Development of 3-D Medical Image Visualization System

Thomas A. Nwodoh
(Email: nwodoh@digitechcorp.com)
Department of Electronic Engineering
University of Nigeria, Nsukka.

ABSTRACT

This paper reports the development of a holographic video (holovideo) rendering system that uses standard 2-D medical imaging inputs and generates medical images of human body parts as holographic video with three-dimensional (3-D) realism. The system generates 3-D medical images by transforming a numerical description of a scene (such as in data from URI, CAT, PET, and X-ray databases) into a holographic fringe pattern and then displays the images on the image volume of a holovideo display system. The system uses specialized digital signal processors to scale up the computation and rendering speed of the holovideo computing system beyond what exists today. Holograms developed under this research have horizontal (holographic) resolution high enough for smooth binocular parallax and a (video) resolution in the vertical direction comparable to NTSC television. Thus, the holovideo rendering and display system provides medical personnel with the information essential for viewing internal organs of humans with accurate 3-D realism. It is envisioned that the commercializable system that will ultimately be developed in the course of this research program will be compact enough to be used as desktop equipment in a medical imaging platform and economical enough to be made available in adequate numbers to hospitals.

KEYWORDS: Synthetic holography, 3-D Imaging, Holographic Video, Video Signal Processing, Stream Processing, FPGA.

1. INTRODUCTION

1.1 3-D Medical Imaging

The visualization of medical image models such as MRI, CAT, PET and X-ray images usually requires an expert to look at and interpret stacks of dimensional (2-D) images. Even with an expert, it is still difficult to accurately register and visualize the models they are in the actual human body part being examined. If the images could be displayed in three-dimension, visualization is much easier. A 3-D model can be rotated and "zoomed" to show areas of interest. A hologram is a diffractive optical element that produces images that give the sensation of 3-D realism to a viewer [5]. With medical images rendered as holographic video, medical personnel can easily use the monocular and binocular vision cues offered by holograms to visualize the images of human body parts with 3-D realism.

1.2 The Hologram Computation Problem

An optical hologram is a diffractive optical element formed by recording the interference pattern between pairs of light beams. Typically, when recording the hologram, a coherent reference beam (such as a laser beam) and the object beam which is locked in phase with the reference beam are both incident on the hologram plate. The two beams interfere and their -phase footprint is recorded on the hologram plate [2]. Every light wave from points on 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 plate. These fringe patterns look like an array of curves when the hologram plate is viewed on a microscope. When the recorded hologram is illuminated with an optical replication of the reference beam used during the recording phase, the fringe patterns diffract the illuminating light. The diffracted light produces the holographic image which reconstructs the optical wavefronts that fell on the plate from objects in the scene during the recording phase [13].

A Computer-Generated Hologram (CGH) frame has its holographic fringes represented as an array of grey scale data. These grey scale values are used to modulate an illuminating beam to produce an image on the display. The generation of the CGH frame starts with a 3-D numerical description of the 3-D scene such as obtained with: computer-aided design models of objects, stereogram's perspective views of a 3-D scene, or such medical imaging data as in MRI, CAT, and PET databases. The data are then encoded (computed) to generate the holographic fringes [4]. Hologram computations generally involve a series of computations on the data obtained from the scene descriptions.

The generation of computer generated holographic fringes for real-time holographic video display is very computation-intensive requiring the development of specialized systems to generate and display holovideo at video rates. It takes the order of ten tera-operations (computational) to fully compute the hologram of even small objects [2]. Available computational and modulation tools, and communication bandwidth cannot generate that magnitude of required data throughput in real-time - the main bottleneck being the computation hardware that is connected with content processing between the user interface and the fringe computation [6]. It will take today's even top-of-the-line processors of the order of several days to fully compute one hologram frame! However, scientists and engineers have adopted some techniques that make the generation of holograms possible within reasonable computation time limits. These techniques include: data reduction, exploiting parallelism, and by using specialized processors [5]. This research exploits these techniques to generate and display holographic videos of human body parts at video rate (up to thirty frames per second). While many commercial frame buffer cards exist in today's market, none can handle the magnitude of the bandwidth or processing power for holographic video.

