In Proceedings Spring Conference on Computer Graphics SCCG 2006. Bratislava: Univerzita Komenského, 2006. s. 66-73. ISBN 80-223-2175-3.

Scanline Rendering of Digital HPO Holograms and Hologram Numerical Reconstruction

Martin Janda*
University of West Bohemia

Ivo Hanák†
University of West Bohemia

Václav Skala‡
University of West Bohemia

Abstract

This paper presents a fast and complete rendering method that can be used for creating digital horizontal parallax only (HPO) holograms of a scene consisting of triangle mesh objects. This method solves visibility and occlusion problems and is capable of working with triangles as the basic primitive instead of the usually used points. It is fast because it avoids the direct distance computation using the square root operation. Advanced numerical reconstruction techniques are used for presenting the results of the rendering method.

Keywords: digital holography, hologram synthesis, hidden surface removal, hologram reconstruction

1 Introduction

The two dimensional displays are known for quite a long time. Two dimensions, however, are not enough. One needs three dimensions. There are several research activities in progress whose goal is to create a genuine three dimensional display.

This research has already delivered several solutions like stereoscopic displays or volumetric displays, but the solutions are not complete because each of them delivers only some of the depth cues. The stereoscopic displays have a single viewing position and/or viewer if tracking is utilized; volumetric displays are not able to hide occluded surfaces.

These limitations might be bypassed by the auto-stereoscopic displays based on multiple views. They are able to deliver very appropriate results with all depth cues thanks to the large number of views. Yet, these images are still 2D images not a wavefront emitted by the original scene as in the case of the holography.

Holography is completely different from solutions mentioned before. This technology is able to reconstruct complete wavefront and thus is capable of delivering all depth cues without limitations in ideal case. This fact makes this technology very promising for the future utilization within 3D displays.

However there is always a catch and the holography catch is a huge amount of data needed to describe a hologram. This influences memory consumption, transmission bandwidth, and processing time if a hologram is computed numerically.

Nowadays, there are spatial light modulators that allow modification of light properties similar to that of an optical hologram. Unfortunately, they have several limitations in resolution, size, and range of light property modifications but they can be considered as a basis for future holography-based displays. Nevertheless, it is not a problem of computer graphics, rather it is a problem of the hardware development.

The problem to solve is to compute a hologram of a synthetic scene that would be fast enough. There has been a lot of work done in this field, e.g. Lucente [1994], but the results are quite far from the beauty of the scenes we are used to from games and computer generated movies.

We have developed a method for hologram rendering which is capable of evaluating visibility and occlusion in a scene composed of triangular meshes, which are most convenient since all tools for processing triangle meshes are available.

For the purposes of presenting our results we need to be able reconstruct the scene from a hologram. This is also useful for reusing scene information contained in a hologram for other displaying systems like the already mentioned stereoscopic display, which works on a principle of delivering slightly different images to each eye. In order to use a hologram as a data source for such a display one needs to reconstruct the scene from two different viewing positions and then deliver those images to eyes.

The reconstruction process described in this paper is a combination of existing approaches as described in the following section. The approach described here is simple, yet it provides a reasonable output that allows us to prove the major part of this paper: the rendering algorithm.

Now a little summary of what follows:

- In the section 2 the basic information about holography and the current state of the art is presented.
- Section 3 contains description of the scan line rendering method of a hologram.
- Section 4 contains description of the reconstruction issues.
- Section 5 presents short summary and our conclusions.
- Our intentions for a future work are presented in the section 6.

2 Fast Introduction into Holography

In opposition to the majority of computer graphics approaches, where the light is represented by rays or structures of similar properties, the light is considered a wave in the case of the holography. The mathematical expression for the wave has a form of a phasor notation:

$$\tilde{u}(t) = U \exp(\varphi - i\omega t) \tag{1}$$

where $\varphi$is the basic phase and U is an amplitude of the wave that square is related to its energy according to mean optical intensity. The $\omega$is an angular frequency of the wave and t is time. The first

[*] e-mail: mjanda@kiv.zcu.cz
supported by MŠMT Czech Rep. no. LC06008
[†] e-mail: hanak@kiv.zcu.cz
supported by 6FP EU 3DTV NoE no. 511568
[‡] e-mail: skala@kiv.zcu.cz
supported by MŠMT Czech Rep. no. 1P04LA240

component of the exp-function argument represents a spatial configuration of the wave while the second part represents a time varying configuration of the wave. The latter one is omitted for computational reasons, i.e. the scene and the waves are examined in a single moment of a time with no consideration about the wave propagation speed through a space. The $\tilde{u}()$ has complex values, i.e. it is a vector in a complex plane.

The next important thing that is required for holography is the phenomenon of light diffraction on a diffraction grating. Diffraction is essential to holography because it causes a change of the light wave propagation direction with some energy loss caused by the fact that the diffraction grating divides energy of wave between several waves, each traveling in different direction.

