fig. 11. Therefore, assuming that the spectrum of the original sub-triangle $\mathbf{v}_\uparrow(0,0)$ is $F_{\uparrow(0,0)}$ which is calculated using Eq. (6), the spectrum of triangle $\mathbf{v}_\uparrow(m,n)$ is

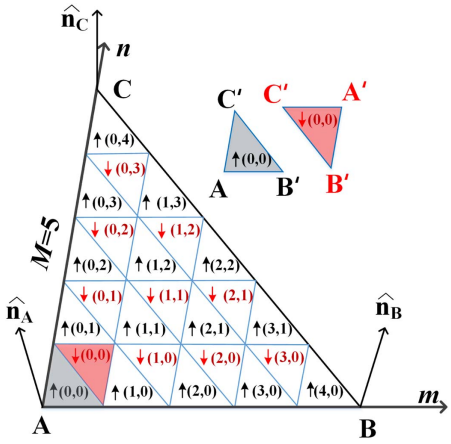

Fig. 11. Schematic of subdividing the mother triangle ΔABC at $M = 5$, yielding 16 up sub-triangles and 9 down sub-triangles. $\Delta AB'C'$ and $\Delta A'B'C'$ are the original upper and lower sub-triangles, respectively.

$$F_{\uparrow(m,n)} = F_{\uparrow(0,0)} \cdot \exp[-j2\pi\mathbf{f}(m\mathbf{e}_{AB} + n\mathbf{e}_{AC})]. \tag{24}$$

Similarly, the spectrum of triangle $\mathbf{v}_\downarrow(m,n)$ is

$$\begin{aligned}F_{\downarrow(m,n)} &= F_{\downarrow(0,0)} \cdot \exp[-j2\pi\mathbf{f}(m\mathbf{e}_{AB} + n\mathbf{e}_{AC})],\\ 0 \le m+n \le M-2 \ &\text{for } m,\ n \in \mathbf{N}_0 \ .\end{aligned} \tag{25}$$

$F_{\downarrow(0,0)}$ is the spectrum of triangle $\mathbf{v}_\downarrow(0,0)$. Because triangle $\mathbf{v}_\downarrow(0,0)$ is symmetric to triangle $\mathbf{v}_\uparrow(0,0)$ about side B′C′, there must exist a simple translation relationship. Therefore, $F_{\downarrow(0,0)}$ can be calculated as

$$F_{\downarrow(0,0)} = F^*_{\uparrow(0,0)} \cdot \exp[-j2\pi\mathbf{f}(\mathbf{v}_{\mathbf{B}'} + \mathbf{v}_{\mathbf{C}'})], \tag{26}$$

where $F^*_{\uparrow(0,0)}$ is the conjugate of $F_{\uparrow(0,0)}$.

Equations (24) and (25) indicate that the spectra of all sub-triangles can be directly obtained by simply shifting the spectrum of the original up sub-triangle $\mathbf{v}_\uparrow(0,0)$. That is, regardless of the number of sub-triangles the mother triangle is subdivided into, we only need to calculate the spectrum of $\mathbf{v}_\uparrow(0,0)$ using Eq. (6). The concise computation was an improvement in Refs. [10,43].

Similar to the interpolation of the sub-triangle vertex coordinates in Eq. (18), vertex normals can also be obtained by interpolating the vertex normals of the mother triangle, $\hat{\mathbf{n}}_A, \hat{\mathbf{n}}_B$, and $\hat{\mathbf{n}}_C$. In this study, we used the average of the vertex normals as the face normal of the sub-triangles. Therefore, we define $\mathbf{n}_\uparrow(m,n)$ and $\mathbf{n}_\downarrow(m,n)$ as the face normals of triangles $\mathbf{v}_\uparrow(m,n)$ and $\mathbf{v}_\downarrow(m,n)$, respectively, as follows:

$$\begin{cases}\mathbf{n}_\uparrow(m,n) = \hat{\mathbf{n}}_A + \frac{\hat{\mathbf{n}}_B - \hat{\mathbf{n}}_A}{M}\left(m+\frac{1}{3}\right) + \frac{\hat{\mathbf{n}}_C - \hat{\mathbf{n}}_A}{M}\left(n+\frac{1}{3}\right)\\ \mathbf{n}_\downarrow(m,n) = \hat{\mathbf{n}}_A + \frac{\hat{\mathbf{n}}_B - \hat{\mathbf{n}}_A}{M}\left(m+\frac{2}{3}\right) + \frac{\hat{\mathbf{n}}_C - \hat{\mathbf{n}}_A}{M}\left(n+\frac{2}{3}\right)\end{cases}$$

$$\Rightarrow \begin{cases}\hat{\mathbf{n}}_\uparrow(m,n) = \frac{\mathbf{n}_\uparrow(m,n)}{||\mathbf{n}_\uparrow(m,n)||}\\ \hat{\mathbf{n}}_\downarrow(m,n) = \frac{\mathbf{n}_\downarrow(m,n)}{||\mathbf{n}_\downarrow(m,n)||}\end{cases}, \tag{27}$$

where $\hat{\mathbf{n}}_\uparrow(m,n)$ and $\hat{\mathbf{n}}_\downarrow(m,n)$ denote the normalized vectors. Based on the Blinn–Phong reflection model, the amplitudes of each sub-triangle, defined as $I_{\uparrow(m,n)}$ and $I_{\downarrow(m,n)}$, are

$$\begin{cases}I_{\uparrow(m,n)} = \text{Blinn-Phong}(\hat{\mathbf{n}}_\uparrow(m,n), \hat{\mathbf{L}}, \hat{\mathbf{V}})\\ I_{\downarrow(m,n)} = \text{Blinn-Phong}(\hat{\mathbf{n}}_\downarrow(m,n), \hat{\mathbf{L}}, \hat{\mathbf{V}})\end{cases}. \tag{28}$$

Consequently, by combining Eqs. (24), (25), and (28), the spectrum of the ith mother triangle in Eq. (7) with smooth shading, can be calculated as follows:

