Digital HPO Hologram Rendering Pipeline

M. Janda†[1], I. Hanák‡[1], V. Skala§[1]

[1]CGDV, University of West Bohemia, Czech Republic

Abstract

This paper describes a rendering pipeline for digital hologram synthesis. The pipeline is capable of handling triangle meshes, directional light sources, texture coordinates, and advanced illumination models. Due to the huge computational requirements of hologram synthesis only the HPO holograms are considered.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Raytracing; Hidden line/surface removal; Color, shading, shadowing, and texture

1. Introduction

Holographic displays are very close to their practical utilization. However, the holographic displays need holograms as an input and holograms have to be either obtained optically or synthesized numerically. The numerical synthesis is addressed in this paper.

The holograms are complex diffraction gratings with very large spatial frequencies comparable with a frequency of diffracted light, which is in a case of visible light approximately 5 MHz. Capturing of such frequency without aliasing needs a lots of samples. This fact makes synthesis very computationally intensive and must be therefore implemented as efficiently as possible.

The presented approach is fast because it avoids costly functions, e.g. the square root function, and performs preprocessing wherever it is possible to speed up the synthesis. Although it is fast it is also capable of handling textures and advanced illumination models. It consumes exactly the same input as the classical rendering pipeline and therefore it can be seamlessly integrated into the existing rendering systems.

2. Digital Holography Essentials

At the beginning, there was a light wave, which can be described by the Equation 1. There are two terms in the equation, the spatial one: $\exp\left[-i\varphi\left(\vec{x}\right)\right]$ and the temporal one: $\exp\left(i\omega t\right)$. Only the spatial one is the important one for the synthesis. The temporal one can be omitted without consequences because the temporally coherent light is assumed.

$$\tilde{u}\left(\vec{x},t\right) = A\left(\vec{x}\right)\exp\left[-i\varphi\left(\vec{x}\right)\right]\exp\left(i\omega t\right) = \tilde{u}\left(r\right)\exp\left(i\omega t\right) \quad (1)$$

The spatial term together with an amplitude $A\left(\vec{x}\right)$ is called the complex amplitude. Each complex amplitude is, obviously, determined by the amplitude and phase and can be therefore written as a complex number, which is more convenient.

The whole holography stands on a phenomenon of interference. The interference occurs if two and more light waves are superposed. The complex amplitude of the resulting wave is obtained as a simple summation of complex amplitudes of the original waves. An optical intensity of the composed wave is computed using the Equation 2. Only the last two terms of the Equation 2 are computed in digital holography, see [Luc94].

$$I = |\tilde{u}_R + \tilde{u}_S|^2 = |\tilde{u}_R|^2 + |\tilde{u}_S|^2 + \tilde{u}_R\tilde{u}_S^* + \tilde{u}_R^*\tilde{u}_S \quad (2)$$

The complex amplitude of a wavefront emerging from a scene has to be computed at each sampled point at a hologram frame. Once the complex amplitude is known, the final hologram is obtained using the Equation 3 where $\tilde{u}_S$ is a scene's complex amplitude and $\tilde{u}_R^*$ is a reference beam's complex amplitude.

† supported by MSMT Czech Rep. No. LC06008
‡ supported by 6FP NoE 3DTV project No. 511568
§ supported by MSMT Czech Rep. No. 1P04LA240

$$I_B = 2\Re\left\{\tilde{u}_R^* \tilde{u}_S\right\} = 2\left|\tilde{u}_R\right|\left|\tilde{u}_S\right|\cos\left(\varphi_R - \varphi_S\right) \quad (3)$$

The last missing item to address is a computation of the complex amplitude of a wavefront emerging from a scene. Unfortunately, it is the hardest one as well because a simple analytical description of the wavefront is possible only for a point light source. The spherical wavefront of a point light source is governed by the Equation 4.

$$\tilde{u}\left(x,y,z\right) = \frac{U}{r}\exp\left(i\varphi\right)\exp\left(ikr\right) \quad (4)$$

U relates to the energy radiated by a point light source
$k = 2\pi/\lambda$
$r = \sqrt{x^2+y^2+z^2}$

Utilizing the spherical wavefront means that surfaces in a scene have to be populated with a lot of point light sources. The final wavefront at a hologram plane is then obtained as a summation of all spherical wavefronts. The difficulty with this is that the amount of light sources has to be huge to obtain reasonably accurate result. Another problem is that the wavefront of some sources may be blocked by some occluding geometry so the summation has to be augmented with some visibility evaluator.

