applied optics

Polygon-based computer-generated holography: a review of fundamentals and recent progress [Invited]

Yaping Zhang,[1,*] Houxin Fan,[1] Fan Wang,[2,5] Xianfeng Gu,[3] Xiaofan Qian,[1] and Ting-Chung Poon[4]

[1] Yunnan Provincial Key Laboratory of Modern Information Optics, Kunming University of Science and Technology, Kunming, Yunnan 650500, China
[2] Graduate School of Engineering, Chiba University, Yayoi-cho 1-33, Inage-ku, Chiba 263-8522, Japan
[3] Computer Science Department, SUNY at Stony Brook, Stony Brook, New York 11794, USA
[4] Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia 24061, USA
[5] e-mail: wangfan@chiba-u.jp
*Corresponding author: yaping.zhang@gmail.com

Received 6 October 2021; revised 22 December 2021; accepted 22 December 2021; posted 4 January 2022 (Doc. ID 444973); published 31 January 2022

In this review paper, we first provide comprehensive tutorials on two classical methods of polygon-based computer-generated holography: the traditional method (also called the fast-Fourier-transform-based method) and the analytical method. Indeed, other modern polygon-based methods build on the idea of the two methods. We will then present some selective methods with recent developments and progress and compare their computational reconstructions in terms of calculation speed and image quality, among other things. Finally, we discuss and propose a fast analytical method called the fast 3D affine transformation method, and based on the method, we present a numerical reconstruction of a computer-generated hologram (CGH) of a 3D surface consisting of 49,272 processed polygons of the face of a real person without the use of graphic processing units; to the best of our knowledge, this represents a state-of-the-art numerical result in polygon-based computed-generated holography. Finally, we also show optical reconstructions of such a CGH and another CGH of the Stanford bunny of 59,996 polygons with 31,724 processed polygons after back-face culling. We hope that this paper will bring out some of the essence of polygon-based computer-generated holography and provide some insights for future research. © 2022 Optica Publishing Group under the terms of the Optica Open Access Publishing Agreement

https://doi.org/10.1364/AO.444973

1. INTRODUCTION

Holography, invented by Gabor in 1948, is a technique based on interference and diffraction to record and reconstruct a real three-dimensional object [1]. Computer-generated holography deals with the methods for digitally generating holographic interference patterns. The resulting interference patterns are called computer-generated holograms (CGHs). The CGH can then be subsequently printed on a high-resolution film or inputted to a spatial light modulator (SLM) for optical reconstruction. In the early stage of CGH development, neither gray-tone plotter nor spatial light modulators (SLMs) were available. Indeed, the first CGH, invented by Brown and Lohmann in 1966, was necessary a binary, amplitude-only hologram [2]. Computer-generated holography has since become a technique for generating holograms with the development of computer technology and modern optics. CGHs have been widely used in various fields such as digital media, microscopy, optical information storages, 3D display and imaging [3], and most recently holographic displays [4] in virtual reality (VR) and augmented reality (AR) using diffractive optical elements [5]. There are also some recent reviews, surveys, and books on computer-generated holography. Park reports recent progress on computer-generated holography for three-dimensional scenes and on fully analytic mesh-based computer-generated holography [6,7]. Sahin *et al.* provide a comprehensive survey for synthesis of CGHs [8]. Yamaguchi reviews light-field display technologies based on both light rays and wavefronts [9]. Corda *et al.* presents an in-depth review of the algorithms for advanced processing and rendering of CGHs [10]. Yoshikawa and Yamaguchi review holographic printers for CGHs [11]. Tsang *et al.* [12] and Nishitsuji *et al.* [13] focus on the review of fast methods for point-based CGHs. Shimobaba *et al.* [14] also have a review on fast algorithms and hardware implementations on CGHs. Wang *et al.* [15] review the hardware implementations of CGHs. While Matsushima presents a comprehensive treatment on one of the classical approaches, namely, the traditional method (to be discussed in the next section), in polygon-based numerical CGHs [16], Shimobaba and Ito focus on point-based CGH in their book [17]. The book by Tsang provides a comprehensive coverage of modern methods for generating phase-only CGHs for 3D display [18]. From those fine reviews, we know that two main practical obstacles still exist for a wider technology adoption of CGH. One is the limited performance of currently available spatial light modulators (SLMs), and the other is the huge amount of computation time for large-sized objects. On the computational side, fast and efficient methods are sought.

The generation of CGHs in 3D imaging are categorized into wavefront-based and ray-based methods, depending mainly on the object model and light propagation model [8,9]. The wavefront-based method calculates the hologram through the use of wave optics to obtain a wave field scattered from the object, such as the point-based method and the polygon-based method. The ray-based method creates holograms mainly by utilizing the intensity distribution of the 2D images propagating from different viewing points and capturing the object information incoherently based on geometric optics, such as holographic stereogram (HS) and multiple viewpoint projection (MVP) holography [9]. In this review, we concentrate on the wavefront-based method.

To calculate the CGH of a 3D object, the algorithms usually decompose the 3D object into a set of simple primitives such as points [12,13], line segments [19], polygons [20], or layers [21–24]. No matter which type of primitives is used, the complex amplitude of these primitives is superimposed in the hologram plane. The aim is to reduce the calculation of the complex amplitude from all the primitives.

In modern CGH, there are two prevalent approaches based on wavefront reconstruction: the point-based approach and the polygon-based approach. In the point-based approach, the 3D object is represented by a collection of self-illuminating points, each point is emitting a spherical wave toward the hologram plane, and the total complex amplitude on the hologram plane is the object wave calculated by summing up the spherical waves emitted by all the point sources. For a most recent review of fast methods for point-based CGHs, interested readers are referred to the paper by Tsang *et al.* [12] and Nishitsuji *et al.* [13].

In the polygon approach, the 3D object is represented by a collection of 2D polygons (a triangle is often used for the method). The object wave on the hologram plane is the summation of all the diffracted fields from each polygon. The central idea of using polygons to represent a 3D object is that the number of polygons representing an object is much smaller than that of points in the point-based approach, which thus drastically speeds up the calculation time required for the generation of a hologram. Hence, in a way, the polygon-based method is kind of an information reduction approach to computer-generated holography. The polygon-based approach is also motivated by the availability of visualization tools or rending software such as 3ds Max. In 3ds Max, we can edit a mesh by adding or deleting the various polygons. Therefore, the feature makes it possible to apply computer graphics to the generation of CGHs. To improve the reality of 3D object reconstructed by polygon-based CGHs, some basic techniques used in computer graphics, such as shading, rendering, occlusion and hidden-face removal, texture mapping, and illumination can be applied to polygon-based CGH. The amplitude and phase function of each polygon can be encoded with the "surface function" introduced by Matsushima and Kondoh [25], and Matsushima and Nakahara [26], as Matsushima first investigated 3D surface objects with shade and texture [27]. Other examples include the works of Tang *et al.* [28] and Park *et al.* [29]. Inspired by the works of Matsushima *et al.*, Ahrenberg *et al.* [30] first presented algorithms for fast rendering of polygons directly in the angular spectrum domain based on an affine transformation. Subsequently, many improved methods on wave-field rendering [31–33], texture [10,34,35], shading [36], and hidden-surface removal have been described [37–39]. The removal of dark line artifacts on the mesh boundary [40] and occlusion handling between polygon surfaces provide correct information about the perception of depth in order to encode 3D scenes [41–43]. Also the calculation of reflectance distributions has been addressed in polygon-based methods [44–46].

Our aim is to decrease the computation time of CGHs and to improve the quality of the reconstructed image. However, the calculation of CGH of a 3D object with rendering, hidden-face removal, and so on is a heavy computer processing task. To reduce calculation time, fast algorithms and hardware implementations are both needed. As the number of polygons increases, there is a significant computational load associated with hologram synthesis, and most calculations target toward real-time 3D display [14,15]. To speed up the calculation, hardware-based acceleration is highly effective. Accelerated hardware platforms, including graphics processing units (GPUs), field-programmable gate arrays (FPGAs), digital signal processors (DSPs), coprocessors, and so on can bring high efficiency in CGH generation [47–51].

When CGH faces its main practical display application with SLMs, the SLM needs to present high enough resolution, as the resolution of a SLM will affect the quality of hologram display [52–54]. For a SLM, it has constant pixel pitch. Given a fixed wavelength, the maximum diffraction angle of the SLM is dependent on the pixel pitch, and the pixel number decides the image size and the viewing angle of the reconstructed 3D holographic image. The space-bandwidth product (SBP) is defined by the product of the spatial and spectral footprints of the holographic signal. When a hologram is displayed on an SLM, the SBP corresponds to the number of pixels, and so the total pixel count of the SLM puts an upper limit on the SBP of the system [9,55].

