EUROGRAPHICS '0x / N.N. and N.N.
(Editors)

Volume 0 (1981), Number 0

Lighting and Occlusion in a Wave-Based Framework

Remo Ziegler, Simone Croci, Markus Gross †

[1] ETH Zurich, Switzerland

Abstract

We present novel methods to enhance Computer Generated Holography (CGH) by introducing a complex-valued wave-based occlusion handling method. This offers a very intuitive and efficient interface to introduce optical elements featuring physically-based light interaction exhibiting depth-of-field, diffraction, and glare effects. Furthermore, an efficient and flexible evaluation of lit objects on a full-parallax hologram leads to more convincing images. Previous illumination methods for CGH are not able to change the illumination settings of rendered holograms. In this paper we propose a novel method for real-time lighting of rendered holograms in order to change the appearance of a previously captured holographic scene. These functionalities are features of a bigger wave-based rendering framework which can be combined with 2D framebuffer graphics. We present an algorithm which uses graphics hardware to accelerate the rendering.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: Three-Dimensional Graphics and Realism, Additional key Words and Phrases: Holography, Lighting, Wave-based Occlusion

1. Introduction

Over the past years, Holography in general as well as Computer Generated Holography (CGH) in particular saw an increasing interest in terms of content creation for 3D display technology. Increasing computational power and the fast development of a multifunctional highly parallel GPGPU makes the huge, but also highly parallel computation efforts required for hologram creation more tractable. The efforts of recent years were put in different areas such as hologram capturing under non coherent lighting, fast approaches for CGH, new physical possibilities of holographic displays or even in holographic rendering, combining holograms with standard 2D framebuffer renderings.

In this paper we introduce two fundamental operations of computer graphics to holography and wave-based rendering, namely occlusion and lighting. Most of the previous methods [ABMW06, JHS06, FLB86, KDPS01, DH98, Mat05a, ZKG07] assume unobstructed wave propagation or simulate visibility in a ray-based way. Besides being a time consuming evaluation, ray-based occlusion has to handle diffraction at object boundaries or refraction due to phase manipulating occlusions as special cases. We propose a novel complex-valued wave-based occlusion simulating opaque, semi-transparent, and refractive objects based on scalar diffraction theory. This representation is useful not only for scene evaluation on a hologram, but also for holographic rendering, where different aperture dependent glare effects can easily be rendered.

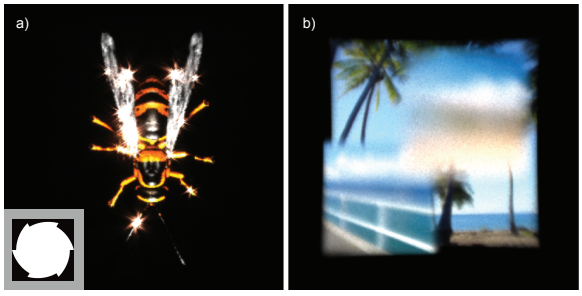

Figure 1: *a) A point-based object rendered using the aperture shape shown on the bottom left. b) A scene consisting of three overlapping planes placed at different depths rendered using wave-based occlusion.*

Another way of enhancing the appearance of a scene is the introduction of lighting. The scene can be rendered for

† {ziegler,sicroci, grossm}@inf.ethz.ch

a specific view under a static illumination setting by altering the wavefield of the primitives. This results in an object which seems to have been painted using the intensities from the light evaluation when seen from different paraxial views. We propose a novel method adapting the reflections according to different parallax views by evaluating the illumination model for every point in the scene on a subset of holographic pixel positions. Furthermore, we propose a novel lighting approach for holographic renderings that allows changing the illumination settings in real-time by evaluations on the GPU.

This work draws upon earlier work [ZKG07, ZBA*07] and assumes the reader is familiar with wave-based propagation and basic concepts of holography. A very brief introduction is given in Sect. 3.1.

2. Related Work

CGH moves towards a more realistic hologram creation by simulating scenes including complex geometry, texture, lighting and occlusion. The scene is subdivided into different primitives which are propagated and evaluated on the hologram. Most methods for CGH use point sampled primitives (cf. [MT00, ABMW06, JHS06, HJS07]), allowing for fast evaluation on the GPU, or subdivide the scene into aperture planes [DH98, Mat05a, ZKG07], which can be propagated efficiently using the Fourier spectrum. Koenig et al. present an efficient way of triangle propagation [KDS01]. However, the propagation is very limited due to a precomputed lookup table. Matsuhima [Mat05a] uses a planar surface and a property function describing shape and texture to propagate a textured triangle. A good introduction to hologram creation and wave-based propagation is given in [Goo68].

Occlusion

Occlusion is typically either not handled at all or the light is treated as rays in order to precompute the occlusion in the scene, such as in [JHS06]. Diffraction due to sharp occlusion boundaries is not simulated by the ray-based approach. Furthermore, a visibility query has to be computed for every new viewing position in order to best approximate the occlusion by ray-based light simulation. Moravec presented a wave-based occlusion approach for paraxial propagation through parallel planes [Mor81], while still using a hybrid wave/ray-based method for image creation. Matsushima [Mat05b] introduces a binary occluder for tilted planes in order to correctly treat scenes built from opaque planes. We extend the binary occluder to a novel complex-valued occluder, allowing the simulation of semi-transparent or refractive objects. Additionally we can generate glare effects simply by applying a semi-transparent occluder at the aperture during wave-based image creation. Nakamae et al. [NKON90] as well as Kakimoto et al. in [KMN*04] presented a way of simulating glare effects. In [KMN*04] the glare effects are, however, only generated for bright points of the scene, which are composited with the rendered scene using billboards.

For objects with high geometric detail, subdivision into planes makes the wave-based occlusion computation very complex. Therefore, we propose to use point-sampled objects represented by surfels as in [PZvBG00] and in order to derive a better visibility evaluation. The scene is rendered by placing point sources at every visible surfel.