2. RESEARCH DESIGN AND METHODS

The first step towards achieving holographic video computation in real time and at video rates using processors that are currently available is the reduction of the enormous amount of data required to compute the holographic fringes. Because of the limited acuity of the human visual system, the amount of data in a hologram can be reduced without adversely affecting the sensation experienced by a viewer looking at the resulting computer generated hologram [1,6,7]. By eliminating vertical parallax, reducing the vertical resolution and the horizontal field of view, and limiting the hologram size, the amount of data required for representation of a hologram is reduced by a factor of up to 50,000 [2,8,13].

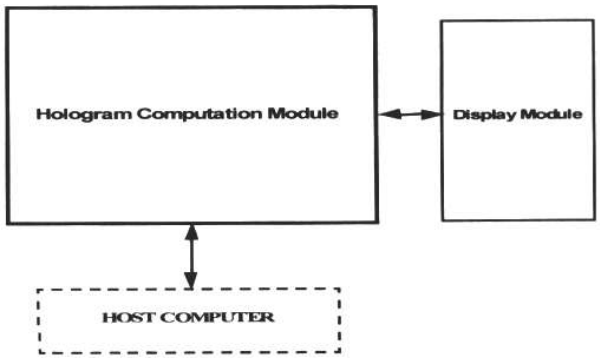

Figure 1: Layout of the Holographic Video Computation and Display System

The enormity of the computation involved in generating a holographic frame is so huge that it requires several specialized processors working in parallel to compute a hologram frame. With HPO (horizontal parallax only) holograms. each hololine (a horizontal slice of the hologram frame) has all the information needed to produce the corresponding part of the image from it for display. Hence each hololine can be rendered and displayed independently - thus making it possible to represent the holoframe as an array of independent hololines. The independence of the hololines facilitates processing parallelism where many processing units work in parallel with the overall computation task

shared between all the processors. Also specialized hardware is needed to perform the rapid computations required to render holograms in real-time at video rates.

The system being reported consists of a hologram computation module and a hologram display module. The computation module contains the electronics required for processing, storage, and formatting of the gray-scale values of the holographic fringes while the display module contains the optics required to display the holographic frame using the gray-scale values supplied by the computation module. This system: has a capacity to compute holograms of a moving scene; is equipped with data archival / retrieval from a host system capability; and is capable of displaying the hologram in the image volume of a holographic display system. Figure 1 shows the block diagram of the hardware of major components of the system.

At the display module, the computed hologram fringes from the computation module are first converted to analog form, then passed through a radio frequency signal processing circuit where they are modulated with a high frequency carrier, amplified, and then fed into a bank of AOMs (Acousto-Optic Modulator) crystals. The signals fed into the AOM are then used to 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. An AOM is a spatial light modulator that diffracts light from an illumination beam as a result of a change in the refractive index of the crystal caused by sound waves launched across the crystal. An AOM can process a 256KB data that form a hololine into a holographic image for a line on the image volume. 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. Figure 2 shows a block diagram of the display module for the hologram display system. See (3) for a detailed description of the optics of the holovideo display.

2.1 Holovideo Computation Algorithm

A holographic stereogram can be generated by using *precomputed fringe elements* (which is specific to a display geometry) and a set of rendered or camera-captured parallax views of a scene. The precomputed fringe elements are generated by computing *basis fringes*, encoding them, and generating rendering schemes to match the display-specific reconstruction geometry. The basis fringes which are independent of any image information, provide the diffractive property such that when convolved with data from the scene, the fringes redirect light into a specific span in a viewzone.

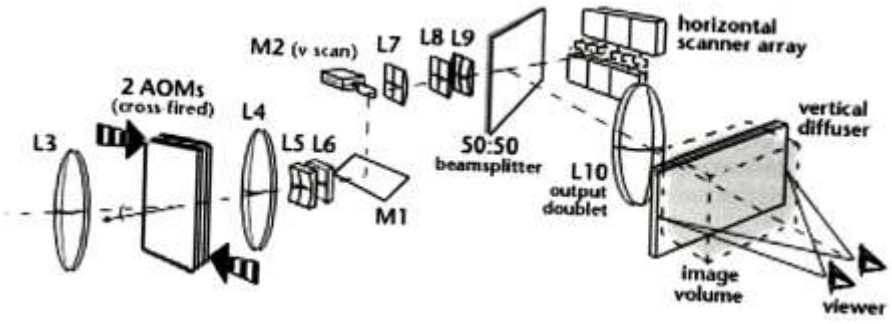

Figure 2: Holographic Display Module

A Horizontal Parallax Only (HPO) stereogram can be constructed to generate a continuous parallax in the horizontal direction by capturing the projection a given scene would have on the viewzone. Since it is impossible to encode infinite parallax, scene parallax is captured from a finite set of perspectives, and is then re-projected back in those same capture-directions. In order to prevent gaps between parallax views in the viewzone, each view is uniformly horizontally diffused over a small angular extent. See [4] for detailed description of stereogram image encoding.