This review paper is primarily concerned with the development in CGH with polygon-based algorithms. There are two classical methods in polygon-based CGH: the traditional method and the analytical method. In Section 2, we will first provide a comprehensive tutorial of the basic theory of the two methods. In Section 3, we will describe some recent progress in the analytical method in which two existing methods and one method proposed by this paper called the fast 3D affine transformation (F3DAT) method are discussed. In Section 4, we will generate CGHs and perform reconstruction using these three analytical methods. Computation time and image quality evaluation are compared among these methods along with some

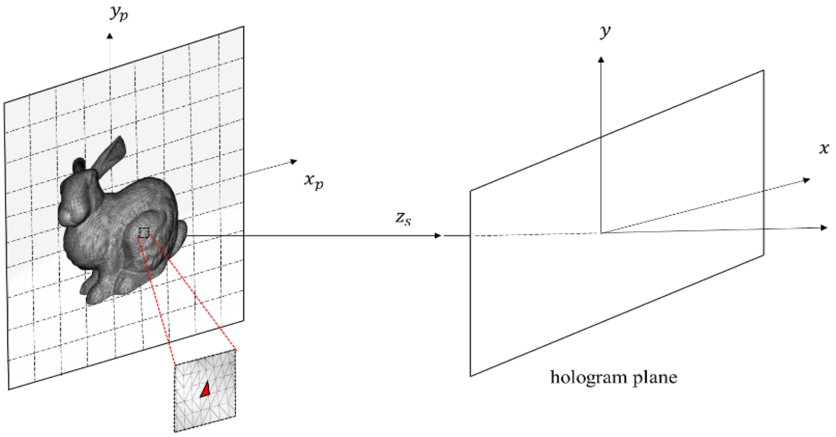

Fig. 1. 3D object as a 3D mesh.

optical reconstruction using a SLM. Real face data obtained by a depth camera will also be used to further test the F3DAT method. In the last section, we make some concluding remarks.

2. TUTORIAL OF TWO CLASSICAL POLYGON-BASED METHODS

Figure 1 illustrates a 3D mesh of the Stanford bunny consisting of triangles. The hologram plane is in the $x-y$ plane. The optical complex field emitted by each polygon (triangle in this case) on the hologram plane is referred to as the *polygon field*. Therefore, we can write the total complex object field on the hologram by summing all the polygon fields from each polygon:

$$u(x, y) = \sum_{i=1}^{N} u_i(x, y), \quad (1)$$

where N is the number of polygons and $u_i(x, y)$ is the polygon field on the hologram plane from the ith polygon. We now discuss two popular polygon-based methods to realize Eq. (1): the traditional method [or the fast Fourier transform (FFT)-based method] and the analytical method. We will start with the traditional method.

A. Traditional Method

Calculations of diffraction between two parallel planes are well known. These can be performed using the angular spectrum method or the Fresnel diffraction formula if small angle approximations are assumed [56]. However, in a general 3D mesh, most polygons do not lie parallel to the hologram plane. Therefore, there was a great need to propagate diffracted fields between tilted planes. Ganci, Patorski, and Rabal *et al.* investigated the diffraction pattern of a tilted plane under the Fraunhofer approximations [57–59]. Leseberg and Frere were the first to investigate the diffraction pattern of a tilted plane using the Fresnel approximations [60]. Subsequently, they extended their investigation for large objects [61]. Tommasi and Bianco first analyzed the relation for the angular spectra of rotated planes [62]. The result is a central idea of most of the popular polygon-based methods. In their subsequent investigation, they applied their approach to calculate the CGHs of off-axis objects and addressed the problems concerning the hologram size and the sampled spectrum [63]. The same problem was further examined and deliberated mathematically in a precise and clear fashion by Matsushima *et al.* [64,65]. Other works on the investigation of diffraction between oriented planes include the ones by Delen and Hooker based on full diffraction theory [66] and by Onural using impulse functions over a 3D surface [67].

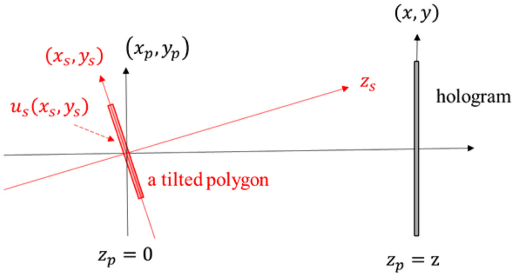

Fig. 2. Coordinate systems: source coordinate system or tilted local coordinate system (x_s, y_s, z_s), parallel local coordinate system (x_p, y_p, z_p), and hologram plane (x, y).

To find the polygon field of the ith polygon $u_i(x, y)$, we need to relate the *surface function of an arbitrary polygon* on a tilted plane $u_s(x_s, y_s)$ to that on the hologram plane [25–27]. Figure 2 shows a single tilted polygon, where coordinate system (x_s, y_s, z_s) is referred to as a *tilted local coordinate system* or a *source coordinate system.* A *parallel local coordinate system* (x_p, y_p, z_p) is also defined, which shares the origin of the tilted local coordinates. In describing the coordinate systems, we are using the same terminology originally introduced by Matsushima [27]. The parallel local coordinate system is parallel to the hologram plane (x, y). So, given $u_s(x_s, y_s)$, we find $u_i(x, y)$. From $u_s(x_s, y_s)$ we can first find the complex field on the parallel local coordinate system. Subsequently, the field on the parallel local coordinate system diffracts to the hologram plane to form $u_i(x, y)$.

The complex polygon field with angular plane wave spectrum $U_s(k_{sx}, k_{sy}) = U_s(k_{sx}, k_{sy}; z_s = 0)$ propagating along z_s is

$$u_s(x_s, y_s, z_s) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} U_s(k_{sx}, k_{sy}; 0) \times \mathrm{e}^{-j\left(k_{sx}x_s + k_{sy}y_s + k_{sz}z_s\right)} \mathrm{d}k_{sx}\mathrm{d}k_{sy}, \tag{2a}$$

where

$$U_s\left(k_{sx}, k_{sy}; z_s = 0\right) = \mathcal{F}\left\{u_s\left(x_s, y_s; z_s = 0\right)\right\} = \iint_{-\infty}^{\infty} u_s\left(x_s, y_s\right)\mathrm{e}^{j\left(k_{sx}x_s + k_{sy}y_s\right)}\mathrm{d}x_s\,\mathrm{d}y_s$$

with $\mathcal{F}\{\cdot\}$ representing the Fourier trasnform of the quanity being bracketed. In vector notations and using the dot product, Eq. (2a) becomes

$$u_s\left(\boldsymbol{r_s}\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} U_s\left(k_{sx}, k_{sy}; 0\right)\mathrm{e}^{-j\boldsymbol{k_s}\cdot\boldsymbol{r_s}}\mathrm{d}k_{sx}\mathrm{d}k_{sy}, \tag{2b}$$

where $\boldsymbol{r_s} = (x_s, y_s, z_s)$ and $\boldsymbol{k_s} = (k_{sx}, k_{sy}, k_{sz})$ are a position vector and a propagation vector of u_s in the source coordinates, respectively, defined by row matrices. Now the two local coordinates $\boldsymbol{r_s} = (x_s, y_s, z_s)$ and $\boldsymbol{r_p} = (x_p, y_p, z_p)$ can be mutually transformed by coordinate rotation using *transformation matrix* T:

$$\boldsymbol{r}_{\boldsymbol{p}}^{\mathrm{t}} = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \begin{pmatrix} a_1 & a_4 & a_7 \\ a_2 & a_5 & a_8 \\ a_3 & a_6 & a_9 \end{pmatrix}\begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} = T\boldsymbol{r}_{\boldsymbol{s}}^{\mathrm{t}}, \tag{3a}$$

or

$$\boldsymbol{r}_{\boldsymbol{s}}^{\mathrm{t}} = T^{-1}\boldsymbol{r}_{\boldsymbol{p}}^{\mathrm{t}}, \tag{3b}$$

where T is a rotation matrix or the product of rotation matrices. The idea is that upon proper rotations, the polygon under consideration will be on the $x_p y_p$ plane that is parallel to the hologram plane. From Eq. (3b), after taking the transpose, we have

$$\boldsymbol{r_s} = \boldsymbol{r_p}(T^{-1})^{\mathrm{t}} = \boldsymbol{r_p}T \tag{4}$$

as $T^{-1} = T^{\mathrm{t}}$ for any rotation matrix. Hence, with Eq. (4), the complex field in Eq. (2a) in the parallel local coordinates becomes

$$u_p\left(\boldsymbol{r_p}\right) = u_s\left(\boldsymbol{r_s}\right)\big|_{\boldsymbol{r_s}=\boldsymbol{r_p}T} = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} U_s\left(k_{sx}, k_{sy}; 0\right)\mathrm{e}^{-j\boldsymbol{k_s}\cdot\boldsymbol{r_p}T}\mathrm{d}k_{sx}\mathrm{d}k_{sy}. \tag{5}$$

