First publ. in: Proceedings / IEEE Conference on Information Visualisation, August 1997, S. 272-278

Holographic Imaging of Lines: a Texture Based Approach

Alf Ritter, Oliver Deussen, Hubert Wagener, Thomas Strothotte
Department of Simulation and Computer Graphics
Otto-von-Guericke University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
{alf|deussen|wagener|tstr}@isg.cs.uni-magdeburg.de

Abstract

Holography is a method for three-dimensional imaging of objects. In this paper we present an approach for the generation of holograms, exploiting standard graphics rendering methods and hardware. We focus on the visualization of objects composed of line segments, which allow for certain simplifications, and thus hologram generation speed-up.

Our method is based on the derivation of a holographic geometric equivalent of the object to be imaged. In this equivalent, the object is represented as a set of geometric primitives combined with precomputed textures. The equivalent is built up under prescribed conditions, thus simulating certain wave characteristics. The hologram of the object to be imaged is gained just by rendering its holographic geometric equivalent. Problems of achieving the necessary resolution of the output hologram are addressed.

Keywords: 3D Visualization, Synthetic Holography, Texture Mapping.

1 Introduction

The visualization of objects in their three-dimensional extent including natural depth cues is a common goal in applications like Virtual Reality. According to the survey of McKenna and Zeltzer [14], a number of display techniques (lenticular screens, slice stacking devices and others) exist to provide depth cues.

The perception of distance or depth is a complex phenomenon. Mechanisms of the eye as well as the brain are involved. A couple of cues or patterns of stimuli provide us with information about the depth and shape of objects in the real world. They work as well for objects presented to us in an image. Depth cues include binocular disparity, convergence, motion parallax, accommodation, and pictorial depth cues (overlap, perspective projection, etc.).

Three-dimensional displays are classified into two major groups: *stereoscopic* and *auto-stereoscopic*. Both present different views to the eyes, but the latter do not require special viewing aids as, for instance, a pair of stereo glasses. VR displays more or less provide depth cues.

Holographic displays in general provide *all* the depth cues. They are auto-stereoscopic. For perceiving depth or motion parallax, the user does not need any further viewing aids. Thus, synthetic holography is a very attractive display method to study from the standpoint of computer graphics.

This paper is organized as follows: first, the principles of optical and synthetic holography are briefly discussed. Then several approaches of synthetic holography are introduced, leading on to the method we concentrate on—the holographic imaging of objects composed of line segments. Based on the analysis of the mathematical background and existing approaches, we derive a new technique which exploits computer graphics rendering methods. Implementation details of our HoloRenderer system and the results are discussed, followed by an outline of future work.

2 Holography as a 3D visualization method

2.1 Optical holography

The principles of holography were presented for the first time by Dennis Gabor [8] in 1948. Holography, as described in physics text books, for instance by Fowles [4], is a two-step process for the *recording* and *reconstruction* of three-dimensional objects. The hologram is a special diffraction screen. It is used to reconstruct in detail the wave field emitted by the object to be imaged. In hologram recording the output from a laser is separated into two beams, one of which illuminates the object. The other beam (called the reference beam) is directed onto a fine-grained photographic film. The film is exposed simultaneously to the reference beam and to the laser light reflected from the object (see Figure 1(a)). An interference pattern results. It is recorded on the film and constitutes the hologram. The hologram contains all the information needed to reproduce the wave field of the object.

For reconstruction, the developed hologram is illuminated again with the reference beam (Figure 1(b)). Part

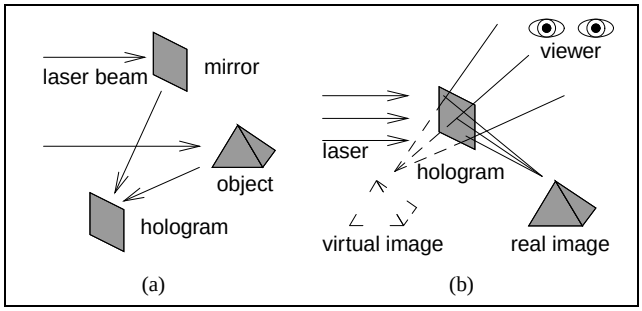


Figure 1: Recording (a) and reconstruction (b)

of the resulting diffracted wave field is a precise, three-dimensional copy of the original wave reflected by the object. The viewer looking at the hologram sees the image of the object in depth. The perspective of the view can be changed just by moving the head.

