

Article

# A Layered Method Based on Depth of Focus for Rapid Generation of Computer-Generated Holograms

Xiandong Ma<sup>1</sup>, Jinbin Gui<sup>1,2,\*</sup>, Junchang Li<sup>1</sup> and Qinghe Song<sup>1</sup>

<sup>1</sup> Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China; 3153834066gg@gmail.com (X.M.); jcli99@163.com (J.L.); songqinghe@kust.edu.cn (Q.S.)

<sup>2</sup> Yunnan Provincial Key Laboratory of Modern Information Optics, Kunming 650500, China

\* Correspondence: guijinbin@kust.edu.cn

**Abstract:** In this paper, a layered method based on focal depth is proposed for the fast generation of computational holograms. The method layers objects with focal depth as spacing and approximates triangles on the object as projections on the layers based on the physical properties of the focal depth to simplify the computation. Finally, the diffraction distributions of all layers are calculated via angular spectral diffraction and superimposed to obtain the hologram. The proposed method has been proven to be about 20 times faster on a CPU than the analytical polygon-based method. A hologram containing tens of thousands of triangles can be computed on a GPU in a fraction of a second. In addition, this method makes it easy to attach complex textures, which is difficult with polygon-based analysis methods. Finally, holograms of objects with complex textures were generated, and the three-dimensionality of these holograms was confirmed by numerical and optical reconstruction.

**Keywords:** computer-generated hologram; holographic 3D display; depth of focus; polygon-based method; layer-based method



**Citation:** Ma, X.; Gui, J.; Li, J.; Song, Q. A Layered Method Based on Depth of Focus for Rapid Generation of Computer-Generated Holograms. *Appl. Sci.* **2024**, *14*, 5109. <https://doi.org/10.3390/app14125109>

Academic Editor: Davood Khodadad

Received: 21 April 2024

Revised: 21 May 2024

Accepted: 23 May 2024

Published: 12 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Holographic display technology is widely considered to be the most promising three-dimensional (3D) display technology since it can completely reconstruct the light field of a three-dimensional scene [1–8]. A computer-generated hologram (CGH) simulates the physical interference process of holography for numerically represented 3D objects [9–12], and it plays an important role in holographic display technology because these objects can be conveniently generated by computers and can be used to reconstruct virtual objects. According to the sampling method used for the numerical representation of 3D objects, the methods for generating CGH can be divided into point-based methods, polygon-based methods, and layer-based methods.

The point source method is the most widely used because of its simple and intuitive principle. However, to ensure image quality, the number of discrete points in three-dimensional objects is usually millions of orders of magnitude, which leads to too long of a computation time for generating holograms. To solve this problem, many algorithms have been proposed to simplify the process and reduce calculation time [13–21]. However, the large number of samples is still an important factor slowing down the calculation speed. The polygon-based method treats three-dimensional objects as a collection of polygons [22,23], which can reduce the number of samples by two to three orders of magnitude compared to the point source method. Due to this method of discretizing objects into polygons, rendering information such as lighting, shadows, and textures in computer graphics can be utilized to enhance the rendering of the scene [24]. For many years, scholars have been dedicated to the research of polygon-based methods, proposing many advanced and efficient methods [25–31]. The layer-based method performs layer sampling on 3D objects [32–35], and compared to point source and panel methods, it has the least number of samples. When the number of layers is fixed, the computational complexity

of layer-based methods is not affected by the scene's complexity, while point-based and polygon-based methods do not have this advantage. Layer-based methods consume less memory and compute faster, thus having great potential in real-time holographic 3D displays. At present, these methods have achieved good results in computational efficiency and imaging quality after years of development and iteration. For example, a convolutional symmetric compressed look-up table (CSC-LUT) [36] method proposed by Wei et al. can achieve real-time (>24 fps) color holographic display corresponding to three perspectives of a 3D scene. The WRP-like method [37] proposed by Wang et al. can compute holograms with tens of thousands of triangles in seconds, even with a CPU. Liu et al. proposed a dual-channel parallel neural network (DCPNet) [38] that can generate 2 k phase holograms with high fidelity in 36 ms.

In this paper, a new approach is proposed based on the study of focal depth. This method utilizes the property wherein the diffraction of a point source in the focal depth range can be approximated as its projection. The triangles on the object are layered at intervals of the focal depth, and their projections on the layers are considered their diffractions onto the layers. After layering and projecting all the triangles on the object, the object is approximated as a set of planar light sources with focal depth spacing, which are parallel to the holographic plane. Therefore, the diffracted field of an object in the hologram plane is approximated as a superposition of the diffracted fields of these parallel planes. The diffracted field is calculated using angular spectrum theory, and this calculation is accelerated by the fast Fourier transform (FFT). The proposed method projects the tilted triangles onto the layers without considering the diffraction calculation of the tilted triangles, so the calculation speed is greatly improved. This paper compares the computational efficiency of the proposed method with a polygon-based analytical method and demonstrates a computational speed increase of about 20 times on a CPU. In addition, the polygon-based parsing method makes it difficult to generate holograms with complex textures because it relies on the analytic form of the spectrum, and many scholars have researched this challenge [39]. In contrast, the proposed method is based on FFT and can easily generate holograms with complex textures. In this paper, holograms with complex textures are generated using the UV mapping method, and the effectiveness and three-dimensional sense of the holograms are proved by numerical and optical reconstruction.

## 2. The Layered Method Based on the Depth of Focus

### 2.1. Method

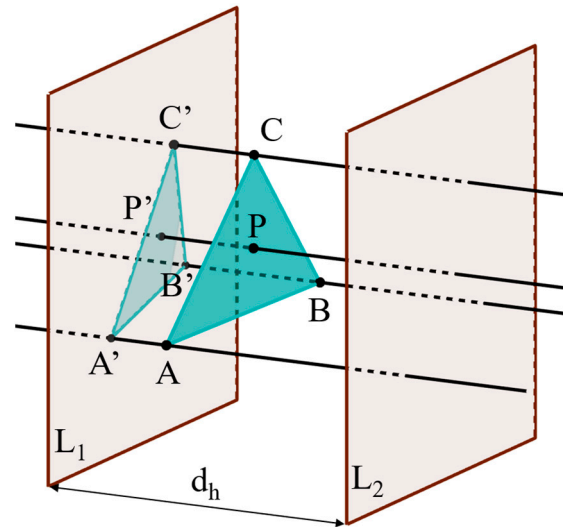
In [40], it has been theoretically demonstrated that the focal depth characteristics of an LCOS imaging system exhibit the property where, within the depth of field, the diffracted light field of a point source can be approximated as a direct projection of the point source, and an expression for calculating the depth of focus has been derived:

$$d_h = \frac{2\lambda d^2}{(N\Delta)^2}, \quad (1)$$

where  $d_h$  represents the depth of focus,  $\lambda$  stands for the wavelength of light,  $d$  denotes the distance from the object to the hologram plane,  $N$  signifies the number of samples taken across the hologram, and  $\Delta$  represents the sampling interval. In this paper, this theory is used to approximate and simplify the algorithm for generating holograms of objects composed of polygons.

Suppose two mutually parallel planes exist,  $L_1$  and  $L_2$ , with the distance between them equal to the depth of focus  $d_h$ . Within the space between these parallel planes lies a triangle labeled  $ABC$ , as shown in Figure 1. Since the size of the triangle lies within the range determined by planes  $L_1$  and  $L_2$ , any point on the triangle has a distance to plane  $L_1$  that is less than the depth of focus  $d_h$ . According to the physical properties of the depth of focus mentioned above, the diffraction of the light wave from the point source  $P$  on the triangle  $ABC$  onto the plane  $L_1$  can be approximated as the projection of the point  $P$  onto

the plane  $L_1$ . In this way, after projecting all points on the triangle, a projection triangle  $A'B'C'$  can be obtained on the plane  $L_1$ , and this projected triangle can be considered as the diffraction distribution of the triangle  $ABC$  on the plane  $L_1$ .