Similarly, a propagation vector in the source and parallel local coordinates can also be transformed like position vectors as follows:

$$\boldsymbol{k_s} = \boldsymbol{k_p}T, \tag{6}$$

where $\boldsymbol{k_p} = (k_{px}, k_{py}, k_{pz})$. With this equation, the exponential term original in Eq. (2b) and now in Eq. (5) becomes

$$\boldsymbol{k_s}\cdot\boldsymbol{r_s} = \boldsymbol{k_s}\cdot\boldsymbol{r_p}T = \boldsymbol{k_p}T\cdot\boldsymbol{r_p}T.$$

If vector $\boldsymbol{a}$ and $\boldsymbol{b}$ are defined with row matrices, we can write the dot product as a matrix product as follows:

$$\boldsymbol{a}\cdot\boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^{\mathrm{t}}.$$

Hence,

$$\boldsymbol{k_s}\cdot\boldsymbol{r_s} = \boldsymbol{k_p}T\cdot\boldsymbol{r_p}T = \boldsymbol{k_p}T(\boldsymbol{r_p}T)^{\mathrm{t}} = \boldsymbol{k_p}T(T^{\mathrm{t}}\boldsymbol{r}_{\boldsymbol{p}}^{\mathrm{t}}) = \boldsymbol{k_p}\boldsymbol{r}_{\boldsymbol{p}}^{\mathrm{t}} = \boldsymbol{k_p}\cdot\boldsymbol{r_p}, \tag{7}$$

and with this, we rewrite Eq. (5) to become

$$u_p\left(\boldsymbol{r_p}\right) = u_s\left(\boldsymbol{r_s}\right)\big|_{\boldsymbol{r_s}=\boldsymbol{r_p}T} = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} U_s\left(k_{sx}, k_{sy}; 0\right)\mathrm{e}^{-j\boldsymbol{k_p}\cdot\boldsymbol{r_p}}\mathrm{d}k_{sx}\mathrm{d}k_{sy} = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} U_s\left(k_{sx}, k_{sy}; 0\right) \times \mathrm{e}^{-j\left(k_{px}x_p + k_{py}y_p + k_{pz}z_p\right)}\mathrm{d}k_{sx}\mathrm{d}k_{sy}, \tag{8a}$$

where

$$k_{pz}(k_{px}, k_{py}) = \sqrt{k_0{}^2 - k_{px}^2 - k_{py}^2}, \tag{8b}$$

with k_0 being the wavenumber of the light. The exponential function in Eq. (8a) represents plane wave propagating along the z_p direction (or the z direction) on the parallel local coordinates as shown in Fig. 2. We now need to fully convert Eq. (8) in terms of (k_{px}, k_{py}, k_{pz}). In other words, the terms

$$U_s\left(k_{sx}, k_{sy}; 0\right)dk_{sx}dk_{sy} \tag{9}$$

still need to be converted. From Eq. (6), we have

$$\boldsymbol{k}_{\boldsymbol{s}}^{\mathrm{t}} = T^{-1}\boldsymbol{k}_{\boldsymbol{p}}^{\mathrm{t}},$$

which is

$$\begin{pmatrix} k_{sx} \\ k_{sy} \\ k_{sz} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix}\begin{pmatrix} k_{px} \\ k_{py} \\ k_{pz} \end{pmatrix}.$$

Therefore, we have

$$k_{sx} = k_{sx}(k_{px}, k_{py}) = a_1 k_{px} + a_2 k_{py} + a_3 k_{pz}(k_{px}, k_{py}) \tag{10a}$$

and

$$k_{sy} = k_{sy}(k_{px}, k_{py}) = a_4 k_{px} + a_5 k_{py} + a_6 k_{pz}(k_{px}, k_{py}). \tag{10b}$$

Therefore, $U_s(k_{sx}, k_{sy}; 0)$ from Eq. (8a) can be expressed in terms of (k_{px}, k_{py}, k_{pz}) explicitly as

$$U_s\left(k_{sx}, k_{sy}; 0\right) = U_s\left(k_{sx}(k_{px}, k_{py}), k_{sy}(k_{px}, k_{py}); 0\right) = U_s\left(a_1 k_{px} + a_2 k_{py} + a_3 k_{pz}, a_4 k_{px} + a_5 k_{py} + a_6 k_{pz}; 0\right). \tag{11}$$

Now, using Eq. (10), the differential element of Eq. (8a) is achieved by

$$dk_{sx}dk_{sy} = \left|J\left(k_{px}, k_{py}\right)\right| dk_{px}dk_{py}, \tag{12}$$

where $J(k_{px}, k_{py})$ is the *Jacobian* of the coordinate transformation of k_{sx} and k_{sy} with respect to k_{px} and k_{py}, given by

$$J\left(k_{px},k_{py}\right)=\begin{vmatrix}\frac{\partial k_{sx}}{\partial k_{px}} & \frac{\partial k_{sx}}{\partial k_{py}}\\ \frac{\partial k_{sy}}{\partial k_{px}} & \frac{\partial k_{sy}}{\partial k_{py}}\end{vmatrix}=\frac{(a_2a_6-a_3a_5)k_{px}}{k_{pz}(k_{px},k_{py})}$$
$$+\frac{(a_3a_4-a_1a_6)k_{py}}{k_{pz}(k_{px},k_{py})}+(a_1a_5-a_2a_4)$$
$$\approx(a_1a_5-a_2a_4), \tag{13}$$

for paraxial approximations as k_{px} and k_{py} are much smaller than k_{pz}. With Eqs. (11) and (12), Eq. (9) becomes

$$U_s\left(k_{sx},k_{sy};0\right)dk_{sx}dk_{sy}$$
$$=U_s\left(k_{sx}(k_{px},k_{py}),k_{sy}(k_{px},k_{py});0\right)\left|J\left(k_{px},k_{py}\right)\right|dk_{px}dk_{py}$$
$$=U_s\left(a_1k_{px}+a_2k_{py}+a_3k_{pz},a_4k_{px}+a_5k_{py}+a_6k_{pz};0\right)$$
$$\times\left|J\left(k_{px},k_{py}\right)\right|dk_{px}dk_{py}. \tag{14}$$

Substituting Eq. (14) into Eq. (8a), we have

$$u_p(\boldsymbol{r_p})=\frac{1}{4\pi^2}\iint_{-\infty}^{\infty}U_s\left(a_1k_{px}+a_2k_{py}+a_3k_{pz},a_4k_{px}\right.$$
$$\left.+a_5k_{py}+a_6k_{pz};0\right)e^{-j\left(k_{px}x_p+k_{py}y_p+z_p\sqrt{{k_0}^2-k_{px}^2-k_{py}^2}\right)}$$
$$\times\left|J\left(k_{px},k_{py}\right)\right|\mathrm{d}k_{px}\mathrm{d}k_{py}, \tag{15}$$

where, again, $U_s(k_{sx},k_{sy};z_s=0)=\mathcal{F}\{u_s(x_s,y_s;z_s=0)\}$ and $u_s(x_s,y_s;z_s=0)=u_s(x_s,y_s)$. For a given surface function of a polygon $u_s(x_s,y_s)$, Eq. (15) gives us the complex field propagating on the (x_p,y_p,z_p) coordinate system.

Therefore, we can write the polygon field $u_i(x,y)$(due to the ith polygon) on the hologram plane at $z_p=z_i$ with (x_p,y_p) replaced by (x,y) in Eq. (15):

$$u_i(x,y)=\mathcal{F}^{-1}\left\{U_{s,i}\left(a_{1,i}k_{px}+a_{2,i}k_{py}+a_{3,i}k_{pz},a_{4,i}k_{px}\right.\right.$$
$$\left.\left.+a_{5,i}k_{py}+a_{6,i}k_{pz};0\right)\left|J_i(k_{px},k_{py})\right|\mathrm{e}^{-jz_i\sqrt{{k_0}^2-k_{px}^2-k_{py}^2}}\right\}, \tag{16}$$

where $U_{s,i}(k_{sx,i},k_{sy,i};0)=\mathcal{F}\{u_{s,i}(x_s,y_s;z_s=0)\}$ is the angular spectrum on the source plane and $u_{s,i}(x_s,y_s;z_s=0)$ denotes the ith polygon that is z_i away from the hologram. We observe that numerical calculations of each of the polygon field $u_i(x,y)$ on the hologram require two FFTs, and finally the total polygon field on the hologram is computed by Eq. (1). The use of Eq. (16) to find the total polygon fields is what we call the traditional method—the same name that it is called by Pan *et al.* [68], where the first FFT is applied to the tilted local coordinate system or a source coordinate system, and the second FFT is applied in the frequency domain. Clearly, the two processes involve expensive numerical calculations, and hence we can also call the traditional method the FFT-based method [34]. The main drawback is the need to perform two FFTs for each polygon as mentioned. In addition, most seriously, the transformation of the spectrum from the local coordinate system to the parallel local coordinate system through rotation is nonlinear [see Eq. (10)], and that requires interpolation to allow FFT sampling at even intervals to alleviate sampling distortion (see Ref. [16], Sections 9.5 and 9.6 for detailed discussion). However, one of the biggest advantages of the traditional method is that shading and texture mapping of each polygon are naturally incorporated into the surface function, which leads to the unparalleled rendering of high-resolution and realistic 3D reconstructed images.

