Chidi holographic video system.

Thomas A. Nwodoh and Stephen A. Benton.

MIT Media Laboratory

20 Ames Street, Cambridge MA 02139 USA

ABSTRACT

Holo-Chidi is a holographic video processing system designed at the MIT Media Laboratory for real-time computation of Computer Generated Holograms and the subsequent display of the holograms at video frame rates. It's processing engine is adapted from Chidi which is a reconfigurable multimedia processing system used for real-time synthesis and analysis of digital video frames. Holo-Chidi is made of two main components: the sets of Chidi processor cards and the display video concentrator card. The processor cards are used for hologram computation while the display video concentrator card acts as frame buffer for the system. The display video concentrator also formats the computed holographic data and converts them to analog form for feeding the acousto-optic modulators of the Media Lab's Mark-II holographic display system. The display video concentrator card can display the computed holograms from the Chidi cards loaded from its high-speed I/O (HSIO) interface port or precomputed holograms loaded from a PC through the Universal Serial Bus (USB) port of its communications processor at above video refresh rates. This paper discusses the design of the display video concentrator used to display holographic video in the Mark-II system.

Keywords: Holographic video, frame buffer, fringe computation, hololines.

1. INTRODUCTION

A hologram is a record of the complex interference patterns between light from objects in a scene illuminated by a suitable coherent light source and another mutually coherent light source (called the reference source). The interference pattern recorded contains information about both the amplitude and phase of the light scattered from the objects in the scene. Beam pairs interfere and their "phase footprint" is recorded on the hologram plate. Every light wave from a point of the 3-D object(s) in the scene interferes with the reference beam and with one another to produce a "picket-fence-like" super-imposition of interference patterns (called fringe patterns) which are imprinted on the hologram plate. To display the hologram, it is illuminated by a suitable reconstruction light beam. The resulting diffracted light produces the holographic image which reconstructs the optical wavefronts that fell on the plate from the objects in the scene during the recording phase [1 2].

A computer-generated hologram (CGH) has an array of grey-scale data equivalent to an optical hologram's fringe patterns. The digital samples represented by the grey scales are generated by modeling and sampling the complex field in the hologram plane. The generation of the CGH frame starts with a 3-D description of the scene such as obtained with: computer-aided design models of objects, stereographic perspective views of a 3-D scene, or from other scientific databases. The grey scales are then generated by sampling at suitable intervals.

In Holo-Chidi, two basic approaches are used for hologram computation - the fully-computed hologram approach and the holographic stereogram approach. The fully-computed hologram approach renders the hologram fringes by simulating the light interference process. Three steps are used for this approach: scene modeling, occlusion processing, and interference modeling [3 4 5]. The holographic stereogram approach involves the angular multiplexing of a finite number of two-dimensional perspective views of the scene and the encoding of the result into the holographic fringe pattern.

At the MIT Media Laboratory our horizontal parallax only (HPO) computer-generated holograms have a size of 36MB organized as 256KB per hololine (a horizontal line of the fringe pattern) by 144 hololines. The holographic video (holovideo) images displayed on the Mark-II holographic display system has an image volume of width 135mm, height 70mm, and a view zone of 30^o. Hologram computation is carried out on the Chidi processor cards, where a large field programmable gate array (FPGA) called RP (Reprogrammable Processor) is configured to implement the hologram computation algorithm. The computed hologram is either stored for later display or transmitted to the display video concentrator for display in real time. The display video concentrator formats the computed hologram from multiple Chidi processor cards into the the format needed to display the hololines. The card acts as a frame buffer for the holograms to be displayed. It also converts the digital grey scales to analog form used to drive the Mark-II. In the Mark-II display, the computed hologram fringes now in analog form, are passed through a radio frequency signal processing circuit where they are modulated onto a high frequency carrier, amplified, and then fed into the AOM (Acousto-Optic Modulator) array. The signals fed into the AOM then phase modulate a spatially filtered, expanded, collimated, coherent illumination laser beam so that when a hololine of the hologram frame is fed through the AOM, it produces a diffracted wave conforming to the fringe patterns of the hololine. The AOM is a spatial light modulator that diffracts light from an illumination beam by a change in the refractive index of the crystal caused by sound waves launched across the crystal. In the Mark-II display there are eighteen AOMs , each processing the 256KB data that form a hololine into a holographic image for a line on the image volume. The hololine is a horizontal slice of the frame. The image of the entire object is then optically "assembled" by scanning and spatially multiplexing the diffracted beams from the AOMs to the display using mirrors and lenses. To a viewer looking at the image volume, the image appears tangible, displaying the depth cues of horizontal motion parallax and partial accommodation. See [6] [7] [8] [9] [10] for the setup used for MIT Media Lab's Mark-II system.

