3DTV Conference proceedings, IEEE ISBN 978-1-4244-0722-4, 2007

HPO HOLOGRAM SYNTHESIS FOR FULL-PARALLAX RECONSTRUCTION SETUP

Martin Janda, Ivo Hanák, and Václav Skala

University of West Bohemia
Department of Computer Science and Engineering
Univerzitní 8, Pilsen, 306 14, Czech Republic

ABSTRACT

Digital hologram synthesis is a task that simulates a physical process. Currently, there is no working solution capable of handling continuous surfaces, solving visibility, and applying a local intensity variation efficiently. Due to the computational complexity an approximation is required. Our solution tries to address this issue by extending the most efficient HPO hologram synthesis in such way that the computed hologram does not require a special optical setup for reconstruction.

***Index Terms*—** Holography, Computer graphics, Rendering

1. INTRODUCTION

Holography is a scientific area that deals with light field capturing using the phenomena of diffraction and interference. It utilizes the wave nature of light as the fundamental building block. If coherent light is used it is possible to capture monochromatic light waves scattered from an object. It is done by interfering the reference wave with the scattered one. The image of the resulting interference pattern captured on a photosensitive material then acts later on as a diffraction grating. This grating, if illuminated by the reference wave, produces a light field which contains the original scattered wave. In the ideal case this can lead to the impression of a real scene even thought only the hologram is present, see [1] for reference. Holography is therefore a promising technology for a genuine 3D display.

Digital holography is a numerical version of the optical holography. It employs equations and relations derived for light propagation to numerically simulate the physical phenomena [2]. It deals with processing of captured holograms, extracting information stored within, and finally with the numerical synthesis of holograms. The latter one is targeted by this paper.

Currently, there is no approach to synthesis available that would be able to simulate the physical phenomena accurately enough in reasonable time. This is because of the computational complexity of the problem. To illustrate the fact, without any approximations or assumptions, to numerically simulate a propagation in a free space between two parallel planes that are sampled by $N \times N$ samples an algorithm of the computational complexity $\mathcal{O}(N^4)$ is required. Due to the nature of the problem, it is not possible to reduce the complexity without loosing some information.

One of the most drastic approximations used is the reduction or elimination of the vertical parallax in a hologram. Such holograms are called horizontal parallax only (HPO) holograms and they bring reduction of complexity by one order of magnitude. The HPO hologram is a hologram that lacks vertical parallax, i.e. the wave from a single point is not propagated vertically [3]. Therefore, the viewer does not experience parallaxing of the image if the viewer changes her position relatively to the Y-axis, i.e. up-vector.

Nevertheless, the vertical parallax is not so important as the horizontal one. It is because the human eyes are organized horizontally, thus the viewer does experience the binocular disparity in the HPO hologram, hence she does sense the depth. The drawback of HPO holograms is the necessity of a special reconstruction setup composed of cylindrical lenses [4], otherwise the reconstructed image is heavily vertically blurred.

Our main goal therefore was including additional information back into the HPO hologram so that it could be reconstructed on a common reconstruction setup without the need of special lenses or other special display device except the spatial light modulator such as LCD or DMD [2] with reasonably small element pitch. Our extension is therefore a compromise between the performance of the HPO hologram synthesis and the simple optical setup used for reconstructing the full parallax holograms. It should be emphasized again that the actual synthesis approach **is not** the main topic of this paper but the enrichment of the HPO hologram content **is**.

In order to improve the performance of our approach we used a graphical processing unit (GPU). Currently, the GPU is able to replace a specific part of the graphics pipeline with a user supplied program that can perform various tasks. Thanks to the specialization of the GPU, it offers a greater computational power than the comparable CPU [5]. The GPU has two

This work has been supported by EU within FP6 under Grant 511568 with the acronym 3DTV and by MSMT CR projects CPG No. LC06008 (Centre for Computer Graphics) and No. 1P04LA240.

programmable stages: the vertex stage that process vertices and the pixel stage that process individual pixel fragments. As the pixel stage is able to access every element on a target plane it is the usual place where the most of the useful computations are done.

The rest of the paper continues with a description of our solution, results, and conclusion with notes on further speed up of the approach. The description of the solution includes a brief introduction to the approach that our solution is based on and the results section offers a comparison of our approach to the full parallax digital holograms.

2. SOLUTION

Our solution is based on the HPO hologram synthesis approach described in [6]. The synthesis approach imposes some assumptions on physical properties of the simulation. First it assumes that the scene is aligned so that the normal of the recording plane is pointing towards the Z-axis and the center of the recording plane is at the origin. The assumption on the center is not strict and the approach allows recording the scene from any position by application of a transformation. Second, it assumes that the scene consists of triangular mesh surfaces. Other kinds of surfaces can be converted to a triangular mesh by approaches well known in computer graphics. Also, it assumes that no part of the scene lies behind the hologram plane nor it is intersecting it. This can be assured by clipping operation on geometry also known from the computer graphics.