B. Analytical Method

We see from Eq. (16), that for an arbitrary polygon with surface function $u_{s,i}(x_s,y_s;0)$, its spectrum is $\mathcal{F}\{u_{s,i}(x_s,y_s;0)\}=U_{s,i}(k_{sx},k_{sy};0)$, which is calculated numerically by FFT. If we have an analytical function describing the spectrum of an arbitrary polygon, we will save time in calculations, and then Eq. (16) can be calculated using a single FFT numerically. In addition, each spectrum of the polygon can be precalculated to speed things up. Ahrenberg *et al.* [30] pioneered a method to analytically compute the spectrum of an arbitrary polygon through the use of *affine transformation* [69]. An affine transformation is a *geometrical transformation* that maps input coordinates (x,y) into output coordinates (x_s,y_s) according to

$$\begin{pmatrix}x_s\\y_s\end{pmatrix}=\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}a_{13}\\a_{23}\end{pmatrix}. \tag{17}$$

Typical affine transformations consist of various operations: scaling, reflection, shear, and translation. Figure 3(a) shows a general triangle $f_\Gamma(x_s,y_s)$ on the source coordinate system with the vertex coordinates as (x_1,y_1), (x_2,y_2), and (x_3,y_3). In Fig. 3(b), we also show a unit right triangle on the (x,y) coordinates. The affine transform relating the two coordinates can be written as

$$\begin{pmatrix}x_s\\y_s\end{pmatrix}=\begin{pmatrix}x_2-x_1&x_3-x_2\\y_2-y_1&y_3-y_2\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}x_1\\y_1\end{pmatrix}. \tag{18}$$

We see that transforming the three vertices of the unit right triangle $f_\Delta(x,y)$ gives three new vertices, which forms the triangle $f_\Gamma(x_s,y_s)$. Note that the function $f_\Gamma(x_s,y_s)$ and $f_\Delta(x,y)$ are constant valued 1 inside the triangles and 0 everywhere else.

The central idea of the 2D affine transform analytical method is that we can relate the Fourier transform (or the spectrum) of an arbitrary triangle to that of a unit right triangle, as the spectrum of the unit right triangle is given analytically. The spectrum

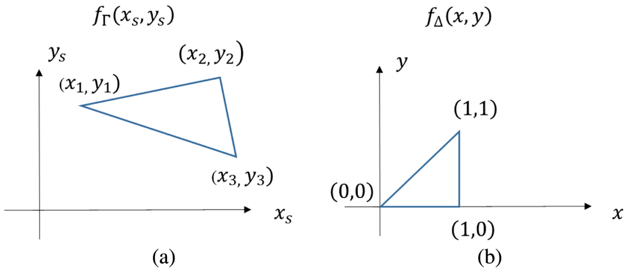

Fig. 3. (a) General triangle on the source coordinates (x_s,y_s); (b) unit right triangle.

of $f_\Gamma(x_s, y_s)$ is

$$\mathcal{F}\{f_\Gamma(x_s, y_s)\} = \mathrm{F}_\Gamma(k_{sx}, k_{sy}) = \iint_{-\infty}^{\infty} f_\Gamma(x_s, y_s) \mathrm{e}^{j(k_{sx}x_s + k_{sy}y_s)} \mathrm{d}x_s \, \mathrm{d}y_s, \quad \textbf{(19)}$$

and similarly, the spectrum of $f_\Delta(x, y)$, which has an analytical expression, is given by

$$\mathcal{F}\{f_\Delta(x, y)\} = F_\Delta(k_x, k_y) = \iint_{-\infty}^{\infty} f_\Delta(x, y) \mathrm{e}^{j(k_x x + k_y y)} \mathrm{d}x\mathrm{d}y$$
$$= \int_0^1 \int_0^x \mathrm{e}^{j(k_x x + k_y y)} \mathrm{d}x\mathrm{d}y$$
$$= \begin{cases} \frac{1}{2}, & k_x = k_y = 0 \\ \frac{1-e^{jk_y}}{{k_y}^2} + \frac{j}{k_y}, & k_x = 0,\ k_y \neq 0 \\ \frac{e^{jk_x}-1}{{k_x}^2} - \frac{je^{jk_x}}{k_x}, & k_x \neq 0,\ k_y = 0 \\ \frac{1-e^{jk_y}}{{k_y}^2} - \frac{j}{k_y}, & k_x = -k_y,\ k_y \neq 0 \\ \frac{e^{jk_x}-1}{k_x k_y} + \frac{1-e^{j(k_x+k_y)}}{k_y(k_x+k_y)}, & \text{elsewhere} \end{cases} \quad \textbf{(20)}$$

Now, for a general triangle relating to the unit right triangle, we use Eq. (17) to obtain affine operations $x_s = a_{11}x + a_{12}y + a_{13}$ and $y_s = a_{21}x + a_{22}y + a_{23}$. With a change of variables to (x, y), Eq. (19) becomes

$$\mathrm{F}_\Gamma(k_{sx}, k_{sy})$$
$$= \iint_{-\infty}^{\infty} f_\Gamma(x_s, y_s) \mathrm{e}^{j(k_{sx}x_s + k_{sy}y_s)} \mathrm{d}x_s \, \mathrm{d}y_s$$
$$= \int_0^1 \int_0^x f_\Delta(x, y) \mathrm{e}^{j[k_{sx}(a_{11}x + a_{12}y + a_{13}) + k_{sy}(a_{21}x + a_{22}y + a_{23})]}$$
$$\times |J(x, y)| \, \mathrm{d}x\mathrm{d}y$$
$$= \int_0^1 \int_0^x \mathrm{e}^{j[k_{sx}(a_{11}x + a_{12}y + a_{13}) + k_{sy}(a_{21}x + a_{22}y + a_{23})]}$$
$$\times |J(x, y)| \, \mathrm{d}x\mathrm{d}y,$$

where the Jacobian is

$$J(x, y) = \begin{vmatrix} \frac{\partial x_s}{\partial x} & \frac{\partial x_s}{\partial y} \\ \frac{\partial y_s}{\partial x} & \frac{\partial y_s}{\partial y} \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}. \quad \textbf{(21)}$$

Rearranging the equation $\mathrm{F}_\Gamma(k_{sx}, k_{sy})$ from above and using the definition of $F_\Delta(k_x, k_y)$ in Eq. (20), we can write

$$\mathrm{F}_\Gamma(k_{sx}, k_{sy}) = |a_{11}a_{22} - a_{12}a_{21}| \, \mathrm{e}^{j(k_{sx}a_{13} + k_{sy}a_{23})}$$
$$\times \int_0^1 \int_0^x \mathrm{e}^{j[k_{sx}(a_{11}x + a_{12}y) + k_{sy}(a_{21}x + a_{22}y)]} \mathrm{d}x\mathrm{d}y$$
$$= |a_{11}a_{22} - a_{12}a_{21}| \, \mathrm{e}^{j(k_{sx}a_{13} + k_{sy}a_{23})}$$
$$\times F_\Delta\left(a_{11}k_{sx} + a_{21}k_{sy}, a_{12}k_{sx} + a_{22}k_{sy}\right). \quad \textbf{(22)}$$

This is an analytical expression of the spectrum of $u_s(x_s, y_s) = f_\Gamma(x_s, y_s)$, i.e., $\mathcal{F}\{u_s(x_s, y_s)\} = U_s(k_{sx}, k_{sy}) = \mathrm{F}_\Gamma(k_{sx}, k_{sy})$, with F_Δ given by Eq. (20) analytically. Therefore, the polygon field due to the ith polygon z_i away from the hologram, according to Eq. (16), becomes

$$u_i(x, y) = \mathcal{F}^{-1}\Big\{U_{s,i}\big(a_{1,i}k_{px} + a_{2,i}k_{py} + a_{3,i}k_{pz}, a_{4,i}k_{px}$$
$$+ a_{5,i}k_{py} + a_{6,i}k_{pz}; 0\big) \left|J_i(k_{px}, k_{py})\right| \mathrm{e}^{-jz_i\sqrt{{k_0}^2 - k_{px}^2 - k_{py}^2}}\Big\}, \quad \textbf{(23)}$$

where $U_{s,i}(k_{sx,i}, k_{sy,i})$ is now given analytically as

$$U_{s,i}(k_{sx,i}, k_{sy,i})$$
$$= \mathcal{F}\{u_{s,i}(x_s, y_s; z_s = 0)\} = \mathrm{F}_{\Gamma_i}(k_{sx,i}, k_{sx,i})$$
$$= \left|a_{11,i}a_{22,i} - a_{12,i}a_{21,i}\right| \mathrm{e}^{j(k_{sx}a_{13,i} + k_{sy}a_{23,i})}$$
$$\times F_\Delta\big(a_{11,i}k_{sx,i} + a_{21,i}k_{sy,i}, a_{12,i}k_{sx,i} + a_{22,i}k_{sy,i}\big).$$

Again, the total polygon field on the hologram can now be computed using Eq. (1). As compared to the traditional method [see Eq. (16)], this method has the advantage that it only needs to perform a single inverse FFT and avoids the need of interpolation. By using Eq. (23) along with Eq. (1), we can obtain the total polygon field on the hologram, and the first FFT used in the traditional method is replaced by an analytical expression, i.e., $\mathrm{F}_{\Gamma_i}(k_{sx,i}, k_{sx,i})$. Therefore, the method described here is called the analytical method. However, one of the biggest drawbacks of the method is that texturing a polygon leads to a convolution in the angular spectrum of the polygon, which slows down the calculation speed.