2.2 Synthetic holography

Synthetic holography embodies the computer simulation of the overall holographic process or parts of it. In the hologram generation step, the computer calculates the transmittance of the holographic plate. The resulting hologram is usually transferred to photographic film for optical reconstruction. Exceptions are the reconstruction of holograms by means of numerical simulation, using systems like DigiOpt [1] or the direct connection of the computer generating the hologram to a real time display, as implemented in the MIT holovideo system, discussed by Lucente and Galyean in [13].

3 Holograms of objects composed of line segments

3.1 Hologram computation methods

Synthetic holography imposes high requirements on computational power and storage capacities. The reduction of effort is therefore a common goal. Several approaches to hologram generation exist, which address the demand for the reduction of effort more or less. In *direct simulation* [9] the object to be imaged is represented by a number of radiating points. Each of these points is a source of a spherical wave. The complex amplitude of this wave is determined for all locations at the hologram plate. For the generation of synthetic holograms using this method, one computation step per hologram pixel and luminous object point has to be performed. Each computation step includes complex additions and multiplications. Considering a size of 5000×5000 hologram pixels and an object consisting of 1000 points, 25 billion steps have to be performed.

One solution for decreasing the computational effort is the determination of point contributions in a separate step. Lucente describes a method in [12], in which the contributions of all possible 3D object points are precomputed and stored in a look-up table. In the actual computation step these point contributions are just read from the table and accumulated in the hologram. Based on this method, also the rapid generation of holographic stereograms is possible [13].

The methods described so far generate images of objects consisting of luminous points. In case the 3D object consists of line segments, it is useful to decompose the object information into lines instead of points. A line can represent a large number of points. Composing the objects of lines leads to a considerable reduction in computational effort. Since line drawings are a powerful method in computer graphics (as stated by Strothotte et al. in [16]) and since they can be effectively transferred into holographic images, we focus on the computation of holograms taken from objects composed of line segments.

3.2 Holographic imaging of lines–mathematical background

Objects to be imaged in a hologram consist of luminous geometric primitives such as points, lines or surfaces. These primitives emit waves which result in the recording step in a certain pattern on the hologram. During hologram reconstruction the waves are again generated and focus in the original primitives, thus reconstructing these primitives. As stated, for instance, by Frère in [5], a mapping of primitive and wave type exists:

- a point source emits spherical waves,
- infinite lines emit cylindrical waves, and
- plane waves are emitted by infinite planes.

For the description of the holographic patterns (see Frère et al. in [7]) generated by cylindrical waves originating from lines, we introduce a Cartesian coordinate system (see Figure 2). The hologram is in the *xy* plane of the coordinate system. The *z* axis is the optical axis. If the line is parallel to the *x* axis and located at a distance $z = R$ to the hologram plane, the complex amplitude $u(x,y,0)$ in the hologram plane is described by

$$u(x,y,0) = exp\left(i\frac{\pi}{\lambda R}y^2\right) rect\left(\frac{x}{2a}\right), \qquad (1)$$

which consists of the Fresnel approximation and a clipping function to restrict the line influence to a rectangular area on the hologram, thus generating line segments of finite length in the reconstruction step. The effects occurring at the edges of the hologram portions are disregarded [7]. λ is the wavelength.

This representation is restricted to line segments parallel to the *x* axis. An arbitrary line segment *L* is described as follows:

1. the angle α of L relative to the x axis,
2. the angle γ of L relative to the hologram plane, and
3. the coordinates (x_0, y_0, z_0) of the center point of L.