Lighting

In 1995, Lucente and Galyean introduced the holographic stereograms [LG95], where the lighting for different views is taken from the rendered parallax views. Matsushima precomputed the shading of the scene [Mat05a] by using a fixed viewpoint in the middle of the hologram, and adjusted the amplitudes of the primitives accordingly. More recently, Hanak et al. [HJS07] proposed an angular sampling of the scene using the GPU, where a local lighting model is evaluated per hologram pixel. Since we evaluate the scene using wave-based occlusion handling, the lighting model has to be evaluated for every scene point and not every angular sample. This being rather time consuming, we evaluate the lighting for a subset of the hologram pixels and interpolate for values in between. This approach could not be applied [HJS07], since the correspondences between angular samples of different viewpoints are not given.

In our proposed lighting method for hologram renderings, we make the assumptions that the BRDF can be freely chosen and the holographic rendering shows an ambient color. Although these are strong assumptions, they could be alleviated in future work by estimating the BRDF from different parallax views and geometry such as in [SWI97, YDMH99, LGK*01, MGW01]. To the best of our knowledge, there has not been any previous work about lighting in hologram renderings.

3. Overview

In this paper we present different ways to improve CGH, hologram rendering, and wave-based rendering (see Fig. 2). CGH describes the wave-based evaluation of a scene on a hologram from which different images can be created from different viewpoints. The hologram is a real-valued 2D-function equal to the intensity of the static interference pattern of a reference and an object wave, from which the complex valued wavefront of the object wave can be reconstructed. Hologram rendering and wave-based rendering evaluate the complex-valued wavefront on the camera aperture, creating an image from a single viewpoint. The former takes only one hologram as an input, while the latter creates an image of an entire scene, which could also contain a hologram.

The two areas of improvements are occlusions and lighting in a wave-based framework. First, we introduce a complex-valued occluder that can handle non-paraxial wave propagation to tilted planes, allowing arbitrary scene creation. With this wave-based approach we are able to simulate defocusing and diffraction according to scalar diffraction theory. Note that wave-based occlusion handles all the occlusion from all possible viewing directions in a single step,

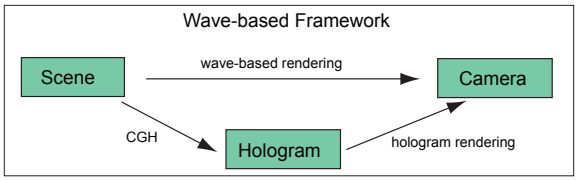

Figure 2: *In our wave-based framework we distinguish between three different wave propagation steps, namely CGH, wave-based rendering, and hologram rendering.*

as opposed to view dependent ray-based occlusion. Second, we present a lighting approach for wave-based rendering as well as a way of lighting holographic renderings in real-time. The real-time lighting allows the user to enhance the appearance of geometrical details by moving a point light over the holographic rendering.

3.1. Wave-based Propagation

In our wave-based framework, we subsample the input into point and planar sources. The propagated wavefront of these sources is evaluated on a plane, which can either be an occluding plane, a hologram, or an aperture of a camera. The spherical wave emitted by a point source at position P_0 is represented as

$$u(P) = A_0 \frac{e^{\mathbf{i}kr+\varphi_0}}{r}\,, \tag{1}$$

where A_0 is the real-valued amplitude at P_0, k is the wave number, φ_0 is the initial phase at P_0, $u(P)$ is the complex wave at the position of evaluation P, and $r = ||P - P_0||_2$. To evaluate the complex-valued wave of a point source on a hologram or on an aperture of a camera, Eq.(1) has to be evaluated for every point P of the target plane. A discrete wavefield $u(P)$ given on a plane S_A is propagated to point P' based on the Rayleigh-Sommerfeld formula:

$$u(P') = \frac{\mathbf{i}}{\lambda} \iint_{S_A} u(P) \frac{e^{\mathbf{i}kr}}{r} (\mathbf{r} \bullet \mathbf{n}) ds\,. \tag{2}$$

This equation can be interpreted as a superposition of point sources $\frac{e^{\mathbf{i}kr}}{r}$ located at the aperture S_A with amplitude $\frac{u(P)}{\lambda}$ multiplied by a phase shift of 90° resulting from the multiplication by imaginary unit $\mathbf{i}$. Additionally, the spherical waves are multiplied by a directional factor $(\mathbf{r} \bullet \mathbf{n})$, with $\mathbf{r} = P' - P$ and $\mathbf{n}$ being the surface normal of S_A. The complexity of the propagation step from a source plane of resolution $N \times N$ to a target plane of the same resolution is therefore $O(N^4)$. We use a fast convolution approach based on Eq.(2), solving the plane to plane propagation in $O(N^2 \log N)$ (see Sect. 4.2).

More details on wave-propagation can be found in [Goo68, ZKG07].

4. Wave-based Occlusion

Handling occlusion in a wave-based manner allows correct simulation of diffraction at object boundaries and provides view-independent occlusion handling. Building on work presented by Matsushima [Mat05b], we approximate the scene by piecewise planar objects acting as occluders of propagated waves as well as emitters of new waves. In contrast to [Mat05b], we extend the binary occluder to a complex-valued occluder in order to simulate semi-transparent objects as well as optical elements such as thin lenses.

4.1. Complex-Valued Occluder

In our framework an occluder is always a plane. This limitation is due to the applied propagation function based on Rayleigh-Sommerfeld which requires the wavefield to be defined on a planar surface.

Every planar surface is split into two parts, an occluding and an emissive part, both of which are complex-valued. The **occluding part** modifies every complex value of the incoming wavefield u, including the amplitude and phase. We model the occluding part by a complex-valued texture o, which is multiplied by u. There are three classifications of occlusion: binary mask, semi-transparent object, and a thin optical element altering phase and amplitude. As shown in [Mat05b], o can be a binary mask, which simulates arbitrary planar shapes, and therefore includes any polygonal shape. Thus an occluding triangle can be represented by a binary mask occluding the incoming wave. An example is