**Figure 1.** Approximating the diffraction of a triangle as its projection in the depth-of-focus range.

We noticed that the depth of a triangle is not always less than the depth of focus, and it may also be greater than the depth of focus (the depth of a triangle is the length of the triangle along the direction of light wave spreading). In this case, we need to split the triangle into polygons with depths not exceeding the focal depth and then project these polygons onto the nearest layer, as shown in Figure 2. After the projection is complete, a tilted triangle light source can be transformed into a set of plane light sources that are planar to the hologram plane. The diffraction distribution of this set of planes in the hologram plane can be calculated by angular spectrum theory. The angular spectrum method is a common method for calculating the diffraction of light waves between two parallel planes, which can be expressed by the equation

$$U_H = F^{-1} \left\{ F\{U_o\} \exp \left[ j \frac{2\pi}{\lambda} d \sqrt{1 - (\lambda f_x)^2 - (\lambda f_y)^2} \right] \right\}, \quad (2)$$

where,  $U_H$  represents the optical field on the hologram plane,  $U_o$  signifies the optical field on the object plane.  $j$  denotes the imaginary unit,  $\lambda$  is the wavelength of light,  $d$  indicates the propagation distance, and  $f_x$  and  $f_y$  denote the coordinates in the frequency domain. Here,  $F\{\}$  and  $F^{-1}\{\}$  represent the Fourier transform and its inverse, respectively. Therefore, as shown in Figure 3, each layer is parallel to the hologram plane, so the propagation of the light wave from each layer to the hologram plane can be calculated using the angular spectrum method. Finally, the light waves from these layers in the hologram plane are summed to obtain the object diffraction distribution  $U_H$ .

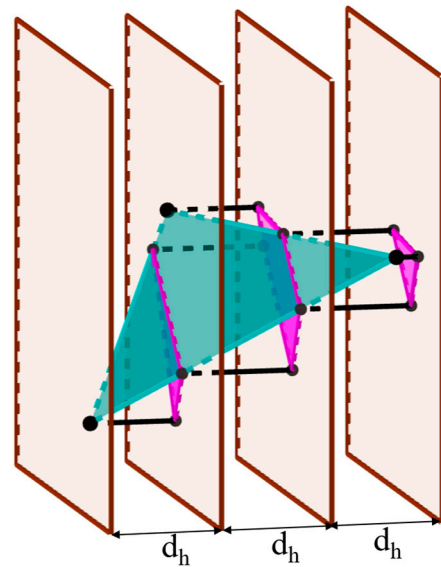


Figure 2. Layered projection of triangles with depth greater than focal depth.

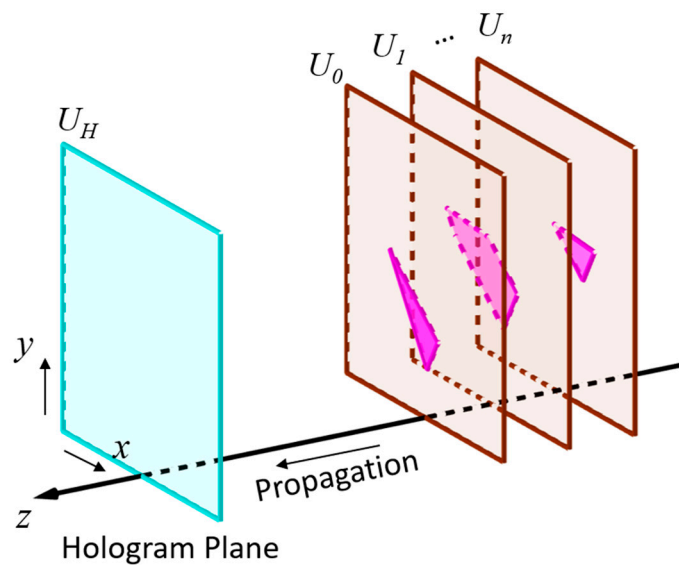


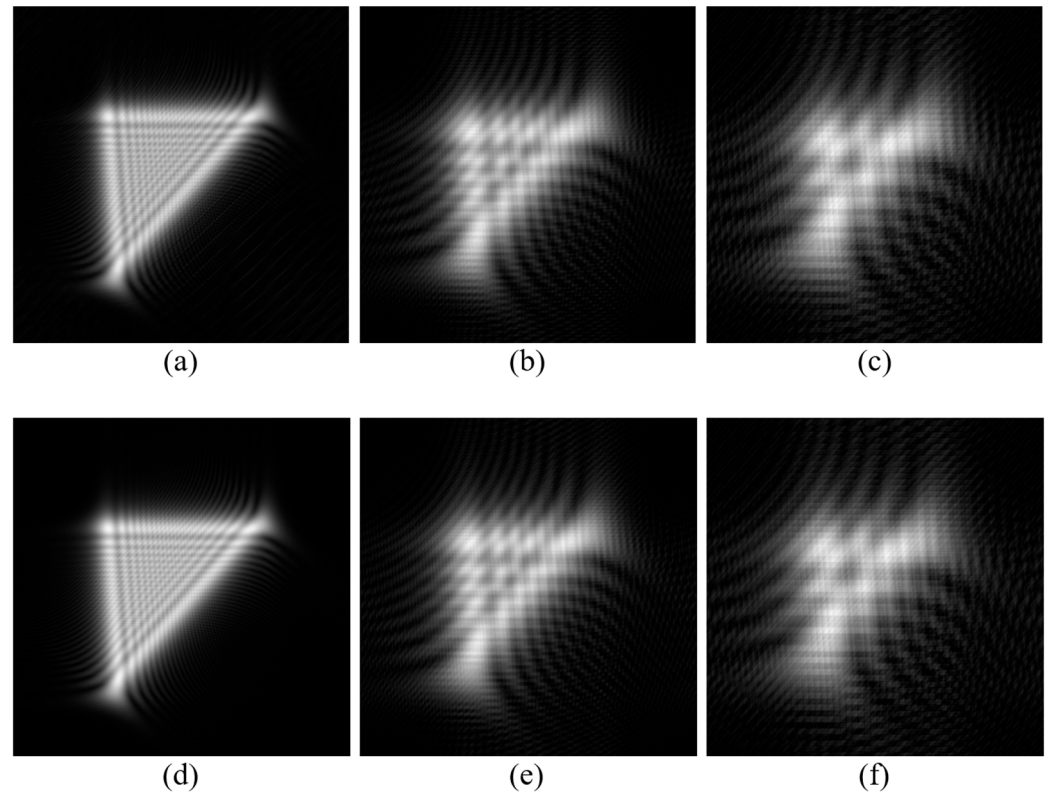
Figure 3. Schematic of the angular spectrum method for calculating the propagation of each layer to the holographic plane.

### 2.2. Verification

In order to verify the feasibility of the layered projection method based on depth of focus, the diffraction distributions of random triangles computed by this method and the analytical polygon-based method [41] are compared. The analytical polygon-based method calculates the diffracted field of tilted triangles through 3D affine transformations and the analytical form of the spectrum, and there is no approximation and interpolation in the calculation process, so the calculation results of this method are more accurate. Therefore, the computational results of this method are used as a reference to verify the feasibility of the proposed method.