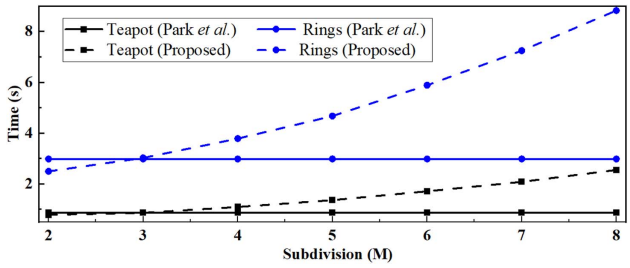

Fig. 12. Efficiency comparison between the Park *et al.*'s method (solid line) and the proposed method (dashed line) at different values of M. From Fig. 11, M denotes the subdivision level of the mother triangle. The Park *et al.*'s method is independent of M; thus, it behaves as a constant as M increases.

$$F_i(f_x, f_y) = \sum_{n=0}^{M-1}\sum_{m=0}^{M-n-1} I_{\uparrow(m,n)}F_{\uparrow(m,n)} + \sum_{n=0}^{M-2}\sum_{m=0}^{M-n-2} I_{\downarrow(m,n)}F_{\downarrow(m,n)}, \tag{29}$$

which indicates that uniformly varying amplitudes are determined by the subdivision level of the mother triangle, i.e., the value of M. Although the amplitude gradient is theoretically discontinuous, larger M values result in a smoother rendering of the object surface. Figure 10 shows the reconstructed results of the teapot with 1560 triangles and rings with 5760 triangles for different values of M. We can see that when $M \geq 5$, the reconstructed images of the teapot no longer exhibit a significant smoothness change, whereas for the rings, it does not change from $M = 3$. In Fig. 12, the dashed lines show the elapsed time of the proposed sub-triangle method at different values of M. For comparison, the solid lines represent Park *et al.*'s method [32], which is independent of M. From Fig. 12, as M increases, the elapsed time of the proposed method will exceed that of Park *et al.*'s method. However, within the smoothing range perceptible to the human eye, we can choose an appropriate M to allow the most economical calculation.

In summary, the sub-triangle shading method simulates the Phong shading model by replacing pixel normals with sub-triangle normals, which outperforms the method proposed by Park *et al.* [32] in terms of realistic rendering. Park *et al.*'s method uses the Gouraud shading model that interpolates pixel amplitude, and it also omits the specular reflection term, resulting in a loss of gloss. In terms of computational efficiency, the improved sub-triangle shading method simplifies the computational process, and thus also outperforms the traditional method of rendering by increasing the number of mother triangles [29,30] and the convolution-based method proposed by Yoem and Park [33].

5. OCCLUSION CULLING

A. Basic Mechanism of Occlusion Culling

Occlusion provides the most powerful monocular depth cues. As in all the reconstructed images shown in the previous sections, 3D objects appear as unusually bright areas because of the overlap of triangles along the same light of sight, such as the dashed circles in Fig. 6 and the interlocking rings shown in Fig. 10, which is known as incorrect occlusion.

In the polygon-based hologram, Matsushima *et al.* proposed an impressive method for occluded surface removal: the silhouette method [38–40]. However, the silhouette method requires the sorting of surfaces from front to back and at least two propagation processes for each surface, which is not economical. Askari *et al.* improved the silhouette method in the frequency domain using an analytical spectral expression [41], which avoids propagation twice and enhances the efficiency. However, this method employs three FFTs for each triangle. We argue that this is not very economical. In this section, we propose a novel occlusion culling method to effectively merge into the rendering pipeline of polygon-based holograms, while maintaining a full analytical frequency spectrum.

We first cull the backfaces that follow $\hat{\mathbf{V}} \cdot \hat{\mathbf{n}} \leq 0$, where $\hat{\mathbf{V}}$ is the observation direction, $\hat{\mathbf{V}} = (0, 0, 1)$, and $\hat{\mathbf{n}}$ is the surface normal. Assume that the 3D object has N_0 triangles, and N_f forward triangles remain after performing backface culling. However, some forward triangles are still invisible, even if their normal points to the observer, such as multiple objects in front–back alignment, concave objects, and inter-occlusion triangles, as shown in Fig. 13(a). Therefore, we used collision detection to implement occlusion culling. Unlike computer graphics, where collisions are detected for each rasterized pixel, the proposed method performs collision detection for the vertices of all the forward triangles.

Figure 13(b) illustrates collision detection for ΔDEF. Detected vertices D, E, and F emit a ray into the hologram plane, and if this ray intersects a triangle, the vertex is occluded. Note that an orthogonal projection is used here, i.e., all the ray vectors are parallel to z axis. Therefore, the task becomes a ray–triangle intersection test in geometry. In this study, we employ the intersection test algorithm optimized by Moller and Trumbore [56], hereafter referred to as

$$\mathrm{TF} = \mathrm{RayTriIntersect}(\mathbf{v}_i, \mathrm{T}, \mathbf{e}_r), \tag{30}$$

where $\mathbf{v}_i$ denotes the coordinate of the detected vertex, T denotes the triangular set of potential occluders, $\mathbf{e}_r$ denotes the unit vector of the ray, and $\mathbf{e}_r = \hat{\mathbf{V}}$. RayTriIntersect $(\mathbf{v}_i, \mathrm{T}, \mathbf{e}_r)$ returns TF = true, implying that the vertex $\mathbf{v}_i$ intersects one of the triangles, otherwise, "false", implying no intersection. Generally, determining whether a vertex is occluded requires traversing all triangles and performing an intersection test. In other words, for N_f triangles with N_v vertices, the algorithm RayTriIntersect repeats $N_f \times N_v$ times, which has significant redundancy.