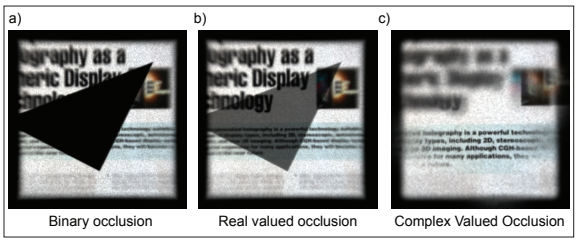

Figure 3: *Depending on the values of the complex-valued occluder, three different material properties, opaque (a), transparent (b), and refractive (c), can be simulated. c) has a slightly shifted occluder as well as a different focal length of the observing camera as compared to a) and b) in order to best visualize the refraction effect.*

shown in Fig. 3a. Reducing o to real values leads to a function that allows a modulation of the incoming amplitudes and permits semi-transparent objects, as depicted in Fig. 3b. By finally applying the full complex function we can alter the phase and therefore the direction of the propagating waves (see Fig. 3c). Choosing the values appropriately lets us simulate arbitrary thin optical elements. They have the characteristic that the incoming and outgoing positions of light are the same. One possible element is the thin lens,

where the phase shift is defined by $o(x,y) = e^{\mathbf{i}\frac{2\pi}{k}\sqrt{x^2+y^2+f^2}}$, with f being the focal length of the lens, x and y the coordinates on the occluder, and k the wave number defined as $k = \frac{2\pi}{\lambda}$ with λ being the wavelength. In our implementation, we provide an interface to create a simple thin lens model, as described in [Goo68], by specifying the two radii R_1 and R_2. The output of varying thin lenses is shown in Fig. 4. Our system also allows complex-valued textures to be used as a complex-valued occluder input, such that the user can specify its own optical elements.

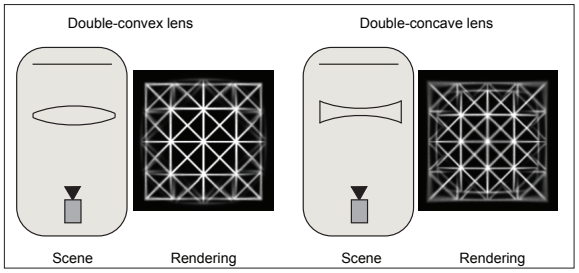

Figure 4: *We provide an intuitive interface for thin lens models that creates the complex-valued texture automatically.*

The **emissive part** is also a complex-valued function able to simulate an arbitrary wavefront propagating in any arbitrary direction in the positive hemisphere. This wavefront is added to the occluded part leading to the outgoing wavefront u_B.

4.2. Plane Propagation

In this section we describe the wave propagation method we apply in our implementation. This propagation is the same for any arbitrary plane pair.

The propagation $\mathscr{P}_{A\to\hat{B}}(u_A)$ from A to B consists of four steps. In the first step, the propagation cone of every plane is evaluated as shown in Fig. 5a. In the second step, the wavefield u_A is transformed into the angular spectrum $U_A = \mathscr{F}\{u_A\}$ using the Fourier transform. As shown in [ZKG07], the angular spectrum can be rotated by applying a Matrix M such that the rotated spectrum is given by $U_A^\circ = M \cdot U_A$. The new plane A° lies parallel to plane B as depicted in Fig. 5b, but might not be paraxial to B. In the third step, we apply an off-axis propagation from A° to $\hat{B}$. $\hat{B}$ corresponds to the extended plane B limited by the union of B and the propagation cone PC_A shaded in grey. Different ways of off-axis propagation have been presented in previous work. The most frequently used one is the Fourier-Shift approach as presented in [DH98, ZKG07]. The Fourier-Shift propagation has the disadvantage requiring a lot of padding when the wavefield has to be propagated over big distances. Therefore, we use the propagation method presented in [SJ05], which is based on a convolution of the wavefield with a propagation function $g(\cdot)$ as shown in Eq.(4).

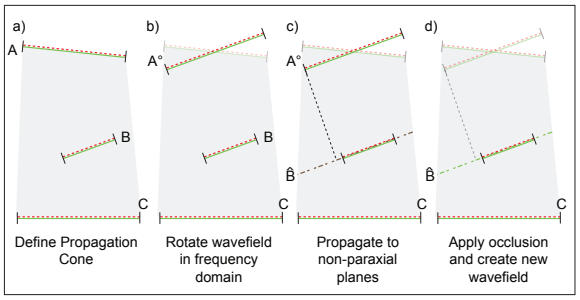

Figure 5: *a-d shows the required steps to propagate the wavefield from one plane to the next. Refer to Fig. 6 for the legend of the symbols.*

$$u'_b = u_a * g(\cdot) \tag{3}$$
$$= \mathscr{F}^{-1}\{\mathscr{F}\{u_a\} \cdot G(\cdot)\} \tag{4}$$

$G(\cdot)$ denotes the Fourier transform of $g(\cdot)$. This approach evaluates the exact Rayleigh-Sommerfeld formula without any approximations in $O(n^2 logn)$ time, where n is the number of samples of a square wavefield. Since there is an analytical form of the Fourier transform $G(\cdot)$ of the propagation function $g(\cdot)$, we only require one Fourier transform of the wavefield, a multiplication with $G(\cdot)$ and an inverse Fourier transform to propagate wavefield u_a. The convolution approach only requires a padding which is as big as the propagation function $G(\cdot)$, independent of the propagation distance. All the incoming wavefields to $\hat{B}$ are named u'_b. The occlusion function o_B can now be multiplied by with u'_B to apply the occlusion to the complete incoming wavefield.

4.3. Scene Evaluation

The scene is subdivided into planar objects which are evaluated from back to front. Starting at the furthest plane A, as depicted in Fig. 6, the wavefront u_A is propagated to the next closer plane B. In order to capture parts of the wavefront u_A which are not occluded by B, we have to evaluate the wave u'_B on a bigger plane $\hat{B}$ leading to $u'_B = \mathscr{P}_{A\to\hat{B}}(u_A)$. The extent of $\hat{B}$ is limited by the the propagation cone PC_A as well as the extend of B. The propagation cone PC_A is defined as the convex hull containing the hologram plane, plane C, as well as plane A. In our implementation we choose a conservative propagation cone. It is defined as an orthographic cylinder that tightly bounds all the planes while its ground plane is coplanar with the final evaluation plane, e.g. C.