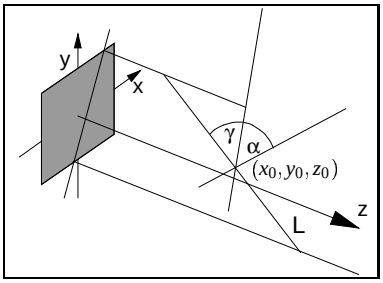


Figure 2: Line parameters

The point (x,y) on the hologram plane is transformed by a 2D rotation to a point (X,Y) with $X = x\cos\alpha - y\sin\alpha$ and $Y = x\sin\alpha + y\cos\alpha$. The angle γ is incorporated by

$$R(X,\gamma) = z_0 - X\tan\gamma. \tag{2}$$

The function $R(X,\gamma)$ describes the distance between the line and the hologram points (pixels). The reconstruction of lines outside the optical axis (z axis in Figure 2) is achieved by introducing a linear phase factor

$$exp\left[i\frac{2\pi}{\lambda z_0}(x_0X + y_0Y)\right]. \tag{3}$$

The resulting complex amplitude is

$$\begin{aligned} u(x,y,0) &= exp\left[i\frac{2\pi}{\lambda z_0}(x_0X + y_0Y)\right] \exp\left[i\frac{\pi}{\lambda}\frac{Y^2}{R(X,\gamma)}\right] \\ &\quad \times rect\left(\frac{X}{2a}\right). \end{aligned} \tag{4}$$

Figure 3 shows an example of a holographic image obtained with the method described above. Three lines perpendicular to each other constitute a coordinate system (3(a)). In the hologram (3(b)) the rectangular areas representing the wave patterns are to be seen. The clipping is performed in the x and y direction to prevent spatial frequencies which cannot be transferred to the hologram.

4 A new approach

4.1 The holographic geometric equivalent

A straight line emits a cylindrical wave, provided there is a constant phase distribution along the line (see [11], [7]). The phase of individual points on a line is a free parameter. Its modification does not alter the perception of the geometry in the optical reconstruction. Linear phase distributions result in conical waves. The relationship between the line segment to be reconstructed and the conical

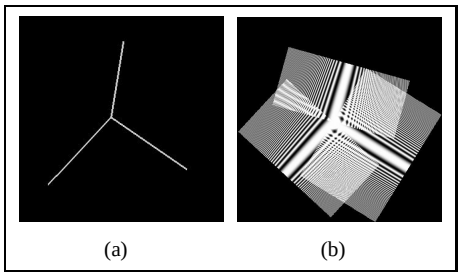


Figure 3: Image of a coordinate system, original object (a) and its hologram (b)

wave is not uniquely determined [10]. In the reconstruction step each line focus can be obtained by an infinite number of different conical waves. The cone angle β is the free parameter. A boundary condition is that β has to be chosen such that in the recording step the light emitted by the luminous line segment reaches the hologram.

In particular, we can assume a linear phase distribution along a line segment such that the hologram plane is parallel to a tangential plane of the emitted conical wave. The resulting pattern is shown in 4(b). The pattern can be gen-

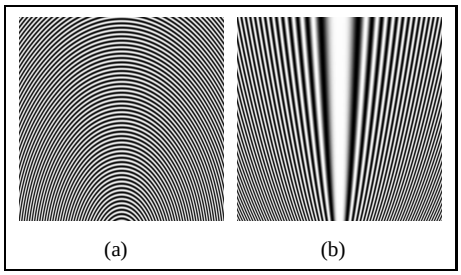


Figure 4: Hologram of an inclined line (a), the same line with a linear phase distribution applied (b)

erated with the method described in Section 3.2. It represents conic sections (the cone is cut at the top), the aspects of which are described by Aumann and Spitzmüller in [2].

How can these patterns be obtained using computer graphics methods? In a first approach, we explore the approximation of these holograms by "linear" patterns. A line parallel to the hologram emits a cylindrical wave (see Figure 5(a)). As soon as the line is inclined to the hologram, the pattern is more or less distorted. At first sight, the distortion seems to be similar to a perspective projection of the cylindrical pattern, if the linear phase distribution is chosen as explained above (see Figure 4(b)). In order to simulate this behavior, the object to be imaged in a hologram is transformed into its *holographic geometric equivalent*. This equivalent consists of a set of textured rectangles which are constrained in their orientation. The texture represents the "reference pattern" (Figure 5(a)), generated by