B. Construct an Octree for Collision Detection

To avoid redundant queries for a more accurate intersection test, we developed an octree spatial data structure to organize the object geometry. An octree is a type of hierarchical structure, that is, the data are arranged at different levels from top to bottom. The topmost level contains eight children, each of which has its own eight or no children. For example, in Fig. 14, a cubical box that encloses the object is split into eight children boxes with the same size along the three axes, and each children box continues to split further into eight unless this box

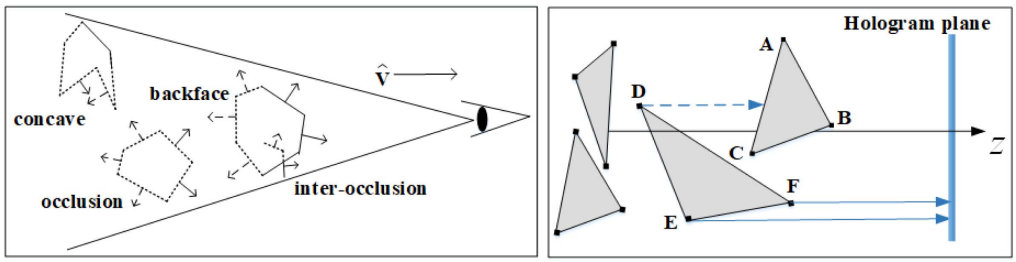

Fig. 13. (a) Schematic diagram of backface culling and occlusion culling. (b) Schematic diagram of collision detection.

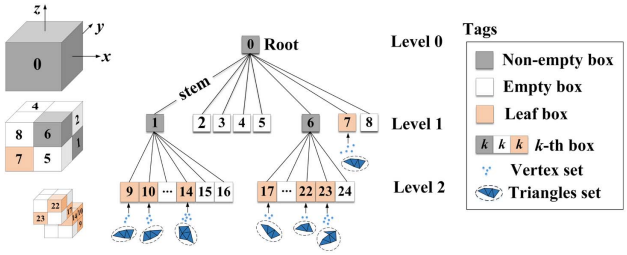

Fig. 14. Schematic diagram of an octree structure. A cube encloses the object, and then splits separately into eight children boxes. Empty boxes, such as boxes 2 and 3, and leaf boxes, such as box 7, stop splitting. However, the non-empty boxes (1 and 6) continue to split until a leaf box or empty box appears. Each leaf box contains a vertex set, which forms a set of triangles. The triangle set affiliated with a certain leaf box is the object of performing the intersection test.

is empty or leaf. The leaf box is identified by certain criteria, such as reaching the maximum recursion depth or containing fewer than a certain number of vertices. The bottommost level of each stem is either a leaf box with some vertices or an empty box. The vertices in the leaf box are shared by some triangles; therefore, we consider that these triangles are affiliated with this leaf box.

In this study, we used a publicly available program to construct the octree data structure of 3D objects [57], hereafter referred to as

$$\text{Octree}(\mathbf{v}, \text{“leafCriteria”}), \tag{31}$$

where $\mathbf{v}$ are the coordinates of all vertices, and property “leafCriteria” represents the criteria for being a leaf box. For example, a teapot consisting of 1560 triangles (containing 821 vertices) was divided into 441 boxes with up to five levels. The leaf box contained no more than 10 vertices. Suppose k^l represents the kth box located at the l level, as shown in Fig. 14. If k is an odd number, then the $(k+1)^l$th box is the potential collision box of the k^lth box. Conversely, if k is an even number, there is no potential collision box at the l level and the query should be directed to the $l-1$ level.

The hierarchical structure of the octree allows us to query potential collision boxes along the stem to the root rather than search through the entire object, which implies a reduction in performing the intersection test from $N_f \times N_v$ times to $\log(N_f) \times N_v$ times. During the query, as soon as a potential collision box appeared, the triangles affiliated with this box were immediately tested for intersections using Eq. (30). If the intersection with any triangle turns out to be “true,” the query is stopped, and the conclusion that this vertex is occluded is returned. In summary, the entire query is recursive in nature and ends in two cases: either the intersection test returns “true,” which means occlusion, or the query reaches the root, which means non-occlusion.

C. Occlusion Culling Based on Sub-Triangles

Collision detection returns the result of whether each vertex is occluded and the collision box k^l if it is occluded. Among the triangles formed by these vertices, we classify them into four types: the first type is the fully visible triangle, i.e., none of the three vertices is occluded, defined as the set T_0; the second and third types are those with one and two invisible vertices, respectively, defined as the sets T_1 and T_2, which are partially occluded; the fourth type is the fully invisible triangle, three vertices of which are occludees, defined as T_3. Figure 15(a) shows the reconstruction results of only T_0 set for the teapot and rings, which appear fragmented because of the lack of partially occluded triangles. Therefore, the triangle sets T_1 and T_2 must be considered in the calculations.

In the case of occlusion culling for the triangle sets T_1 and T_2, we follow the idea of subdividing the triangles presented in Section 4 to be able to use the analytic spectrum of triangles. Figure 16(a) shows two exemplary cases of T_1 triangles, and Fig. 16(b) shows two exemplary cases of T_2 triangles. Taking Fig. 16(a) as an example, ΔABC is labeled as occludee after collision detection due to the invisibility of vertex A, and the corresponding collision box is k^l. According to the description in Section 4, the occluded triangles are divided into M^2 congruent sub-triangles. For differentiation, we define M_0 as the subdivision of T_0-type triangles, M_1 as the subdivision of T_1-type triangles, and M_2 as the subdivision of T_2-type triangles. The vertex coordinates of each sub-triangle can be solved using Eq. (23). Then, starting from vertex A, each sub-triangle vertex is again fed into the collision detection loop for the interaction test. If the vertex is obscured, all sub-triangles using this vertex are culled, as shown by the blue-shaded sub-triangles in Figs. 16(a)–16(d). Therefore, the final spectrum of the object should be the sum of the triangles retained after occlusion culling, expressed as

$$F_{\text{visible}} = F_{\mathrm{T}_0} + F_{\mathrm{T}_1} + F_{\mathrm{T}_2}, \tag{32}$$

where F_{T_0} is the spectrum of the set of T_0 triangles, and F_{T_1} and F_{T_2} represent the spectrum after occlusion culling for the