The horizontal parallax only (HPO) holograms were introduced to reduce the amount of operands involved in the summation, see [Luc94]. Instead of considering every source in a scene only those having the same Y coordinate as the currently computed hologram sample point are considered. Moreover, sampling rate in a vertical direction can be reduced.

For more details about holography see [Har96] and [Goo05]

3. Previous Work

The rendering method introduced in [JHS06] consists of several steps:

- Obtaining a slice of scene's geometry
- Removing backfacing parts
- Sampling the semicircle above each hologram sample
- Choosing and accumulating a complex amplitude of the closest scene's point from each sampled direction
- Computing the interference pattern

The whole process of synthesis is depicted in the Figure 1. From each sampled point rays are cast into a scene and the closest points found are accumulated to obtain a guess of the final wavefront.

In this paper a description of the enhancements of this method are presented. The enhanced method is capable of handling textures and advanced illumination models so a complete rendering pipeline is achieved. A description of all acceleration features is also included.

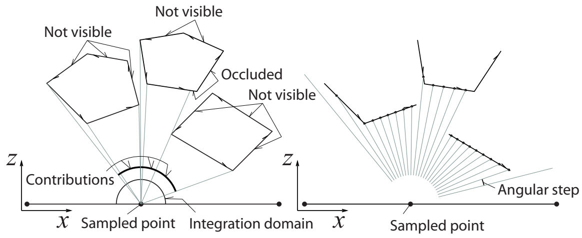

Figure 1: *Raycasting method for wavefront estimation.*

4. Rendering Pipeline

All relevant stages of the pipeline are described in this section. They are:

- Efficient edge based scanline slice acquiring
- Surface data interpolation
- Efficient angular sampling
- Advanced illumination model evaluation

4.1. Geometry Slicing

Because only HPO holograms are assumed a scene is reduced to its single planar slice consisting of polygons and polylines which are then utilized in the angular sampling stage. The method for obtaining the slice is almost the same as the scanline conversion of a triangle. The scanning is performed from up to down along Y axis.

The scanning is edge based. A list of active edges is maintained for each slice position. Each one of the active edges has indices pointing at the opposing edge to the left and to the right. The left and right side is determined by the Y axis and the triangle's normal.

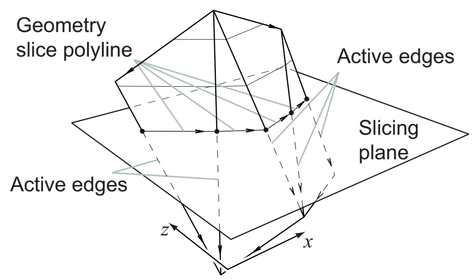

Figure 2: *Geometry slicing.*

The intersections of active edges with a slicing plane are computed from the previous intersections by adding a vector. The vector represents a step of linear interpolation performed between edge's endpoints and it is constant since it is determined by a slicing step that is constant as well.

For each scanline and for each active edge a corresponding slice vertex is created. Then slice edges are generated using the opposing edge information. Since triangles are aware of their orientation the slice edges are all oriented from left to right. This orientation is utilized when visibility is evaluated, see [JHS06]. The vertices and edges from all objects are joined into a single list and passed into a next stage of the pipeline.

4.2. Surface Data Interpolation

For illumination purposes some data has to be interpolated across a triangle's surface, e.g. texture coordinates. The data needs to be interpolated form values stored in vertices.

The first phase of interpolation is performed during the geometry slicing stage. Each scanned edge has a data record associated and the interpolated values are obtained in a same fashion as the computation of intersections described above.