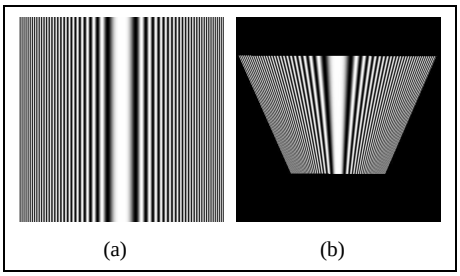


Figure 5: Cylindrical wave (a), rendering of an inclined textured rectangle (b)

a cylindrical wave. One textured rectangle (Figure 5(b)) per original line represents the wave emission of the line. The rectangles are oriented and positioned in the same way as their line counterparts which build up the original object (see Figure 7, the rectangles are centered around the original lines). The rectangle orientation according to the original line leads to a distortion of the rectangle together with its texture (see Figure 5(b)) in the perspective projection. Thus, a line inclined to the hologram plane is simulated. The rectangles mimic the clipping of the wave pattern accomplished by the rectangle function shown in Equation 1.

Due to the fact that geometry processing and texture mapping are implemented in hardware on computer graphics workstations, the hologram generation is fast. It can be performed in nearly real time for reasonable resolutions.

4.2 Refinement

The approach as demonstrated so far uses linear distortions of a "reference pattern" to image lines inclined to the hologram plane. However, this is not a valid "simulation" of conic sections. The distortion along the line has to be rather quadratic instead of being linear.

In a refinement of the existing approach the texture coordinates in the holographic geometric equivalent are altered according to the angle γ (recall Figure 2) describing the element orientation. Chen and Williams [3] define morphing as a simultaneous interpolation of shape and texture. In this case an effect similar to morphing is gained, forcing the textures to display patterns typical for conic sections (see Figure 7). The texture coordinates are altered along the line. The shape of the underlying rectangles remains undistorted.

Each element (rectangle) of the holographic geometric equivalent is subdivided into a set of stripes the texture coordinates of which can be altered independently. Furthermore, the reference texture displaying a cylindrical pattern is now mapped to each stripe instead just once to the entire rectangle. In each display step the texture coordinates for each stripe are altered according to the *position* of the element, and its *orientation*. The new texture coordinate $s'(v)$ assigned to a stripe vertex v of the equivalent's element E is determined from the original texture coordinate $s(v)$ for a line parallel to the hologram plane and the square of the distance $dist(v)$ of the vertex from the hologram plane. The distance $dist(v)$ depends on the line position $L_0 = (x_0, y_0, z_0)$, where $v_z = z_0$, and the line orientation γ with respect to the hologram plane (see Figure 2).

$$s'(v) = f(s(v), dist(v)^2) \tag{5}$$

Thus, a quadratic distortion is achieved.

4.3 Examples

The equivalent of a single line (linear distortion) is already to be seen in Figure 5(b). The refined version to display the shape of conic sections is shown in Figure 6. The pattern converges in a quadratic fashion. Figure 7 illustrates the transformation of three lines perpendicular to each other into their holographic geometric equivalent. It

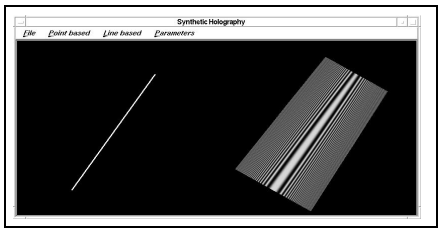


Figure 6: Original line (left) and its holographic geometric equivalent (right)

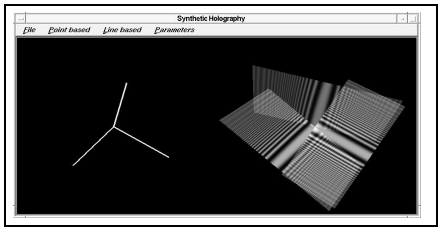


Figure 7: Three lines forming a coordinate system (left), the equivalent (right)

consists of three rectangles which are rendered translucent. Thus, overlap without introducing further disturbances in the reconstruction step is accomplished. A more complex example is shown in Figure 8. The original object consists of about 200 line segments.