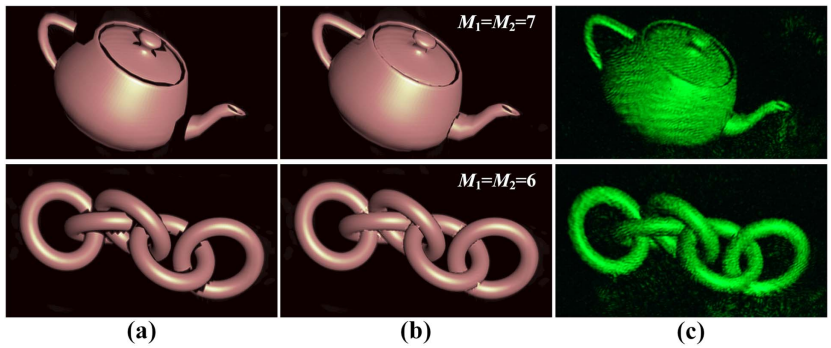

Fig. 15. Reconstructed results by the proposed occlusion culling method. (a) Results of computing only the set of T_0 triangles whose three vertices are visible. The teapot and ring were subdivided with $M_0 = 5$ and $M_0 = 3$, respectively, to smooth the shading. (b) Results of filling the set of T_1 and T_2 triangles after occlusion culling. (c) Reconstructed images from the optical experiments.

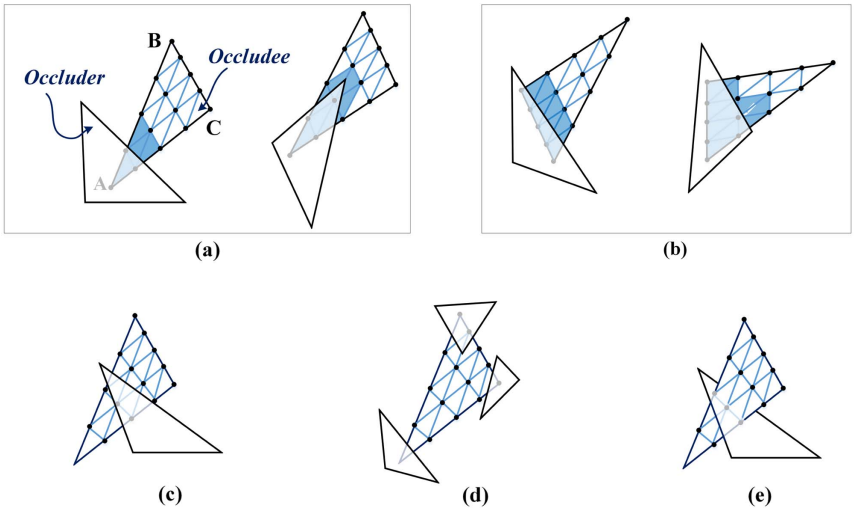

Fig. 16. Schematic diagram of occlusion culling. (a) Triangle set T_1 with one vertex obscured and divided into M_1^2 congruent sub-triangles. (b) Triangle set T_2 with two vertices obscured and divided into M_2^2 congruent sub-triangles. Blue sub-triangles are culled because they have at least one occluded vertex. (c)–(e) Three special cases that are occluded but cannot be properly addressed by the proposed method.

set of T_1 triangles and for the set of T_2 triangles, respectively. All terms in the above equation can be solved analytically using Eqs. (6) and (29).

In summary, the proposed occlusion culling method removes some of the occluded sub-triangles from the sets of partially visible triangles T_1 and T_2 by continuing collision detection on the vertices of all sub-triangles. In this way, we cannot perfectly eliminate the occluded region along the boundary line, but we can adjust M_1 and M_2 to control the trajectory of the occlusion culling. Figure 15(b) shows the results of occlusion culling with $M_1 = M_2 = 7$ for the teapot and $M_1 = M_2 = 6$ for the rings. The edges of the occlusion appear very tight compared to those in Fig. 15(a), which confirms the proposed method. In addition, it should be noted that the proposed occlusion culling method does not apply to some exceptional cases: in Fig. 16(c), although the rear triangle is partially occluded, the collision detection mechanism cannot be triggered because all three vertices are visible; in Fig. 16(d), since all three vertices of the rear triangle are invisible, it is treated as fully invisible, although only the three angles are obscured; and in Fig. 16(e), two inlaid triangles occlude each other on the inside, but both are treated as fully visible. The proposed method can serve all the 3D models mentioned in this paper because they are closed, and there are no isolated groups of triangles. All 3D models were generated using the 3D modeling software Blender.

Figure 17(a) shows the numerically reconstructed images of all the 3D models mentioned in this paper after occlusion culling. The number of triangles and the shading rendering parameters of the objects are presented in Tables 2 and 3. The triangles T_0, T_1, and T_2 are subdivided by different values of M_0, M_1, and M_2, respectively (see the legends). Each reconstructed image exhibited appropriate occlusion and a strong sense of realism in shading rendering, which confirms the generalizability of the proposed method. The supplementary Visualization 1 shows numerical reconstructions of three objects at different distances, where the objects automatically cull occlusions during rotation and apply constant illumination.

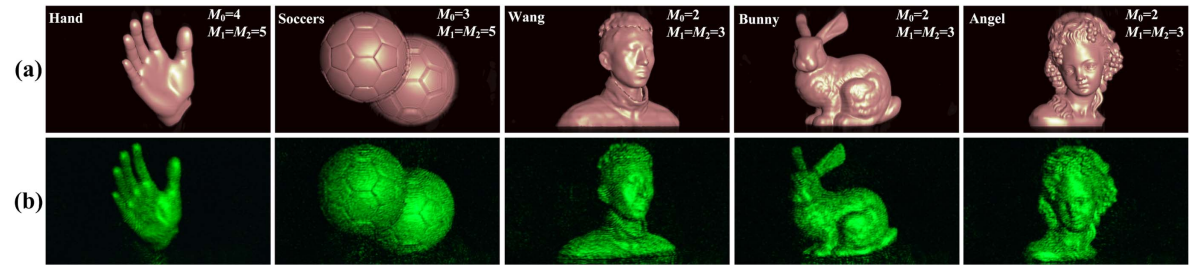

Fig. 17. Reconstructed images of all 3D objects referred in this paper. They are subjected to the proposed occlusion culling at the subdivision of M_0, M_1, and M_2. The number of triangles for each object is given in Table 2. Shading rendering follows the method proposed in Section 4. (a) Numerically reconstructed images, and (b) experimentally reconstructed images.