Since the proposed method involves two distinct scenarios during its computation process (wherein one scenario allows for direct projection of the triangle, while the other necessitates the division of the triangle prior to projection), the experiment encompassed both cases by selecting multiple different propagation distances for the diffraction calculations. Thirty triangles were used for the experiment, and the orientations and shapes of these triangles were randomized, and diffraction distributions of 100 mm, 400 mm, and

700 mm were calculated for each triangle. Figure 4 shows experimental results for one of the random triangles. Figure 4a–c represent the diffraction distributions calculated by the proposed method at distances of 100 mm, 400 mm, and 700 mm, respectively. Figure 4d–f represent the diffraction distributions calculated by the analytical polygon-based method at 100 mm, 400 mm, and 700 mm, respectively. Figure 4 shows that there is almost no difference in diffraction distribution between the two methods when light waves propagate at the same distance.



**Figure 4.** A set of results of the diffraction distributions calculated by the two methods. (a–c) are the diffraction distributions at 100 mm, 400 mm, and 700 mm, respectively, calculated by the proposed method. (d–f) are the diffraction distributions at 100 mm, 400 mm, and 700 mm, respectively, calculated by the analytical polygon-based method.

In order to describe the similarity between the diffraction distributions calculated by the two methods under the same conditions, the peak signal-to-noise ratio (PSNR) between them was calculated. PSNR is a common and objective method used to determine the similarity between two images, which can be calculated by Equation (3):

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right), \quad (3)$$

where  $MAX_I$  refers to the maximum possible value of a pixel in the image. In this paper, since the images are 8-bit grayscale images,  $MAX_I$  is taken as 255. MSE stands for mean square error, and its computation formula is as follows:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2, \quad (4)$$

where  $m$  and  $n$  are the height and width of the image, respectively.  $I(i, j)$  and  $K(i, j)$  are the two images for comparison, respectively. The diffraction distributions obtained from each triangle at the same distance using both methods were used to calculate the PSNR using

Equation (3) to obtain the results shown in Table 1. In Table 1, the first column represents the numbering of the random triangle, and the second, third, and fourth columns represent the PSNR between the diffraction distributions calculated by the two methods at light wave propagation distances of 100 mm, 400 mm, and 700 mm, respectively.

**Table 1.** PSNR of diffraction distributions of 10 random triangles at different distances calculated by two methods.

ID	Diffraction of 100 mm	Diffraction of 400 mm	Diffraction of 700 mm
1	32.61	30.96	28.88
2	31.92	31.31	28.68
3	30.50	29.84	28.18
4	29.79	29.57	27.46
5	30.98	30.56	28.71
6	35.67	33.46	30.11
7	35.27	32.23	29.22
8	32.28	31.27	29.59
9	30.16	29.31	27.28
10	33.19	32.13	30.77
11	34.88	33.61	30.93
12	33.00	32.87	31.19
13	32.58	32.29	30.76
14	32.75	32.75	30.87
15	35.03	33.74	30.97
16	35.28	33.34	30.54
17	32.53	31.16	29.55
18	37.98	36.24	32.83
19	29.75	28.75	26.69
20	35.50	34.91	32.18
21	32.17	31.46	29.79
22	31.67	31.19	29.29
23	30.92	30.74	28.93
24	32.56	31.26	29.94
25	36.24	35.86	31.10
26	29.92	28.89	27.05
27	33.83	32.71	30.21
28	30.31	29.53	27.48
29	33.39	33.03	31.33
30	37.39	30.56	26.07

From Table 1, we can observe a phenomenon that there is a little decrease in the PSNR value as the diffraction distance increases. We will explain the phenomenon. The analytic polygon method calculates the diffraction distribution by analytic equation without approximation, so its results are more accurate, while the proposed method uses the focal depth as the spacing to layer the triangles, and from Equation (1), we know that the focal depth is proportional to the square of the distance. Thus, the further the diffraction distance and the smaller the number of layers, the larger the error between the calculated diffraction distribution and the actual diffraction distribution. Note that the experiment has included the case of dividing the triangles into only one layer, which is the case with the largest error. We calculated the mean and standard deviation of these PSNRs, as shown in Table 2. This means that at different propagation distances, the PSNR of the diffraction distributions of the two methods is around 31.5, indicating that the diffraction distributions calculated by the two methods are extremely similar. Thus, the feasibility of dividing the triangles into layers based on the depth of focus is demonstrated.

**Table 2.** The mean and standard deviation of PSNR.

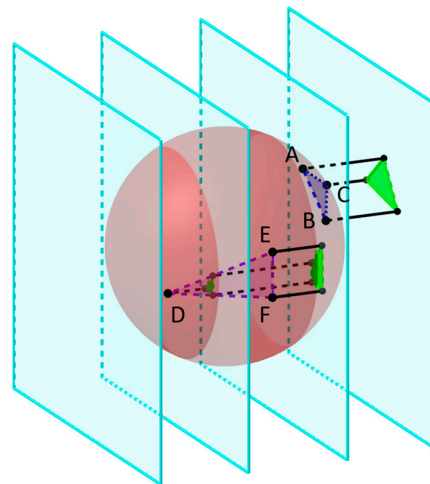
The Mean of PSNR	The Standard Deviations of PSNR
31.46	2.42

### 2.3. Use on Three-Dimensional Objects

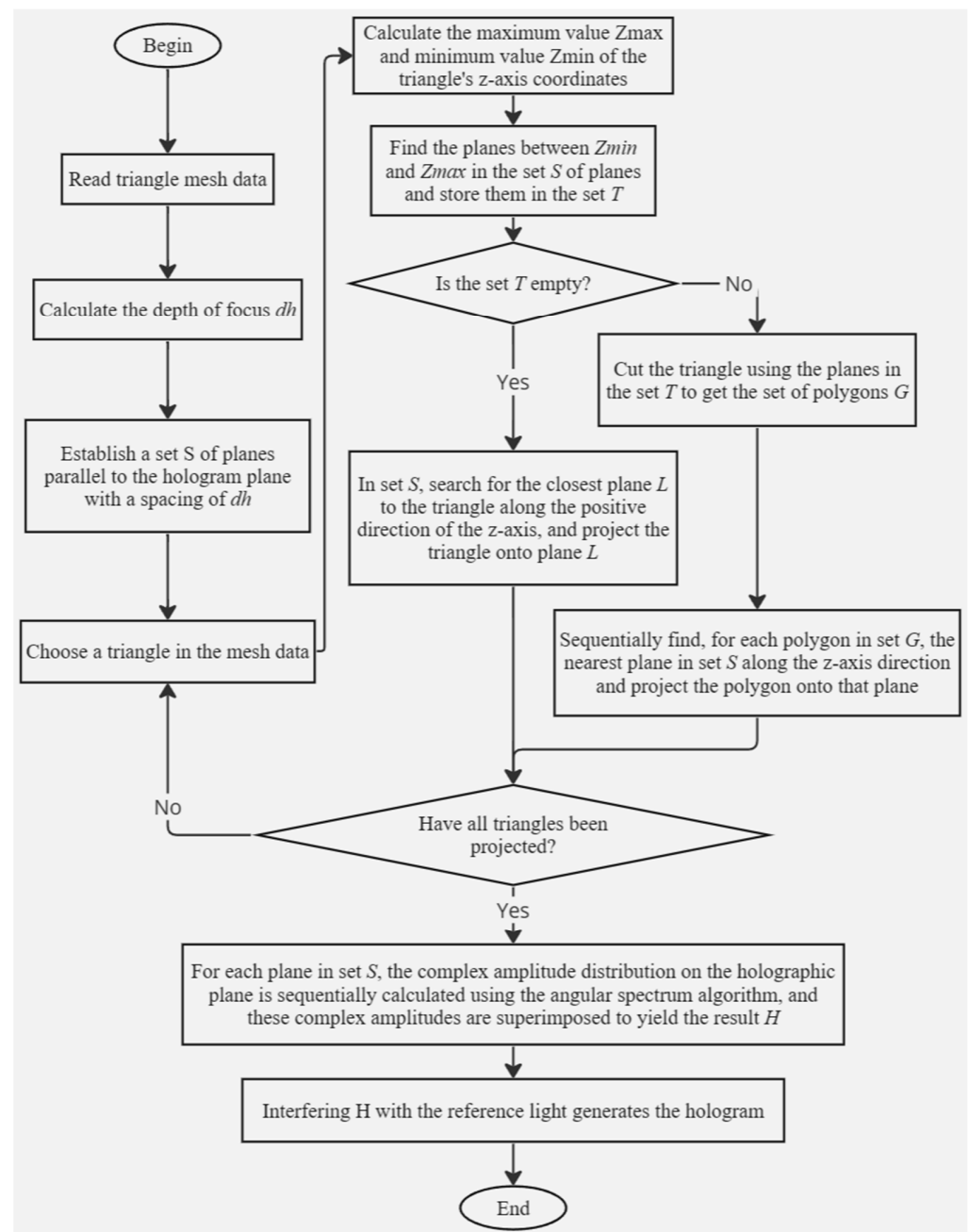
The feasibility of the proposed method to compute the diffraction distribution of individual triangles has been verified above, and we will next use the method to compute the hologram of the 3D object.

