A GENERALIZED PIPELINE FOR PREVIEW AND RENDERING OF SYNTHETIC HOLOGRAMS

Ravikanth Pappu Carlton Sparrell* John Underkoffler
Adam Kropp Benjie Chen Wendy Plesniak

{pappu, carltonj, jh, akropp, benjie, wjp}@media.mit.edu

Spatial Imaging Group, MIT Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139.

*Interval Research Corporation
Palo Alto, CA 94304.

ABSTRACT

We describe a general pipeline for the computation and display of either fully-computed holograms or holographic stereograms using the same three-dimensional database. A rendering previewer on a Silicon Graphics Onyx allows a user to specify viewing geometry, database transformations, and scene lighting. The previewer then generates one of two descriptions of the object - a series of perspective views or a polygonal model - which is then used by a fringe rendering engine to compute fringes specific to hologram type. The images are viewed on the second generation MIT Holographic Video System. This allows a viewer to compare holographic stereograms with fully-computed holograms originating from the same database and comes closer to the goal of a single pipeline being able to display the same data in different formats.

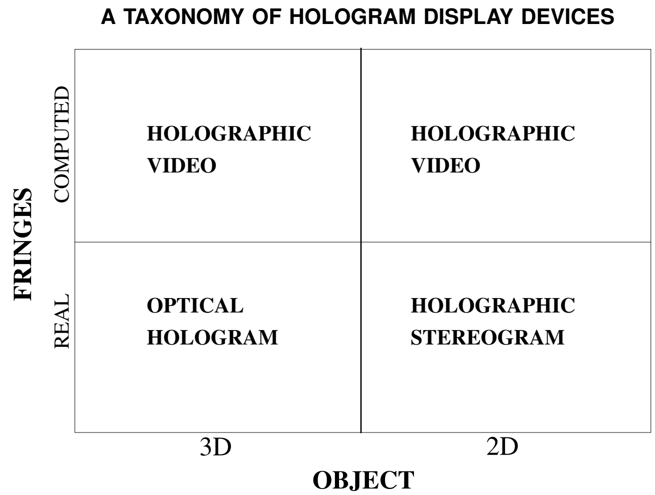
KEYWORDS: computer-generated holograms, holographic stereograms, holographic video

1. INTRODUCTION AND TAXONOMY

The MIT Holographic Video System (holovideo) is designed to display computer-generated holograms. At the heart of the system is an acousto-optic spatial light modulator which modulates a laser beam with a computed fringe pattern. The computational and opto-mechanical aspects of the display are documented elsewhere [1], so we will not dwell on them here. For the present purpose, we may consider holovideo to be a black box which accepts two inputs: a computer-generated hologram (CGH) and light. The output of the black box is a horizontal parallax only (HPO) three-dimensional image which intimately depends on how the CGH was computed.

In order to situate holovideo in the pantheon of holographic display devices we offer a taxonomy of display devices for holograms. The nature of both the object and the fringe pattern is the basis for categorizing these devices. The recognition that objects may either be three-dimensional (physical or modeled) or a sequence of perspectives and that fringe patterns also exhibit a dichotomy (real or computed) allows us to apportion the space of all display holograms into four smaller spaces, as shown in the table below. For example, the production of an *optical hologram* demands the existence of a *3D object* and results in a *real fringe pattern* recorded in photosensitive material. At the other extreme, it is possible to start with a computer-graphic *three-dimensional model* of a scene and generate a *computed holographic fringe pattern* from it. In this case, the fringe pattern is simply an array of numbers; it is incapable of diffracting light until is loaded into an SLM which modulates a beam (or beams) of light. One of the two intermediate states - 2D object/computed fringes - also requires an electronic holographic display. In general, an electronic holographic display is required to turn *any* computed fringe pattern into an image.

In this paper, we will consider two distinct ways of generating the CGH from the same three-dimensional database and describe a pipeline for dispatching the computed fringe pattern to the display. The source and destination of the pipeline are identical for both types of CGHs but the intervening algorithmic and electronic subsystems are distinct. We are implementing a generalized pipeline from the database to the display for two reasons. First, there is a practical trade-off to be considered. Fully-computed holograms offer the greatest image realism but necessitate an exact simulation of the interference process to generate the fringe pattern. In this instance, continuous surfaces on the object are



modeled as a collection of closely-spaced luminous points specified by their location, amplitude, and phase. Holographic stereograms, on the other hand, furnish but an approximation to the required object wavefront while their computation is far more rapid. In this instance, the object is represented as a series of perspectives which are input to a fringe rendering system. In a practical visualization system, the viewer must be able to exercise the option of making the trade-off. The second and more philosophical ground for implementing this pipeline is the fact that holovideo is designed to reconstruct *any* wavefront - it does not know anything about the fringe pattern; rather, the fringe pattern is aware of holovideo parameters. This being the case, it should be possible to use holovideo to display the database in different formats. The implementation presented in this paper is an example of the display's generality of purpose.

