A fast analytical algorithm for generating CGH of 3D scene

Yuanzhi Liu
Jianwen Dong
Yiying Pu
Bingchu Chen
Hexiang He
Zilan Deng
Hezhou Wang
A fast analytical algorithm for generating CGH of 3D scene

Yuanzhi Liu, Jianwen Dong*, Yiying Pu, Bingchu Chen, Hexiang He, Zilan Deng and Hezhou Wang**

State Key Laboratory of Optoelectronic Materials and Technologies, Sun Yat-sen (Zhongshan) University, Guangzhou 510275, China

ABSTRACT

We present a set of analytical formula on describing the diffraction field of the three dimensional (3D) triangular-mesh-based model. The advantage of the proposed method is that it can avoid using the numerical algorithm -- Fast Fourier Transform, which leads to a depth-of-field limitation by the Whittaker-Shannon sampling theorem. We employ the proposed method to generate the hologram of 3D texture model derived from the real scene or 3D design software. In order to further increase the computation speed, we have rendered a real scene by employing the GPU platform. Our homemade GPU algorithm performs hundreds of times faster than those of CPU. As we developed a general phase adjustment technique for polygon-based algorithm, the holographic reconstructed scenes possess high performance.

Keywords: Computer generated hologram, 3D display, analytical algorithm, graphic processing units

1. INTRODUCTION

Three-dimensional display has drawn more and more attentions [1]. Computer generated hologram (CGH) is one of the promising technologies of 3D display [2, 3]. It can use the digital 3D object data stored in the computer to produce the wavefronts, which create the most accurate depth cues of 3D objects [4, 5].

However, one of the problems in holographic encoding for 3D object is time consuming. Several fast algorithms have been performed to reduce the calculation time [6-10]. These methods consider that the 3D models are composed of individual self-luminous points. Recently some researchers have developed other fast algorithms that regard 3D objects as a combination of spatial planar segments. They use the fast-Fourier-transform (FFT) for calculating the angular spectrum [11-16]. Researchers also studied the hardware to accelerate the computation. The research group in MIT Media Lab developed a special purpose computational board for holographic rendering [17-19]. At that time, the implementation of this technique was 50-times faster than workstation. A group at the Department of Medical System Engineering at Chiba University developed another special purpose hardware architecture, HORN, since 1992 [20]. They also achieved a remarkably performance. Recently some researchers presented parallel algorithms for CGH based on commodity graphic processing units (GPUs) [21-24].

In this paper, we propose a novel algorithm to dramatically decrease the computation time. This approach is according to our analytical theory, which describes the diffraction field distributions of the 3D scene. Since FFT is not used here, our approach can get rid of the short depth-of-field limitation [25-29]. Further more, we propose an efficacious phase adjustment for polygon-based encoding to remove the visible mesh edges. Hence, a 3D object with smooth reconstructed surface can be accomplished. The algorithm is implemented on GPU using CUDA (Compute Unified Device Architecture) technique [30], and performs hundreds of times faster than those on CPU.

2. THREE DIMENSIONAL INFORMATION PROCESS

2.1 Three dimensional data acquisition

Image-based modeling and photogrammetry approach is an inexpensive and flexible way to record a real scene. Here we adopt the software- ImageModeler® [31] to obtain the 3D data from ordinary 2D digital photos.

* Corresponding author: Jianwen DONG, dongjwen@mail.sysu.edu.cn;
**Corresponding author: Hezhou WANG, stswzh@mail.sysu.edu.cn
Fig. 1 shows the process of 3D data acquisition. Firstly, we take two digital photos from two arbitrary different aspects of a real scene, and then mark the corresponding points in these two photos (as the blue points shown in Fig. 1). The software will calibrate the coordinate automatically. After that, the chess and box model are built, and photorealistic texture maps for the model are automatically extracted from the source images. Till then, we obtain the textured model of 3D scene.

Figure 1. Different aspects digital photos for the 3D scene modeling

2.2 Analytical theory

To deduce the analytical theory for the diffraction field of 3D objects, we have to convert the 3D model to triangle meshes first. If the light field diffracted by each triangle can be calculated, the whole objects field distribution on the hologram can be achieved through superposition.

We consider a diffraction of monochromatic plane wave by a spatial finite triangular-shape aperture within an infinite opaque screen. $O_o(x_i, y_i)$ is used to denote this screen function. Local coordinate system and global coordinate system are set as $(x_i, y_i, z_i)$ and $(x_g, y_g, z_g)$ respectively. A rotation matrix $R'$ connects these two coordinate systems, whose element $i, j$ is expressed as $r_{ij}$. The center of mass of the triangle in global coordinate system is denoted as a group of constant $(x_c, y_c, z_c)$. Hologram is located at $z_g = 0$ and perpendicular to $z = z_g$. The plane wave propagates along the axis $z = z_g$, whose complex amplitude in local coordinate system is expressed as \( \exp[i2\pi(r_{11}x_i + r_{12}y_i + z_c)/\lambda] \). Through the Rayleigh-Sommerfeld diffraction integral [5], the diffraction field on the hologram can be obtained,