Table 4. Elapsed Time Using the Proposed Occlusion Culling Method

		Occlusion Culling (s)			
Models[a]	Backface Culling Only (s)	Octree	Detection	Total	Extra Costs
Teapot	1.42	0.005	0.075	1.82	28.2%
Rings	3.79	0.02	0.13	4.23	11.6%
Hand	4.88	0.04	0.14	5.16	5.7%
Soccer	6.51	0.07	0.16	6.98	7.2%
Wang	9.94	0.10	0.42	10.86	9.3%
Bunny	21.78	0.54	1.18	23.53	8.0%
Angel	26.64	0.68	1.53	28.39	6.6%

[a]The subdivision parameter is given in Fig. 17.

Next, we discuss the efficiency of the proposed occlusion culling method. Table 4 lists the elapsed time for each object under occlusion culling. The three main added processes in the proposed method are octree construction, collision detection, and intersection tests of T_1 and T_2 sets. However, the proposed occlusion culling method incurs only a small extra cost compared to backface culling. This is because collision detection identifies triangles with all three vertices occluded while their surface normals face forward. These triangles do not require computation in occlusion culling, whereas they do so in backface culling. The increase in time consumed by the teapot is more pronounced because of the larger size of the occluded triangles, which requires more subdivision of the sub-triangles to sew them together tightly.

6. DISCUSSION

A. Contributions

Based on the fully analytic algorithm introduced in Section 2, we investigate three crucial aspects of generating polygon-based holograms from Sections 3–5: accelerated computation, shading rendering, and occlusion culling. These three aspects constitute a comprehensive, high-speed rendering pipeline for arbitrarily complex 3D objects. Figure 18 shows a full view of the rendering pipeline. First, the compact frequency sampling region $(f_{x\mathrm{CE}}, f_{y\mathrm{CE}})$, where the spectral energy is mainly concentrated, is calculated according to Section 3.A. The resampled frequency coordinates enter the pipeline and are used for subsequent analytical calculations. Second, backface culling of 3D objects was performed. If the remaining triangles are injected directly into the rendering pipeline, occlusion will result in a discordant brightness. Therefore, we performed collision detection using an octree structure to identify the occluded vertices. The set T_0 of fully visible triangles goes directly to the computation of the analytic spectrum. Then, shading rendering is activated with the Blinn–Phong reflection model according to the triangle subdivision method proposed in Section 4.B. For the partially occluded T_1 and T_2 sets, collision detection will continue for the subdivided triangle vertices until a fully visible sub-triangle is determined, and the shading is rendered in the same manner. Finally, all calculated triangle spectra are summed to obtain the diffraction spectrum of the object, and the hologram is obtained by performing the NUFFT given in Eq. (12).

The holograms obtained by the proposed methods were encoded as a double phase [58] and imported into an SLM for optical reconstruction, as shown in Figs. 15(c) and 17(b), accordingly. The reconstructed images were displayed on a phase-only SLM with 4096×2400 pixels (Jasper SRK4K) and captured by a camera (Sony α7R – II) at 250 mm from the SLM. The reconstructed images in the experiments show well a similar rendering to the simulated results, but some noise is always introduced due to the encoding. Several studies have suggested methods for the evaluation of hologram reconstructions and distortion correction [59,60]. To further demonstrate the generality of the proposed rendering pipeline, we imported a Thai statue model with one million triangles into the pipeline. A high-resolution hologram of 7680×4320 1 μm pixels was generated in approximately 10 minutes and reconstructed numerically, as shown in Fig. 19. All triangles are subdivided by $M_0 = M_1 = M_2 = 3$, which means that the object contains a total of nine million sub-triangles. The reconstructed distance