The simplest grating to describe is a cosine grating, which is a grating resembling the cosine function where –1 means full transparency and +1 full opacity of the material. The equation, which describes a diffraction of a plane wave whose direction of propagation is perpendicular to the grating, is called a diffraction condition:

$$\sin\Theta = g\frac{\lambda}{\Lambda},\ g \in \{0,1,2,\ldots,\infty\}, \qquad (2)$$

where Θ is the angle measured from the normal of a grating plane, λ is an incident light wavelength, and Λ is a length of the cosine function period of the grating.

The g is an integer number and represents an order. Even though the g is in the range of $<0;\ +\infty)$, only value of $g = 1$ is considered. The values larger than 1 are omitted due to very low energy they carry and in the case of holography they are a cause of noise. The g = 1 order has approximately the energy of one fourth of the original plane wave energy. The highest energy has the zero order that is basically the undiffracted plane wave. This order is ignored as well because of the fact that it is not possible to control its direction by means other than by changing the original wave's direction. In holography, this zero order may damage the resulting image due to its high energy.

Besides the diffraction holography also depends on a phenomenon of interference that appears if two waves are superposed together. According to phases of incident waves, various cases of interference ranging from constructive, i.e. both phases are the same, to destructive, i.e. difference of phases is π, appear.

This effect is important for holography because it allows construction of a structure that acts as a diffraction grating later on. And this is the essence of holography. When a plane considered as a screen is illuminated by a coherent light and by a light coming from a scene an interference pattern also called fringe pattern is generated. The coherent light is usually referenced as a reference wave. The whole situation can be described by the following equation:

$$I = \left|\tilde{u}_R + \tilde{u}_s\right|^2 = \left|\tilde{u}_R\right|^2 + \left|\tilde{u}_s\right|^2 + \tilde{u}_s^*\tilde{u}_R + \tilde{u}_s\tilde{u}_R^*, \qquad (3)$$

where I is an energy density, $\tilde{u}_S$ is the wavefront from a scene, $\tilde{u}_R$ is the reference wavefront and superscript * denotes complex conjugate. Rather than the equation itself, its expansion is important for holography. The first component represents energy of the reference wave and influences only variation in brightness of the whole hologram and thus can be omitted in numerical simulations. The second component is denoted as a scene self-interference and causes unwanted artifacts such as a noise. For simplification reasons this component is usually omitted as well.

The most important parts of the Equation (3) are the last two components that are the actual cause of the fringe pattern. These components, respective their real parts are usually denoted as bipolar intensity and can be expressed as real part of complex number multiplication:

$$I_B = 2\Re\left\{\tilde{u}_R^*\tilde{u}_s\right\} = 2\left|\tilde{u}_R\right|\left|\tilde{u}_s\right|\cos(\varphi_R - \varphi_s). \qquad (4)$$

If such pattern, i.e. I_B, is recorded on a light-sensitive material such as a high-resolution photographic film, the result is a diffraction grating, i.e. hologram. When this diffraction grating is illuminated by a reference wave of the same propagation direction, distance, and wavelength, waves are created thanks to the diffraction. A part of these waves resembles the original waves that were coming from the scene when recording was made. The only difference is in the brightness of the resulting image caused by the fact that the energy of the image is part of reference wave energy utilized in reconstruction.

The direction of the reference wave relative to the recording plane together with a position of the scene is an important aspect of the optical holography. The simplest approach is to put the source of reference wave, the scene and the recording plane in a line. This configuration is denoted as in-line hologram and requires a scene to be transparent in a majority of its volume, see Hariharan et al. [1996]. Due to the nature of the scene, this configuration allows reconstruction of silhouettes mostly.

Another configuration is denoted as off-axis, i.e. direction of the reference wave propagation and the direction towards the scene does not follow normal of the recording plane. This configuration allows to record surface information of objects in scene and when reconstructed, the image is not disturbed by the first component from the equation (3) at all in an ideal case. Nevertheless, for computational reasons we will consider only in-line holography.

2.1 Digital Hologram Computation

From the numerical point of view, both construction of a hologram and a reconstruction of an image can be simulated by considering points in the scene as point sources of light, i.e. the scene has to be decomposed into reasonable number of reasonable small points. Each point emits a spherical light wave that propagates through the space according to a following expression:

$$\tilde{u}(x,y,z) = \frac{U}{r}\exp(\varphi)\exp(kr), \qquad (5)$$

where U is related to an energy of given point source and φ is its starting phase. In a majority of cases the scene sampling is significantly larger than the wavelength and thus this starting phase is considered as zero, i.e. the component containing this phase is omitted, see Lucente [1994]. The k denotes a wave-number and depends on a wavelength, i.e. $k = 2\pi/\lambda$. The r is a distance of examined point to the point source, i.e. $r = \sqrt{x^2 + y^2 + z^2}$. Wavefront coming from a scene is computed as a superimposition of individual point sources' spherical light waves.