\[
O_h(x_h, y_h) = (1/i\lambda) \iint O_o(x_i, y_i) \exp[i2\pi(r_{11}x_i + r_{12}y_i + z_c)/\lambda] \left[ \exp(ikr)/r \right] dx_idy_i
\]

where $k$ is the wave number and $\lambda$ is the wavelength; $r = \sqrt{(x_h-x_g)^2 + (y_h-y_g)^2 + (z_h-z_g)^2}$ indicates the distance between the hologram pixels and points within the triangle. $(x_{ch}, y_{ch})$ denotes the hologram coordinate system. If the assumption

\[
(x_h-x_c)^2 + (y_h-y_c)^2 + z_c^2 \gg (x_h-x_c)^2, \text{and} (y_h-y_c)^2
\]

is satisfied, then $r$ can be expanded into:

\[
r = r_0 - \frac{x^2 + y^2}{2r_0} - \left( r_{11}(x_h-x_c) + r_{12}(y_h-y_c) - r_{13}z_c \right) x_i + \left( r_{15}(x_h-x_c) + r_{16}(y_h-y_c) - r_{17}z_c \right) y_i
\]
where \( r_0 = \sqrt{\left( x_H - x_c \right)^2 + \left( y_H - y_c \right)^2 + z_c^2} \) is the distance between the triangular mass center and the hologram pixel. In fact, Eq. (2) can be deemed to a modified paraxial approximation. The optical axis here is the perpendicular line from the mass center of triangle to the hologram plane. It requires the linear scale of each triangle should be much less than the distance between the mass center of triangle and the hologram plane. Because the scene usually contains numbers of triangles and each triangle is small, this restriction can be easily satisfied. In this way, Eq. (1) becomes,

\[
O_H(x_H, y_H) = \exp \left[ \frac{ik(\omega + \nu_0)}{i\lambda r_0} \right] \iint O_o(x_0, y_0) Q(x, y) \exp \left[ -\frac{2\pi}{\lambda r_0} \left( x_H x_t + y_H y_t \right) \right] dx_0 dy_0,
\]

where \( Q(x, y) = \exp \left[ ik \left( \frac{x^2 + y^2}{2r_0} \right) \right] \) is the quadratic phase factor; and \( x_H = r_{11}(x_H - x_c) + r_{21}(y_H - y_c) - r_{31}z_c - r_{10}, \), \( y_H = r_{12}(x_H - x_c) + r_{22}(y_H - y_c) - r_{32}z_c - r_{10} \). Note that Eq. (4) is similar in form to the well known Fresnel diffraction between two parallel planes [5]. As we know the Fresnel diffraction formula can be simplified to Fraunhofer form when an infinity aperture lens set immediately in front of the object [5]. Numerous experiment results of Fraunhofer holography reveal that the quadratic phase factor within Fresnel integral does not affect the reconstruction performance [5]. So in the CGH, the quadratic phase factor \( Q(x, y) \) in Eq. (4) can also be discarded but would not affect the reconstruction result. Thus, Eq. (4) will be simplified as

\[
O_H(x_H, y_H) = \exp \left[ \frac{ik(\omega + \nu_0)}{i\lambda r_0} \right] F \left[ O_o(x_0, y_0) \right]
\]

where \( F \left[ O_o(x_0, y_0) \right] \) is the Fourier transform of the triangle, evaluated at frequencies \( \left( \frac{x_H}{\lambda r_0}, \frac{y_H}{\lambda r_0} \right) \). Although the triangle is arbitrary, we can still derive an analytical form from the Fourier transform of a special triangle with fixed vertices, with the help of affine transformation. One of the options of special triangle can be found in Ref. 15 or Ref. 16. Then we can compute the diffraction field of an arbitrary tilted triangle analytically. This is a very important result, as it allows us to generate the hologram directly pixel by pixel without the need for numerical algorithm FFT, which will limit the depth-of-field because of the Whittaker-Shannon sampling restriction [5, 25-29]. And our theory can also be easily performed on the parallel computation platform. Finally, the whole objects field can be obtained through the linear superposition of the diffraction field per triangle.