Once the slice edges are obtained, the interpolated data values d are obtained as $d(l) = A + l(B - A)$ where A are data at the first vertex of a slice edge, B are data at the second vertex of a slice edge, l is the interpolation parameter acquired as $l = \mathbf{L}_S/\mathbf{L}$, see Figure 4. Difference $(A - B)$ is stored in the slice edge.

The texture coordinates were mentioned as an example of interpolated data, but any other attributes can be added such as vertex colours. Normals are also usually interpolated but the normals have to be renormalized, which is quite costly operation. As an alternative, the normals can be taken from a normal map providing a texture fetch is faster than the normal normalization.

4.3. Angular Sampling

Next stage of the pipeline is the angular sampling. It is responsible for evaluating occlusion, illumination model and a phase at a hologram plane, see [JHS06].

The sampled angles are the same for all hologram samples. During the sampling a vector representing a direction from the sampled point at every sampled angle is needed for illumination model evaluation. It is the viewer vector. These vectors are precomputed into a table.

Sine's of the sampling angles are also needed for aligning Z-depth to the nearest wavelength multiple. This aligning is not performed in a Z direction but it is rather performed in a sampled direction. This distance correction d_s is computed from the triangle depicted in the Figure 3 as $z_s / \sin\alpha$.

Edges in a slice are sampled using the sine law in a triangle. The triangle is depicted in the figure 4. The β angle is gradually decreased and the α angle is gradually increased as the sampled angle advances counter clock wise. The α angle is used for recalculation of the triangle ratio. The γ angle is used for evaluation of the distance d and the β angle is used for evaluating the length $\mathbf{L_S}$ utilized for computing the parameter of a data record's interpolation.

The uniform change of the sampling angle is an advantage

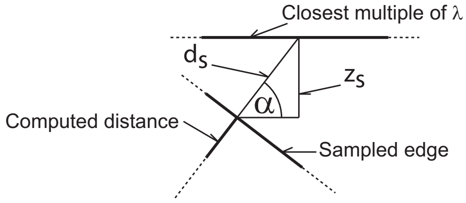

Figure 3: *Aligning of the Z-depth.*

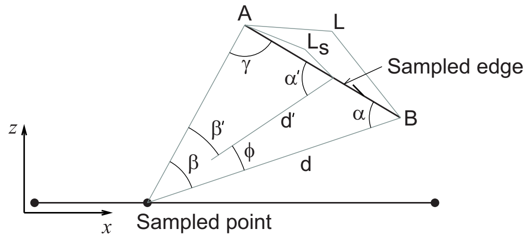

Figure 4: *Distance evaluation.*

because the differential equation for cosine evaluation can be used. This differential equation is based on the well known reflection vector evaluation $\mathbf{R} = 2\mathbf{N}(\cos\alpha) - \mathbf{L}$. The cosine corresponds to the X coordinate of the reflection vector. The new cosine value is obtained from the two previous values as it is apparent form the Equation 5. The sine function can be computed similarly only a value of $\pi/2$ is subtracted from an initial angle.

$$\cos\beta_i = 2\cos\beta_{i-1}\underbrace{\cos\phi}_{const} - \cos\beta_{i-2} \tag{5}$$

4.4. Illumination Models

Data needed for evaluating the illumination model are interpolated using the principle described in the Section 4.2. A viewer vector is precomputed in a table and since all light sources are assumed directional, light vectors are therefore also known. The one last missing component is a normal.

The normal can be interpolated in the same way as the texture coordinates but linear interpolation does not preserve the required unity of the normal vector. The interpolated vector has to be normalized first which requires square root operation. This can make the normal evaluation slow.

An alternative is to use normal map. Texture coordinates can be interpolated without problem and the corresponding normal can be obtained form a texture. On the other hand, a texture fetch operation is not trivial either. So several tests were performed and the results are in the Table 1.

All necessary components are known so the illumination model can be evaluated. A simple Phong illumination model and Phong shading [Pho75] was used for the purposes of testing. The results proved great improvement in a comparison to the originally used constant shading in [JHS06].