As it was already mentioned, the hologram is constructed by superimposing of a wavefront coming from the scene together with a reference wave. In optical holography, the reference wave can be of any form with the requirement of a coherence and monochromacity, i.e. only single wavelength. In the case of digital holography, the situation is not different, yet, for simplification reasons the reference wave is considered to be a plane wave. For such a wave, the propagation in the space of constant refraction index is defined as:

$$\begin{aligned}\tilde{u}_R &= U\exp(i\varphi_R)\exp(ikr)\\ &= U\exp(i\varphi_R)\exp\left[i(k_x x + k_y y + k_z z)\right]\end{aligned} \qquad (6)$$

where k is a wave-number and it is equal to a size of the wave-vector (k_x, k_y, k_z).

Of course, during the process of rendering, one has to, consider the visibility of points. The points are infinitesimally small thus all points are always visible. That introduces transparency into the scene and in most cases, this is an unwanted phenomenon.

Some hidden surface (points) removal method has to be applied before computing the wavefront. However, this is quite an expensive operation. Moreover, the visibility of points changes across the hologram plane. Each point on a hologram can be considered as a viewpoint and the visibility of points has to be evaluated according to that viewpoint. This makes it very demanding in regard to algorithmic and computation efficiency.

One trick to cut down the computation demands is to get rid of the vertical parallax. It is not as important as the horizontal one because human eyes are also arranged horizontally. By omitting the vertical parallax, one has to consider only the slice portion of a scene for a given hologram row, i.e. only points having Y coordinate equal y may contribute to a hologram row having the same coordinate y. This practically removes one order of complexity and also reduces dimension of all rendering problems to two dimensions. Holograms with horizontal parallax only are denoted HPO.

2.2 Digital Hologram Reconstruction

Reconstruction, i.e. computation of an image, means numerical propagation of an optical field from one plane to other. In its basic form, the solution is similar to the construction process, i.e. superposing of individual points sources where each point source is a point on a hologram plane. If a source plane is set to the coordinate system's origin and directed to direction of positive Z-axis, then one obtains the following equation:

$$\tilde{u}(x,y,z) = \iint \frac{1}{r}\tilde{u}_0(x_0,y_0)\exp(ikr)\,dx_0\,dy_0\ , \qquad (7)$$

where r is the distance between the source point $(x_0, y_0, 0)$ and the examined point (x, y, z), k is a wave-number, and $\tilde{u}_0$ describes an optical field at a source place. If a source is a hologram, i.e. an intensity function $h(x_0, y_0)$, the source optical field is usually estimated as $\tilde{u}_0(x_0,y_0) = \tilde{u}_R(x_0,y_0)^* h(x_0,y_0)$.

It can be clearly seen from the Equation (7) that a direct implementation of the equation leads to an algorithm of $O(N^4)$ complexity and becomes unusable even for common sizes of 1024×1024 samples. For these reasons, this algorithm is not employed at all.

Nevertheless, the reconstruction can be successfully performed using the Fourier transformation (FT) together with the Fresnel approximation, see Goodman [2004]. If a distance to the source plane is much larger than the maximum extent in the source plane, i.e. $z<<(x^2+y^2)$, it is possible to approximate the variable r in the Equation (7) by an expression that does not contain square root.

In the case of the variable r utilized for modification of the amplitude it can be even approximated by the variable z because the amplitude is not as important as the phase, see Matoba et al. [2002]. In the case of the r variable in the phase component, is it possible to approximate it by $r \approx z + \frac{x^2+y^2}{2z}$ as the first two components of the Taylor expansion for square root in the expression $r = \sqrt{x^2+y^2+z^2} = z\sqrt{1+a}$.
An important requirement is that the third component should not cause changes larger than 1 rad. If fulfilled, this leads to an expression which, if expanded, resembles the 2D FT of an input optical field multiplied by a chirp function, see Goodman [2004]:

$$\begin{aligned}\tilde{u}(x,y,z) = {}& \frac{\exp(ikz)}{z}\exp\left(ik\frac{x^2+y^2}{2z}\right)\\ & \iint \tilde{u}_0(x_0,y_0)\exp\left(ik\frac{x_0^2+y_0^2}{2z}\right)\exp\left(-ik\frac{xx_0+yy_0}{z}\right)dx_0\,dy_0\end{aligned} \qquad (8)$$

Another possibility is to interpret FT of an input optical field as a set of plane waves, see Esmer et al. [2004]. Each sample $U(k_x, k_y)$ on such FT-transformed plane is basically a complex amplitude, i.e. amplitude and phase, of a plane wave that is propagated in direction of its wave-vector k_x, k_y, k_z, where $k_z = \sqrt{k^2 - k_x^2 - k_y^2}$.
After that, the image is computed using a backward FT over propagated set of plane waves. This solution requires FT twice. Yet, it does not utilize approximation of any kind and thus it is able to handle reconstruction that violates condition required for the Fresnel approximation. An interesting feature of this approach is the fact that it is able to handle other cases than just parallel plane, i.e. it allows estimating optical field on plane that is rotated in respect to the source plane.