The propagated wavefront u'_B is multiplied by the complex-valued occlusion function o_B of plane B leading to $\bar{u}'_B = u'_B \cdot o_B$. The wavefield u_B emitted by B is added to $\bar{u}'_B$ and propagated to C. So, the final propagated wavefront on plane C is defined as $w'_C = \mathscr{P}_{B\to C}(\bar{u}'_B + u_B)$. Iterating over all the planes from back to front leads to the wavefront evaluation on the hologram.

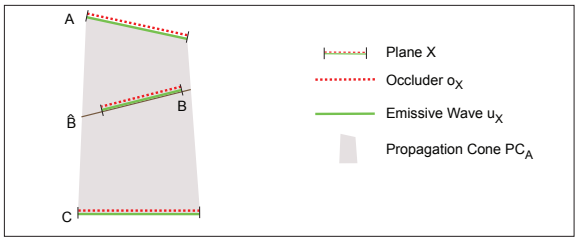

Figure 6: *A simple scene with three occluding planes. The wavefront is propagated from A over B to C.*

Applying wave-based occlusion allows one to evaluate the visibility for all different viewing directions at once, saving a lot of computational time. Fig.7c illustrates this effect in a small example consisting of a point source P occluded by a plane C. In a ray-based approach, the visibility would have to be evaluated for all possible rays, such as the sample rays shown in Fig. 7c. In a wave-based approach, the wavefront evaluated on W (see Fig. 7a) is set to zero at the occluder C as depicted in Fig. 7b. Since the frequencies lost due to the occlusion correspond to directional components, the occlusion can be considered as a loss of rays in different directions.

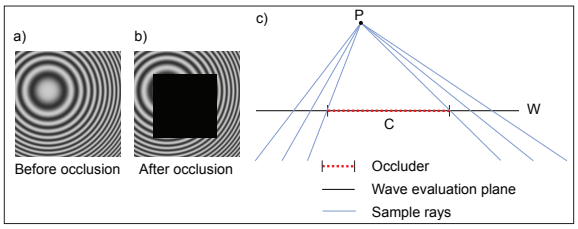

Figure 7: *a) shows the wavefront from a point source P evaluated on plane W. b) depicts the same wavefront from a) occluded by C. c) illustrates the setup of this small example scene.*

4.4. Occlusion for Point-based Objects

If a hologram or a wave-based rendering of a point-based object has to be evaluated, it is important to take occlusions into account. Not doing so results in transparent looking objects [ABMW06], where it is hard to make out any shape. Instead of evaluating the visibility of points using a wave-based approach, we use surfels and a splatting step as described in [PZvBG00] to determine visible points. The surfels have multiple parameters, such as a position, a radius for the disc size, a normal, and a color. Generating a hologram from a point-based object would require computing the visibility from every holographic pixel position. This approach is computationally very expensive. Instead, we generate the visibility only for the central view of the hologram and use it as an approximation for the rest of the paraxial views. In the case of wave-based rendering we only have one viewpoint and the aperture is usually fairly small compared to the object size, such that computation of the visibility from one viewing position is enough.

Once the visibility has been determined for all the points, we generate a point source per visible surfel and build the wavefield as a superposition of all these point sources. Having the normal and the color of the point source provided, we can also evaluate a lighting scheme as described in Sect. 5.

4.5. Glare

Glare is directly dependent on the obstruction of the wavefield entering the camera or the eye. The aperture shape of the camera, or eye lashes, and the pupil of the eye lead to characteristic glare effects. The aperture of a camera can easily be simulated by a complex-valued occluder, allowing the modeling of an arbitrary aperture shape combined with a thin lens model. The eye lashes can either be modeled on a separate plane than the pupil function for more accuracy, or be combined by multiplication to one occluder.

Glare occurs for all the pixels of the final image. The reason that glare effects are not seen for the entire scene observed by the human eye is due to the high dynamic range of the input. Only the very brightest points lead to visible glare. To illustrate the glare effects we scaled the low dynamic range input data by multiplying it with a user defined transformation function.

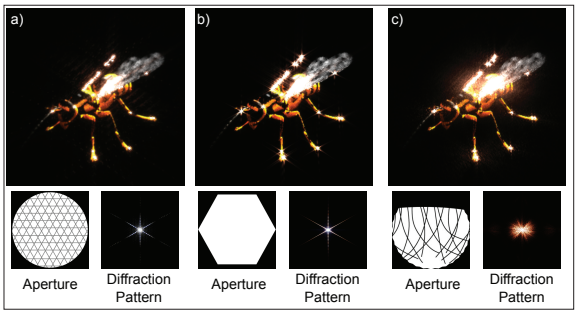

Figure 8: *Every aperture shape leads to a very characteristic glare pattern. a) and b) show geometrical apertures, whereas c) simulates the vision of an eye when looking through the eyelashes. On the bottom right the diffraction pattern of one point source is depicted. To fully see the diffraction pattern we recommend viewing the electronic version of the pictures. (Resolution: 512x512, Time per iteration: 8s, # of iterations: 60)*

Various aperture dependent glare functions can be simulated easily by changing the complex-valued occluder function. In Fig. 8, we show different aperture shapes and occlusion functions, partly taken from [KMN*04], together with the resulting glare when applying the complex-valued occluder and wave propagation. Unlike in [KMN*04], we generate the final image by applying the occlusion due to the

aperture shape or eye lashes to the entire wavefield of the scene and do not require a compositing step using billboards.

5. Lighting

Rendering full-parallax images from a hologram reveals varying reflection properties depending on the BRDF of an object. To generate the complete spectrum of reflections, the illumination model of the scene has to be evaluated per hologram pixel [JHS06, HJS07]. Depending on the evaluation of the illumination model, the amplitude of the corresponding object point is scaled. In our implementation, we focus on two approximations for this evaluation. One evaluates the lighting only for the central position of the image aperture, adapts the amplitude of the scene, and renders the scene on the entire image aperture. This leads to the same lighting as would be obtained using a rasterizing pipeline for image generation. Fig. 9a shows a scene under ambient illumination, while Fig. 9b and Fig. 9c show spotlights as well as directional lights.