The fact that the HPO hologram is synthesized allows simplifying the evaluation of both occlusion and light field computation because the HPO hologram is actually a group of one dimensional hologram lines, where each line is treated individually without any concern about other lines. Thus, the scene has to be sliced first by the plane perpendicular to the hologram plane that contains a given hologram line. As this may be a complicated operation for a general surface, in the case of triangular mesh it is not: it is just a matter of proper data structures, see [6].

The geometry contained in each slice consists of oriented lines. The point sources are generated on the lines by the means of a ray casting. All generated points have the same phase, i.e. the approach is not an exact simulation of the recording process used in optical holography. The intensity of each point source is computed according to the used illumination model which is another feature well known from the computer graphics. It is assumed that the point sources do not interact with other point sources in the scene. The ray casting approach is illustrated in the Fig. 1.

As it is described in [6], after a point source constituting the first intersection of the ray directed in a given angle originating at current sample on the hologram line and the geometry of the slice is obtained, the distance from the hologram sample to the intersection is computed to get the proper phase.

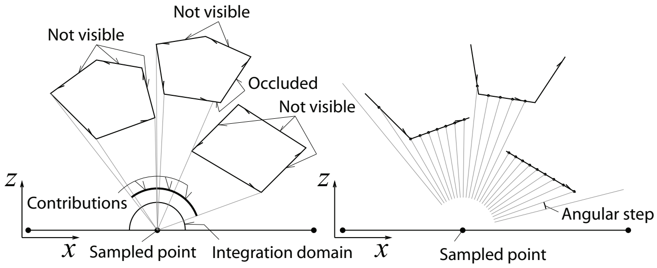

Fig. 1. Ray casting approach for hologram synthesis. [6]

The actual sample is estimated according to an approximation of equation for a spherical wavefront:

$$\tilde{u}(x) \approx A \exp\left\{-i\frac{2\pi}{\lambda}\left[z_s^2 + (x - x_s)^2\right]^{\frac{1}{2}}\right\}, \qquad (1)$$

where (x_s, z_s) is the intersection point, $(x, 0)$ is the point at the hologram line, and A is evaluated according to the illumination model. The resulting complex amplitude $\tilde{u}(x)$ is accumulated. After processing all rays cast from the current hologram sample, the accumulated value represents a sample of resulting light field and the next sample is processed similarly. Our approach modifies the part, where the contribution from a ray is processed.

Since the information about the point source is stored only in one line of the hologram the vertical parallax is lost and an attempt to reconstruct an HPO hologram on a full parallax hologram reconstruction setup will end up as an image vertically blurred beyond recognition. The information about the point has to be distributed into all other lines. This distribution is done by superposing the proper column from the diffraction pattern of a single point source located at the proper distance from the plane into all hologram lines, i.e. into the samples of the same x coordinate as the currently processed one. The x value of the column is evaluated from the knowledge of the angle and the z coordinate, i.e. the column index is equal to the x coordinate of the point source relative to the current sample on the hologram.

Individual values along the column can be computed directly. Yet, in such case a computationally expensive functions such as square root have to be evaluated. Therefore our approach adapts the hologram synthesis based on the principle of combining pre-computed diffraction pattern of a single point source at the distance z_0 stored in a complex texture, see [7] and [8]. The main idea of this principle is that the diffraction pattern of an arbitrary point is obtained by transforming the pre-computed one. The center of the pattern coincides with the (x, y) position of the point source and the z position is used to compute the scaling factor s of the pre-computed pattern:

$$s = \left(\frac{z_0}{z}\right)^{\frac{1}{2}}, \qquad (2)$$

where z_0 is the distance along the z-axis of the point source that was used to compute the pattern. The pattern has to be magnified if $z > z_0$ otherwise it has to be minified.

Our approach uses the pre-computed diffraction pattern of a single point as well but instead of accumulating the whole pattern it accumulates only one column out of it. The principle is illustrated in the Fig. 2. The modified processing of the ray is therefore following. A point source is obtained from the intersection of the current ray and the geometry in the slice. The illumination model is evaluated to obtain the proper amplitude scaling factor for the pattern. The distance is not evaluated since only the z and x coordinates are required. The z coordinate determines the scale coefficient s. The scaled x coordinate pinpoints the proper column in the diffraction pattern texture. The appropriate part of the column is added into the result according to the height of the hologram frame and the y coordinate of the currently processed line.