3 Hologram Scan Line Rendering

This section describes our method for rendering a hologram of a triangle mesh object. This method is capable of computing some sort of form factors which relates an occlusion – a point is not considered if it is occluded by some other point – and visibility – less area is visible if a triangle is tilted or far away – of triangles and makes the captured and reconstructed scene more real like.

The method is tailored for the HPO holograms because the problem of rendering is much easier for HPO holograms. This is due to reduced dimension of the problem. Instead of three dimensional problem of general form factor computation, one has only two dimensional problems in the case of HPO. The Figure 1 depicts the situation for one single row of a hologram.

If a HPO hologram is considered, then triangle mesh objects are reduced into a set of planar polygons or polylines if mesh is not manifold. The bipolar intensity of a hologram's sample is then obtained as an integral of contributions from all polygon edges over half a circle, see Figure 1, providing the edges faces the sampled point. However, the problem of such integration is that the integrated function is quite complex. More over, this function must exclude those edge parts, which are occluded by closer edges and thus are invisible from the sample's position.

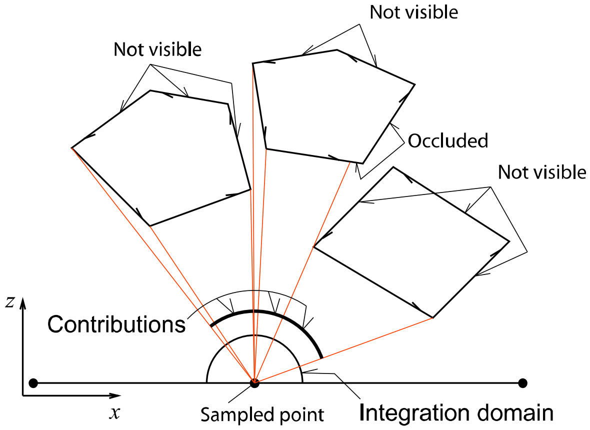

Figure 1: Rendering of the HPO hologram.

The integral can, however, be guessed by sampling the scene angularly. This sampling is depicted in the Figure 5. The integration is then substituted by a summation. Since this summation has to be performed for each hologram sample, it has to be as efficient as possible. Fortunately, we have developed a method which performs this summation very efficiently and it is described in detail in the following sections.

3.1 Geometry Slicing

First of all, the scene slice corresponding to the currently evaluated hologram row has to be computed. This task is quite easy and can be performed in a similar way as the triangle scan line conversion. Scanning is performed in the Y axis direction.

Vertices of each triangle have to be sorted by their Y coordinate. The first and the last vertex define the leading edge, the first and the second one define the top edge and the second and the third one define the bottom edge. A step vector corresponding to the scan line spacing is computed for each of the triangle's edges. Then the initial intersection of the edge with the nearest scan line is computed, after that all consequent intersections are computed simply by adding the step vector to the previous intersection.

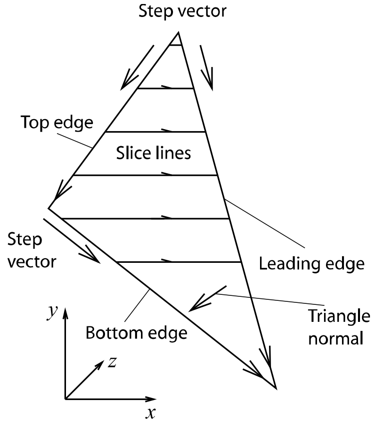

Figure 2: Modified triangle scanline slicing.

The slice of a triangle, which is a line, is defined by the intersection points of the current scan line with the leading edge and the currently active opposing edge. The acquired line is required to be oriented in such a way that its first vertex is on the left side and the second one on the right side. The left respective right side is derived from the triangle normal.

An easy way to find out if the leading edge determines the first or the second vertex of a slice line is to use some vector algebra. One has to compute a dot product of a triangle's normal and a vector product of the leading edge and the top edge considered as vectors. The sign of the dot product determines if the vertex obtained from the leading edge is the first one or the second one. This method of course requires uniform orientation of all triangles.

In the current method's implementation, each triangle is treated individually so if two triangles share one edge then this edge is actually processed two times. This can be optimized by taking the triangles adjacency into consideration.

3.2 Lines Visibility Preprocessing

The result of the slicing is a set of oriented lines. The situation is depicted in the Figure 1. It is obvious that some of the lines are invisible from every sample point of the current row. The question is how to detect invisibility from a given point. Once this question is sorted out then the line is a potential contributor if it is visible from at least one of the hologram extremes. See Figure 3 for reference.