5 Implementation

5.1 Software architecture

The HoloRenderer system is implemented in C++ on Silicon Graphics workstations, thus exploiting hardware

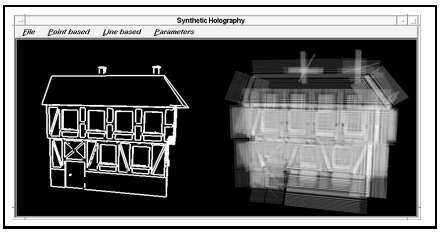


Figure 8: A house (left) and its equivalent (right)

support for rendering. It is capable of generating holograms of line objects using the graphics libraries OpenGL and IRIS Performer. In order to study conventional line-based and point-based approaches (see Section 3.1), corresponding computation methods have been implemented as well.

The modules of the HoloRenderer (see Figure 9) form a holographic pipeline which is fed with the geometry data of the object and produces the holographic image at the end. The *viewing module* loads the object to be imaged,

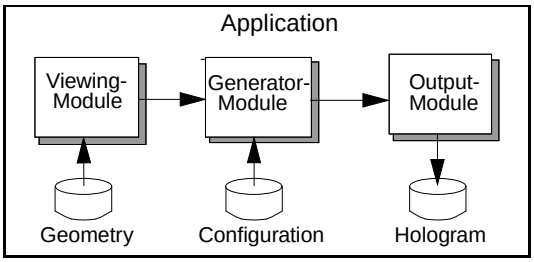


Figure 9: Architecture of the HoloRenderer

displays it, builds, and renders the holographic geometric equivalent. For conventional hologram computation, implemented in the *generator*, it extracts line parameters or the point-based information. The *configuration* determines parameters like wavelength and the hologram dimensions. The *output module* collects the computation results produced by the *generator* and stores the hologram in a file (pixel-based formats TIFF, BMP, Postscript, IRIS RGB).

5.2 Hologram reconstruction

To display the holographic image, the hologram generated with the system described above has to be reconstructed. The reconstruction can be accomplished in two ways: *simulation*, and *optical reconstruction*. The results of a simulation are already to be seen in Figure 3. The simulation was performed with the system *DigiOpt*, which was developed by Aagedal et al. [1] at the University of Karlsruhe. The advantage of a simulated reconstruction is that the results can be quickly obtained. On the other hand, only holograms of reduced resolution could be simulated because of memory limits.

To visualize holographic images in their three-dimensional extent, they have to be optically reconstructed. In order to prepare the holograms for optical reconstruction, we transfer them to photographic material using a film recorder. In our setup for the optical reconstruction of synthetic holograms a laser emits coherent light needed for reconstruction. The widened laser beam penetrates the hologram, and the real image of the object appears in the space behind the hologram.

5.3 Implementation details

The “reference texture” showing the cylindrical pattern is computed using the Fresnel approximation (see Equation 1). The texture is generated in a separate pass and stored persistently in a file as a pixmap. It is loaded again as soon as the holographic geometric equivalent is displayed.

The geometry of the original object composed of line segments is stored in the *scene graph* which is an IRIS Performer specific tree-like data structure. In the process of the generation of the equivalent, an additional scene graph is built up which consists of rectangles having the same length as their line counterparts in the read-out scene graph. The rectangles are subdivided into stripes as described in Section 4.2. The precomputed textures are applied to the rectangles. Additional attributes of the rectangle encompass the rectangle width to control overlap and spatial frequencies in the hologram, and the alpha value to control the translucent appearance. Alpha blending techniques, as described by Neider et al. in [15], are provided by OpenGL and can therefore be used by the HoloRenderer system.

The position and orientation of the rectangles is chosen in accordance with the lines from which they originate. First, the rectangle is centered around the line, and second, the plane normal of the rectangle is mapped under parallel projection to a vertical line segment, i.e. it has an x-component of zero. Otherwise, distortions due to the perspective projection introduce corruptions of the depth reconstruction of the line segments. Thus, the rectangles are longitudinally inclined according to the line orientation, but normal to the viewing plane in the other direction.

The object to be imaged can be interactively transformed, rotated and positioned relative to the camera. The equivalent is subject to the same transformation. During rotation, the rectangles of the equivalent are tilted. The tilt with respect to the axis specified by the original line has to be compensated in order to obey the orientation constraint mentioned above. This is accomplished by using a special node provided by IRIS Performer. *Performer Billboards*

are utilized to guarantee the desired orientation of the affected geometry (the rectangles).