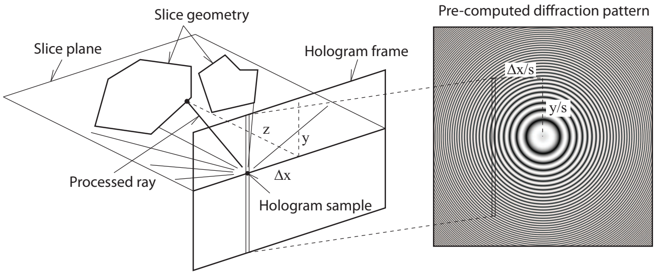

Fig. 2. The accumulation of a column from the diffraction pattern texture.

In exchange for a small amount of additional processing this method significantly extends the HPO synthesis described in [6]. The synthesized holograms can be replayed using the standard reconstruction setup used for full parallax holograms. This makes HPO holograms more feasible.

Besides that, it is easily implementable with graphical processing unit (GPU) support that improves the overall performance. The visibility is handled in a screen space naturally by the GPU using the depth-buffer. The only bottleneck is the accumulation of the proper column of the diffraction pattern because it assumes an extensive access to the texture of the pattern. However, this operation access the texture sequentially, so it is expected that the texture cache hit/miss ratio will be kept reasonably low so that the cache is able to reduce the memory latency [5].

The GPU-aided solution is a straightforward analogy of the pure software solution with several differences. Unlike the software solution it evaluates all intersections for a given angle at the same time. This allows an exploitation of the depth-buffer facility to solve the occlusion. Also, the propagation of the column of the pattern to other hologram lines is modified because it requires a scatter operation which is not available on current GPU's. It is replaced by an appropriate gather operation, i.e. a sample on the hologram has to accumulate all contributions obtained from all sampling rays that originate at the same column as the currently processed sample.

The computation is done in the pixel stage of the GPU. The algorithm is implemented in two passes and assumes a hardware with dynamic flow control support:

1. According to the given angle ξ the intersections of rays and scene geometry are computed. The output of this operation is the z coordinate of the intersections and a color for later use.

2. According to the output of the previous step the samples are accumulated. This step exploits two loops, each inspecting a column in a different direction. This step consumes the most of the computation time.

3. RESULTS

We have implemented the algorithm in our experimental renderer running in software mode to verify the principle. We also implemented a GPU-aided version to verify our assumption on speedup. The reconstructions presented in the Fig. 3 were computed numerically as a propagation in angular spectrum perpendicular along the Z-axis [9].

The test scene contained one directional light. Geometry consisted of 4128 faces. The local center of the scene was located at the distance of 200 mm from the hologram plane. The angular sampling step was $0.003°$, angular range $2.0°$, and the actual size of the hologram is 10×10 mm sampled by 1000×1000 samples.

The computation times of the software solution are not included since the efficiency of implementation was not our goal. The experimental software implementation was used for validation purposes only. The GPU-aided solution required 47 minutes for computing the scene using the ATI Mobility Radeon X1600 hardware.

4. CONCLUSION

We have enhanced the HPO rendering algorithm towards the full parallax application. Thanks to this enhancement we are able to render holograms of synthetic scenes in reasonable time and we expect that the reduction of the quality and/or the information contained within the hologram will not affect the viewer experience significantly. As the computer graphics is able to deliver almost photorealistic images for interactive visualizations our hologram may create such impression as well because our approach is compatible with computer graphics for interactive visualization. The algorithm benefits from the use of the GPU. The algorithm is also easily scalable: it can handle arbitrarily large scenes and it can be distributed on tile basis with almost linear speedup.