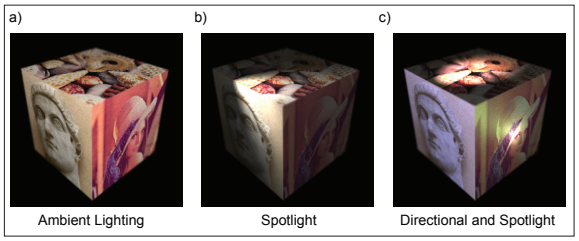

Figure 9: *A scene is rendered with ambient illumination a), with one spot light in b), and using colored spotlights and directional lights in c). (Resolution: 512x512, Time per iteration: 100s, # of iterations: 800)*

The second approach evaluates the lighting model on a subset S of hologram pixels and interpolates the amplitudes for the hologram pixels in between. We implemented a lighting evaluation where the number of samples of S can be chosen by the user. The samples $s_i \in S$ are placed on a regular grid over the entire aperture or hologram, as depicted in Fig. 10a, while using a bilinear interpolation for the illuminations corresponding to hologram pixels in between. This way, the lighting of the scene changes with varying viewpoint, while limiting computational costs.

The maximal measurable frequency of the BRDF is limited by the maximal sampling distance Δs_i between all positions s_i and the distance z of the closest point of the scene to s_i as depicted in Fig. 10b. The angle β is defined as $\beta = 2 \cdot arctan\frac{\Delta s_i}{2 \cdot z}$. A very similar limitation applies to the employed camera. For points in focus, all the rays passing through the aperture are integrated to a final pixel intensity. This implies that the maximum sampling angle limiting the maximal frequency is determined by the aperture size and the distance to the object. This is also true for a physical camera capturing a scene. In Fig. 11, we show a rendering

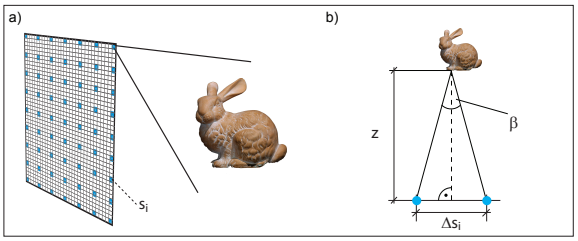

Figure 10: *a) the lighting is only evaluated for a subset (marked in blue) of the hologram pixels. The maximum frequency that can be captured is directly dependent on the angle β depicted in b).*

of the same hologram using a camera with different sized apertures. The size of the specular reflection in red shrinks with a smaller aperture, and therefore, the maximal measurable frequency increases.

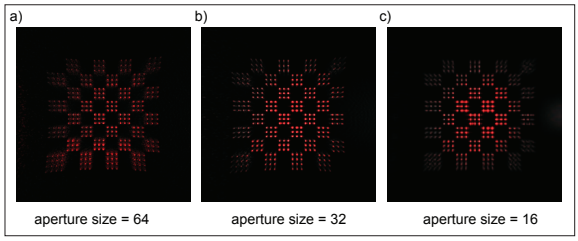

Figure 11: *By decreasing the aperture size from a) to c), higher frequencies of the BRDF become visible, since the area of integration per pixel is decreasing. (Resolution Hologram: 2048, Resolution Image: 256x256 for aperture size 16, Time per iteration: 20s, # of iterations: 1)*

5.1. Lighting of Hologram Renderings

Holograms are recorded using a static illumination setting, consisting either of a laser source or an arbitrary illumination for Computer Generated Holography (CGH). However, when rendering holograms, the observer might want to choose a different illumination for reasons such as revealing more information about geometrical properties of the holographic scene. Therefore, we propose a novel real-time lighting approach for hologram renderings.

In a first step, the image of a hologram is rendered by propagating the wavefront of the hologram to the image aperture. Using the approach presented in [ZKG07], we are able to extract a color image as well as its corresponding depth map. Based on these values, a local lighting model requiring the normal and reflectance properties at every point of the object surface is evaluated.

The points in the scene p_i can be computed from the corresponding points in the depth map d_i by applying a back projection matrix M_P^{-1} such that $p_i = M_P^{-1} d_i$.

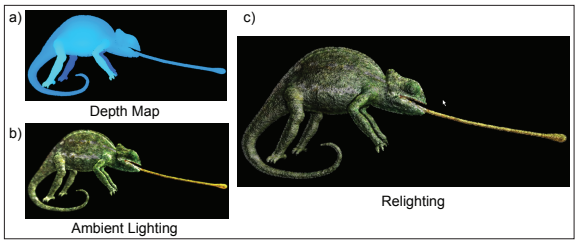

Figure 12: *Using the reconstructed depth map a) and the color map b) the lighting can be performed in real-time c).*

We consider two different approaches for the evaluation of the normal vector. The first approach requires three points P_1, P_2 and P_3 on the surface and computes the normal $\mathbf{n}_0$ by evaluating the cross product of the two vectors defined by the chosen points as $\mathbf{n}_0 = (P_1 - P_0) \times (P_2 - P_0)$. Although this method is very fast, it is also very sensitive to noise. Therefore, a filter may need to be applied to avoid a speckled depth map. The second approach fits a tangential plane defined by

$$z - z_0 = a \cdot (x - x_0) + b \cdot (y - y_0) \tag{5}$$

through a given set of points such that the normal of this plane can be used as the surface normal. The parameters a and b can be evaluated by solving a least squares minimization problem leading to a surface normal $\mathbf{n}_0 = (a, b, -1)^T$. In our implementation, we provide a plane fitting for five, nine or thirteen neighboring points. A possible alternative for normal estimation would be to use MLS-based reconstruction [GG07].

Since this kind of scene illumination is based on the rendered image stored in the color buffer and on the depth map stored in the depth buffer, the evaluation of the lighting model can be performed entirely on the GPU using pixel shaders. The user can interact with the scene by changing the position of the point light source or the direction of the rays of a directional light source in real-time. Please refer to the video of the paper to see a user changing the light position in real-time.

6. Results

We integrated the occlusion handling and lighting into a bigger wave-based propagation framework. An easy setup of the scene consisting of different lighting possibilities as well as different object representations can be defined in an XML-format file. The application is written in C++ and makes frequent use of the Graphics Hardware programmed using HLSL. Performing computations on the Graphics Card increases the speed for point source evaluation, occlusion handling, and lighting considerably. The renderings are generated using an Intel Core 2 CPU 6700 at 2.66GHz with a NVidia 7950GT graphics card.