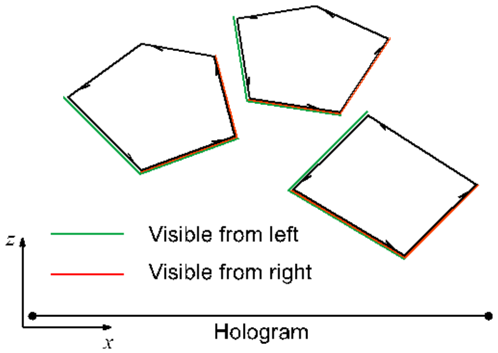

Figure 3: Candidate lines are those visible from at least one hologram row extreme.

And now, let's deal with the question of visibility. Because the lines have uniform orientation their visibility is computed from the angles between the line's vertices and the positive X axis in respect to the sample point position. If the angle α corresponding to the first vertex of then line is larger then the angle β corresponding to the second vertex then the line is visible from the sample point and invisible otherwise. See Figure 4 for a reference.

The angles are recomputed for each processed sample thus their computation must be as efficient as possible. One way to achieve such efficiency is to pre-compute a huge table which returns the angle directly from the provided offset from a sampled point at the X axis and the vertex's z coordinate.

Now, all the lines that never contribute to any hologram sample points have been removed. Everything is set for the next, most computationally demanding phase, which is angular sampling.

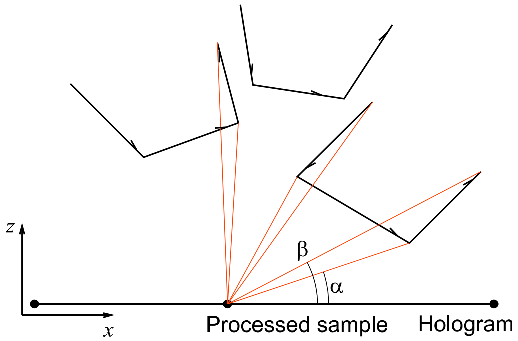

Figure 4: Before angular sampling all lines invisible from currently evaluated sample are removed.

3.3 Angular Scene Sampling

One more preprocessing task, which is the lines length computation, has to be done. The direct distance computation, utilizing the square root operation, is performed because the lines' length has to be computed only once for each row. The length information is reused later on for indirect distance computation. Now everything is ready for computing the hologram samples at the current row.

The angles for each line have to be computed with respect to the current sample x position. This can be done again using the pre-computed table of angles. Once the angles are acquired, the lines have to be sorted by the β angle of the second vertex. This sorting does not have to be so terrible operation to do because the angles are changing slowly since the sample position is changing slowly as well. The intensive sorting has to be therefore done for the first sample point only. The quick sort algorithm is used in this case. However, from this sample point a further one can use a bubble sort, which is very efficient in sorting almost sorted set, which is our case.

The sorted list of lines is then used for maintaining the list of active lines, which is a list of lines that are potential contributors to the currently sampled angle, as the angle of angular sampling is growing. Whenever the angle is increased, all the lines, with the β angle smaller then the currently sampled angle, are moved from the sorted list into the list of active lines. Then the lines with the α angle smaller then the currently sampled angle are removed from the list of active lines. It should be noted, that the lines which are invisible from the current sample point are ignored, it is those where angle α is smaller than the angle β.

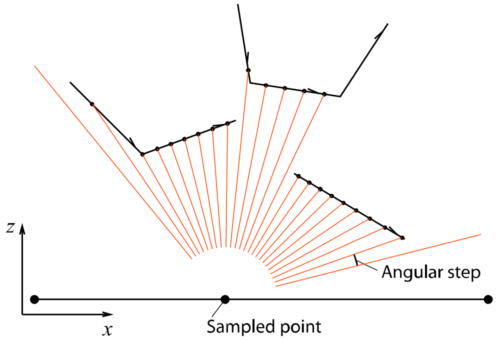

Figure 5: Angular sampling of a scene. Only the closest angular samples are accumulated.

For each line in the active lines list the distance of the current angular sample to the hologram sample is computed. Then the closest sample is eventually accumulated because it is the non-occluded one. This is repeated for each sampled angle and in the end the phase and intensity at the sampled hologram position is evaluated and written into an output. The bottleneck of this process is the distance evaluation. But it can be significantly accelerated using the method described in the following section.

The advantage of this angular sampling is that the lines, which are facing to the hologram and are closer, are sampled more densely then the tilted or distant ones. This complies with the fact that less area is visible from the tilted or distant triangles and therefore such triangles contribute less to the final sample value. One can object that this is only two-dimensional case but in three dimensions triangles may tilt in two directions. But this is actually handled during the scan line conversion. Tilted triangles end up with less scan lines then those facing the hologram plane more directly.