### 2.3 Parallel algorithm based on GPU

A parallel coding based on GPU is considered to decrease the computation time in an inexpensive way. We adopt the CUDA compiler as a programming environment, which is developed by NVIDIA. It can compile a C-like language source code for parallel application on the GPU [30]. One of the problems should be considered when using the parallel calculation is the data communication. Since we obtain a full analytic expression of the object field, each sampling points on the hologram can be calculated individually. It means that we can suppress the time of data communication furthest. Fig. 2 indicates our parallel CUDA algorithm. M blocks are set and each block contains N threads on the GPU for calculation. M, N depends on the number of data, sampling points on the hologram and the capability of the chip. When the 3D data sent into the global memory we use N threads in each block to read the data of N triangles. Each thread computes independently at the same time. And then the feature data of each triangle, such as rotation and affine transform matrix are saved into the shared memory. To access the shared memory is much faster to access the global memory. Then each thread in the same block reads the same feature data of the same triangle. The all M by N threads independently compute each pixel of the whole hologram simultaneously. After that, we begin another N triangles calculation loop. For the final holographic data exist in the GPU, we use the OpenGL to directly display the hologram. If there are P triangles and the hologram contains I by J pixel, it needs P*I*J times loop in the usual CPU serial operation. In our algorithm, however, it only needs P/N times loop.
The Dell server PE2900 with Quad-Core Intel Xeon Pro X5460 2X3.16GHz, and GPU chip ‘GeForce GTX 285’ made by NVIDIA are employed for comparison. Fig. 3 indicates the computation time between parallel algorithm based on CUDA in GPU, parallel algorithm based on MPI in CPU, and serial algorithm in CPU. The resolution of the hologram calculated here is 1024 by 768. As can be seen, while the triangles are not too many, the inherent transmission cost most of the time but CUDA algorithm still performance one hundred times faster than that serial algorithm in CPU. When the number of triangles becomes huge, the accelerated effect by CUDA is encouraging. It runs more than several hundred times faster than that serial algorithm in CPU.

![Figure 2. Schematic diagram of the algorithm](image1)

![Figure 3. Calculation times comparison between CPU and GPU](image2)
2.4 Shading and occlusion

Some other specific should be taken into account to achieve a better performance. For avoiding some unexpected shading, we adopt a modified Lambert brightness as introduced in [14]. The real scene usually contains more than one object, where the occlusion becomes much more important and troublesome. In this paper we use a simple geometric-like occlusion method- each triangle is projected on the hologram plane and then sorted by the distance of each triangle’s mass center. The shielded rear triangles are discarded. This method is quite well for our situation because of the narrow bandwidth of SLM nowadays.

2.5 Phase adjustment

Until now, if we use the analytic formula directly, some ugly edges of the mesh will be visible in the reconstruction [15, 16]. That is because of the interference between two diffraction fields of the same edge- it belongs to two conjoint triangles. We propose a simple and useful approach here to remove these visible edges- Phase Adjustment. What we do is to adjust the $z_c$ of each triangle to an integral multiple of wavelength. For this adjustment is so slight that it will not affect the macroscopical reconstruction positions of triangles. However it can effectively make the visible edge disappear.

We calculate a simple case, a teapot, to validate the effect of phase adjustment. Fig. 4 shows the comparison of numerical reconstruction. It can be seen that the phase adjustment technique removes the visible edges effectively.

![Figure 4. Reconstruction comparison](image)

3. RESULTS

For proving the algorithm, numerical reconstruction of the 3D real scene is shown in Fig. 5. Figs. 5 (a-c) show three different views of the reconstructed scene to demonstrate the enhancement on the realistic effect. The reconstructed views in Figs. 5 (a) and (c) are consistent with the original photos in Fig. 1, respectively. Further more, we can reconstruct other aspects besides the two original ones, as shown in Fig. 5 (b). The occlusion effects between chesses cause a strong depth sensation. For instance, the king in front is appeared to be on the right side of the right bishop in Fig. 5 (a); when the scene rotates a slight angle, the king comes to the left side of the right bishop [Fig 5 (b)]; when it rotates more, the king becomes to shield the back pawn [Fig 5 (c)]. It demonstrates our occlusion methods works quite well.

On the other hand, Fig. 5 (a) focus on the back pawn. It can be seen that the king and the front part of the chessboard blur out when the pawn and the rear part of the chessboard is clear. While focus on the right bishop [Fig 5 (b)], which lies between the front and rear pieces, the pawns become indistinct. When focus on the king as shown in Fig. 5(c), the king and the front part of the chessboard is clear. Contrarily, the pawns are defocus blur. This defocus effect further enhances the depth sensation, and further demonstrates that the 3D true-life scene is faithfully reconstructed.

As has been shown in Fig 4 (b), our phase adjustment technique removes the artificial edges mentioned in [15, 16]. The reconstructed scene performs a well solid effect. Meanwhile, the objects retain their positions and are not affected by the phase adjustment. The shading technique used here also adds to the impression of surface curvature.
4. CONCLUSION

An analytical algorithm for fast generating hologram of 3D real scenes is proposed. In this method, each sampling point for the object wave field across the hologram plane can be exactly calculated. Hence, it can free from the distance limitation caused by FFT. We also propose a phase adjustment technique to remove the artifacts visual edges. This technique is suitable for all polygon-based algorithms. Finally we employ the GPU to accelerate the calculation and obtain several hundred times faster than the serial algorithm based on CPU.

Figure 5. Different aspects digital photos for the 3D scene modeling
5. ACKNOWLEDGE

This work is supported by the National Natural Science Foundation of China (10874250, 10674183, 10804131), and Ph.D Degrees Foundation of Education Ministry of China (20060558068).

REFERENCES