2. CHIDI PROCESSOR CARD

In the Holo-Chidi system, holograms are computed on the Chidi processor cards. The main architectural blocks in Chidi are shown in Fig. 1 [11] [12]. The GPP is a PowerPC 7400 processor running at up to 500MHZ with 64 data bits and 32 address bits. Its function is to control the operation of the entire Chidi processor card by running programs and reading/writing the control registers of the specialized processors.

The PCI bridge interfaces the Chidi card to a PCI bus. Through this interface, the Chidi card can access data on a host machine and the host can access Chidi's memory. The PCI bus, which is capable of a sustained bandwidth of 264MB/s, was chosen because of its ubiquity and robust bus features. With the PCI interface, Chidi can be interfaced to different host platforms which can furnish the system with data archival capabilities. The PCI interface is implemented with an MPC107 - PowerPC/PCI bridge configured for multiple PowerPC processors. In addition to acting as the PCI bus gateway the MPC107 also acts as the system's memory and ROM controller. The main memory has a 128MB DRAM and 512KB flash EPROM. The EPROM holds the system boot code while the DRAM stores programs and data. The PCI bus is compliant with PCI standard 2.1 with 32 bit data/address bus running at up to 66MHz.

The Stream Address Generator (SAG, an FPGA) is configured as an additional processor tied to the PowerPC bus. The SAG generates the stream addresses for data moving to the RP and DS circuits. Recorded hologram data being displayed from a disk resident on the host flows in a stream fashion from the PCI interface through the DS and RP to the HSIO port, whence they are transmitted for display. The SAG which is under the control of the GPP sets up this data transmission by generating all the addresses needed by the RP and the DS with all the handshake signals needed to implement the data flow. If the hologram is stored in the local main memory, the SAG which can

Figure 1: Chidi processor blocks

read/write from/to memory also controls the data flow through the DS and RP to the HSIO port. Additionally, the SAG acts as a local bus slave controller on the PowerPC bus. When the bus is in the slave mode, the DS using the SAG's slave control signals can access the PowerPC bus.

The DS (Data Shuffler) controls the flow of stream data between the PowerPC bus and the RP. It basically contains buffers, FIFOs, comparators, data flow state machines, and registers needed to move stream data using the Media Lab's Stream Processing mechanism [13]. Sixty four bits of data can flow through the DS in either direction simultaneously.

The RP is the specialized computing hardware in this architecture. It is dynamically reconfigurable by the processor and can be used to compute a wide range of algorithms including discrete cosine transform, matrix transpose operation, fast Fourier transform, and superposition stream processing. It can input or output data from/to either the DS or the HSIO port. When it is configured for a given algorithm, it processes the input data, storing intermediate results on the 2MB external SRAM tied to it, and outputs its results either to the HSIO port or to main memory. It can simultaneously take 64 bits of data from or to the DS and has 32 bit data access to and from the HSIO port. Interface to the HSIO is implemented with National's LVDS (Low Voltage Differential Signaling) chipset. Each transmitter or receiver can transmit/ receive 18 bits of TTL data at a rate of up to 170MB/s. Using two transmitters, a CHIDI processor card can transmit data to a display card at rate of up to 340MB/s.

Both the SAG and the DS are implemented with Altera 10K50 SRAM-based FPGA chip, which has 274 usable IOs on a 356 pin BGA package, while the RP is Altera 10K100 SRAM-based FPGA with 403 usable IOs on 503 pin PGA package. The DS and SAG are programmed after system initialization using a serial EPROM tied to their configuration ports. The RP is programmed by the processor with controls generated by the SAG using configuration data loaded in memory. In effect, the RP is the only dynamically reprogrammable device in Chidi as only it can be

Figure 2: Data input for holovideo computation

reprogrammed as needed by the processor at runtime.