This angular sampling also provides uniform sampling for the whole scene. As a result, one can use radiance values obtained from illumination model as the amplitude value. And because the hologram is normalized at the end, the intensity distribution of a scene is maintained and it is independent on the angular sampling step.

3.4 Faster Distance Evaluation

So far so good, the last remaining problem is the distance evaluation. The distance evaluation needs that nasty square root operation and since distance computation is the most frequent operation during hologram rendering one should avoid direct distance computation whenever it is possible.

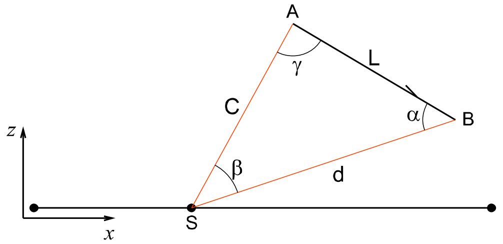

Figure 6: Sine law is used for computing the line samples' distance to the hologram's sample.

The solution is to use the sine law which states that the ratio of sine of an angle and the length of a line which subtends this angle is constant in a triangle: $a : b : c = \sin\alpha : \sin\beta : \sin\gamma$. Our triangle is depicted in the Figure 6. L is the length of a line segment, which was computed in the preprocessing step, d is the distance we want to know, C is constant length for a given sample position and line.

The β is already known, it is the difference of the end points' angles used for visibility check. The triangle ratio can be already evaluated from L and β. The distance C is the only one computed using the square root because there is no other option. Now we have all the information needed for computing the γ and α angles and eventually d. The triangle ratio r is then $r = L / \sin\beta$. The α is computed as $\alpha = \arcsin(C / r)$ and the γ angle is what is left to π. $\gamma = \pi - \alpha - \beta$. Finally, the desired d is computed as: $d = r \sin\lambda$. There is, however, one little glitch, witch is that sine of α and $\pi-\alpha$ is the same value therefore one can never know if the α is correct or not. This can be solved using the dot product:

$$\cos\gamma = \frac{(B-A)\cdot(C-A)}{CL}. \tag{9}$$

We can now compute the γ directly using the arccos function but since we already know the angles. It is better to use only the sign of the cosine to determine the correct values of the already computed α and γ angles.

Since we know all the required parameters we can compute d for each sampled angle using the sine law. To compute d for the next angular sample, we just have to re-compute the triangle ratio and from this ratio and the sine of γ angle, which is constant for this triangle, we can compute d just by one multiplication. To compute the ratio we need the sine function call, but the sampling angle is changing with the constant step (the angular sampling step) and therefore so does the β and α. For each angle step, β is decreased for the amount of the angle step and α is increased for the same amount. One can easily utilize a differential method to compute the new value from the old ones.

3.5 Scene sampling

During the testing of our algorithm we have experienced a rather strange behavior in the reconstructed image. If a scene that consists of a vertical line or several distinguishable lines with thickness equals to a sampling step of the HPO with distance to the hologram of $z = (n+0.5)\lambda,\ n \in \mathrm{N}$, a line and/or lines of such thickness are reconstructed. However, as the thickness of the line increases, the intensity of the reconstructed line compared to the background decreases. At a certain point, the reconstructed image resembled a scene consisting of an object that shadows a reference planar wave, see the Figure 7.

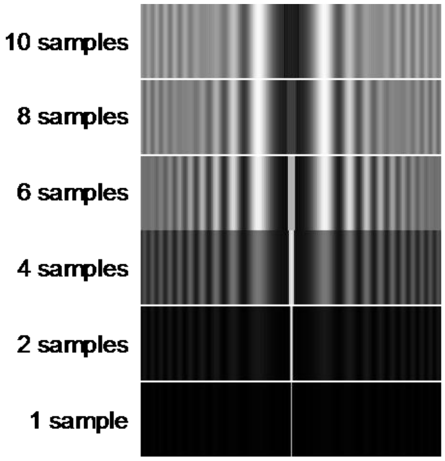

Figure 7: Line of various thicknesses measured in hologram samples at distance $z = (n+0.5)\lambda,\ n \in \mathrm{N}$.

Nevertheless, this behavior was not observed when the line is positioned at $z = n\,\lambda,\ n \in \mathrm{N}$. This ensures that a constructive interference appears at a point of its orthogonal projection to a hologram plate. Based on that observation, we have adjusted the rendering method to displace the position of the angular samples so that to the z position is the nearest smaller integer multiple of the recording wavelength.

3.6 Results

A knot mesh consisting of 2880 faces was used as a test scene. The model has about 6 mm in diameter and was placed about 20 cm from the hologram plane. The computed HPO hologram plate was 12 x 6 mm and samples pitch in both directions was set to 10µm, which is a reasonable value for an inline hologram. The computed images had therefore resolution 1200×600. Wavelength used for rendering was 633 nm.