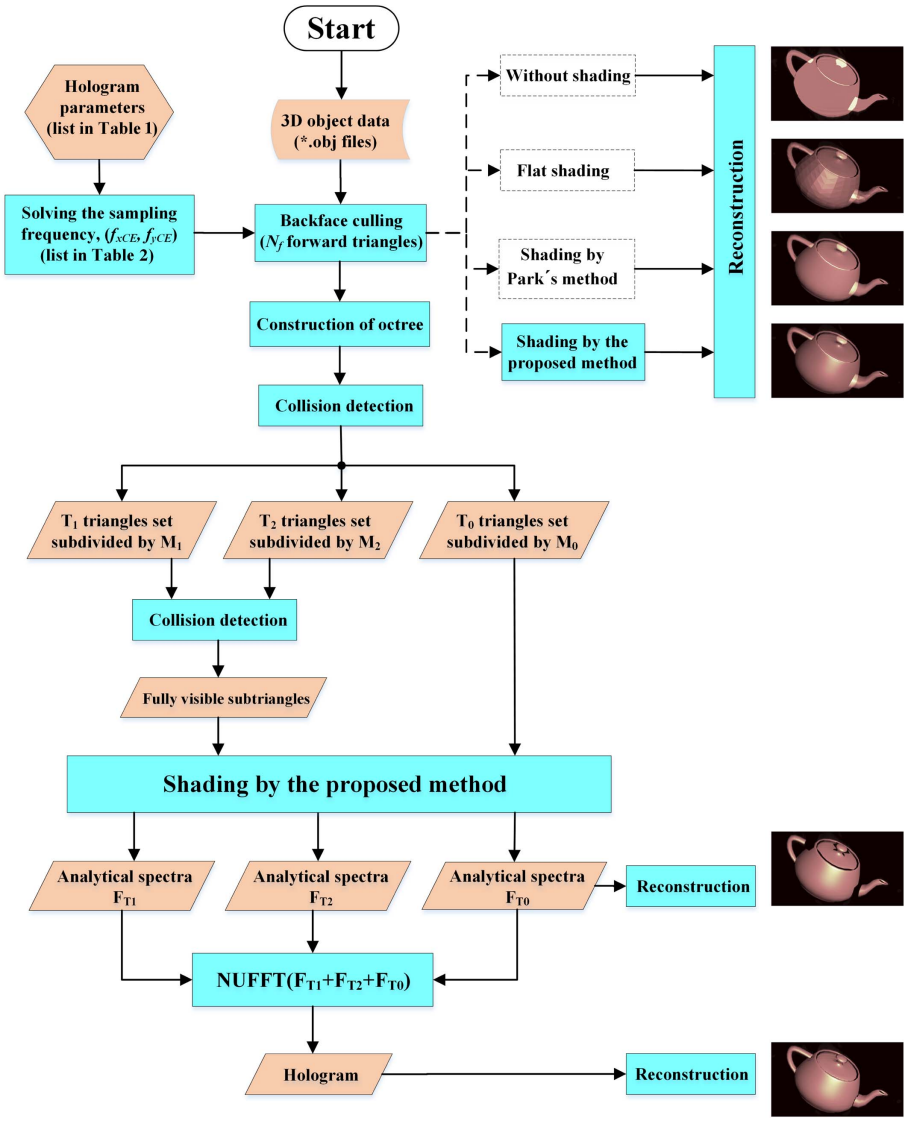

Fig. 18. Flowchart of the proposed high-speed rendering pipeline for the polygon-based holograms.

is at the front of the statue, whereas the rear appears blurry because it is out of focus. The animation of changing the focus position during reconstruction can be seen in Visualization 2. The supplementary Visualization 3 includes the reconstruction of the rotating Thai statue with occlusion culling. It is worth noting that online computing only takes up approximately 300 megabytes of memory for the GPU, but BE-ASM requires over 10 gigabytes of memory and takes approximately 5 h.

B. Limitations

We have not been able to achieve real-time computation, but the primary problem is the use of MATLAB, and we believe that with full-scratch development using C++ and CUDA can be achieved in real time. Our proposed method is a light-weight algorithm owing to the analytical solution of the shading and occlusion processes using an octree. This computational complexity is less than that of the conventional method using FFTs and interpolation [9] and therefore has advantage in real-time computations.

The proposed rendering pipeline uses a certain degree of approximation at each session. Appropriate approximations perform very well in practical applications but seem to have limitations for more general situations. For example, although the spectral energy focus method skips unwanted spectrum calculations by controlling the value of η given in Eq. (9), smaller values of η lead to a lower reconstruction quality. In particular, shorter propagation spreads more spectral energy over the high-frequency components, which requires greater values of η for fidelity and leads to a drop in acceleration. The M values used to subdivide the mother triangles achieve different degrees of smooth shading, whereas excessive subdivision increases workload. Therefore, the appropriate M can only be weighed empirically. Although the proposed occlusion culling method can eliminate the invisible localities of partially occlusions triangles, it is sometimes powerless, as shown in Figs. 16(c)–16(e).

The results of the in-focus and out-of-focus reconstruction of holograms confirmed monocular parallax, which is essential

Fig. 19. Numerical reconstruction of the ultra-high-resolution hologram of the Thai statue. The Thai statue contains 1,000,000 triangles and is subdivided by $M_0 = M_1 = M_2 = 3$. The full running time in the pipeline is 607 s.

for 3D perception. However, objectively, the proposed CGH pipeline does not show reconstructions with motion parallax, that is, it fails to offer scenes viewed from multiple viewpoints. There are two reasons for the lack of motion parallax. First, the light wave emitted from the triangular mesh, as given in Eq. (3), propagates in a single direction so that illumination shading and occlusion culling will be processed only along this direction, and the other is the absence of a random phase assigned to each triangle leading to low-frequency signals that cannot spread to a larger field of view (FOV). By contrast, although the silhouette method [38–40] and its improved version [41] compromise computational lightness, they offer motion parallax for multiple viewpoints. Yeom *et al.*'s shading method [33] enables observation over a large FOV owing to the inclusion of multi-directional carrier waves through convolution calculations. For the proposed CGH pipeline, one way to address the problem of motion parallax or large FOV is to recalculate holograms from different viewpoints, including the particular carrier wave, shading reflection, and occlusion culling.

Alternatively, the light-field stereogram technique has a clear advantage in implementing full parallax [61–63], because it executes occlusion and shading processes for each viewpoint with well-refined 3D computer graphics techniques. Recently, light-field-based hologram calculations using deep learning have been proposed to achieve unparalleled image quality [27]. However, light field methods have the problem of requiring re-rendering for each element image, while the proposed polygon-based method is a timely rendering of the raw 3D object within the pipeline.

7. CONCLUSION

This study introduced a dedicated pipeline for computing polygon-based holograms and is capable of rendering realistic holograms with smooth shading at a high speed. The proposed pipeline involves three major novel methods, in which the compact sampling strategy for accelerating the computation is an extended application of that proposed in the authors' previous publications [34]. The triangle subdivision method improves smooth rendering, particularly the specular effect that has not yet been considered in other studies [10,43], and the sub-triangle-based occlusion culling method is pioneering.

Compact spectra were adopted to narrow the sampling range using energy-focused approximation. Although we discarded high-frequency information, the computational results showed no significant degradation in the quality of the reconstructed images. Importantly, this significantly reduced the computational effort, achieving a speed-up of nearly 30 times.