To capture or render scene parallax information using cameras for example, the cameras are positioned along a linear track, with the view normal also normal to the capture plane. *N* views are generated from locations along the track that correspond with center output directions of the basis fringes. Once *N* parallax views have been captured, they are combined with the *N* pre-computed basis fringes to assemble a holographic stereogram. The same basis fringes can be used for any sweep of parallax views generated with the same capture geometry. Each "pixel" of the hologram (*holo-pixel*) is computed as a sum of basis fringes scaled by corresponding image pixel values. The resulting hologram frame, when properly illuminated, sends *N* different image pixels in *N* different directions at the display, resulting in a kind of light-field reconstruction of the captured scene. The pixels of each parallax view are projected in the same direction from which they were captured, and uniformly spread a little to prevent dead space in the viewzone. When a viewer observes the reconstructed image, sampled scene parallax information is relayed to each eye, and this stereo view changes appropriately and smoothly with side-to-side head movement.

To generate the holographic fringes, the 3-D scene is modelled 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 (such as the stereogram's perspective views) are *Hogel Vectors*, each of which describe the spectrum of the associated Hogel. The fringes are computed by having each Hogel Vector used as an array of coefficients of the associated Hogel. The Hogel Vector component (i.e. the perspective data) is multiplied with a component from a set of pre-computed *Basis Fringes* (i.e. the diffraction component). Each basis fringe is computed such that it contains energy in a particular region of the spectrum and control the directional behavior of diffracted light [3]. By multiplying each Basis Fringe with its corresponding Hogel Vector component and summing the results, the resulting fringes contain the spectral energy specified by the Hogel Vector. This generally involves performing a matrix inner product operation.

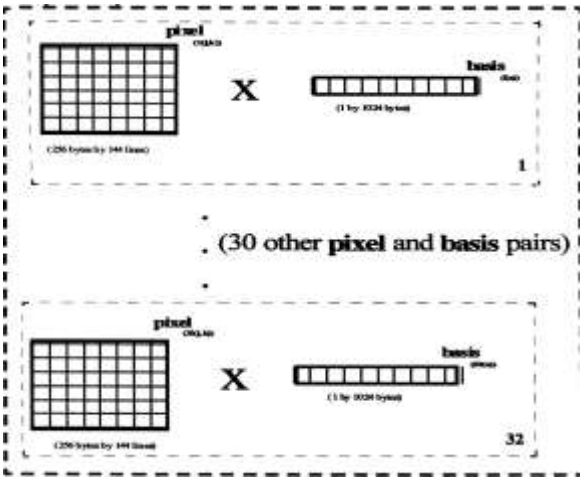

Figure 3: Data Input for Holovideo Computation

For a holographic stereogram generated using thirty two 256 by 144 byte perspective images (as in figure 3), the holovideo algorithm involves a series of: additions of the product of a byte from the 1 by 1024 basis function array and a byte from a 256 byte by 144 line perspective view of the scene whose hologram is being computed. In this particular implementation, there are thirty two image frame stereograms representing thirty two different 2-D perspective views of the scene and there are thirty two pre-computed basis

functions. However, the system being reported can support an arbitrarily large number of perspective view and basis function pairs. The multiplications are carried out between image frame (i.e. **pixel** in figure 3) and basis function (i.e. **basis** in figure 3) byte pairs as illustrated in figure 3. 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, as in:

$$\text{holo-value}_{(j\times n,k)} = \sum_{i=1}^{32} \text{pixel}_{i(j,k)} \times \text{basis}_{i(n)} \qquad \ldots(1)$$

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. In figure 3, there are thirty two (i.e. the i-values) **pixel** byte arrays of size 256 (i.e. the j-values) by 144 (i.e. the k-values) and there are thirty two (i.e. the i-values) **basis** byte arrays of size 1 by 1024 (i.e. the n-values). For any fixed *i* and *k*, 1 by 256K values are obtained by multiplying each 256 j-bytes in pixel with each 1024 n-bytes in basis. Then for all of 144 k-lines, 256K by 144 values are obtained. To generate a holo-value value, each of the corresponding 256K by 144 values are summed over all 32 *i*-perspective views. Hence, this is a superposition processing operation.