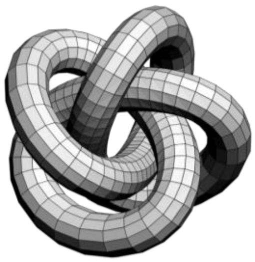

Figure 8: Input object mesh

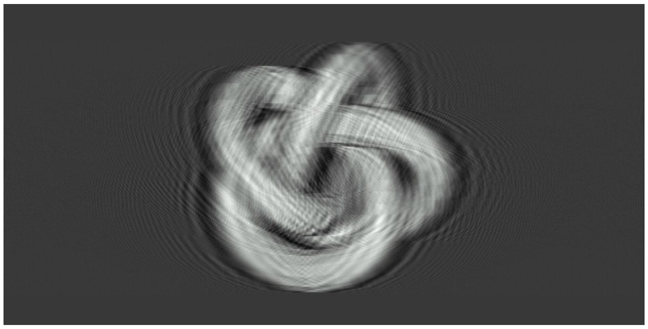

Figure 9: Object's hologram

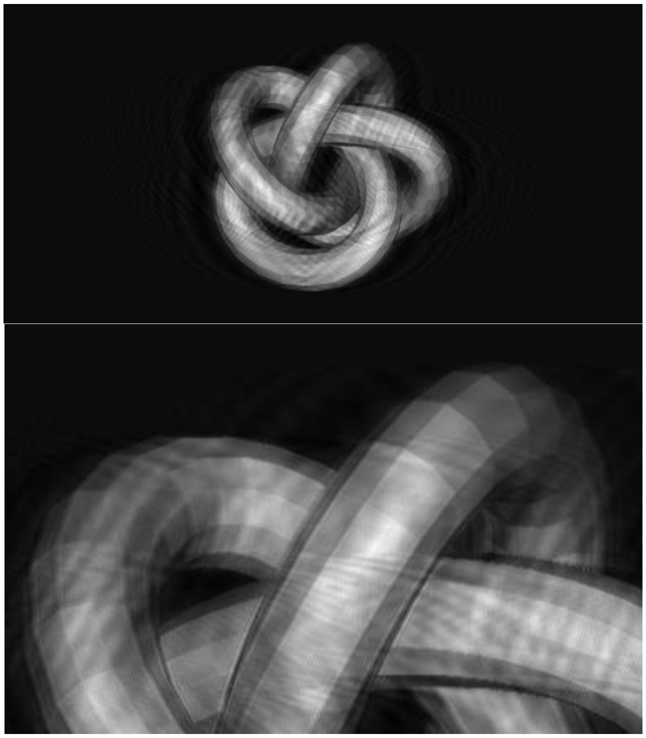

Figure 10: Reconstructed image (top) and its close up (bottom).

4 Numerical Reconstruction

The reconstruction approach we utilized in order to validate our results resembles an approach of Esmer et al. [2004]. It is based on the same assumption, i.e. that FT of the original optical field decomposes such field to a set of plane waves of different propagation direction and complex amplitude.

In order to simulate basic optical system we have incorporated a lens to the reconstruction process in a setup depicted in the Figure 11.

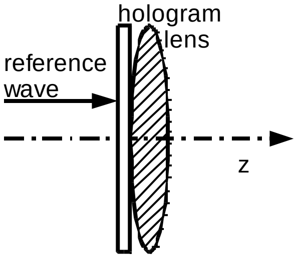

Figure 11: Reconstruction setup with lens, see Goodman [2004].

If the hologram's extent in the X and Y axis is smaller then the extent in the Z axis, then the lens in a spatial space is defined by an equation that contains two components: a constant phase modification and source plane dependent phase modification, see Goodman [2004]. Due to the fact that the constant phase modification is applied over the whole plane it is usually omitted and thus one gets:

$$\tilde{l}(x_0, y_0) = \exp\left(-ik\frac{x_0^2 + y_0^2}{2f}\right), \qquad (10)$$

where f is a focal distance. In order to obtain image without disturbance caused by magnification, the focal distance is set to $f = z/2$. We also tried to apply non-approximated version of the lens equation as described in Goodman [2004]. but we did not recognize any differences in the range of distances we have used in performed computations.

Due to the fact that we are both computing and reconstructing HPO, we have decided to omit the Y-axis completely and thus threat each line individually. This assumption is based on a definition of the HPO and a construction of a device capable of displaying the HPO, see Lucente [1994]. For such device, the viewpoint moving along a vertical line sees always the same image and thus there is no need for performing propagation in the Y-axis similar to the case of full-parallax holograms. Also, this approach allows us to parallelize the reconstruction pretty simply, if speedup is required.