To put things into perspective, we want to point out that almost at the same time Ahrenberg *et al.* published their paper, another group, Kim *et al.* [70], proposed another analytical angular spectrum representation of image light field emitted from the model of a triangle-mesh-modeled 3D surface objects. Using simple geometric transform, they are also able to get the analytical 2D Fourier transform of an arbitrary triangle on the source plane. Other analytical methods include the work of Zhang *et al.* using conformal geometry theory [71], and Sakata and Sakamoto deriving 3D affine transformation in the Fresnel region [72]. In addition, a semi-analytical method [34] has also been developed by Lee *et al.*

As a final note to this subsection, we want to point out that while we have discussed the two classical polygon-based methods, i.e., the traditional method and the analytical method, there are other modern polygon-based methods that build on the ideas of these two methods. For example, in the analytical method, we could take the affine transform of the source function before rotation to align the source coordinate system to be parallel with the hologram plane. Pan *et al.* [68,73,74] have developed a method using 3D affine transformation, which contains the rotational, translational, and scaling transformation. They have reduced the computation time as compared to that from Ahrenberg *et al.* Zhang *et al.* [20] have recently introduced a fast generation method that performs polygon rotation and 2D affine transformation to obtain full 3D affine transformation, which further reduces the computation time by avoiding the time-consuming aspect of solving a pseudo-inverse matrix encountered in Pan *et al.*'s work. In the next section, we will discuss and compare further on the methods by Pan *et al.* and Zhang *et al.* with the addition of a F3DAT method proposed in this paper. The F3DAT method avoids the time-consuming nature of solving the pseudo-inverse matrix and at the same time avoids using the multiple three-dimensional rotations used in Zhang *et al.* Hence, the computational efficiency is further improved. We want to mention that, with all these analytical methods investigated, we have not considered any texture mapping, as we simply try to seek ways to improve the calculation time for uniform surface function of each polygon. Therefore, these analytical methods including F3DAT are not a universal technique to calculate the object field of polygon-meshed objects. The traditional method, all the analytical methods mentioned, and other novel techniques are aiming for the calculation of CGHs with the ultimate goal of high-quality reconstruction with efficient calculations.

3. RECENT PROGRESS

A. 3D Affine Transformation/Pseudo-Inverse Matrix Method

Inspired by 2D affine transformation, 3D affine transformation has been proposed by Pan *et al.* [68,74]. For convenience, according to the terminology used by Pan *et al.*, the hologram, located at $z = 0$, is in the $x - y$ plane, and the xyz coordinates are called the global coordinates. Again we denote coordinate system (x_s, y_s, z_s) as a tilted local coordinate system or a source coordinate system. They also define the unit right triangle $f_\Delta(x_s, y_s)$ at $z_s = 0$ in the local coordinates, where we know its spectrum analytically as discussed earlier. The right triangle can be related to the arbitrary polygon to the global coordinate system via a 3D affine transformation as follows:

$$[x, y, z, 1]^t = T[x_s, y_s, z_s, 1]^t, \tag{24}$$

where $[x, y, y, 1]^t$ and $[x_s, y_s, z_s, 1]^t$ are the global coordinate vector and the local coordinate vector of a given point. T is an affine coordinate transformation matrix given by

$$T = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{25}$$

Hence, we can relate a pairwise correspondence of the vertices between the two triangles according to

$$\begin{cases} x = a_{11}x_s + a_{12}y_s + a_{13}z_s + t_1 \\ y = a_{21}x_s + a_{22}y_s + a_{23}z_s + t_2 \\ z = a_{31}x_s + a_{32}y_s + a_{33}z_s + t_3. \end{cases} \tag{26}$$

Since the unit right triangle is assumed to lie in the plane $z_s = 0$, the above equation becomes

$$\begin{aligned} x &= a_{11}x_s + a_{12}y_s + t_1 \\ y &= a_{21}x_s + a_{22}y_s + t_2 \\ z &= a_{31}x_s + a_{32}y_s + t_3. \end{aligned} \tag{27}$$

We need to compute the $a's$ and $t's$ of the above equations. However, since $[x_s, y_s, z_s, 1]^t = [x_s, y_s, 0, 1]^t$, we cannot invert Eq. (24) to find the $a's$ and $t's$. Pan *et al.* then bring on the concept of a pseudo-inverse matrix to perform inversion using the singular decomposition method [68,74]. By employing the 3D affine transformation in Eq. (24) along with the concept of a pseudo-inverse matrix and the analytical spectrum of a right unit triangle, they have been able to speed up the calculation process.

B. Full Analytical 3D Affine Transformation

The introduction of the pseudo-inverse matrix has produced calculation errors and slowed down the calculation speed. Zhang *et al.* have introduced a fast generation of the full analytical method to avoid the use of the pseudo-inverse matrix [20]. The method includes three core steps: rotation transformation for the tilted polygon until it is parallel to the hologram plane, 2D affine transform of the rotated polygon, and finally the computation of the field distribution on the hologram by using the angular spectrum (AS) method for diffraction. The end result of the method has enabled the computation of a complex 3D objects with thousands of polygons, and at the same time the computation speed is much faster than the traditional method and the analytical method in Section 2.A and 2.B, respectively, and the method proposed by Pan *et al.* discussed in the previous section.

C. Fast 3D Affine Transformation Method

We have currently attempted a revised method based on the 3D affine transformation based on Pan *et al.*'s to improve the computation efficiency. To solve the singular matrix problem, they use the concept of a pseudo-inverse matrix, which is very time consuming and not precise for the calculations. The existence of the noninvertible matrix is that the unit right triangle is defined to locate at $z_s = 0$ plane, so the value of z_s of the triangle has been set to be zero in the formalism. In other words, they have not considered the relative position between the global system and the local system. What is it to be if we let the unit right triangle be located at the $z_s = z_0 \neq 0$ plane? Once we have $z_s \neq 0$, the matrix T can now be fully inverted. The resulting consequence is the fast calculation time, and we call it the fast 3D affine transformation (F3DAT) method. In the next section, we will compare the simulation and experimental results of all the methods discussed in this section.

4. COMPUTATIONAL AND EXPERIMENTAL RESULTS

A 3D acquisition system consisting a camera [Dalsa CA-D6-0512W with resolution 532 × 516 pixels of pixel size 10 μm] and a fringe projector [Kodak DP900 projector with available wavelengths of 360–700 nm] have been used to extract depth points and texture information of real objects to make a point cloud model [75]. The viewing volume is $342 \times 376 \times 658\ \text{mm}^3$ with the actual size of "Sophie" being $137.8 \times 180.6 \times 75.1\ \text{mm}^3$. The point cloud model is then converted to a triangular-mesh model by using triangulation

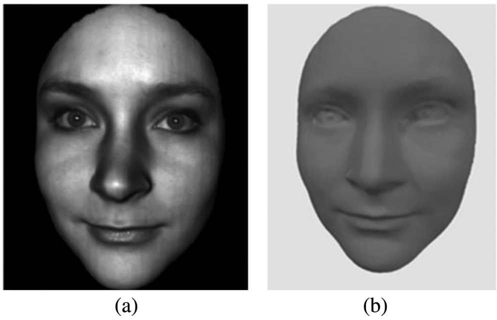

Fig. 4. (a) Picture of a real person, (b) "Sophie," geometric surface of (a). The geometric surface with the texture image in (a) is from a publicly accessible geometric archive from the 3D Scanning Laboratory in Stony Brook University [76].