6.1. Occlusion

In Fig. 13 we created a scene consisting of three textured planes occluding each other. The focal length is set to each plane in turn in Fig. 13a to Fig. 13c. Note the correct handling of the occlusion boundary for the defocused front plane. This means that points laying directly behind the edge of the front plane will have a partial influence on the final rendering, which corresponds to the physical image creation using a limited aperture size. Although only one propagation has been applied per plane, we obtain all possible occlusion directions for viewpoints placed on the target aperture, resulting in view-independent occlusion handling. This is true independent of the complex-valued function of the occluder. Fig.14 shows renderings for different complex occlusion functions leading to refraction effects. Being able to model any planar occluder leads to a very intuitive and easy simulation of arbitrary aperture shapes allowing visible glare effects for bright scene points (Fig. 8).

6.2. Lighting

Most of our scenes are direct wave-based renderings, meaning no hologram is created for the image generation. When creating an image for a specific viewpoint we approximate the lighting of the scene by only evaluating the illumination for this viewpoint (Fig. 9). We have shown that this is an approximation which only holds for a theoretical pinhole camera. A realistic camera simulation requires the lighting evaluation for every point on the aperture. The variation over the aperture of the BRDF of a surface point is integrated to one pixel value for points in focus. This effect can be seen as a low-pass filtering of the BRDF depending on the aperture size. The change in the specular reflection is clearly visible for a scene rendered from a hologram in Fig. 11, which has been evaluated using our lighting evaluation described in Sect. 5.

We present several examples of real-time lighting of hologram renderings in Fig. 15. Altering the lighting can be used to reveal more information about the surface shape as well as to generate more pleasing images.

7. Conclusion

We presented several novel methods to improve wave-based image generation and pointed out their usefulness for CGH. Handling occlusion based on the propagation of waves allows view independency and requires only one propagation step. Furthermore, defining the properties of an occluder becomes very intuitive, since the transmission function can be described in the form of a complex-valued texture. In this way, opaque, transparent, and refractive objects can be integrated as new primitives when creating a wave-based rendering. Additionally, we showed that the occluder interface can be employed to generate diffractive effects of the camera aperture causing glare in the final rendering. The arbitrary aperture shape can also be defined by a texture.

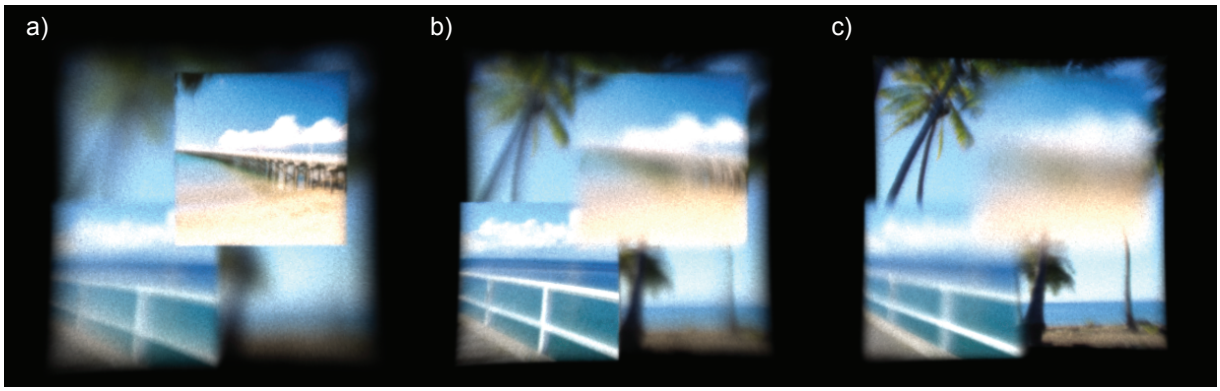

Figure 13: *Wave-based occlusion treats the plane boundaries correctly, even for defocused objects. The three images depict the same scene rendered with three different focal lengths. (Resolution: 512x512, Time per iteration: 248s, # of iterations: 700)*