First, a set of planes parallel to the hologram plane is defined to divide the 3D object, and the spacing of this set of planes is equal to the depth of focus. Then, we need to project the triangles on the 3D object onto these planes according to the following rules:

1. If the triangle lies between two neighboring planes, project the triangle along the direction of propagation of the light wave onto the closest plane, such as triangle *ABC* in Figure 5.
2. If the triangle passes through one or more planes, use the plane that passes through it to cut the triangle, and then the polygon obtained by the cutting is projected along the direction of the optical axis onto the nearest neighboring plane, as shown in triangle *DEF* in Figure 5.

**Figure 5.** Layered projections of triangles on three-dimensional objects.

Assuming that the object's light wave travels in the positive direction along the *z*-axis, the process for the proposed method to compute the hologram of a 3D object is visually presented in Figure 6, which enables a more lucid comprehension of the entire calculation procedure.



**Figure 6.** The division of a three-dimensional object using parallel planes.

### 3. Performance

The core problem of the polygon-based method is the diffraction of tilted triangles. In contrast, the proposed method approximates the diffraction calculation of tilted triangles to the diffraction calculation between parallel planes by projecting the triangles into layers according to the depth of focus, thus reducing the computational burden of the polygon-based method. To test the computational efficiency of the proposed method, the computational time consumption of the method is compared with that of the analytical polygon-based method [41] under the same conditions.

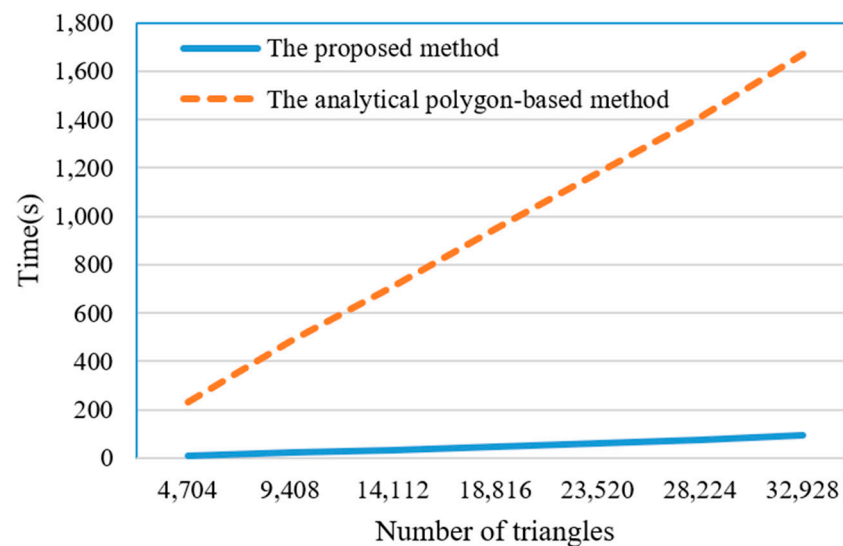
The experiment was performed on a Windows 10 system, where algorithms were programmed in C++ within Visual Studio 2019 and then executed and displayed using MATLAB 2022. The computation time was measured using the “tic” and “toc” commands in MATLAB. The CPU version is the 11th Gen Intel(R) Core (TM) i7-11800H. For accuracy and consistency, we limited the computation to a single thread. We created a 3D model



using Blender and adjusted all models to have the same length along the z-axis. We set the hologram resolution to  $1024 \times 1024$ , the pixel spacing to 0.018 mm, the light wavelength to 532 nm, and the diffraction distance to 200 mm. The experimental results are shown in Table 3 and Figure 7. Table 3 shows the computational time consumption of the proposed method and the analytical polygon-based method, as well as the ratio of the time consumption of the analytical polygon-based method to that of the proposed method when computed on a CPU. Figure 7 shows the change in computation time with the number of triangles for both methods when computed on the CPU.

**Table 3.** Comparison of time consumption between two methods on a CPU.

The Number of Triangles	The Proposed Method	The Analytical Polygon-Based Method	Acceleration Ratio
4704	10.7 s	233.9 s	21.9
9408	22.2 s	483.9 s	21.8
14,112	35.1 s	709.4 s	20.2
18,816	48.5 s	947.2 s	19.5
23,520	60.3 s	1178.9 s	19.6
28,224	76.3 s	1411.7 s	18.5
32,928	92.3 s	1670.8 s	18.1



**Figure 7.** The curve depicting the relationship between the time taken and the number of triangles for two methods on the CPU.

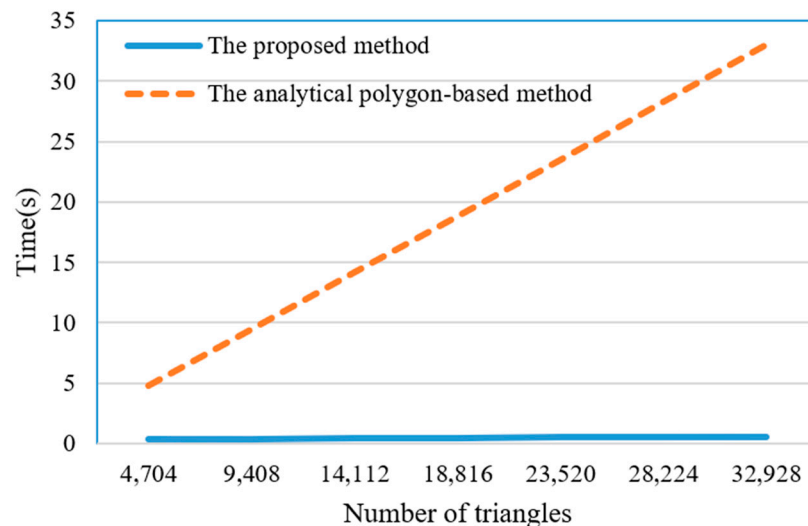
The experimental results show that the computational speed of the proposed method is improved by about 20 times compared to the analytic polygon-based method, which confirms that it has a very high computational efficiency. The analytic polygon-based method, in order to calculate the diffraction of tilted triangles, requires a three-dimensional affine transformation of each triangle and the solution of complex analytic equations. Therefore, the computational time of this method is greatly affected by the number of triangles. Meanwhile, the proposed method avoids the direct computation of the diffraction of tilted triangles by approximating the triangle mesh as a set of planar light sources parallel to the holographic plane through the layered projection of the triangles according to the depth of focus. In the proposed method, the projection calculation of triangles does not involve complex calculations. It has a very low computational complexity, so the step can be accomplished in a very short time. In addition, the diffraction between the layer and the holographic plane can be quickly calculated using angular spectral diffraction with the support of the fast Fourier transformation. Therefore, the method has high computational efficiency.