To obtain a byte of a fringe in the final holographic frame, each holo-value$_{(jHn,k)}$ is subsequently normalized to yield a byte of the final hologram frame. Normalization serves to fit the value of holo-value$_{(jHn,k)}$ to the display system used. The system being reported has an 8 bit dynamic range for fringe display. Normalization involves the computation of:

$$\text{holo-pixel}_{(j\times n,k)} = \left(\frac{\text{holo-value}_{(j\times n,k)} - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}} \right) \times 255 \qquad \ldots(2)$$

where α_{max} is the largest un-normalized value for the entire hologram frame (that is maximum value of holo-value$_{(jHn.k)}$ and α_{min} is the least un-normalized holo-value$_{(jHn,k)}$ (which is typically zero). With α_{min} = 0, equation 2 reduces to:

$$\text{holo-pixel}_{(j\times n,k)} = \left(\frac{\text{holo-value}_{(j\times n,k)}}{\alpha_{\max}/255} \right) \qquad \ldots(3)$$

2.3 Hardware Model of the Computation Module

The hardware module consists of nine standard PCI form-factor Processor Cards and three Video Concentrator Cards (VCCs). The processor cards are PCI cards since they are required to plug into a host's PCI slots. The Processor cards are the basic hologram computation units in this architecture while the VCCs control the display of a holoframe by scheduling the transmission to the display of the data for each hololine received from each of the different Processor Cards working in parallel. The computed hologram is subsequently transmitted to the VCC where the computed holographic data from multiple Processor cards are formatted into the format required to display the hololines. The VCC also converts the digital grey scales into the analog form used to drive the display.

2.3.1 The Processor Card

Each Processor card in the system consists ot a PCI bridge, SDRAM memory, Field Programmable Gate Arrays (FPGAs), and high speed I/O (HSIO), all resident on a standard PCI form-factor card. The FPGA on the Processor card is used for performing computations on streams of data to generate the holographic fringes and for the control of the transmission of the results through a HSIO interface port to the VCCs. Data archival/retrieval capability is provided through the PCI interface via which the system can download/upload data from/to a host system while real-time input data streams are received through frame grabbers sitting on the host's PCI bus and attached to cameras capturing a scene.

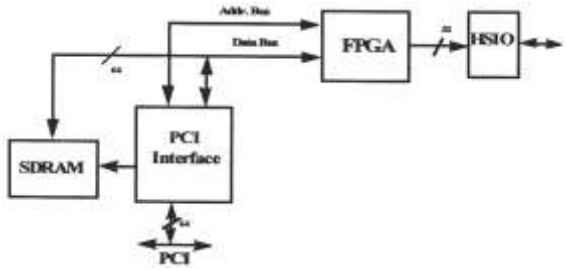

Figure 4: Block Diagram of the Processor Card

The PCI bridge interfaces the Processor card to a PCI highway. Through this interface, the Processor card can access data on a host machine and the host can access Processor card's memory. The PCI bus which is capable of a sustained bandwidth of 528MB/s is used because of its ubiquity and robust bus features. With the PCI interface, the system can be interfaced to different host platforms. The PCI bus used is compliant with PCI standard 2.1 with 64 bit data bus running at 66MHz.

Using data from a scene generated by cameras hooked to the host or data from a 3-D model of an object in the host's hard disk, the computation module computes the holographic fringes. The generated hologram fringes are subsequently formatted into the format required to display the hololines. Before being transmitted to the display the computed digital holographic fringes having digital grey scale values are converted into analog form by D-to-A converters and used to drive the display.