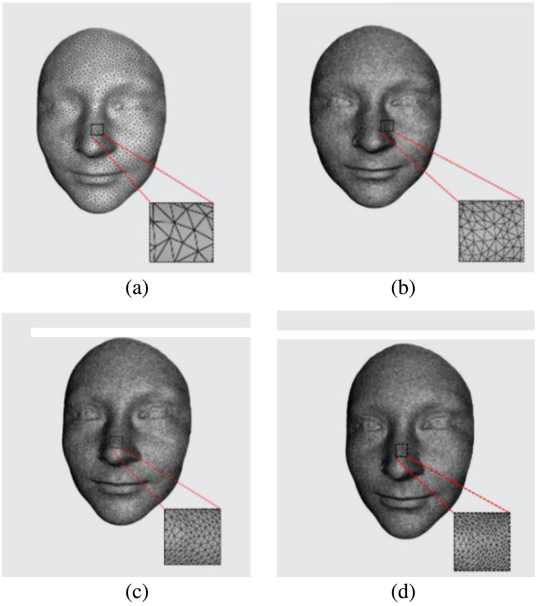

Fig. 5. Sophie consisting of (a) 7270 polygons, (b) 21,402 polygons, (c) 35,403 polygons, and (d) 49,272 polygons.

along with computer graphics. The captured face of a real person is shown in Fig. 4(a), and the geometric surface of Fig. (4a) is called "Sophie," which is shown in Fig. 4(b). Figures 5(a)–5(d) show the face of Sophie consisting of 7270, 21,402, 35,403, and 49,272 polygons, respectively. Note that in the present investigation, these polygons have no texture mapping. However, limited approaches have been investigated to include texture mapping in the analytical techniques [34,35].

We generate holograms with a resolution of 1024 × 1024, and the hardware includes Intel Core i7-11700 at 4.8 GHz, 16 G-byte RAM with RTX 3060Ti under the environment of MATLAB 2018b. In order to reproduce the details of the object, instead of using polygons with constant amplitude, we render the 3D mesh with shading using a simple method to judge the normal and assign values to the triangles with different constant values of reflectance according to the following formula:

$$A_i = k_1 \cos\alpha + k_2 \cos\beta + k_3 \cos\gamma, \tag{28}$$

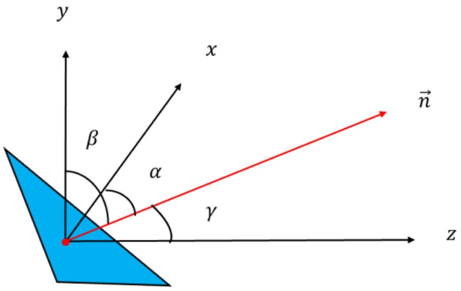

Fig. 6. Polygon with normal vector and angles, β and γ with respect to the x, y, and z axis, respectively.

Table 1. Comparison of Calculation Times for Different Methods

Number of Polygons for Sophie	Traditional	Ahrenberg	Pan	Zhang	F3DAT
7270	//	//	499 s	410 s	363 s
14,254	//	//	978 s	816 s	720 s
21,402	//	//	1470 s	1213 s	1060 s
28,611	//	//	1966 s	1621 s	1389 s
35,403	//	//	2434 s	2007 s	1750 s
42,205	//	//	2931 s	2398 s	2093 s
49,272	//	//	3413 s	2786 s	2463 s

where A_i is the amplitude of the surface function of the ith polygon, and, according to Fig. 6, the normal vector $\vec{n}$ of each triangle is at angles α, β, and γ with the xyz axis in the global coordinate system. k_1, k_2, and k_3 are the weight factors, ranging from 0 to 1 under the condition that $k_1 + k_2 + k_3 = 1$.

In Table 1, we compare the calculation time with the different methods we have discussed. Numerical reconstruction distance is at 200 mm away from the hologram. The calculation times using the traditional and Ahrenberg *et al.*'s methods are too long to tabulate. Note that for all the cases, the F3DAT method is the fastest. Note that in all calculations, we reduce the number of polygons that need to be processed using back-face culling, i.e., we only calculate the polygons that satisfy $\vec{n_h} \cdot \vec{n_a} > 0$, where $\vec{n_h}$ is the normal vector of the hologram plane and $\vec{n_a}$ is the normal vector of the arbitrary polygon. For most 3D objects, back-face culling is an effective way to remove hidden surfaces.

Since Pan *et al.*'s method is a precursor of the F3DAT method, we compare their image reconstruction quality and similarity, and we tabulate the results in Table 2. In the table, PSNR is the peak signal-to-noise ratio, and SSIM is the structural similarity index measure. We see that the two methods are of high quality with great similarity but with the F3DAT method being faster.

In Fig. 7, we show the numerical reconstructions from the F3DAT transformation method. Clearly we observe that the reconstruction from the 49,272-polygon Sophie provides finer details. Finally, regarding numerical reconstruction, we want to point out the importance of shading introduced in Eq. (28). Figure 8 shows the reconstructions with and without shading in (a) and (b), respectively, for the 3D mesh with 49,272 polygons.

In Fig. 9, we show the optical display of the hologram with 49,272-polygon Sophie at optical reconstruction distance of

Table 2. Table of PSNR and SSIM Comparing Pan's Method and the Fast 3D Affine Transformation Method

Number of Polygons (Sophie)	PSNR	SSIM
7270	39.68 (dB)	1.0
21,402	39.54 (dB)	1.0
35,403	39.51 (dB)	1.0
49,272	39.50 (dB)	1.0

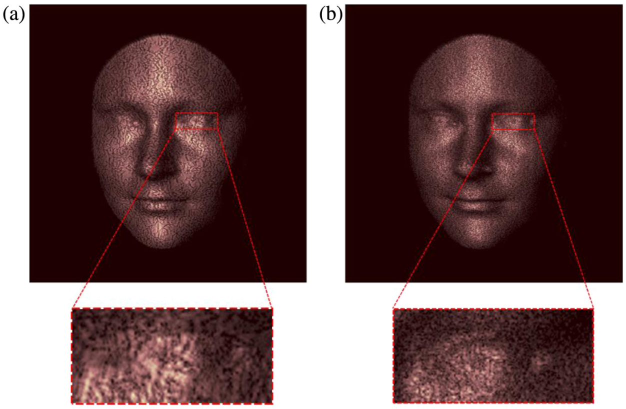

Fig. 7. Reconstructed Sophie consisting of (a) 7270 polygons; (b) 49,272 polygons of the original 3D mesh, from the F3DAT transformation method.

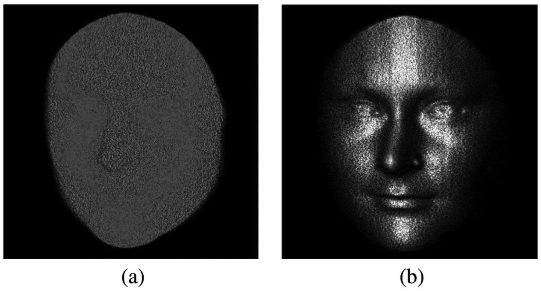

Fig. 8. Reconstruction of Sophie (a) without shading, (b) with shading.

200 mm from the hologram. Note that the actual size of Sophie was first reduced to $5.17 \times 7.11 \times 3.01\ \text{mm}^3$ before the generation of the hologram. The hologram is of 1024 × 1024 pixels. The phase-only hologram of size about $8.2 \times 8.2\ \text{mm}^2$ is then displayed by a phase-only SLM. The SLM used is a HOLOEYE PLUTO(NIR-011) phase-only SLM with a resolution of 1920 × 1080 (full HD 1080 p) with a pixel size of 8 µm. This SLM provides a refresh rate of 60 Hz (monochrome) with a bit depth of 8 bits with diffraction efficiency of over 80%. We use a green laser with wavelength of 532 nm.

We also show the numerical and optical reconstructions of the Stanford bunny. The bunny consists of 59,996 polygons originally. After back-face culling, it has 31,724 polygons. Using the F3DAT method, it takes 2463 s to complete the CGH of 1024 by 1024. Figures 10(a) and 10(b) show the numerical and optical reconstructions, respectively.

As a conclusion to this section, we want to point out that Table 1 has tabulated the calculation time of the two basic frameworks in polygon-based holography, i.e., the traditional method and the analytical method of three approaches. In general, the analytical methods produced shorter calculation time against the traditional method. The reason is that in the traditional method, as pointed out in Sections 2.A and 2.B, it requires two FFTs to obtain each polygon field on the hologram, whereas in the analytical method, it only needs to use a single FFT. In addition, the transformation of the spectrum from the local coordinate system to the parallel local coordinate system is nonlinear, which requires interpolation. Indeed, one estimate indicated that interpolation can take up to 44% of the CPU time [77]. Other estimations reported that interpolation occupies from 61% to 78% of the total calculation time [78,79]. The severity of interpolation is object dependent, as different objects have different surface curvatures. However, with the traditional method, one can achieve high-resolution computer-generated holograms for realistic reconstruction because shading and texture are naturally incorporated into the surface function. Further research on parallel computation with dedicated computing hardware can accelerate the numerical calculations for the traditional method.