The proposed shading rendering method divides a triangle into multiple sub-triangles and calculates the amplitude of each sub-triangle according to the reflection model. In this method, we only need to calculate the spectrum of one of the sub-triangles regardless of the number of times it is subdivided, which is an important improvement over the approach presented by Kim [10], a pioneer of the subdivision method. Compared to Park *et al.*'s proposed continuous rendering [32], the proposed rendering model is not theoretically consecutive. However, by controlling the value of the subdivision level M, a very smooth effect can be achieved. More importantly, we can simulate more realistic highlight reflections with only a small increase in the subdivision load compared to Park *et al.*'s method.

The concept of triangular subdivision is also used for occlusion culling. Although the proposed occlusion culling method approximates culling occlusions along the edges of the occluders, the results show that proper subdivisions can be tightly stitched together to reveal the true occlusion relationship. Owing to the fast query function of the octree structure, the proposed occlusion-free culling method requires little additional overhead compared to occlusion-free culling.

Many complex 3D objects could be calculated and reconstructed, thereby confirming the effectiveness of the proposed rendering pipeline for polygon-based holograms. We also analyzed the limitations existing in each step of the pipeline and discussed potential solutions, which will be validated in the near future.

Funding. Japan Society for the Promotion of Science (19F01097, 22H03607).

Acknowledgment. Thanks to Stanford Computer Graphics Laboratory for sharing the "bunny" and "Thai statue" 3D models, and Stony Brook University for sharing the "angel" model.

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

REFERENCES

1. L. Onural, F. Yaraş, and H. Kang, "Digital holographic three-dimensional video displays," Proc. IEEE **99**, 576–589 (2011).
2. S.-C. Kim and E.-S. Kim, "Effective generation of digital holograms of three-dimensional objects using a novel look-up table method," Appl. Opt. **47**, D55–D62 (2008).
3. T. Shimobaba, N. Masuda, and T. Ito, "Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane," Opt. Lett. **34**, 3133–3135 (2009).
4. R. Kukołowicz, M. Chlipala, J. Martinez-Carranza, M. S. Idicula, and T. Kozacki, "Fast 3D content update for wide-angle holographic near-eye display," Appl. Sci. **12**, 293 (2022).
5. Y. Zhao, L. Cao, H. Zhang, D. Kong, and G. Jin, "Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method," Opt. Express **23**, 25440–25449 (2015).
6. J.-S. Chen and D. Chu, "Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications," Opt. Express **23**, 18143–18155 (2015).
7. H. Zhang, Y. Zhao, L. Cao, and G. Jin, "Layered holographic stereogram based on inverse Fresnel diffraction," Appl. Opt. **55**, A154–A159 (2016).
8. 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).
9. K. Matsushima, "Computer-generated holograms for three-dimensional surface objects with shade and texture," Appl. Opt. **44**, 4607–4614 (2005).
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. 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).
12. Y.-Z. Liu, J.-W. Dong, Y.-Y. Pu, B.-C. Chen, H.-X. He, and H.-Z. Wang, "High-speed full analytical holographic computations for true-life scenes," Opt. Express **18**, 3345–3351 (2010).
13. Y. Pan, Y. Wang, J. Liu, X. Li, and J. Jia, "Fast polygon-based method for calculating computer-generated holograms in three-dimensional display," Appl. Opt. **52**, A290–A299 (2013).
14. Y.-P. Zhang, F. Wang, T.-C. Poon, S. Fan, and W. Xu, "Fast generation of full analytical polygon-based computer-generated holograms," Opt. Express **26**, 19206–19224 (2018).
15. T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram of line-drawn objects without FFT," Opt. Express **28**, 15907–15924 (2020).
16. D. Blinder, T. Nishitsuji, T. Kakue, T. Shimobaba, T. Ito, and P. Schelkens, "Analytic computation of line-drawn objects in computer generated holography," Opt. Express **28**, 31226–31240 (2020).
17. D. Blinder, T. Nishitsuji, and P. Schelkens, "Real-time computation of 3D wireframes in computer-generated holography," IEEE Trans. Image Process. **30**, 9418–9428 (2021).
18. P. Tsang, T.-C. Poon, and Y. Wu, "Review of fast methods for point-based computer-generated holography," Photonics Res. **6**, 837–846 (2018).
19. P. Tsang and T.-C. Poon, "Review on theory and applications of wavefront recording plane framework in generation and processing of digital holograms," Chin. Opt. Lett. **11**, 010902 (2013).
20. 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).
21. T. Nishitsuji, Y. Yamamoto, T. Sugie, T. Akamatsu, R. Hirayama, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, "Special-purpose computer horn-8 for phase-type electro-holography," Opt. Express **26**, 26722–26733 (2018).
22. C. Edwards, "Holograms on the horizon?" Commun. ACM **64**, 14–16 (2021).
23. R. Horisaki, R. Takagi, and J. Tanida, "Deep-learning-generated holography," Appl. Opt. **57**, 3859–3863 (2018).
24. Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," ACM Trans. Graph. **39**, 185 (2020).
25. L. Shi, B. Li, C. Kim, P. Kellnhofer, and W. Matusik, "Towards real-time photorealistic 3D holography with deep neural networks," Nature **591**, 234–239 (2021).
26. J. Wu, K. Liu, X. Sui, and L. Cao, "High-speed computer-generated holography using an autoencoder-based deep neural network," Opt. Lett. **46**, 2908–2911 (2021).
27. S. Choi, M. Gopakumar, Y. Peng, J. Kim, M. O'Toole, and G. Wetzstein, "Time-multiplexed neural holography: a flexible framework for holographic near-eye displays with fast heavily-quantized spatial light modulators," in *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 1–9.
28. H. Nishi and K. Matsushima, "Rendering of specular curved objects in polygon-based computer holography," Appl. Opt. **56**, F37–F44 (2017).
29. H. Nishi, K. Matsushima, and S. Nakahara, "Rendering of specular surfaces in polygon-based computer-generated holograms," Appl. Opt. **50**, H245–H252 (2011).
30. K. Yamaguchi and Y. Sakamoto, "Computer generated hologram with characteristics of reflection: reflectance distributions and reflected images," Appl. Opt. **48**, H203–H211 (2009).
31. Y. Tsuchiyama and K. Matsushima, "Full-color large-scaled computer-generated holograms using RGB color filters," Opt. Express **25**, 2016–2030 (2017).
32. J.-H. Park, S.-B. Kim, H.-J. Yeom, H.-J. Kim, H. Zhang, B. Li, Y.-M. Ji, S.-H. Kim, and S.-B. Ko, "Continuous shading and its fast update in fully analytic triangular-mesh-based computer generated hologram," Opt. Express **23**, 33893–33901 (2015).
33. H.-J. Yeom and J.-H. Park, "Calculation of reflectance distribution using angular spectrum convolution in mesh-based computer generated hologram," Opt. Express **24**, 19801–19813 (2016).
34. F. Wang, T. Shimobaba, T. Kakue, and T. Ito, "Controllable energy angular spectrum method," arXivarXiv:2203.10966 (2022).
35. T. Akenine-Moller, E. Haines, and N. Hoffman, *Real-Time Rendering* (AK Peters/CRC Press, 2019).
36. 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).
37. H. Zhang, N. Collings, J. Chen, B. A. Crossland, D. Chu, and J. Xie, "Full parallax three-dimensional display with occlusion effect using computer generated hologram," Opt. Eng. **50**, 074003 (2011).
38. K. Matsushima, "Exact hidden-surface removal in digitally synthetic full-parallax holograms," Proc. SPIE **5742**, 25–32 (2005).
39. A. Kondoh and K. Matsushima, "Hidden surface removal in full-parallax CGHS by silhouette approximation," Syst. Comput. Jpn. **38**, 53–61 (2007).
40. K. Matsushima, M. Nakamura, and S. Nakahara, "Silhouette method for hidden surface removal in computer holography and its acceleration using the switch-back technique," Opt. Express **22**, 24450–24465 (2014).
41. M. Askari, S.-B. Kim, K.-S. Shin, S.-B. Ko, S.-H. Kim, D.-Y. Park, Y.-G. Ju, and J.-H. Park, "Occlusion handling using angular spectrum convolution in fully analytical mesh based computer generated hologram," Opt. Express **25**, 25867–25878 (2017).
42. F. Wang, T. Shimobaba, Y. Zhang, T. Kakue, and T. Ito, "Acceleration of polygon-based computer-generated holograms using look-up tables and reduction of the table size via principal component analysis," Opt. Express **29**, 35442–35455 (2021).
43. Y. Pan, Y. Wang, J. Liu, X. Li, and J. Jia, "Improved full analytical polygon-based method using Fourier analysis of the three-dimensional affine transformation," Appl. Opt. **53**, 1354–1362 (2014).
44. X. Zhang, *Matrix Analysis and Applications* (Cambridge University, 2017).