3. HOLOGRAM COMPUTATION

In the Holo-Chidi system, the principle used for the generation of the hologram fringes is the Diffraction-Specific Fringe Computation algorithm reported in [14], which is based on the discretization of space and spatial frequency in the fringe pattern. The 3-D scene is modeled with an array of data called *hogels* (for holographic element), each having a uniform spectrum and discretized into equal regions. Then, generated from a 3-D description of the scene are *Hogel Vectors* (such as the stereogram's perspective views), each of which describe the spectrum of the associated Hogel. To generate the fringes, each Hogel Vector is used as an array of coefficients of the associated Hogel. The Hogel-Vector encoding technique, which is associated with the Diffraction-Specific Fringe Computation algorithm, is used to generate the holographic fringes from the scene description. Each Hogel Vector component (ie. the perspective data) is multiplied with a component from a set of precomputed *Basis Fringes* (which supply the diffraction component). Each basis fringe is computed such that it contains energy in a particular region of the spectrum and controls the directional behavior of diffracted light [15]. By multiplying each Basis Fringe with its corresponding Hogel Vector component and summing the results, the resulting fringes contains the spectral energy specified by the Hogel Vector. This generally involves performing a matrix inner product operation.

3.1. Holovideo algorithm

The RP configured as a superposition stream processor performs the computationally intensive operation of generating the 36MB fringes by multiplying and accumulating the product of the Hogel Vectors and the Basis Fringes. As shown in Fig. 2, the Basis Fringe is a 1 by 1024 array of data while the Hogel Vector is a 256 by 144 image frame array representing the individual views of the scene.

The holovideo algorithm was implemented for the RP, written in VHDL, and simulated and synthesized using SYNOPSYS. Instantiations of multipliers, accumulators, and dividers were used to process the data streams coming from memory.

The holovideo algorithm involves a series of: additions of the product of a byte, ($\mathbf{basis}_{(i(n))}$ in Fig. 2), from a 1 by 1024 basis function array and a byte, ($\mathbf{pixel}_{(i(j,k))}$ in Fig. 2), from a 256 byte by 144 line stereogram view of the scene whose hologram is being computed. There are thirty-two image frame stereograms representing thirty two different 2-D perspectives views of the scene and there are thirty two precomputed basis functions. However, Holo-Chidi can support arbitrarily large number of perspective views and basis functions. The multiplications are carried out between image frame and basis function byte pairs as illustrated in Fig. 2. Each multiplication results in a 16 bit product whose corresponding values are accumulated over all 32 frame and basis function pairs yielding a 21 bit value, $\mathbf{holo\_value}_{(j*n,k)}$ as in:

$$\mathbf{holo\_value}_{(j*n,k)} = \sum_{i=1}^{32} \mathbf{pixel}_{(i(j,k))} * \mathbf{basis}_{(i(n))} \tag{1}$$

Where j has values from 1 to 256, k is the hololine index with range from 1 to 144, and n has range from 1 to 1024.

Each $\mathbf{holo\_value}_{(j*n,k)}$ is subsequently normalized to yield a byte of the final hologram frame. Normalization serves to fit the value of $\mathbf{holo\_value}_{(j*n,k)}$ to the display system used. Mark-II uses 8 bit fringes. Normalization involve the computation of:

$$\mathbf{holo\_pixel}(j*n,k) = \left(\frac{\mathbf{holo\_value}_{(j*n,k)} - \alpha_{min}}{\alpha_{max} - \alpha_{min}}\right) * \mathbf{255} \tag{2}$$

where α_{max} is the largest un-normalized value for the entire hologram frame (that is maximum value of $\mathbf{holo\_value}_{(j*n,k)}$) and α_{min} is the least un-normalized value (typically zero).