The core of the analytical method employs a uniform surface function, which would affect the image quality as well as the creation of realistic 3D images. In other words, shading and

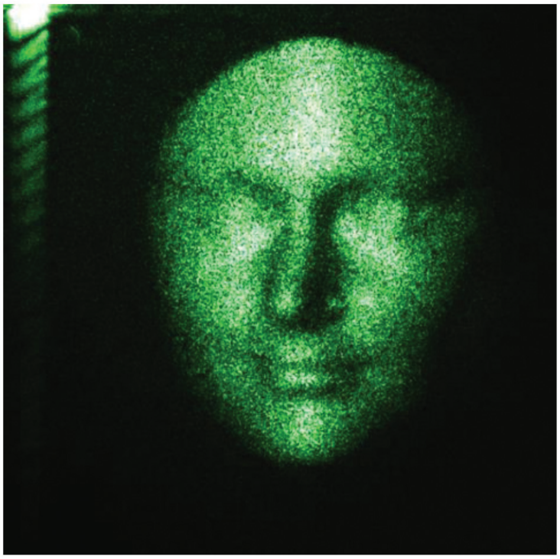

Fig. 9. Optical reconstruction of Sophie, which consists of 49,272 polygons as a 3D mesh.

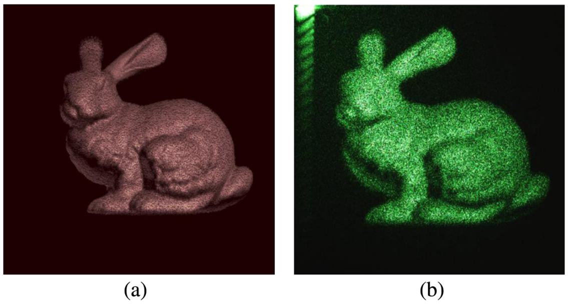

Fig. 10. (a) Numerical reconstruction; (b) optical reconstruction.

texture mapping are not included in the analytical methods considered. However, these techniques allow for computer-generated holograms suitable for SLM display. To achieve realistic reconstructed objects, shading and texture-mapping are needed. Surface diffuseness can be included in the analytical methods fairly easy. For example, Kim *et al.* [70] and Pan *et al.* [68] divided each triangle in the local coordinate system into a set of smaller triangles with different amplitudes and phases. In the present paper, we have included a simple shading method for each polygon [see Eq. (28)]. However, texture mapping is not as straightforward, and texturing of a surface function is paramount. In general, texturing a polygon leads to a convolution in the frequency domain, as the texture pattern multiplies the shape of the polygon, resulting in the slowdown of the overall calculation time. Clearly, there is a trade-off between computational efficiency and texture mapping, which needs to be further investigated, as analytical texturing algorithms remain fairly unexplored.

5. CONCLUDING REMARKS

This review has an emphasis on the algorithms and development of polygon-based computer-generated holography in the area of hologram synthesis. Particular attention is given to the investigation of their numerical implementation of each method. Two classical polygon-based CGH generation algorithms have been fully discussed. Performance of three most recent methods is also evaluated by using high-resolution 3D real face data captured by a depth camera—a state-of-the-art experimental result in polygon-based CGH. A summary of the present research progress is also provided. Complexity of rendering and hidden-surface removal have a direct impact on the efficiency of the CGH computation and the resulting holographic reconstructed image quality. Objective quality assessments have also been evaluated in terms of PSNR and SSIM. However, holographic data have very different signal properties compared to 2D images, and other quality assessments appropriate for holographic data should be investigated [80,81].

The challenge of high-quality and high-resolution hologram synthesis, real-time holographic display, fast algorithms, artificial intelligence (AI), and acceleration of computation with hardware implementation are all needed. We believe that the algorithm development with advanced hardware will open exciting avenues for significantly advancing computer-generated holography and its applications.

Funding. National Natural Science Foundation of China (11762009, 61865007); Yunnan Provincial Program for Foreign Talent (202105AO130015); Yunnan Provincial Science and Technology Department (2019FA025).

Acknowledgment. The authors thank Prof. Chongguang Li and Prof. Yongan Zhang from Kunming University of Science and technology for their helpful discussions. They also thank Wenlong Qin and Qingyang Fu for their help on the experiment. Yaping Zhang also acknowledges the support from Virginia Tech for offering an open access database which enables her to do the up-to-date research work. We appreciate the use of the SLM, which is on loan from Prof. Yanfei Lv, Department of Physics and Astronomy, Yunnan University.

Disclosures. The authors declare no conflicts of interest.

Data Availability. Original data are available in Ref. [76].

REFERENCES

1. D. Gabor, "A new microscopic principle," Nature **161**, 777–778 (1948).
2. B. R. Brown and A. W. Lohmann, "Complex spatial filtering with binary mask," Appl. Opt. **5**, 967–969 (1966).
3. T.-C. Poon, ed., *Digital Holography and Three-dimensional Display: Principles and Applications* (Springer, 2006).
4. P. W. M. Tsang and T.-C. Poon, "Review on the state-of-the-art technologies for acquisition and display of digital holograms," IEEE Trans. Ind. Inf. **12**, 886–901 (2016).
5. Z. He, X. Sui, G. Jin, and L. Cao, "Progress in virtual reality and augmented reality based on holographic display," Appl. Opt. **58**, A74–A81 (2019).
6. J.-H. Park, "Recent progress in computer-generated holography for three-dimensional scenes," J. Inf. Disp. **18**, 1–12 (2017).
7. J.-H. Park, "Recent progress on fully analytic mesh based computer-generated holography," Proc. SPIE **10022**, 100221G (2016).
8. E. Sahin, E. Stoykova, J. Makinen, and A. Gotchev, "Computer-generated holograms for 3D imaging: a survey," ACM Comput. Surv. **53**, 1–35 (2020).
9. M. Yamaguchi, "Light-field and holographic three-dimensional displays [Invited]," J. Opt. Soc. Am. A **33**, 2348–2364 (2016).
10. R. Corda, D. Giusto, A. Liotta, W. Song, and C. Perra, "Recent advances in the processing and rendering algorithms for computer-generated holography," Electronics **8**, 556 (2019).
11. H. Yoshikawa and T. Yamaguchi, "Review of holographic printers for computer-generated holograms," IEEE Trans. Ind. Inf. **12**, 1584–1589 (2016).
12. P. W. M. Tsang, T.-C. Poon, and Y. M. Wu, "Review of fast methods for point-based computer-generated holography," Photon. Res. **6**, 837–846 (2018).
13. T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Review of fast calculation techniques for computer-generated holograms with the point-light-source-based model," IEEE Trans. Ind. Inf. **13**, 2447–2454 (2017).
14. T. Shimobaba, T. Kakue, and T. Ito, "Review of fast algorithms and hardware implementations on computer holography," IEEE Trans. Ind. Inf. **12**, 1611–1622 (2016).
15. Y. Wang, D. Dong, P.-J. Christopher, A. Kadis, R. Mouthaan, F. Yang, and T.-D. Wilkinson, "Hardware implementations of computer-generated holography: a review," Opt. Eng. **59**, 102413 (2020).
16. K. Matsushima, *Introduction to Computer Holography: creating Computer-Generated Holograms as the Ultimate 3D Images* (Springer, 2020).
17. T. Shimobaba and T. Tto, *Computer Holography Acceleration Algorithms and Hardware Implementations* (CRC Press, Tayler and Francis Group, 2019).
18. P. W. M. Tsang, *Computer-generated Phase-only Holograms for 3D Display* (Cambridge University, 2021).
19. Ch. Frère, D. Leseberg, and O. Bryngdahl, "Computer-generated holograms of three-dimensional objects composed of line segments," J. Opt. Soc. Am. A **3**, 726–730 (1986).
20. 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).
21. A. Gilles and P. Gioia, "Real-time layer-based computer-generated hologram calculation for the Fourier transform optical system," Appl. Opt. **57**, 8508–8517 (2018).
22. H. Zhang, L. Cao, and G. Jin, "Computer-generated hologram with occlusion effect using layer-based processing," Appl. Opt. **56**, F138–F143 (2017).
23. 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).
24. J.-S. Chen and D.-P. Chu, "Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications," Opt. Express **23**, 18143–18155 (2015).
25. K. Matsushima and A. Kondoh, "Wave optical algorithm for creating digitally synthetic holograms of three-dimensional surface objects," Proc. SPIE **5005**, 190–197 (2003).