Furthermore, we can observe that the computational time consumption of both methods is linearly related to the number of triangles. This is easy to explain because both methods have a separate computation that needs to be performed for each triangle, and the computational effort for this part of the computation is not affected by factors such as the shape or size of the triangles, so the computational effort grows linearly with the number of triangles.

To further increase the computational speed of the proposed method, using GPU parallel computing was considered. Therefore, the parallelized program of the proposed method and the parallelized program of the resolved polygon method were implemented through CUDA C++ for comparison. The GPU used was the NVIDIA GeForce RTX 3060 Laptop GPU. The experimental platform, development environment, and experimental parameters were the same as those of the computational performance experiment on the CPU above. The experimental results are shown in Table 4 and Figure 8. Table 4 shows the computational time consumption of the proposed method and the analytical polygon-based method, as well as the ratio of the time consumption of the analytical polygon-based method to that of the proposed method when computed on a GPU. Figure 8 shows the change in computation time with the number of triangles for both methods when computed on a GPU.

**Table 4.** Comparison of time consumption between two methods on a GPU.

The Number of Triangles	The Proposed Method	The Analytical Polygon-Based Method	Acceleration Ratio
4704	0.33 s	4.83 s	14.64
9408	0.36 s	9.45 s	26.25
14,112	0.41 s	14.17 s	34.56
18,816	0.45 s	18.85 s	41.89
23,520	0.49 s	23.54 s	48.04
28,224	0.53 s	28.31 s	53.42
32,928	0.57 s	33.05 s	57.98



**Figure 8.** The curve depicting the relationship between the time taken and the number of triangles for two different methods on a GPU.

The proposed method experiences a more efficient computation on GPUs; generating a hologram of 30,000 triangles only takes about 0.5 s. Further, the acceleration ratio of the proposed method over the analytic polygon-based method gradually increases as the number of triangles increases. This indicates that the computational efficiency of the analytic polygon-based method is more sensitive to the number of triangles. In contrast, the proposed method is less affected by the number of triangles. This is mainly because

GPUs are good at simple operations such as addition and subtraction, while they are weak at complex operations such as calculating sine, cosine, and exponentials. There are a lot of complex exponential operations used in the analytic polygon-based method, which is the shortcoming of the GPU. In the proposed method, only the hierarchical projection is involved in the number of triangles. These calculations can be performed using basic mathematical calculations only and can be highly parallelized. Therefore, the proposed method can perform the computations extremely fast on GPUs. However, since this method is an approximation, it is more suitable for scenarios that require high computational speed but lower accuracy requirements.

It is worth noting that the computation speed of the proposed method is not solely dependent on the number of triangles in the input object. This is because the method includes calculations such as the fast Fourier transform for each layer, meaning that the computation speed is also related to the number of layers into which the three-dimensional object is divided. The depth of focus and the length of the object determine the number of layers, while the depth of focus is related to the wavelength of light, the LCOS (liquid crystal on silicon) size, and the diffraction distance, all physical parameters that will affect the calculation speed of the proposed method. However, the number of triangles remains the main factor affecting the calculation speed.

#### 4. Texture Mapping

Unlike the analytical polygon-based method, the proposed method in this paper utilizes FFT to perform angular spectrum diffraction, which consequently affords it substantial convenience when applied to texture-mapping tasks. In this section, we present the application of one of the most common texture-mapping methods, UV mapping, to the proposed method in this paper.

In 3D modeling software, it is possible to paint texture maps onto 3D models, which can then be exported as triangle meshes where each vertex of the triangles is assigned a set of UV coordinates. However, merely knowing the UV coordinates at the vertices of a triangle is insufficient for texturing purposes because we need to determine the texture information at any point within the triangle. Barycentric interpolation is one method that addresses this issue, enabling us to calculate the attribute values at any arbitrary point within a triangle based on the attributes of its three vertices. The barycentric interpolation can be expressed by the following equation:

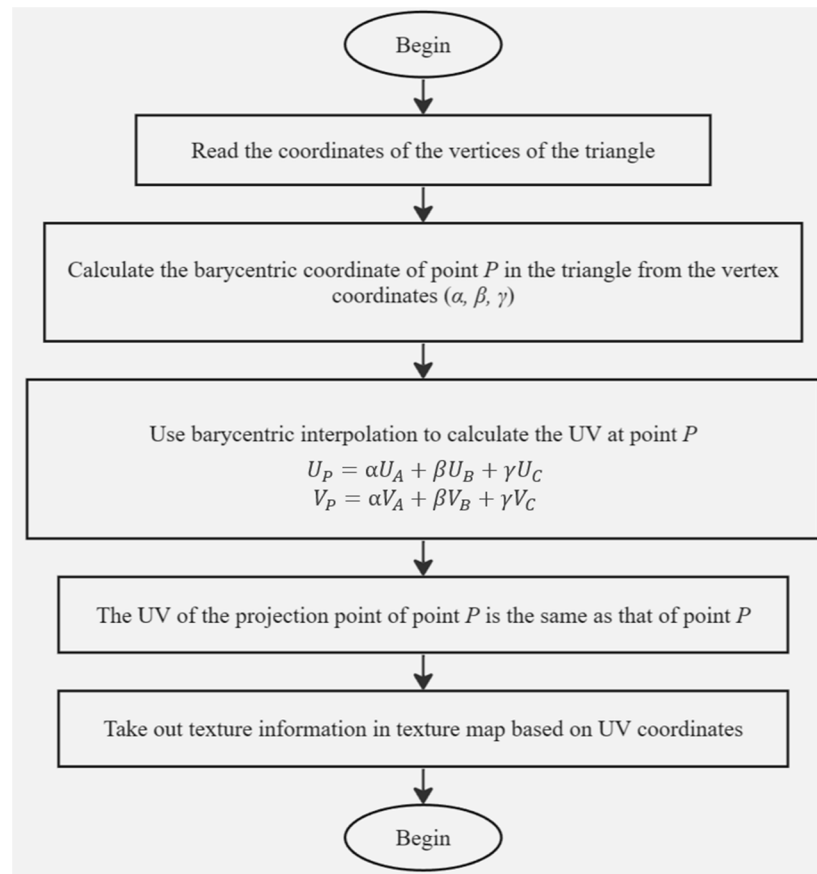
$$P = \alpha A + \beta B + \gamma C. \quad (5)$$