A multiply-accumulate module is used to generate the sum of the product of the image frame - basis function byte pairs while a divider is used for normalization. Fig. 3 shows the functional blocks used for holovideo computation in the RP. The comparator is used to generate the value of α_{max}. The multiply-accumulate-compare process uses several pipelines, each of which is made of 8 bit by 8 bit multipliers producing 16 products, 16 bit by 21 bit adders with 21 bit sums, and 21 bit comparators. Each normalizer pipeline has a divider with 21 bit numerators and 13 bit denominators giving an 8 bit quotient representing the final hologram fringe. The numerator used for the division are the $\mathbf{holo\_value}_{(j*n,k)}$ values while the denominator used is $\frac{\alpha_{max}}{255}$, which is a constant. The normalization phase of the computation commences after the value of α_{max} and α_{min} have been determined. The comparator placed at the output of the multiply-accumulate pipeline compares each $\mathbf{holo\_value}_{(j*n,k)}$ with the current value α_{max} to determine the α_{max} for the entire hologram frame. α_{min} is assumed to be zero here. Because the holovideo algorithm is not very sensitive to the amplitude of the fringes, a divider denominator value of of 2^{13} is used in this implementation. This is the maximum value that $\frac{\alpha_{max}}{255}$ can attain.

4. HOLO-CHIDI DESIGN

The Holo-Chidi system uses Chidi processor cards for hologram computation. The video concentrator cards are used to control display the computed holograms in Mark-II. The video concentrator can take in data from two main sources: the HSIO port whence data from Chidi processor cards can be loaded, and the USB port whence precomputed holograms may be loaded from a host PC.

4.1. The display video concentrator

The display video concentrator interfaces Chidi processor cards with the Mark-II display. The following constraints determined the design of the display card:

Figure 3: Functional blocks for holovideo application

Figure 4: Layout of Holo-Chidi system

Figure 5: Block diagram of the display video concentrator

- Since the Mark-II display has no memory, a holoframe being displayed must be refreshed at above the critical flicker frequency (30 frames per second). A 36MB frame being refreshed at 30 frames per second requires a 1.14 gigabytes per second display bandwidth.

- Mark-II display requires 18 inputs to the acousto-optic modulators. The system needs to support this data width so as to transmit the 18 hololines simultaneously to the display. The bandwidth of each AOM is 110Mbytes/s. The display card must be able to feed each AOM channel with data at that rate.

- A hololine in Mark-II must be read without interruption until the end of the line. Any gaps in the bit stream would be visible on the image volume [5]. Hence 256KB hololine has to be fed to each channel of the AOM as a continuous data stream. To achieve this, a buffer capable of holding data for the 8 hololines (2MB) of a frame being displayed per AOM channel is required on the display video concentrator card.

- A Chidi card supports a data width of up to 4 bytes at its output port. 4 bytes is thus the maximum width of data that can be sent from a Chidi card to the display card in one Chidi HSIO clock cycle.

Fig. 5 is a block diagram of the display card. Chidi's HSIO port is an LVDS interface circuit, hence the input port of the display card has an LVDS interface chipset. An LVDS receiver, DS90CF364 receives two bytes every LVDS clock cycle from a Chidi card. The LVDS receivers convert the signals from LVDS signals to TTL. Nine LVDS channels on the display video concentrator card receive eighteen bytes of TTL data per LVDS clock cycle. The D-to-A Converters (DACs) that feed the acousto-optic modulators of Mark-II are loaded 4 bytes at a time. Consequently, each byte lane from the LVDS interface is formated into 4 bytes lanes by using latches. The 4 latched bytes are then loaded into 4 FIFOs that are used as a buffer for data for a hololine coming from the HSIO port.

Data received in the FIFO are either sent to the DACs for transmission to the display for display in real-time or they are assembled in the SDRAM which acts as a frame buffer. The SDRAM has two 16 Meg by 16bit chips per AOM channel. The SDRAMs on the card can hold up to thirty two 36MB hologram frames.

Precomputed holograms can also be loaded from a host PC through the USB port of the display video concentrator. The data is assembled in the display video concentrator's main memory (DRAM) and moved to the SDRAMs for display.

The DACs convert the 8 bit digital data fed from the FIFOs to an analog video signal which are then transmitted to a radio frequency signal processing circuit resident in the Mark-II display.

A communications processor, MPC850 control the operation of the display video concentrator. It has a PowerPC core, a RAM controller, and several IO interfaces [(including USB and Inter-IC Controller (I^2C)]. The USB port is used for loading precomputed holograms from a host PC to the frame buffer. The card's main memory is the 128MB DRAM.