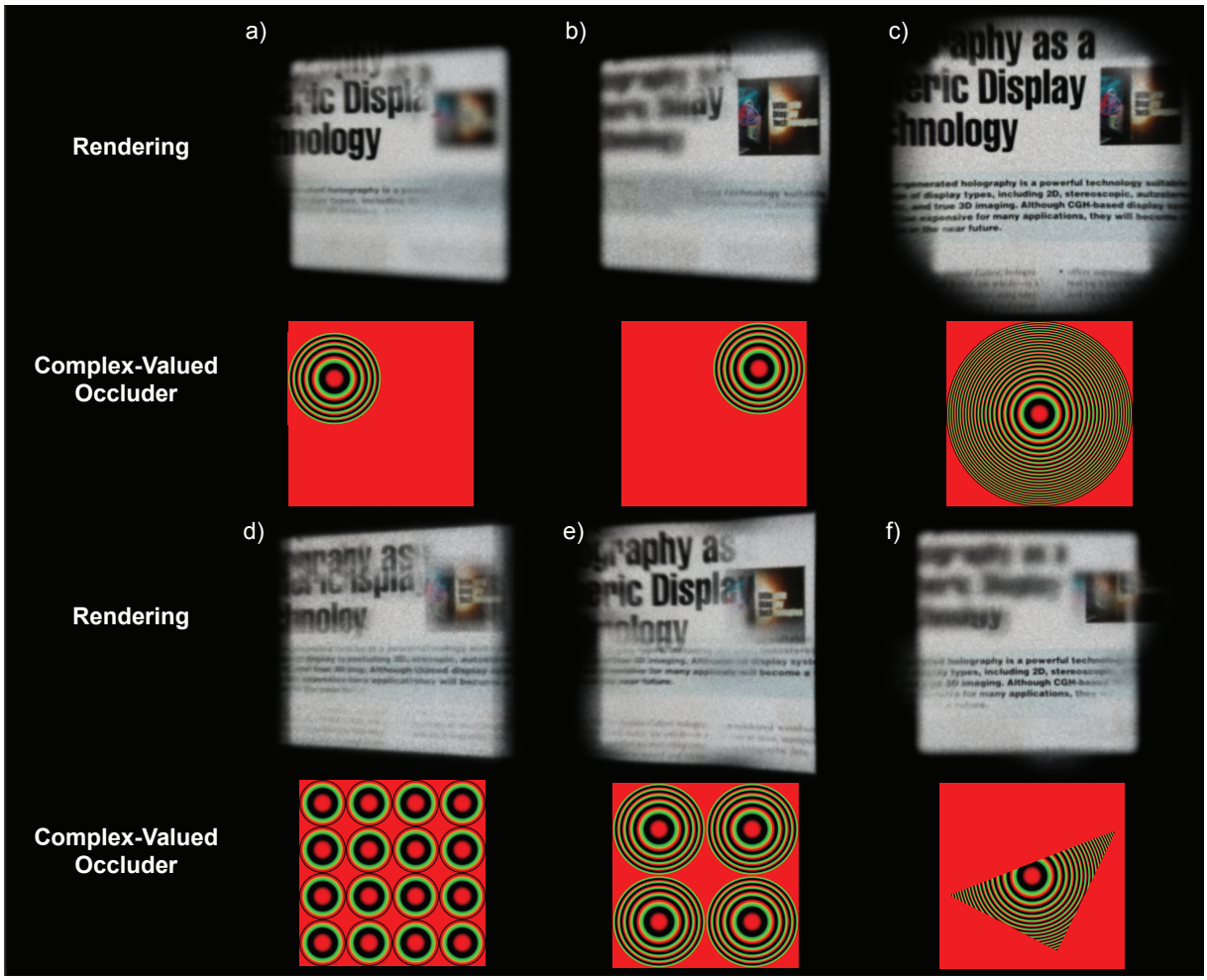

Figure 14: *The complex-valued occluders are illustrated with the red and green textures, where the red channel is used to store the real value and the green channel the imaginary value. The renderings are based on a tilted textured aperture behind an occlusion plane. (Resolution: 512x512, Time per iteration: 195s, # of iterations: 500)*

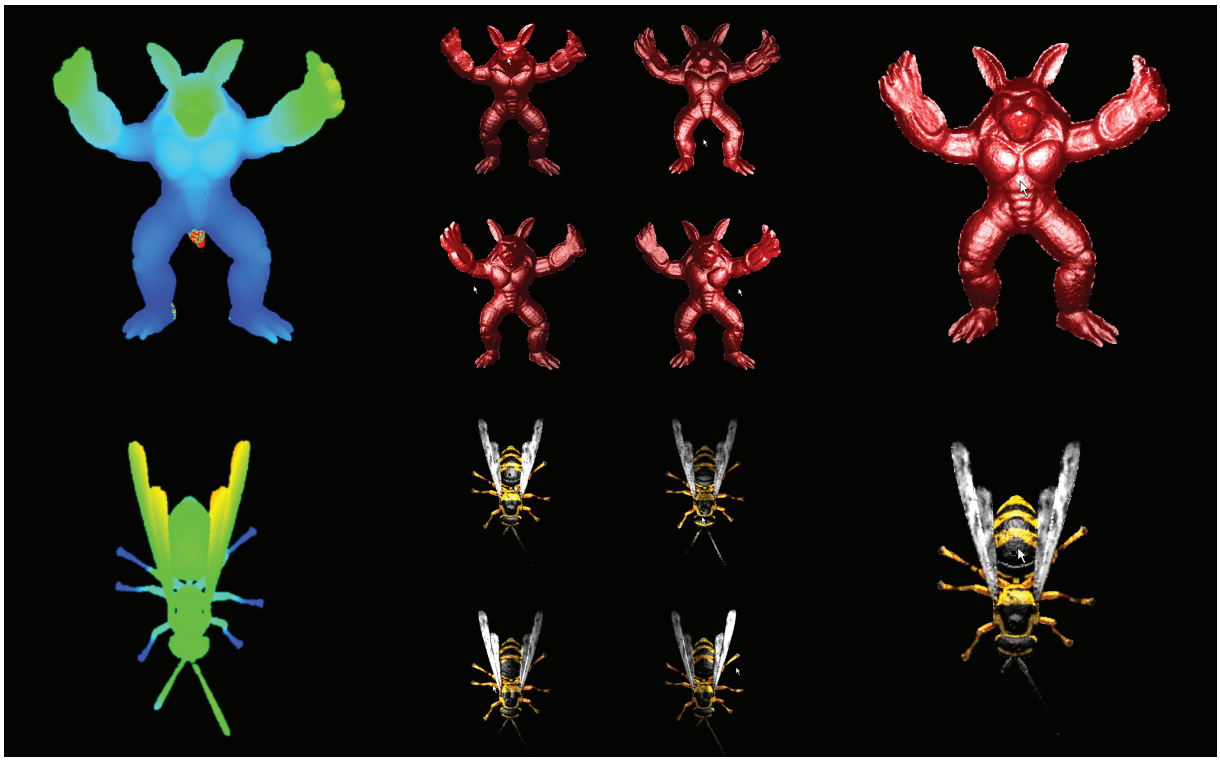

Figure 15: *On the left, the color-coded depth map of the scene is shown, which is used as a base for the normal estimation. The scene can then be rendered for different light positions in real-time.*

Furthermore, we presented a way of evaluating the lighting of the scene for a full-parallax hologram, and demonstrated the influence of the aperture size of the camera on the highest measurable frequency of the BRDF. Computing the lighting only for a subset of hologram pixels yields a big speed up in hologram evaluation while approximating the BRDF only for very high frequencies.

Rendered scenes from holograms suffer from a fixed illumination scheme. Our novel real-time hologram lighting of holograms allows interactive user-defined lighting based on a normal map created from the reconstructed depth field of the hologram. By altering the lighting, geometric details become more visible, which could be used for analyzing object surfaces as well as creating appealing renderings. Furthermore, the hologram could be integrated in a new scene and rendered under arbitrary illumination.