Here,  $A$ ,  $B$ , and  $C$  denote a specific property of the triangle's three vertices, which could be coordinates, UV coordinates, colors, or normals, among others;  $P$  represents the same property of a point within the triangle; and  $(\alpha, \beta, \gamma)$  denote the barycentric coordinates of that point. The following relationship exists between  $\alpha$ ,  $\beta$ , and  $\gamma$ :

$$\alpha + \beta + \gamma = 1, \quad (6)$$

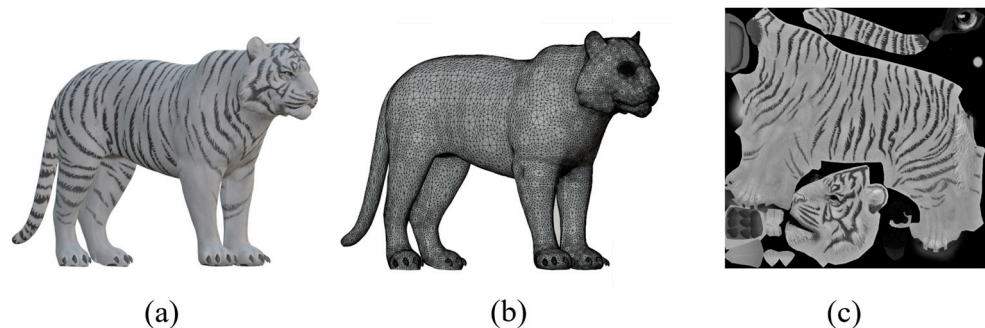
where,  $\alpha$ ,  $\beta$ , and  $\gamma$  can be calculated using the method detailed in [42], and hence, the UV coordinates of point  $P$  can be obtained through barycentric interpolation, as represented by Equation (5), utilizing the UV coordinates of points  $A$ ,  $B$ , and  $C$ . In the proposed method, the triangles need to be projected, but the projection does not change the UV coordinates of the points.

If you want to calculate the texture information of the projection point of a point  $P$  within a triangle, you can refer to the steps shown in Figure 9. In the figure,  $(U_P, V_P)$  denote the UV of point  $P$ , and  $(U_A, V_A)$ ,  $(U_B, V_B)$ , and  $(U_C, V_C)$  denote the UV of the three vertices of the triangle, respectively.

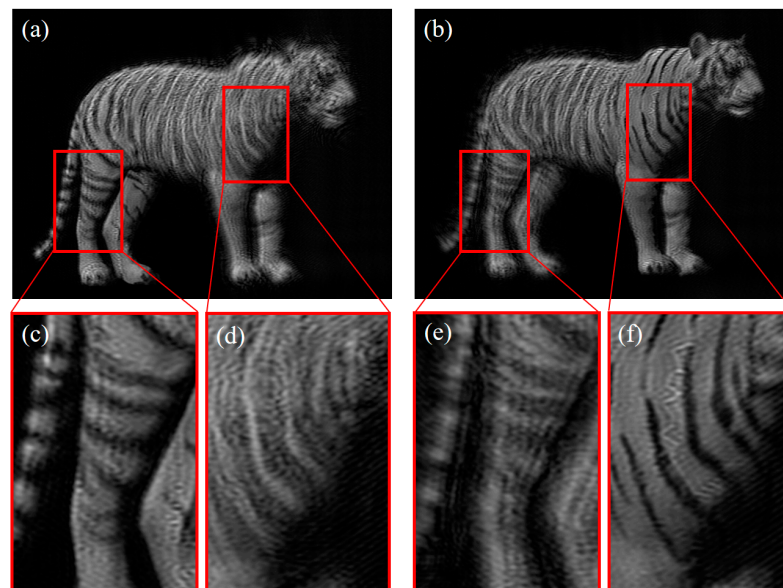


**Figure 9.** Process for calculating the texture of a projected point.

In order to inspect the effect of texture mapping applied in the proposed method, a hologram of the tiger model, as shown in Figure 10a, was generated. Figure 10a shows the tiger model with texture mapping applied, Figure 10b shows the triangle mesh of the model, and Figure 10c shows the texture map of the tiger model. In the experiment, the diffraction distance was set to 300 mm, the resolution of the hologram was  $1920 \times 1080$ , the pixel pitch was 0.045 mm, and the light wavelength was 532 nm. The generated holograms were numerically reconstructed at different distances to obtain the results shown in Figure 11. The 3D object carrying the texture is successfully reconstructed in Figure 11. In addition, the 3D sense of the hologram was verified by focusing on different places.

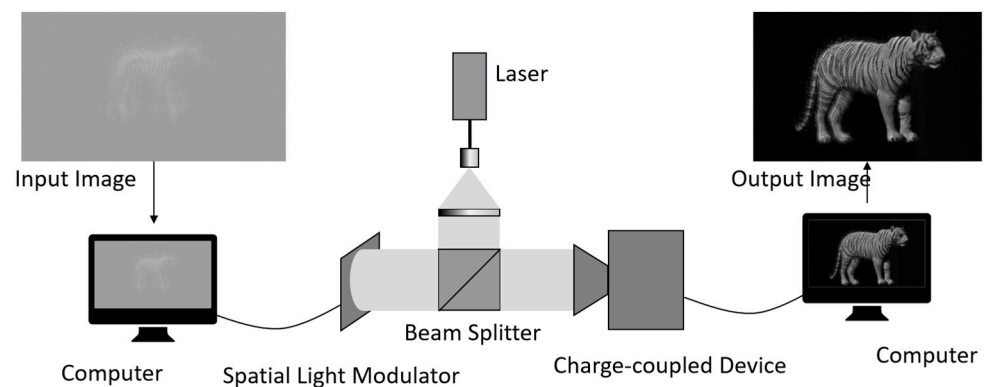


**Figure 10.** The 3D model of the tiger. (a) The original model with additional textures. (b) The triangle mesh of the model. (c) Texture map.

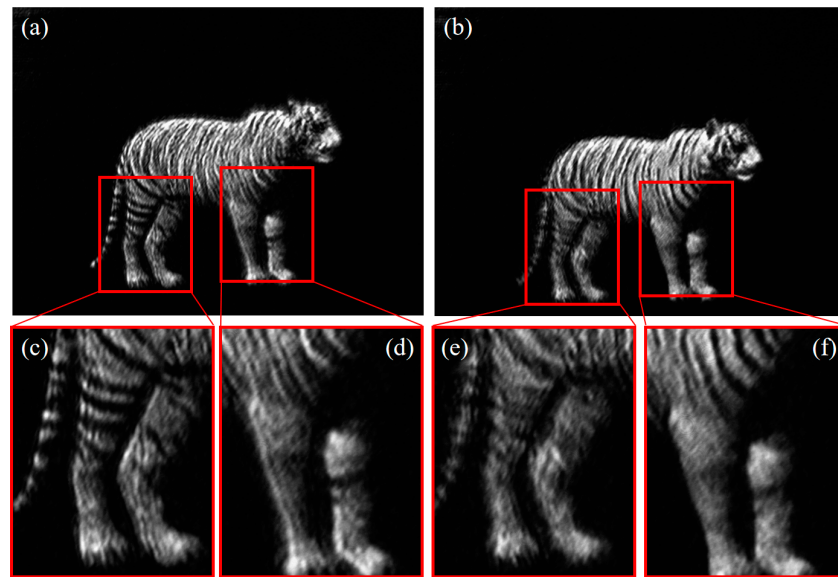


**Figure 11.** Numerical reconstruction of a hologram generated by the proposed method. (a) The numerical reconstruction focusing on the tiger's hind legs. (b) The numerical reconstruction focusing on the tiger's forelimbs. (c,d) and (e,f) are enlarged portions of the box in (a) and (b), respectively.

In order to test the effect of the textured holograms in real optical experiments, an optical path was constructed, as shown schematically in Figure 12. In order to be able to give the reconstructed image higher diffraction efficiency, the hologram was encoded as a phase hologram and loaded onto a phase LCOS. Some previous studies [43,44] can help us to better select a suitable LCOS as well as understand its principles. The phase LCOS used in the experiment had a resolution of  $1920 \times 1080$  with a pixel pitch of 0.045 mm, and the laser wavelength used was 532 nm. The reconstructed image was received through a CCD (charge-coupled device) and displayed on a computer connected to the CCD. Since the CCD used only senses light intensity and not color, grayscale results shown in Figure 13 were obtained.