This paper reports on the implementation of the pipeline from a system-integration viewpoint. It pulls together work done at the Spatial Imaging Group over the past few years and presents a unified picture of the holographic video signal chain. The structure of this paper is as follows. In the following two sections, the computation of fully-computed holograms and holographic stereograms is described and the issues involved in the process are delineated. These sections will not present the algorithms in minute detail; rather, an overview of the computation process is presented and appropriate references are cited to whet the appetite of the interested reader. Section four will introduce the pipeline and describe details of its implementation. Results are presented in section five. We offer concluding remarks and suggest future directions in the final section.

2. FULLY-COMPUTED HOLOGRAMS

By "fully-computed hologram" we are referring to a fringe pattern which is the result of a computational process that models the actual interference process as closely as possible. In the ideal case, the wavefront reconstructed by the display will be perceptually indistinguishable from that of the object. In practice, however, there are significant departures from this ideal. These departures, which we will point out in the ensuing discussion, are a consequence of shortcuts taken to make the computation and display of fringe patterns tractable and also due to technological limitations.

The computation of the fringe pattern may be dissected into three subsystems: scene modeling, occlusion processing, and interference modeling, as shown the Figure 1. The scene modeling subsystem uses standard computer-graphics techniques to generate a shaded polygonal description of the scene. Our three-dimensional representational scheme is the Ohio State University (OSU) format. Recalling that the object is viewed as a collection of point sources, we then populate each polygon with a series of points and assign a location, amplitude, and phase to each of them. Artifacts of spatially and temporally coherent illumination may be diminished by randomly varying the inter-point spacing and assigning uniformly distributed random initial phases to them. Approximating continuous surfaces with a multiplicity of point radiators is the first divergence from the ideal case. However, the wavefront reconstructed in this case can be made perceptually indistinguishable from the ideal wavefront by selecting the point spacing based on psychophysical considerations.

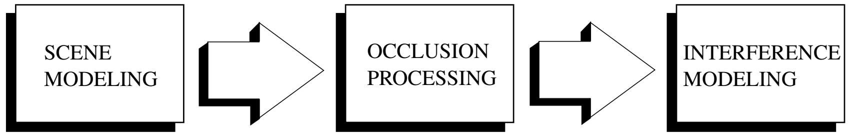


FIGURE 1: FULLY-COMPUTED HOLOGRAMS: COMPUTATIONAL SUBSYSTEMS

The point list is then passed to an occlusion processing system, dubbed *Occfilt*, whose function is to determine the regions on the hololine (recall that we only compute HPO holograms) that are contributed to by each point. Occlusion processing, as implemented in [2], is based strictly on geometrical optics, and no heed is paid to the (weak) diffractive effects at the edges of obstructions. This is a further deviation from the ideal case, although the "edge-wave" is too weak to be perceptible by a human viewer. For each point radiator in the database, Occfilt returns a set of regions on the hololine that it contributes to. All that remains is to render the fringe pattern.

Fringe rendering is accomplished by simulating the classical interference equation for each sample on the hololine. The complex amplitude from each subscribing point radiator is totalled and the intensity at the current sample is determined. The final result is normalized to the range 0-255 because the fringe samples are represented as 8-bit bytes in the framebuffer. Normalization introduces quantization noise into the fringe pattern; another source of wavefront degradation.

3. COMPUTER-GENERATED HOLOGRAPHIC STEREOGRAMS

A holographic stereogram uses wavefront reconstruction to present a finite number of perspectives to a viewer. A fully-computed hologram, in contrast, presents an infinite number of perspectives. This discretization of perspective results in a discrete approximation to the ideal wavefront [3]. Approximating a continuous wavefront by a discrete one permits us to make a significant dent in computation time.