5. Results

The Figure 5 contains a hologram computed by the proposed approach and the numerical reconstruction of a scene. The details can be seen in the Figure 6. The test scene contained two directional lights. Two objects had texture map applied. Geometry consisted of 3492 faces and 100 kHz sampling frequency was used for rendering. The actual size of the hologram is 12×6 mm. Some rendering times are listed in the Table 1.

Option	Time [s]
One light + normal map	568
One light + normal interp.	608
One light + normal interp. and 13968 faces	652
Two light + normal interp.	748

Table 1: *Rendering times for various scene and algorithm configurations on Athlon 64, 3.2 GHz, 3.25 GB RAM.*

6. Summary

A complete pipeline for rendering digital HPO holograms was proposed. It can be seamlessly integrated into the existing rendering systems because it consumes the same scene content description as the standard 3D graphics pipeline and no other information is required.

There are several issues which will be addressed in a future work. The major one is that the presented holograms are quite small. Bigger holograms have to be rendered to obtain more useful viewing aperture. An additional acceleration should be achievable with specialized HW or standard GPU as well.

7. Acknowledgement

We would like to thank to the Faculty of Applied Science at the University of West Bohemia for its support and especially to our colleagues at the department for their support and useful comments.

Great thanks belong to the members of EU 3DTV NoE project for their support, namely Prof. Watson and his research group from University of Aberdeen, especially P. W. Benzie.

This work is supported by EU within FP6 under Grant 511568 with the acronym 3DTV.

This work has been partially supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics).

References

[Goo05] GOODMAN J. W.: *Introduction to Fourier Optics*, third ed. Roberts & Company Publishers, 2005.

[Har96] HARIHARAN P.: *Optical Holography: Principles, techniques, and applications*, second ed. Cambridge University Press, 1996.

[JHS06] JANDA M., HANÁK I., SKALA V.: Scanline rendering of digital hpo holograms and numerical reconstruction. In *Proc. SCCG* (2006), vol. 1, pp. 66–73.

[Luc94] LUCENTE M.: *Diffraction-Specific Fringe Computation for Electro-Holography*. PhD thesis, MIT, 1994.

[Pho75] PHONG B. T.: Illumination for computer generated pictures. In *Communications of the ACM* (June 1975), vol. 18, pp. 311–317.

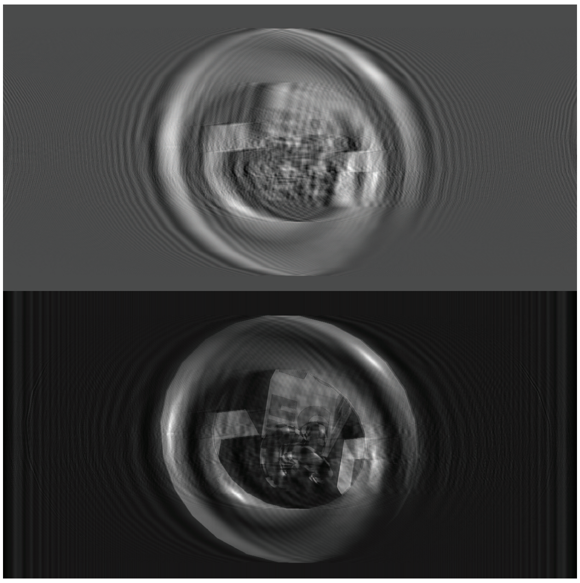

Figure 5: *Hologram in the top. Reconstruction in the bottom.*

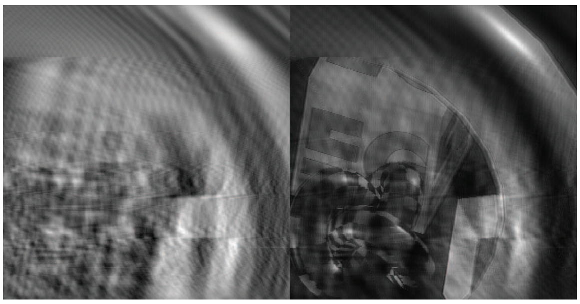

Figure 6: *Close up images. Hologram in the left. Reconstruction in the right.*