As stated above, the hologram is generated by rendering of the equivalent. However, viewport dimensions are constrained to fit typical computer graphics applications. Since holography requires much higher resolutions, the viewport constraint prohibits results in a reasonable hologram resolution. To remedy the situation, the image is tiled. Each tile is rendered in one step within the given constrained viewport dimension. Figure 10 illustrates the approach. The camera is centered for each tile. The rendering setup (camera and viewport) has to be chosen carefully to prevent aliasing effects at the tile borders which would destroy the hologram. The problem is the same as when setting up a picking region in OpenGL (see [15]). This solution is successfully adapted to our problem. It results in the necessity of an additional matrix multiplication per tile rendering. After all the tiles are rendered, they are collected and combined into the final hologram which therefore can be of very high resolution.

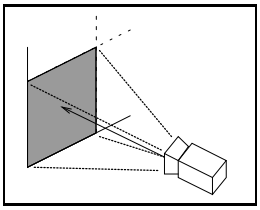


Figure 10: Rendering of tiles

6 Results

The verification of the holograms generated was performed by means of simulation and optical reconstruction. The first example is presented in Figure 11. In the simulated reconstruction the line focus "travels" over the extent of the line. In this case the simulation was performed for two distinct depths. If the whole series of simulations of the line hologram was shown, the line focus would travel almost continuously over the reconstruction. The different line foci appear as "sharp" regions in the reconstruction. In regions out of focus the line does not appear as a narrow slit, but rather widened and blurred.

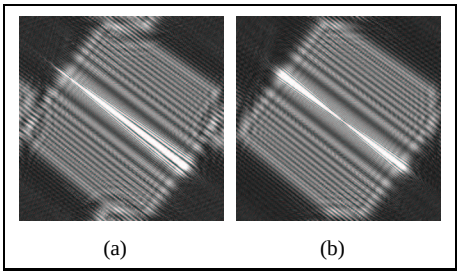


Figure 11: Two simulated reconstructions of a line segment at a (a) nearer and (b) bigger distance from the hologram

Another example of the coordinate system is shown in Figure 12. For comparison, the same object with a conventionally generated hologram was already shown in Figure 3. In the equivalent shown in the left part of Figure 7 each line of the original object is represented as a textured rectangle. In this example, these rectangles have the same "starting point", the origin of the coordinate system to be imaged. The major requirement in this example is that the reconstructed foci of each line converge in this starting point. This requirement is fulfilled in the holographic geometric equivalent: the rectangles have one common point in the origin of the visualized coordinate system (Figure 7). Furthermore, the rectangles are of the same size. The textures which are projected on the rectangles are identical. Thus, by means of perspective projection the textured rectangles are distorted consistently, that is, the distances between pattern maxima (black stripes in the texture) are equal for equal depths in the equivalent. The reconstruction (see Figure 12) proves this assumption. Also in the reconstruction of the "coordinate system example" the three reconstructed lines converge in one point, the "origin". Figure 12(a) illustrates this feature of the holographic geometric equivalent by choosing the reconstruction depth such that the foci are in this example close to the origin. The holographic reconstruction of a house composed of lines is shown in Figure 13. The reconstruction illustrates how the front lines appear in different foci due to different reconstruction depths.

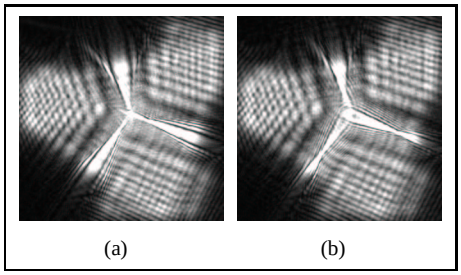
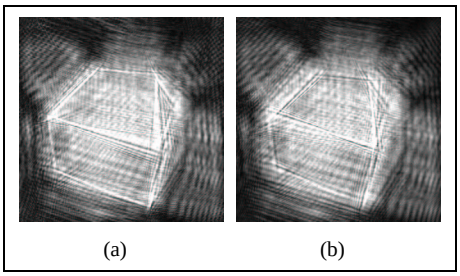


Figure 12: A simple coordinate system, reconstructions at distinct depths

The examples shown in this article are all reconstructed by means of simulation using the DigiOpt system. An optical reconstruction using the experimental setup described in Section 5.2 was performed as well. It verified that the input objects composed of line segments are properly reconstructed. The reconstructions are of considerable depth, which is typical of conical waves, as stated by Leseberg in [10].