**Figure 12.** Schematic diagram of the optical reconstruction experiment.



**Figure 13.** Optical reconstruction of holograms generated by the proposed method. (a) The optical reconstruction focusing on the tiger’s hind legs. (b) The optical reconstruction focusing on the tiger’s forelimbs. (c,d) and (e,f) are enlarged portions of the box in (a) and (b), respectively.

As can be seen, the optical reconstruction results also contain texture. In addition, the optical reconstruction image can focus on different places when reconstructed at different distances. Therefore, texture mapping can work effectively in the proposed method.

Fast generation of polygon-based texture holograms is a major challenge. Currently, one of the latest polygon-based texture hologram generation methods is an analytic-based method [45]. This method is more computationally efficient than the previous ones, but it is still heavily affected by the number of triangles. In this paper, we approximate the diffraction calculation of tilted triangles by a hierarchical projection based on the depth of focus, which also facilitates the mapping of the texture. Although the computation is fast enough to use traditional texture mapping methods, this is an approximation method, so the computation results are not as accurate as those based on the analytic method. It is more suitable for scenes that only need to satisfy human eye observation and do not require high imaging accuracy.

## 5. Conclusions

In this paper, the diffraction calculation of inclined triangles was approximated to the diffraction calculation between parallel planes via hierarchical projection based on the physical properties of the depth of focus. Therefore, the speed of generating holograms using the polygon-based method was improved. The feasibility of the proposed method was verified by comparing the computed diffraction distributions with those computed by the analytic polygon method. The computational efficiency of the proposed method was then verified by comparing the computational performance with that of the analytic polygon method on a CPU and GPU. Finally, texture mapping was applied to the proposed method to generate holograms carrying complex textures, and the three-dimensionality of holograms was verified by numerical and optical reconstruction.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app14125109/s1>.

**Author Contributions:** Conceptualization, J.G. and J.L.; methodology, X.M.; software, X.M.; validation, X.M.; formal analysis, X.M. and J.G.; investigation, Q.S.; resources, Q.S.; data curation, X.M.; writing—original draft preparation, X.M.; writing—review and editing, J.G.; visualization, X.M.; supervision, J.G.; project administration, J.G.; funding acquisition, J.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant numbers 62065010, 62165007, and 61565011.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The image results in the verification experiment can be found in Supplementary Materials, and the remaining images and source code are available upon reasonable request.

**Acknowledgments:** Many thanks to Xi'an Zhongke Micro Star Photoelectric Technology Co., Ltd. for lending the spatial light modulator.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Jin, Z.; Ren, Q.; Chen, T.; Dai, Z.; Shu, F.; Fang, B.; Hong, Z.; Shen, C.; Mei, S. Vision transformer empowered physics-driven deep learning for omnidirectional three-dimensional holography. *Opt. Express* **2024**, *32*, 14394–14404. [[CrossRef](#)] [[PubMed](#)]
2. Chang, C.; Bang, K.; Wetzstein, G.; Lee, B.; Gao, L. Toward the next-generation VR/AR optics: A review of holographic near-eye displays from a human-centric perspective. *Optica* **2020**, *7*, 1563–1578. [[CrossRef](#)]
3. Gopakumar, M.; Kim, J.; Choi, S.; Peng, Y.; Wetzstein, G. Unfiltered holography: Optimizing high diffraction orders without optical filtering for compact holographic displays. *Opt. Lett.* **2021**, *46*, 5822–5825. [[CrossRef](#)] [[PubMed](#)]
4. Shigematsu, O.; Naruse, M.; Horisaki, R. Computer-generated holography with ordinary display. *Opt. Lett.* **2024**, *49*, 1876–1879. [[CrossRef](#)] [[PubMed](#)]
5. Pi, D.; Liu, J.; Wang, Y. Review of computer-generated hologram algorithms for color dynamic holographic three-dimensional display. *Light Sci. Appl.* **2022**, *11*, 231. [[CrossRef](#)] [[PubMed](#)]
6. Liu, J.; Cao, L.; Stoykova, E.; Ferraro, P.; Memmolo, P.; Blanche, P.-A. Digital Holography and 3D Imaging 2020: Introduction to the feature issue. *Appl. Opt.* **2021**, *60*, DH1–DH2. [[CrossRef](#)]
7. Zhang, Y.; Zhang, M.; Liu, K.; He, Z.; Cao, L. Progress of the Computer-Generated Holography Based on Deep Learning. *Appl. Sci.* **2022**, *12*, 8568. [[CrossRef](#)]
8. Xiong, J.; Yin, K.; Li, K.; Wu, S.-T. Holographic optical elements for augmented reality: Principles, present status, and future perspectives. *Adv. Photonics Res.* **2021**, *2*, 2000049. [[CrossRef](#)]
9. Lee, J.; Jeong, J.; Cho, J.; Yoo, D.; Lee, B. Deep neural network for multi-depth hologram generation and its training strategy. *Opt. Express* **2020**, *28*, 27137–27154. [[CrossRef](#)] [[PubMed](#)]
10. Xu, X.; Wang, X.; Luo, W.; Wang, H.; Sun, Y. Efficient Computer-Generated Holography Based on Mixed Linear Convolutional Neural Networks. *Appl. Sci.* **2022**, *12*, 4177. [[CrossRef](#)]
11. Blanche, P.-A. Holography, and the future of 3D display. *Light Adv. Manuf.* **2021**, *2*, 446–459. [[CrossRef](#)]
12. Zhang, Z.; Liu, J.; Jia, J.; Li, X.; Han, J.; Hu, B.; Wang, Y. Tunable nonuniform sampling method for fast calculation and intensity modulation in 3D dynamic holographic display. *Opt. Lett.* **2013**, *38*, 2676–2679. [[CrossRef](#)] [[PubMed](#)]
13. Pi, D.; Liu, J.; Han, Y.; Khalid, A.U.R.; Yu, S. Simple and effective calculation method for computer-generated hologram based on non-uniform sampling using look-up-table. *Opt. Express* **2019**, *27*, 37337–37348. [[CrossRef](#)] [[PubMed](#)]
14. Lucente, M.E. Interactive computation of holograms using a look-up table. *J. Electron. Imaging* **1993**, *2*, 28–34. [[CrossRef](#)]
15. Shimobaba, T.; Masuda, N.; Ito, T. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane. *Opt. Lett.* **2009**, *34*, 3133–3135. [[CrossRef](#)] [[PubMed](#)]
16. Kim, S.-C.; Kim, E.-S. Effective generation of digital holograms of three-dimensional objects using a novel look-up table method. *Appl. Opt.* **2008**, *47*, D55–D62. [[CrossRef](#)] [[PubMed](#)]
17. Jia, J.; Wang, Y.; Liu, J.; Li, X.; Pan, Y.; Sun, Z.; Zhang, B.; Zhao, Q.; Jiang, W. Reducing the memory usage for effective computer-generated hologram calculation using compressed look-up table in full-color holographic display. *Appl. Opt.* **2013**, *52*, 1404–1412. [[CrossRef](#)] [[PubMed](#)]
18. Pi, D.; Liu, J.; Kang, R.; Zhang, Z.; Han, Y. Reducing the memory usage of computer-generated hologram calculation using accurate high-compressed look-up-table method in color 3D holographic display. *Opt. Express* **2019**, *27*, 28410–28422. [[CrossRef](#)] [[PubMed](#)]
19. Jiao, S.; Zhuang, Z.; Zou, W. Fast computer generated hologram calculation with a mini look-up table incorporated with radial symmetric interpolation. *Opt. Express* **2017**, *25*, 112–123. [[CrossRef](#)] [[PubMed](#)]
20. Hasegawa, N.; Shimobaba, T.; Kakue, T.; Ito, T. Acceleration of hologram generation by optimizing the arrangement of wavefront recording planes. *Appl. Opt.* **2017**, *56*, A97–A103. [[CrossRef](#)]
21. Arai, D.; Shimobaba, T.; Murano, K.; Endo, Y.; Hirayama, R.; Hiyama, D.; Kakue, T.; Ito, T. Acceleration of computer-generated holograms using tilted wavefront recording plane method. *Opt. Express* **2015**, *23*, 1740–1747. [[CrossRef](#)] [[PubMed](#)]
22. Zhang, Y.; Fan, H.; Wang, F.; Gu, X.; Qian, X.; Poon, T.-C. Polygon-based computer-generated holography: A review of fundamentals and recent progress [Invited]. *Appl. Opt.* **2022**, *61*, B363–B374. [[CrossRef](#)] [[PubMed](#)]