8. Limitations and Future Work

For highly complex scenes, the wave-based occlusion has several limitations. In order to place the planes in a back to front order, some planes have to be split into smaller ones, leading to an even bigger number of required propagations. Furthermore, the scene evaluation can lead to propagations of very big wavefields. Although we tried out a subtractive approach, where the evaluation of the planes never got bigger than the planes already existing in the scene, it is limited to real-valued occluders not altering the direction of light propagation. A different way of scene evaluation as well as the use of a hardware implementation of the FFT, such as the one presented by NVIDIA® CUDA™, would reduce propagation time considerably.

Evaluating the lighting model on a subset affords multiple propagation steps per plane and limits the BRDF that can be captured by a hologram. Since we are handling occlusion already using a wave-based method, we would like to enhance our framework to fully simulate reflectance and illumination in a wave-based way as well. This would allow using only one propagation step while maintaining view dependent reflectance.

The quality of the lighting of hologram renderings is currently limited by the noise of the reconstructed normals as well as by the limited depth-of-field renderings. Improving the filtering of the depth map, on the one hand, and extending the lighting approach to renderings with a limited depth-of-field, on the other hand, would increase the real-time lighting quality.

Observing a drastic increase in computational performance over the last couple of years, we believe that computationally expensive simulations will become feasible, opening up many possibilities for future work.

Acknowledgements

We thank Bob Sumner for proofreading the paper.

References

[ABMW06] AHRENBERG L., BENZIE P., MAGNOR M., WATSON J.: Computer generated holography using parallel commodity graphics hardware. *Optics Express 14*, 17 (August 2006), 7636–7641.

[DH98] DELEN N., HOOKER B.: Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: a fast Fourier transform approach. *J. Opt. Soc. Am. A 15* (April 1998), 857–867.

[FLB86] FRERE C., LESEBERG D., BRYNGDAHL O.: Computer-generated holograms of three-dimensional objects composed of line segments. *Optical Society of America, Journal, A: Optics and Image Science (ISSN 0740-3232), vol. 3, May 1986, p. 726-730. 3* (May 1986), 726–730.

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Trans. Graph. 26*, 3 (2007), 23.

[Goo68] GOODMAN J. W.: *Introduction to Fourier Optics*. McGraw-Hill Book Company, San Francisco, 1968.

[HJS07] HANÁK I., JANDA M., SKALA V.: Full-parallax hologram synthesis of triangular meshes using a graphical processing unit. In *3DTV Conference Proceedings* (2007).

[JHS06] JANDA M., HANAK I., SKALA V.: Digital hpo hologram rendering pipeline. *Eurographics 2006 Short Papers 1* (2006), 81–84.

[KDPS01] KOENIG M., DEUSSEN O., PADUR V., STROTHOTTE T.: Visualization of hologram reconstruction. In *Proc. SPIE Vol. 4302, p. 80-87, Visual Data Exploration and Analysis VIII* (May 2001), Erbacher R. F., Chen P. C., Roberts J. C., Wittenbrink C. M., Groehn M., (Eds.), pp. 80–87.

[KDS01] KOENIG M., DEUSSEN O., STROTHOTTE T.: Texture-based hologram generation using triangles. In *Proc. SPIE Vol. 4296, p. 1-8, Practical Holography XV and Holographic Materials VII, Stephen A. Benton; Sylvia H. Stevenson; T. John Trout; Eds.* (June 2001), pp. 1–8.

[KMN*04] KAKIMOTO M., MATSUOKA K., NISHITA T., NAEMURA T., HARASHIMA H.: Glare generation based on wave optics. *pg 00* (2004), 133–142.

[LG95] LUCENTE M., GALYEAN T. A.: Rendering interactive holographic images. In *Proc. of SIGGRAPH'95* (New York, NY, USA, 1995), ACM Press, pp. 387–394.

[LGK*01] LENSCH H. P. A., GOESELE M., KAUTZ J., HEIDRICH W., SEIDEL H.-P.: Image-based reconstruction of spatially varying materials. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques* (London, UK, 2001), Springer-Verlag, pp. 103–114.

[Mat05a] MATSUSHIMA K.: Computer-generated holograms for three-dimensional surface objects with shade and texture. *Applied Optics 44* (August 2005), 4607–4614.

[Mat05b] MATSUSHIMA K.: Exact hidden-surface removal in digitally synthetic full-parallax holograms. In *Proc. SPIE Vol. 5742, p. 53-60, Practical Holography XIX: Materials and Applications* (June 2005), pp. 25–32.

[MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 519–528.

[Mor81] MORAVEC H. P.: 3d graphics and the wave theory. In *Proc. of SIGGRAPH'81* (New York, NY, USA, 1981), ACM Press, pp. 289–296.

[MT00] MATSUSHIMA K., TAKAI M.: Recurrence Formulas for Fast Creation of Synthetic Three-Dimensional Holograms. *Applied Optics 39* (Dec. 2000), 6587–6594.

[NKON90] NAKAMAE E., KANEDA K., OKAMOTO T., NISHITA T.: A lighting model aiming at drive simulators. In *Proc. of SIGGRAPH'90* (New York, NY, USA, 1990), ACM Press, pp. 395–404.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *Proc. of SIGGRAPH'00* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 335–342.

[SJ05] SCHNARS U., JÜPTNER W.: *Digital Holography : digital hologram recording, numerical reconstruction, and related techniques*. Springer, 2005.

[SWI97] SATO Y., WHEELER M., IKEUCHI K.: Object shape and reflectance modeling from observation. In *Proc. of SIGGRAPH'97* (August 1997), pp. 379 – 387.

[YDMH99] YU Y., DEBEVEC P., MALIK J., HAWKINS T.: Inverse global illumination: recovering reflectance models of real scenes from photographs. In *Proc. of SIGGRAPH'99* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 215–224.

[ZBA*07] ZIEGLER R., BUCHELI S., AHRENBERG L., MAGNOR M., GROSS M.: A bidirectional light field - hologram transform. In *Computer Graphics Forum* (Prague, Czech Republic, September 2007), Cohen-Or D., Slavik P., (Eds.), vol. 26, Eurographics Association, Blackwell Publishing, pp. 435–446.

[ZKG07] ZIEGLER R., KAUFMANN P., GROSS M.: A framework for holographic scene representation and image synthesis. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 403–415.