However, this simplification may be a problem in a case of a lens because omitting y coordinate would define a cylindrical lens instead of a spherical one and thus define a visual system different from a common human one. Therefore, we have tried to implement both spherical and cylindrical lens and we found out that the difference between resulting images is not recognizable when converted to 8-bit depth.

This effect is caused by the maximum extent in the X- and Y-axis compared to a distance from the hologram plane to the area of interest. We assume that, if $z > x^2 + y^2$ then the change in X and Y axis has only a little influence on outcome of the Equation (10) and thus it is possible to approximate $x^2 + y^2$ by x^2 only. This allows us to pre-compute the Equation (10) and apply such values for each line independently in order to gain a little speedup of algorithm implementation.

As it is common in the reconstruction process, see Goodman [2004], lens is applied in a spatial space by multiplying original optical field by the Equation (10). Afterwards, the resulting optical field is treated similarly to that of Esmer et al. [2004]. The approach is described by the following:

$$\tilde{u}(x_0, y, z) = \mathrm{FT}^{-1}\left\{\mathrm{FT}\left\{\tilde{u}_0(x_0, y)\,\tilde{l}(x_0, y)\right\}\exp(k_z z)\right\}, \qquad (11)$$

where $\tilde{u}_0$ is the optical field of a hologram, $\tilde{l}_0$ is a lens defined by the Equation (10), $k_z = \sqrt{k^2 - k_x^2}$, and z is a distance to the viewer plane.

During the testing of the reconstruction algorithm's performance we have observed a disturbance caused by the lens component as it can be seen in the Figure 12, right column. Due to that and the fact that lenses application did not brought any additional quality to the image, we have decided to omit the lens for reconstruction purposes even though this would cause a loss of perspective. For a comparison of the reconstruction with/without lenses see the Figure 12.

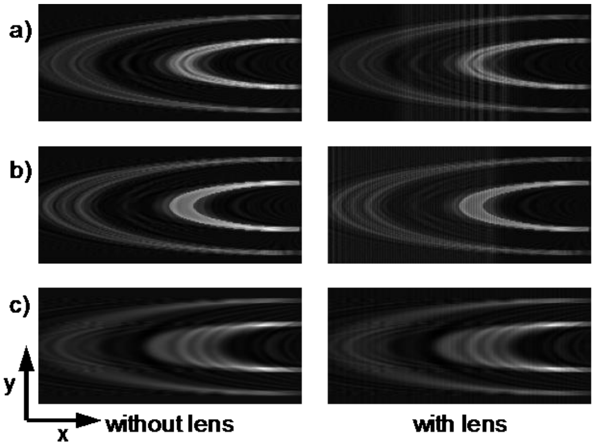

Figure 12: Image of recorded objects (2 half disc, smaller at distance 250 mm, larger at distance 200 mm) with horizontal sample pitch of 5×10^{-3} mm reconstructed at 200 mm (a), at 250 mm (b), and at 1200 mm (c).

5 Summary

We have presented a comprehensive method for computing a digital hologram of a triangle mesh scene. There are no restrictions on the scene content, except for the mesh face orientation which should be counter clockwise. Objects may intersect each other and our method will produce correct results too.

The rendering part successfully solves visibility issues very efficiently and gracefully. Although the sampling rates have to be quite high there is nothing one can do to reduce them. This is due to the diffraction nature of the whole thing.

The most important result we have obtained during the development of this method is the necessity to align the scene samples at the z positions, which are multiplications of the wavelength.

To be able to present the results of the proposed approach, numerical reconstruction approaches, which were also introduced in this paper, have to be used. We perform research in this area as well, but it is a little bit aside from rendering.

6 Future Work

There are a lot of areas in the described method's implementation, where more efficient approaches or preprocessing can be used. We will try to make the implementation as efficient as possible to find out, if real time version can be achieved.

At the moment, the presented method computes only simple constant shading and diffuse illumination. This will be improved in the following versions of the proposed approach.

We would like to include other features of rendering such as advanced illumination models, texture mapping, or transparency handling. These features greatly improve quality of images and thus are the very next improvements of the method.

The inline HPO holograms are easy to process and compute. We do not intend to restrict our work to such holograms only and we plan to create methods for full parallax hologram rendering including the off-axis one.

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.

References

ESMER, G. B., ONURAL, L. 2004. Simulation of scalar optical diffraction between arbitrarily oriented planes. In *Proceedings of ISCCSP 2004*.

GOODMAN, J.W. 2004. *Introduction to Fourier Optics*. Roberts & Company Publishers.

HARIHARAN, P., KNIGHT, P. L., MILLER, A. 1996. *Optical Holography*. Cambridge University Press.

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

MATOBA, O., NAUGHTON, T. J., FRAUL, Y., BERTAUX, N., JAVIDI, B. 2002. Three-dimensional object reconstruction using phase-only information from a digital hologram. In *Proceedings of SPIE*, vol. 4864, 122–128.