The underlying principle in our approach to stereogram computation is to dissect the process into two independently determinable parts [4]. The diffractive part is unvarying and can be computed ahead of time and looked up from a table when required. This is achieved by positing a set of basis functions, each of which has the function of diffracting light in a certain direction. Each of these basis functions is scaled by a number determined from the perspectives. The superposition of these scaled basis functions results in a *hogel* or holographic element: a small segment of the hololine (see Figure 2). If the perspectives change, only the multipliers are to be determined before the superposition can be carried out again. Interference and diffraction have now been functionally replaced by analysis and synthesis: analysis is performed once and new fringe patterns can be synthesized at will by changing the coefficients.

Computer-generated holographic stereograms are extremely versatile in that they admit input data from a variety of sources. All that is needed in order to obtain perspective data is a scanning camera of some sort - no ability to generate a three-dimensional model of a scene is required. Further, occlusion processing may be incorporated into the computer-graphics pipeline and the superposition method described above can readily be implemented in hardware. One such piece of hardware is the *superposition stream processor* [5] that can be used in conjunction with our framebuffer.

Despite their computational appeal, computed stereograms suffer from certain drawbacks. The abrupt change in phase from one perspective to another is readily perceived by the human visual system; a detriment not observed in fully-computed holographic images. This phase jump is also responsible for enhancing speckle when spatially and temporally coherent illumination is used. Finally, increasing the number of perspectives with the same 1 byte/sample framebuffer leads to a decrease in the dynamic range available to each basis function, which in turn manifests itself as a decrease in diffraction efficiency. As for the fully-computed case, intelligent normalization is the key to remedying this problem.

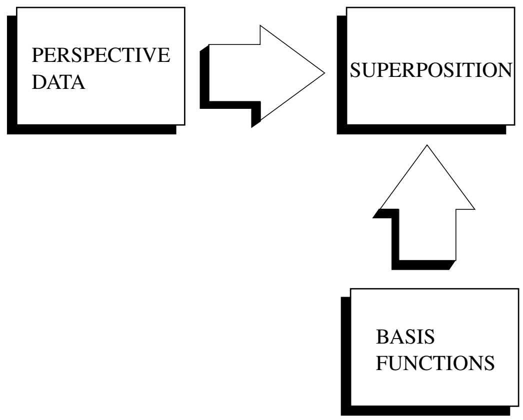


FIGURE 2: HOLOGRAPHIC STEREOGRAMS: COMPUTATIONAL SUBSYSTEMS

4. PIPELINE DESCRIPTION

The final pipeline is fashioned as in Figure 4. We rely on three different platforms to accomplish the required computational duties. The Silicon Graphics Onyx - a specialized graphics machine - runs the rendering previewer. The previewer, christened *Holobuild*, allows a user to compose a scene with several objects and assign unique material properties and lighting information each of them. This previewer enables real-time previewing of images and is a shared front-end for holographic hardcopy, holovideo, and tangible holography. Its functionality includes the generation and storage of either perspective views or a three-dimensional model, control of the position, orientation, and scaling of each object, and complete specification of all display parameters [6]. At the user's urging, Holobuild generates two sets of data: a file in OSU format which is input to Occfilt and several perspectives which are input the stereogram computation program.

The fully-computed hologram routine is currently implemented on an SGI Onyx workstation as shown in Figure 3. The three modules may all be used via the holobuild interface. The computed hologram, which takes on the order of tens of minutes, depending on object complexity, is then dispatched to Cheops via a SCSI link at about 10 MB/s. The image is then viewable on the display. Clearly, these rates are not suitable for interactivity, and work is in progress to expedite the process. The major saving in time will come from porting the computation code to the IBM Power Visualization System (PVS), which will allow a speedup of a factor of 40. This speedup is evident from the stereogram computation times detailed in the following section. Although hologram computation is not an inherently parallelizable process, the fact that we are computing independent lines of the fringe pattern allow us to distribute the computation onto several processors.

The stereogram pipeline makes full use of the processing power of the PVS. The PVS is a 32-processor parallel computer with 16 MB of local memory per processor. This memory is a convenient storage location for the basis functions, which are a mere 32 KB, and allows for rapid access. The perspective data is made available to the stereogram computing program via a HiPPI link whose maximum throughput is 100 MB/s. The stereogram is computed and sent to Cheops via another HiPPI link for display. The total time, excluding perspective generation, is approximately 14 seconds and is independent of object complexity. This time is within an order of magnitude of the time required for full interactivity.