There is a tight control of the reception of the data streams from either the Chidi cards or from the USB port, its formatting into different hololines, and subsequent transmission to Mark-II. An FPGA controls the start/end of both loading and off-loading of the FIFOs. The FPGA also loads control data into the DACs. In addition, the FPGA is used to generate signals that control the scanning mirrors in Mark-II.

5. CONCLUSIONS

The architecture of Holo-Chidi, a research tool being developed by the Spatial Imaging Group of MIT Media Laboratory for holovideo computation and display, has been discussed. Prototype versions of the Chidi cards have been built and tested. A new hardware revision for the Chidi cards, the development of the display video concentrator card, and the integration of both sets of cards with a host system are all currently in progress. The system when fully developed will make possible the computation in real-time and display at video rates of holograms of complex 3-D structures providing binocular and monocular cues to a viewer. The architecture which is scalable allows an increase in the number of Chidi cards as higher data throughput is needed. This makes Chidi very suitable for the holovideo program of the Media Laboratory - bigger sized holograms can be computed, higher resolution can be built in, and full-parallax holograms are possible, just by suitable increases in the number of cards.

ACKNOWLEDGMENTS

The research on Holo-Chidi is supported by the Digital Life Consortium Of the MIT Media Laboratory.

REFERENCES

1. J. W. Goodman, *Introduction to Fourier Optics, Second Edition*, McGraw-Hill, 1996.

2. P. M. Hariharam, *Optical Holography*, Camdridge University Press, 1984.

3. Ravikanth Pappu, Carlton Sparrell, John Underkoffler, Adam Kropp, Benjie Chen, and Wendy Plesniak, "A Generalized Pipeline for Preview and Rendering of Synthetic Holograms", *SPIE Conference on Practical Holography XI*, San Jose, 1997.

4. John Underkoffler, "Occlusion Processing and Smooth Surface Shading for Fully Computed Synthetic Holography", *SPIE Proceedings*, vol. 3011, pp. 19-30, 1997.

5. Mark Lucente, "Interactive Computation of Holograms Using a Look-up Table", *Journal of Electronic Imaging*, vol 2(1), pp. 28-34, 1993.

6. Pierre St.-Hilaire, "Scalable Optical Architectures for Electronic Holography", Ph.D Thesis, Program of Media Arts and Sciences, MIT, 1994.

7. Steve A. Benton, "The Second Generation of the MIT Holographic Video System", *Invited paper: First International Symposium on Three-Dimensional Image Communication Technologies*, 1993.

8. Mark Lucente, Steve Benton, Pierre St. Hilaire, "Electronic Holography: The Newest", *International Symposium on 3-D Imaging and Holography,* Osaka, Japan, 1994.

9. Stephen Benton, "Elements of Holographic Video Imaging", *Proceedings of the 4th International Symposium on Display Holography*, 1991.

10. Stephen Benton, "Experiments in Holographic Video Imaging", *Proc. of the SPIE Inst. on Holography*, 1990.

11. Thomas A. Nwodoh, V. Michael Bove, Jr., John Watlington, and Stephen A. Benton, "A Processing System for Real-Time Holographic Video Computation", *Proc. of SPIE on Reconfigurable Technology*, vol. 3844, pp. 129-140, 1999.

12. V. Michael Bove, Jr., Mark Lee, Yuan-Min Liu, Christopher McEniry, Thomas Nwodoh, John Watlington, "Media Processing with Field-Programmable Gate Arrays on a Microprocessor's Local Bus", *Proc. of SPIE Media Processors*, vol. 3655, 1999.

13. J. Watlington, and V. Michael Bove, Jr., "Stream-Based Computing and Future Television", *SMPTE Journal.*

14. Mark Lucente, “Diffraction-Specific Fringe Computation for Electro-Holography”, Ph.D Thesis, Dept. of Electrical Engineering and Computer Science, MIT, 1994.

15. Mark Lucente, and Tinsley A. Galyean, “Rendering Interactive Holographic Images”, *Computer Graphics Proceedings*, pp. 387-394, 1995.