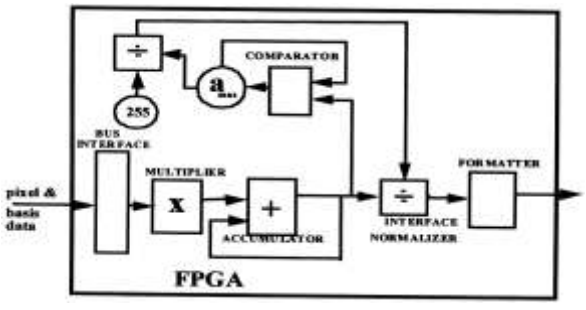

Figure 5: Functional Blocks for Holovideo Application

The FPGA is used as a specialized computing hardware in this architecture. The parallelism and the specialized processing required by the hologram computation algorithm are quite suitable for an FPGA used in generating the holographic fringes. The concurrent functionality of an FPGA can be efficiently used for parallel computation of the superposition operations required for holovideo. The FPGAs are large enough and employs several superposition processing pipelines all working in parallel to generate the fringes of the hologram frame. The FPGAs are dynamically reconfigurable (SRAM-based) 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. Figure 5 shows the functional blocks used by the FPGA in the implementation of the holovideo algorithm.

2.3.2. The Video Concentrator Card

The Video Concentrator Cards in the system interfaces the Processor cards to the display. The VCC also acts as a frame buffer for the system. The VCCs assemble and format data from multiple Processor cards for the display and is made of: a HSIO interface whence data is transferred from the hologram computing Processor cards, a set of FIFOs and video SDRAM used as buffer for data for the hololines being displayed, a microprocessor that handles communication with other VCCs, a co-processor which controls display data formatting, and D-to-A Converters that convert digital fringes to analog form. The co-processor is implemented with an FPGA and controls all the signals needed to format the data from the Multiple Processor Cards into the format required by the display. Figure 6 shows a block diagram of the VCC.

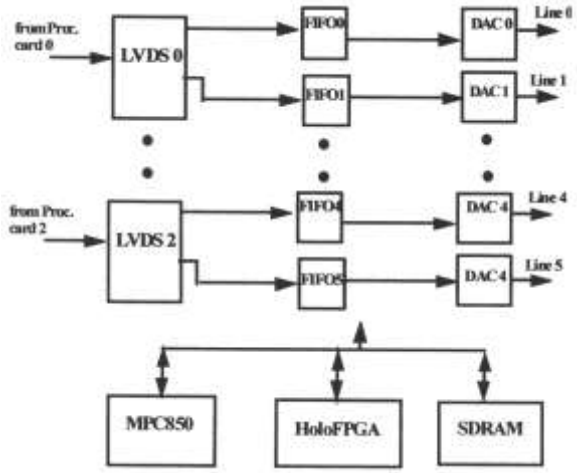

Figure 6: Block Diagram of the VCC

The VCC is designed to ensure that high bandwidths of data generated from multiple Processor Cards are transmitted to the VCC, formatted, buffered, and then passed properly to the display. There is a tight control of the reception of the data streams from the HSIO ports; it's formatting into different hololines, and its subsequent transmission to the display. All these steps are coordinated between the nine Processor cards and the three VCCs. The FPGA on the VCC controls the start/end of both loading and read-out of the FIFOs. While many commercial frame buffer cards exist in today's market, none can handle the magnitude of the bandwidth or processing power for holographic video.

3. MODEL TESTING METHODOLOGY

A design model of the medical image visualization system is generated and the holovideo algorithm running on the FPGAs on the Computation Module is written in VHDL, synthesized and the functionality simulated. VHDL instantiations of multipliers, accumulators, and dividers are used to process the data streams coming from either memory or the PCI bus. Several implementation modes are simulated. A multiply-accumulate module written in VHDL is used to generate the sum of the product of the image frame - basis function byte pan in equation 1 while a VHDL divider module is used for normalization as in equation 3. A comparator is used to generate the value of α_{max}. The multiply-accumulate-compare process uses several VHDL processing pipeline modules 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 dividers with 21 bit numerators and 13 bit denominators giving an 8 bit quotient representing the gray-scale byte of a hologram fringe. The numerator used for the division are the holo-value$_{(jHn,k)}$ values while the denominator used is $\alpha_{max} \div 255$.