5. RESULTS AND DISCUSSION

Before we present the results of the computation, we present a tabular summary of the differences in the computing process. The table is self-explanatory and alludes to remarks made in sections 2 and 3. We point out the critical differences between the two types of computed fringe patterns at various stages in the pipeline: object stage, fringe computation stage, and image stage.

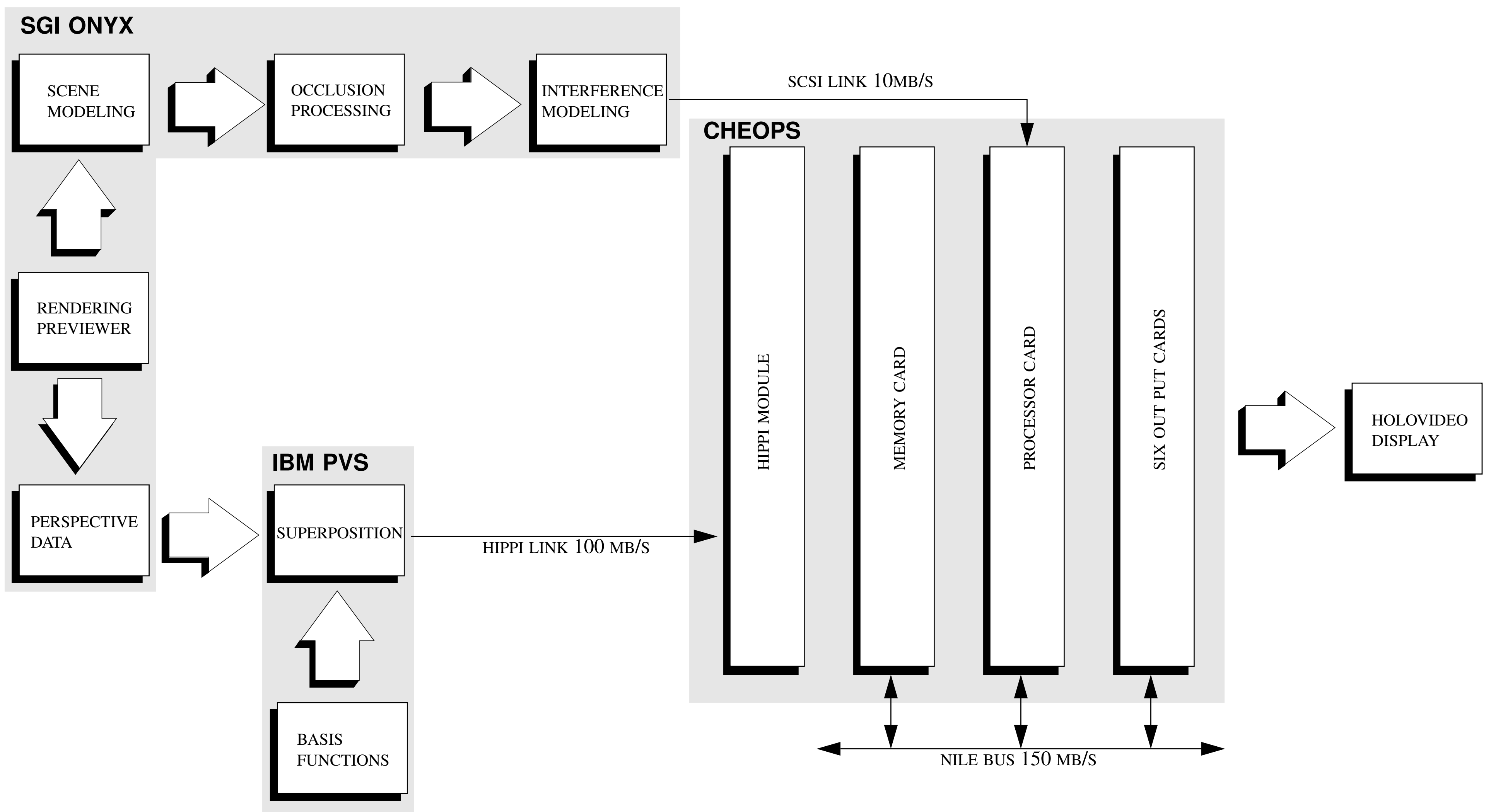


FIGURE 3: GENERALIZED PIPELINE FOR SYNTHETIC HOLOGRAMS

DIFFERENCES BETWEEN THE TWO TYPES OF COMPUTED FRINGE PATTERNS

		FULLY-COMPUTED HOLOGRAMS	COMPUTER-GENERATED STEREOGRAMS
OBJECT	REPRESENTATION	Polygonal model	Sequence of perspectives
	RADIATORS	Point radiators distributed in a volume	Pixels from perspectives; radiating from a fixed plane
	COMPLEXITY	Depends on the number of polygons and the spacing of the points populating them	Independent of the object
	RESOLUTION	Variable; can be made as low/high as required	Fixed at 256x144 pixels per perspective in current implementation
	OCCLUSION	Implemented as a separate subsystem.	Handled by the computer-graphics renderer
FRINGE COMPUTING	METHOD	Simulation of interference	Diffraction-specific fringe computation
	TIME	Proportional to object complexity	Independent of object
IMAGE	REALISM	Infinite number of perspectives; Reconstructed wavefront can approximate object wavefront very closely	Finite number of perspectives; phase-discontinuities present in reconstructed wavefront
	ARTIFACTS	No additional coherent illumination artifacts introduced by computation process	Speckle introduced by phase discontinuities in reconstructed wavefront.

We present a summary of computation times for both types of fringe patterns below. We also include previously published results [5] for stereogram computation using one and two superposition stream processors.From these results, it is evident that, in the near future, interactivity can only be addressed by using stereographic images. Unless one is using a supercomputer, the best possible scenario is when the fringe computation is performed as close to the display end of the pipeline as possible.

COMPARISON OF COMPUTATION TIMES

	OBJECT	PLATFORM	TIME
FULLY-COMPUTED HOLOGRAM	14640 polygons; 10 points per mm	SGI Onyx	1965 seconds
COMPUTER-GENERATED STEREOGRAM	32 perspectives; 256x144 pixels per perspective	IBM PVS	5 seconds
		SGI Onyx	243.9 seconds
		Cheops with 1 superposition stream processor	12 seconds
		Cheops with 2 superposition stream processors	6 seconds

6. CONCLUSIONS AND FUTURE WORK