23. Pan, Y.; Wang, Y.; Liu, J.; Li, X.; Jia, J.; Zhang, Z. Analytical brightness compensation algorithm for traditional polygon-based method in computer-generated holography. *Appl. Opt.* **2013**, *52*, 4391–4399. [[CrossRef](#)] [[PubMed](#)]
24. Wang, F.; Shiomi, H.; Ito, T.; Kakue, T.; Shimobaba, T. Fully analytic shading model with specular reflections for polygon-based hologram. *Opt. Lasers Eng.* **2023**, *160*, 107235. [[CrossRef](#)]
25. Kim, H.; Hahn, J.; Lee, B. Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography. *Appl. Opt.* **2008**, *47*, D117–D127. [[CrossRef](#)] [[PubMed](#)]
26. Ahrenberg, L.; Benzie, P.; Magnor, M.; Watson, J. Computer generated holograms from three dimensional meshes using an analytic light transport model. *Appl. Opt.* **2008**, *47*, 1567–1574. [[CrossRef](#)] [[PubMed](#)]
27. Zhang, Y.-P.; Wang, F.; Poon, T.-C.; Fan, S.; Xu, W. Fast generation of full analytical polygon-based computer-generated holograms. *Opt. Express* **2018**, *26*, 19206–19224. [[CrossRef](#)] [[PubMed](#)]
28. Wang, F.; Shimobaba, T.; Zhang, Y.; Kakue, T.; Ito, T. Acceleration of polygon-based computer-generated holograms using look-up tables and reduction of the table size via principal component analysis. *Opt. Express* **2021**, *29*, 35442–35455. [[CrossRef](#)] [[PubMed](#)]
29. Liu, Y.-Z.; Dong, J.-W.; Pu, Y.-Y.; Chen, B.-C.; He, H.-X.; Wang, H.-Z. High-speed full analytical holographic computations for true-life scenes. *Opt. Express* **2010**, *18*, 3345–3351. [[CrossRef](#)] [[PubMed](#)]
30. Sakata, H.; Sakamoto, Y. Fast computation method for a Fresnel hologram using three-dimensional affine transformations in real space. *Appl. Opt.* **2009**, *48*, H212–H221. [[CrossRef](#)] [[PubMed](#)]
31. Fan, H.; Zhang, B.; Zhang, Y.; Wang, F.; Qin, W.; Fu, Q.; Poon, T.-C. Fast 3D Analytical Affine Transformation for Polygon-Based Computer-Generated Holograms. *Appl. Sci.* **2022**, *12*, 6873. [[CrossRef](#)]
32. Zhao, Y.; Cao, L.; Zhang, H.; Kong, D.; Jin, G. Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method. *Opt. Express* **2015**, *23*, 25440–25449. [[CrossRef](#)] [[PubMed](#)]
33. Chang, C.L.; Xia, J.; Lei, W. One step hologram calculation for multi-plane objects based on nonuniform sampling (Invited Paper). *Chin. Opt. Lett.* **2014**, *12*, 060020. [[CrossRef](#)]
34. Jia, J.; Si, J.; Chu, D. Fast two-step layer-based method for computer generated hologram using sub-sparse 2D fast Fourier transform. *Opt. Express* **2018**, *26*, 17487–17497. [[CrossRef](#)] [[PubMed](#)]
35. Chen, J.-S.; Chu, D.P. Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications. *Opt. Express* **2015**, *23*, 18143–18155. [[CrossRef](#)] [[PubMed](#)]
36. Wei, J.; Wei, C.; Ma, H.; Pi, D.; Li, H.; Liu, X.; Wang, Y.; Liu, J. Convolutional symmetric compressed look-up-table method for 360° dynamic color 3D holographic display. *Opt. Express* **2023**, *31*, 28716–28733. [[CrossRef](#)]
37. Wang, F.; Blinder, D.; Ito, T.; Shimobaba, T. Wavefront recording plane-like method for polygon-based holograms. *Opt. Express* **2023**, *31*, 1224–1233. [[CrossRef](#)] [[PubMed](#)]
38. Liu, Q.; Liu, Q.; Chen, J.; Chen, J.; Qiu, B.; Qiu, B.; Wang, Y.; Wang, Y.; Liu, J.; Liu, J. DCPNet: A dual-channel parallel deep neural network for high quality computer-generated holography. *Opt. Express* **2023**, *31*, 35908–35921. [[CrossRef](#)] [[PubMed](#)]
39. Lee, W.; Im, D.; Paek, J.; Hahn, J.; Kim, H. Semi-analytic texturing algorithm for polygon computer-generated holograms. *Opt. Express* **2014**, *22*, 31180–31191. [[CrossRef](#)] [[PubMed](#)]
40. Li, J.; Tu, H.-Y.; Yeh, W.-C.; Gui, J.; Cheng, C.-J. Holographic three-dimensional display and hologram calculation based on liquid crystal on silicon device [Invited]. *Appl. Opt.* **2014**, *53*, G222–G231. [[CrossRef](#)] [[PubMed](#)]
41. Pan, Y.; Wang, Y.; Liu, J.; Li, X.; Jia, J. Fast polygon-based method for calculating computer-generated holograms in three-dimensional display. *Appl. Opt.* **2013**, *52*, A290–A299. [[CrossRef](#)] [[PubMed](#)]
42. Peter, S.; Marschner, S.R. *Fundamentals of Computer Graphics*; AK Peters: Natick, MA, USA, 2005.
43. Huang, Y.; Liao, E.; Chen, R.; Wu, S.-T. Liquid-Crystal-on-Silicon for Augmented Reality Displays. *Appl. Sci.* **2018**, *8*, 2366. [[CrossRef](#)]
44. Yin, K.; Hsiang, E.-L.; Zou, J.; Li, Y.; Yang, Z.; Yang, Q.; Lai, P.-C.; Lin, C.-L.; Wu, S.-T. Advanced liquid crystal devices for augmented reality and virtual reality displays: Principles and applications. *Light Sci. Appl.* **2022**, *11*, 161. [[CrossRef](#)] [[PubMed](#)]
45. Qin, W.; Fu, Q.; Zhang, Y.; Zhang, B.; Wang, P.; Poon, T.-C.; Gu, X. Rendering of 3D scenes in analytical polygon-based computer holography with texture mapping. *J. Opt. Soc. Am. A* **2024**, *41*, A32–A39. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.