The performance of the holovideo algorithm running on the FPGAs were tested for its real-time capability. Different algorithm implementation options were simulated and tested. For the computation of a 36MB hologram frame with arbitrary brightness, the following options were simulated:

- Option 1: where all computation operations are carried out in one single phase - the multiply, accumulate, and normalize operations are all executed in tandem without the storage of intermediate values. Here, there is no compare stage.
- Option 2: involves the transmission of pre-computed holograms from the FPGA to the HSIO ports. The FPGA just acts as a FIFO buffer for the data flowing from main memory to the HSIO ports.
- Option 3: assumes that the multiply-accumulate values have been obtained and needs to be normalized prior to transmission to the HSIO ports.
- Option 4: simulates normalization circuits using different denominator values. This is because the speed and complexity of the divider circuit is dependent on the number of bit ones in the representation d the denominator. For example, a divide by 7160 is more complex than a divide by 8192 and takes more logic elements to implement.

The FPGAs used on the Computation Module are large enough (up to 10 million logic gates) such that several processing pipelines can be computing fringes in parallel. Each FPGA has at least two hundred and fifty six processing pipelines all working in parallel and there are nine such FPGAs in the Computation Module. With the nine FPGA systems, this is equivalent to having at least 2,304 digital signal processors working in parallel to solve the holovideo fringe computation task.

4. RESEARCH RESULT

The end result of this research makes possible the computation in real-tine and display at video rates of 3-D medical images of body organs. The system tested is capable of computing and displaying 36MB 3-D medical images that are as large as about 150mm in width, 75mm in height, and 100mm depth at video rates. The system uses specialized processors (FPGAs) exploiting parallel processing and specialized processing algorithms to tackle this enormous computation task so as to be able to campute up to thirty holographic frames in one second (i.e. video rate). The architecture modelled herein is scalable allowing incremental addition of hardware as higher data throughput is needed, thus making possible: the computation of bigger sized 3-D medical images, the building of higher resolution into the system, and possibility of generating full parallax 3-D medical images. just by suitable addition of hardware modules.

References

1. Nwodoh, T. A., Bove, V. M., Watlington. J., and Benton, S. A., "A Processing System for Real-Time Holographic Video Computation," Proc. of SPIE on Reconfigurable Technology, vol. 3844, pp. 129-140, 1999.
2. Benton, S. A., "The Second Generation of the MIT Holographic Video System," Invited paper. First Intemational Symposium on Three Dimensional Image Communication Technologies, 1993.
3. Lucente. M.. "Diffraction-Specific Fringe Computation for Electro-Holography," Ph.D Thesis. Dept. of Electrical Engineering and Computer Science. MIT. 1994.
4. St.-Hilaire, P., "Scalable Optical Architectures for Electronic Holography," Ph.D Thesis, Program of Media Arts and Sciences, MIT. 1994.
5. Benton. S., "Experiment in Holographic Video Imaging," Proc. of the SPIE Inst on Holography, 1990.
6. Pappu, R., Sparrell, C., Underkofller, J., Kropp, A., Chen, B., and Plesniak, W., "A Generalized Pipeline for Preview and Rendering of Synthetic Holograms," SPIE Conference on Practical Holography XI, San Jose, 1997.
7. Underkofller, J., "Occlusion Processing and Smooth Surface Shading for Fully Computed Synthetic Holography," SPIE Proceedings. vol. 3011, pp. 19-30, 1997.
8. Lucente, M., "Interactive Computation of Holograms Using a Look-up Table," Journal of Electronic Imaging, vol 2(1), pp. 28-34.1983.
9. Bove, M., and Watlington, J.. "Cheops: A Reconfigurable Data-Flow System for Video Processing," IEEE Transaction on Circuits and Systems for Video Technology. Vol. 5. No.2. 1995.
10. Watlington, J., Lucente, M., Sparrel. C., and Bove, M., "A hardware Architecture for Rapid Generation of Electro-Holographic Fringe Patterns," 1995 IS&T/SPIE Symposium on Electronic Imaging: Science and Technology. Practical Holography, SPIE vol. 2406. paper #2408-23.
11. Watlington, J., and Bove, M.,"Stream-Based Computing and Future Television," SMPTE Journal.
12. Nwodoh, T. A., Prabhakar, A., and Benton, S., "HoIo-Chidi Video Concentrator Card," SPIE Proceeding on Media Processors. vol. 4674, 2002.
13. Lucente. M., and Galyean, T. A., "Rendering Interactive Holographic Images," Computer Graphics Proceedings, pp. 387-394, 1995.