(a) (b)

Figure 13: A house, the reconstructions show different foci of the front lines.

7 Conclusions and future work

The holograms obtained with the presented method, the rendering of holographic geometric equivalents, reconstruct the original objects. The reconstructions are of good contrast. However, the feasibility of our approach has to be demonstrated on more complex input objects (see Strothotte et al. [16]).

Regarding more complex objects, the holographic display of models given in a spline representation is desirable. A spline can be approximated by short segments of straight lines. Straight line segments can be immediately processed by the HoloRenderer system introduced in this article. However, the reconstruction of short line segments is likely to be degraded due to clipping. The shorter the line segments, the more artifacts are introduced to the image by clipping. Another solution is the distortion of the wave pattern according to the shape of the spline as shown in principle by Frère and Bryngdahl [6].

The holographic imaging of splines and, in addition thereto, of line styles enables us to give a 3D insight to techniques applied to computer generated line drawings. An interesting question is how holographic images including line hatchings are perceived by an observer of a hologram. Thus, line drawings and holographic images “live” in a symbiosis.

Acknowledgments

The authors would like to thank Thomas Benziger for establishing the experimental setup for hologram reconstruction.

References

[1] H. Aagedal, T. Beth, H. Schwarzer, and S. Teiwes. Design of paraxial diffractive elements with the CAD system DigiOpt. In I. Cindrich and S. H. Lee, editors, *Diffractive and Holographic Optics Technology IIt*, Proc. SPIE 2404, pages 50–58, 1995.

[2] G. Aumann and K. Spitzmüller. *Computerorientierte Geometrie*. BI-Wissenschafts-Verlag, Mannheim, Leipzig, Wien, Zürich, 1993.

[3] S. C. Chen and L. Williams. View interpolation for image synthesis. In *Proceedings of SIGGRAPH ’93*, pages 279–288, Anaheim, August 1993.

[4] G.R. Fowles. *Introduction to Modern Optics*. Dover Publications, Inc., New York, 1989.

[5] C. Frère. *Verallgemeinerte Konfiguration der Rekonstruktions- und der Hologrammfläche in der digitalen Holografie*. PhD thesis, Fachbereich Physik der Universität-Gesamthochschule Essen, 1988.

[6] C. Frère and O. Bryngdahl. Computer-generated holograms: Reconstruction of curves in 3-D. *Opt. Comm. 60*, pages 369–372, 1986.

[7] C. Frère, D. Leseberg, and O. Bryngdahl. Computer-generated holograms of three-dimensional objects composed of line segments. *Journal of the Optical Society of America (JOSA)*, A3(5):726–730, 1986.

[8] D. Gabor. A new microscopic principle. *Nature*, 161(4098):777–778, May 1948.

[9] W. Lauterborn, T. Kurz, and M. Wiesenfeldt. *Coherent Optics, Fundamentals and Applications*. Springer-Verlag, Berlin, Heidelberg, New York, 1995.

[10] D. Leseberg. Computer generated holograms: Cylindrical, conical, and helical waves. *Applied Optics*, 26(20):4385–4390, 1987.

[11] D. Leseberg and C. Frère. Free positioned and oriented focal lines from computer-generated holograms. *SPIE Proceedings*, 812:113–118, 1987.

[12] M. Lucente. Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1):28–34, 1993.

[13] M. Lucente and T.A. Galyean. Rendering interactive holographic images. In *Proceedings of SIGGRAPH ’95*, pages 387–394, Los Angeles, August 1995.

[14] M. McKenna and D. Zeltzer. Three dimensional display systems for virtual environments. *Presence: Teleoperators and Virtual Environments*, 1(4):421–458, 1992.

[15] J. Neider, T. Davis, and M. Woo. *OpenGL Programming Guide: the Official Guide to Learning OpenGL*. Addison-Wesley, Reading, MA, 1993.

[16] T. Strothotte, B. Preim, A. Raab, J. Schumann, and D.R. Forsey. How to render frames and influence people. *Computer Graphics Forum*, 13(3):455–466, 1994.