45. K. Matsushima and T. Shimobaba, "Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields," Opt. Express **17**, 19662–19673 (2009).
46. W. Zhang, H. Zhang, and G. Jin, "Band-extended angular spectrum method for accurate diffraction calculation in a wide propagation range," Opt. Lett. **45**, 1543–1546 (2020).
47. P. Stoica and R. L. Moses, *Spectral Analysis of Signals* (Pearson Prentice Hall, 2005).
48. Y.-H. Kim, C.-W. Byun, H. Oh, J. Lee, J.-E. Pi, G. H. Kim, M.-L. Lee, H. Ryu, H.-Y. Chu, and C.-S. Hwang, "Non-uniform sampling and wide range angular spectrum method," J. Opt. **16**, 125710 (2014).
49. J.-Y. Lee and L. Greengard, "The type 3 nonuniform FFT and its applications," J. Comput. Phys. **206**, 1–5 (2005).
50. C. Ware, *Information Visualization: Perception for Design*, 4th ed. (Morgan Kaufmann, 2019).
51. T.-S. Chen, C.-C. Chang, and M.-S. Hwang, "A virtual image cryptosystem based upon vector quantization," IEEE Trans. Image Process. **7**, 1485–1488 (1998).
52. B. T. Phong, "Illumination for computer generated pictures," Commun. ACM **18**, 311–317 (1975).
53. J. F. Blinn, "Models of light reflection for computer synthesized pictures," in *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques* (1977), pp. 192–198.
54. R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," ACM Trans. Graph. **1**, 7–24 (1982).
55. K. Yamaguchi, T. Ichikawa, and Y. Sakamoto, "Calculation method for computer-generated holograms considering various reflectance distributions based on microfacets with various surface roughnesses," Appl. Opt. **50**, H195–H202 (2011).
56. T. Möller and B. Trumbore, "Fast, minimum storage ray-triangle intersection," J. Graph. Tools **2**, 21–28 (1997).
57. Sven, "octree-partitioning 3d points into spatial subvolumes," 2022, https://www.mathworks.com/matlabcentral/fileexchange/40732-octree-partitioning-3d-points-into-spatial-subvolumes.
58. C.-K. Hsueh and A. A. Sawchuk, "Computer-generated double-phase holograms," Appl. Opt. **17**, 3874–3883 (1978).
59. Z. He, X. Sui, G. Jin, and L. Cao, "Distortion-correction method based on angular spectrum algorithm for holographic display," IEEE Trans. Ind. Inf. **15**, 6162–6169 (2019).
60. Z. He, X. Sui, G. Jin, D. Chu, and L. Cao, "Optimal quantization for amplitude and phase in computer-generated holography," Opt. Express **29**, 119–133 (2021).
61. T. Yatagai, "Stereoscopic approach to 3-D display using computer-generated holograms," Appl. Opt. **15**, 2722–2729 (1976).
62. K. Wakunami and M. Yamaguchi, "Calculation for computer generated hologram using ray-sampling plane," Opt. Express **19**, 9086–9101 (2011).
63. H. Kang, E. Stoykova, and H. Yoshikawa, "Fast phase-added stereogram algorithm for generation of photorealistic 3D content," Appl. Opt. **55**, A135–A143 (2016).