26. K. Matsushima and S. Nakahara, "Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method," Appl. Opt. **48**, H45–H63 (2009).
27. K. Matsushima, "Computer-generated holograms for three-dimensional surface objects with shade and texture," Appl. Opt. **44**, 4607–4614 (2005).
28. X. Tang, F. Nan, and Z. Yan, "Rapidly and accurately shaping the intensity and phase of light for optical nano-manipulation," Nanoscale Adv. **2**, 2540–2547 (2020).
29. 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).
30. 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).
31. K. Matsushima, "Wave-field rendering in computational holography: the polygon-based method for full-parallax high-definition CGHs," in *IEEE/ACIS 9th International Conference on Computer and Information Science* (2010), pp. 846–851.
32. H. Nishi and K. Matsushima, "Rendering of specular curved objects in polygon-based computer holography," Appl. Opt. **56**, F37–F44 (2017).
33. K. Matsushima, H. Nishi, and S. Nakahara, "Simple wave-field rendering for photorealistic reconstruction in polygon-based high-definition computer holography," J. Electron. Imag. **21**, 023002 (2012).
34. W. Lee, D. Im, J. Paek, J. Hahn, and H. Kim, "Semi-analytic texturing algorithm for polygon computer-generated holograms," Opt. Express **22**, 31180–31191 (2014).
35. Y.-M. Ji, H. Yeom, and J.-H. Park, "Efficient texture mapping by adaptive mesh division in mesh-based computer generated hologram," Opt. Express **24**, 28154–28169 (2016).
36. K. Yamaguchi, T. Ichikawa, and Y. Sakamoto, "Calculation method for CGH considering smooth shading with polygon models," Proc. SPIE **7957**, 795706 (2011).
37. K. Matsushima, "Exact hidden-surface removal in digitally synthetic full-parallax holograms," Proc. SPIE **5742**, 25–32 (2005).
38. 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 **5290**, 90–97 (2004).
39. 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).
40. J.-H. Park, H.-J. Yeom, H.-J. Kim, H.-J. Zhang, B.-N. Li, Y.-M. Ji, and S.-H. Kim, "Removal of line artifacts on mesh boundary in computer generated hologram by mesh phase matching," Opt. Express **23**, 8006–8013 (2015).
41. M.-C. Juan, K. Tomasz, K. Rafal, C. Maksymilian, and I.-M. Sajeev, "Occlusion culling for wide-angle computer-generated holograms using phase added stereogram technique," Photonics **8**, 298 (2021).
42. 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).
43. J.-P. Liu and H.-K. Liao, "Fast occlusion processing for a polygon-based computer-generated hologram using the slice-by-slice silhouette method," Appl. Opt. **57**, A215–A221 (2018).
44. T. Ichikawa, Y. Sakamoto, A. Subagyo, and K. Sueoka, "Calculation method of reflectance distributions for computer-generated holograms using the finite-difference time-domain method," Appl. Opt. **50**, H211–H219 (2011).
45. Y.-G. Ju and J.-H. Park, "Foveated computer-generated hologram and its progressive update using triangular mesh scene model for near-eye displays," Opt. Express **27**, 23725–23738 (2019).
46. 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).
47. H. Kim, J. Kwon, and J. Hahn, "Accelerated synthesis of wide-viewing angle polygon computer-generated holograms using the interocular affine similarity of three-dimensional scenes," Opt. Express **26**, 16853–16874 (2018).
48. D. Im, J. Cho, J. Hahn, B. Lee, and H. Kim, "Accelerated synthesis algorithm of polygon computer-generated holograms," Opt. Express **23**, 2863–2871 (2015).
49. N. Takada, T. Shimobaba, H. Nakayama, A. Shiraki, N. Okada, M. Oikawa, N. Masuda, and T. Ito, "Fast high-resolution computer-generated hologram computation using multiple graphics processing unit cluster system," Appl. Opt. **51**, 7303–7307 (2012).
50. T. Shimobaba, T. Ito, N. Masuda, Y. Ichihashi, and N. Takada, "Fast calculation of computer-generated-hologram on AMD HD5000 series GPU and OpenCL," Opt. Express **18**, 9955–9960 (2010).
51. 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).
52. Z. M. A. Lum, X. Liang, Y. Pan, R. Zheng, and X. Xu, "Increasing pixel count of holograms for three-dimensional holographic display by optical scan-tiling," Opt. Eng. **52**, 015802 (2013).
53. M. Stanley, P. B. Conway, S. D. Coomber, J. C. Jones, D. C. Scattergood, C. W. Slinger, R. W. Bannister, C. V. Brown, W. A. Crossland, and A. Travis, "Novel electro-optic modulator system for the production of dynamic images from giga-pixel computer-generated holograms," Proc. SPIE **3956**, 13–22 (2000).
54. C. D. Cameron, D. A. Pain, M. Stanley, and C. W. Slinger, "Computational challenges of emerging novel true 3D holographic displays," Proc. SPIE **4109**, 129–140 (2000).
55. Y. Ito, M. Mitobe, M. Nagahama, H. Sakai, and Y. Sakamoto, "Wide visual field angle holographic display using compact electro-holographic projectors," Appl. Opt. **58**, G135–G142 (2019).
56. T.-C. Poon and J.-P. Liu, *Introduction to Modern Digital Holography with MATLAB* (Cambridge University, 2014).
57. S. Ganci, "Fourier diffraction through a tilted slit," Eur. J. Phys. **2**, 158–160 (1981).
58. K. Patorski, "Fraunhofer diffraction patterns of tilted planar objects," Opt. Acta **30**, 673–679 (1983).
59. H. J. Rabal, N. Bolognini, and E. E. Sicre, "Diffraction by a tilted aperture," Opt. Acta **32**, 1309–1311 (1985).
60. D. Leseberg and C. Frere, "Computer-generated holograms for 3-D objects composed of tilted planar segments," Appl. Opt. **27**, 3020–3024 (1988).
61. C. Frere and D. Lesberg, "Large objects reconstructed from computer-generated hologram," Appl. Opt. **28**, 2422–2425 (1989).
62. T. Tommasi and B. Bianco, "Frequency analysis of light diffraction between rotated planes," Opt. Lett. **17**, 556–558 (1992).
63. T. Tommasi and B. Bianco, "Computer-generated holograms of tilted planes by a spatial frequency approach," J. Opt. Soc. Am. A **10**, 299–305 (1993).
64. 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).
65. K. Matsushima, "Formulation of the rotational transformation of wave fields and their application to holography," Appl. Opt. **47**, D110–D116 (2008).
66. N. Delen and B. Hooker, "Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: a fast Fourier transform approach," J. Opt. Soc. Am. A **15**, 857–867 (1998).
67. L. Onural, "Exact solution for scalar diffraction between tilted and translated planes using impulse functions over a surface," J. Opt. Soc. Am. A **28**, 290–295 (2011).
68. 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).
69. R.-N. Bracewell, K.-Y. Chang, A. K. Jha, and Y.-H. Wang, "Affine theorem for two-dimensional Fourier transform," Electron. Lett. **29**, 304 (1993).
70. 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).
71. Y. Zhang, J. Zhang, W. Chen, J. Zhang, P. Wang, and W. Xu, "Research on three-dimensional computer-generated holographic algorithm based on conformal geometry theory," Opt. Commun. **309**, 196–200 (2013).

72. H. Sakata and Y. Sakamoto, "Fast computation method for a Fresnel hologram using three-dimensional affine transformations in real space," Appl. Opt. **48**, H212–H221 (2009).
73. Y. Pan, Y. Wang, J. Liu, X. Li, J. Jia, and Z. Zhang, "Analytical brightness compensation algorithm for traditional polygon-based method in computer-generated holography," Appl. Opt. **52**, 4391–4399 (2013).
74. 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).
75. Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang, " High resolution tracking of non-rigid 3D motion of densely sampled data using harmonic maps," in *10th IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, 17–20 October, 2005, pp. 388–395.
76. Stony Brook University 3D Scanning Laboratory, https://www3.cs.stonybrook.edu/~gu/software/holoimage/index.html.
77. K. Matsushima, "Performance of the polygon-source method for creating computer-generated holograms of surface object," in *ICO Topical Meeting on Optoinformatics/Information Photonics* (2006), pp. 99–100.
78. Y. Pan, "Study on Polygon-based methods in computer-generated holography for dynamic holographic three-dimensional display," Ph.D. thesis (in Chinese) (Beijing Institute of Technology, 2015).
79. F. Wang, "Research on computer-generated hologram based on polygon-based method," M.S. thesis (in Chinese) (Kunming University of Science & Technology, 2019).
80. A. Ahar, D. Blinder, T. Bruylants, C. Schretter, A. Munteanu, and P. Schelkens, "Subjective quality assessment of numerically reconstructed compressed holograms," Proc. SPIE **9599**, 95990K (2015).
81. D. Blinder, A. B. Ahar, S. Bettens, T. Birnbaum, A. Symeonidou, H. Ottevaere, C. Schretter, and P. Schelkens, "Signal processing challenges for digital holographic video display systems," Signal Process. Image Commun. **70**, 114–130 (2019).

128.173.90.185 {ts '2022-01-28 12:18:58'}