We have described the current state of the second generation MIT Holographic Video system in terms of a generalized pipeline that allows a viewer to compute one of two types of fringe patterns: fully-computed holograms and computer-generated holographic stereograms. It is possible to view the 3D database with a rendering previewer and make alterations to its position, orientation, and lighting. A stereogram or fully-computed hologram may then be generated and displayed on holovideo. Several differences between the two species of fringe patterns at various stages in the pipeline were identified. Computation times were provided in both cases.

Several issues merit further consideration. The fundamental issue of speeding up the computation needs to be addressed if interactivity is important, and it is. In hologram computing, a system to decouple the fringe computation process from the complexity of the object would be well worth investigating. Some efforts in this direction were made by Underkoffler but there is abundant room for refinement. Normalization of the final computed hologram to occupy the available dynamic range of the SLM is not straightforward and bears further probing. Deciding on efficient basis function for stereogram computing is also a problem waiting to be solved. For both species, efficient representations and reduction of coherent illumination artifacts remain to be determined.

ACKNOWLEDGMENTS

Parts of this research were sponsored by the Honda R&D company, IBM Corporation, NEC Corporation, the Television of Tomorrow Research COnsortium at the MIT Media Laboratory, and the Advanced Research Projects Agency.

REFERENCES

[1] St. Hilaire, P., *Scalable Optical Architectures for Electronic Holography*, Ph.D Thesis, MIT Program in Media Arts and Sciences, Massachusetts Institute of Technology, (1994).

[2] Underkoffler, J., *Toward Accurate Computation of Optically Reconstructed Holograms*, SM Thesis, Media Arts and Sciences Section, Massachusetts Institute of Technology, (1991).

[3] Halle, M., *The Generalized Holographic Stereogram*, SM Thesis, Media Arts and Sciences Section, Massachusetts Institute of Technology, (1991).

[4] Lucente, M., *Diffraction-Specific Fringe Computation for Electro-Holography,* Ph.D Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology (1994).

[5] Lucente, M., *et. al*., A hardware architecture for rapid generation of electro-holographic fringe patterns, in S. A. Benton, ed., Practical Holography IX (1995).

[6] Halle, M., and Kropp, A., Fast computer graphics rendering for full-parallax spatial displays, in S. A. Benton, ed., SPIE Proceedings, Practical Holography XI (1997 - current issue)