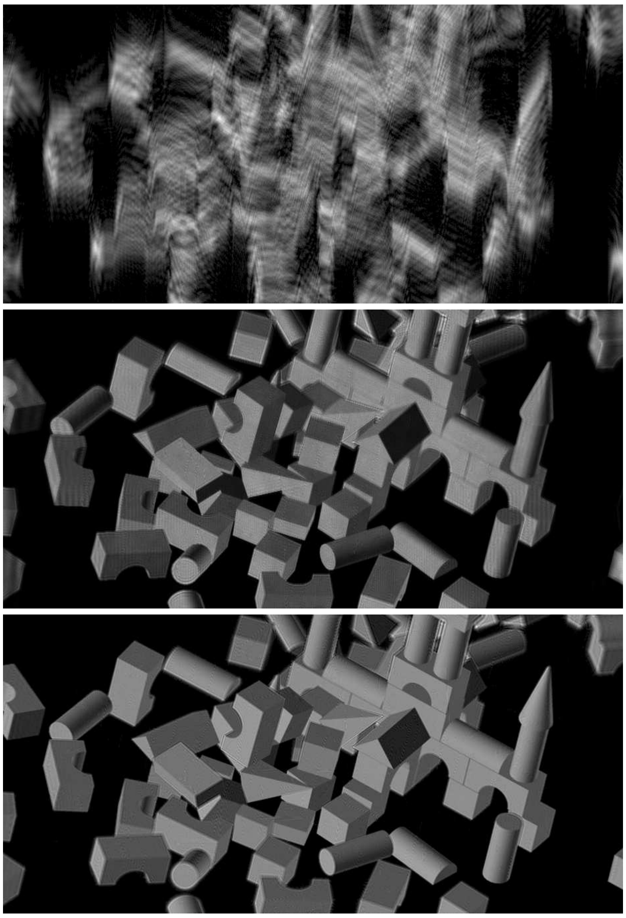

Fig. 3. Comparison of detail from reconstructins of the HPO (top), our HPO (middle), and full parallax (bottom) diffraction patterns using the numerical reconstruction intended for full parallax propagation.

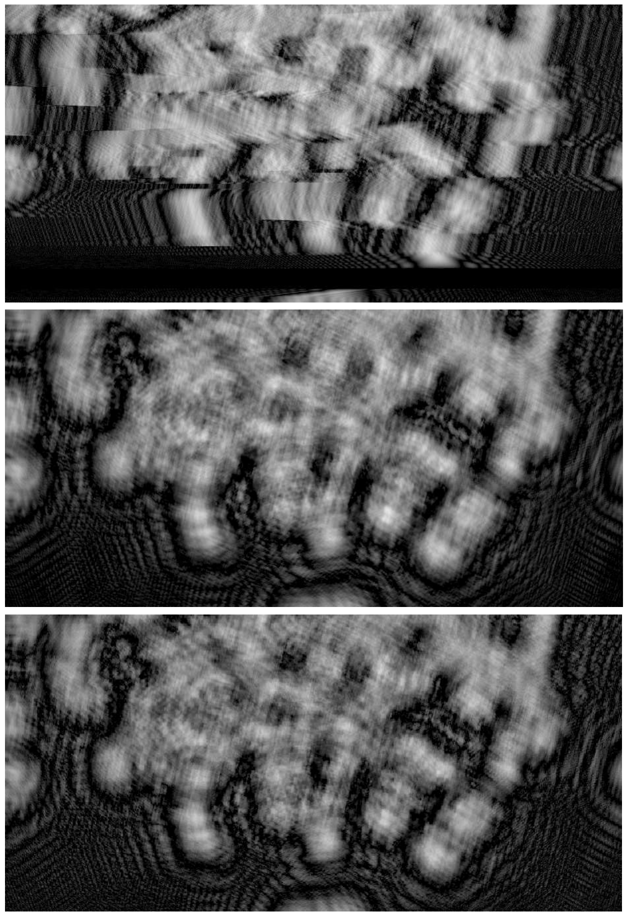

Fig. 4. Comparison of detail from HPO (top), our HPO (middle) and full parallax (bottom) inline holograms.

5. REFERENCES

[1] P. Hariharan, *Optical Holography: Principles, techniques and applications*, Cambridge University Press, 2nd edition, 1996.

[2] J.W Goodman, *Introduction to Fourier Optics*, Roberts & Company Publishers, 3rd edition, 2005.

[3] M. Lucente, *Diffraction-Specific Fringe Computation for Electro-Holography*, Ph.D. thesis, MIT, 1994.

[4] Y. Li, D. Abookasis, and J. Rosen, "Computer-generated holograms of three-dimensional realistic objects recorded without wave interference," *Appl. Opt.*, vol. 40, no. 17, pp. 2864–2870, 2001.

[5] N.K. Govindaraju, S. Larsen, J. Gray, and D. Manocha, "A memory model for scientific algorithms on graphics processors," Tech. Rep. MSR TR 2006 108, Microsoft, 2006.

[6] M. Janda, I. Hanák, and V. Skala, "Digital HPO hologram rendering pipeline," in *EG2006 short papers conf. proc.*, 2006, pp. 81–84.

[7] C. Petz and M. Magnor, "Fast hologram synthesis for 3d geometry models using graphics hardware," in *Practical Holography XVII and Holographic Materials IX*. SPIE, 2003, vol. 5005, pp. 266–275.

[8] A. Ritter, J. Böttger, O. Deussen, M. König, and T. Strothotte, "Hardware-based Rendering of Full-parallax Synthetic Holograms," *Appl. Opt.*, vol. 38, pp. 1364–1369, 1999.

[9] G.B. Esmer and L. Onural, "Computation of holographic patterns between tilted planes," in *Holography 2005*. SPIE, 2006, vol. 6252, p. 62521K.