Quickdraw3D – A front-end to a holographic three-dimensional display.

Andrew Lambert and Nick Grey.

ABSTRACT:

Three-dimensional display devices attempt to present the user with a realistic scene. The realism involves the use of depth cues, so as to convince the user of three-dimensionality. Such display devices can be classified into those that require external aids such as glasses, and those that are viewable unaided. The latter are less cumbersome, and allow many viewers of the display. Holography falls into this class, and is the only display medium that includes all the visual depth cues we expect of a scene.

Recently developed displays involve the computer rendering of three-dimensional models or scenes for use with optical hardware. A result of the computer generation is the display may be animated, either by rendering hologram frames on the fly, or by replaying precomputed holograms like a movie. Quickdraw3D is a useful technology for performing the former, and Quicktime is useful for the latter. Both these software technologies, combined with the emerging compute power of the PowerPC, make an obvious choice for integrating the holographic display with user interaction and modelling tools. We describe our work in this area in developing a three-dimensional display device, and our attempts to harness the Quickdraw3D framework.

INTRODUCTION

Three-dimensional display is a necessary and valuable tool in many disciplines. Devices which provide this functionality offer capabilities for more extensive data visualisation and user-computer interaction. For example, consider a biochemist manipulating a model of a DNA strand, or an architect taking a virtual walk through a house plan. Already consumer electronics devices attempt to improve the user interaction with a game or computer operating system, by using three dimensional graphics rendered to a two dimensional display device such as a television or computer monitor. The success of this interaction is based on making the environment with which the user interacts, realistic, or true to that with which we interact daily.

Various display devices have been designed that attempt to provide the expected visual depth cues we have evolved to expect. These displays fall into two categories; those viewable unaided – *autostereoscopic*, and those requiring either passive sensors to distinguish information destined for each eye, e.g. coloured, or orthogonally polarised glasses, or some active display to present different information to each eye – *stereoscopic*. The former are to be preferred in most situations, especially if there are simultaneous users of the display. Various methods have been exploited to confuse the eye into false perception of depth. The most common of these are the “magic eye” or autostereograms, that exploit subtle changes in repeated patterns or colours. In fact we as users use many depth cues when interpreting a three-dimensional scene[1];

- binocular disparity or stereopsis – slightly different images seen by each eye,
- convergence – rotation of the eyes to match images seen,
- occlusion – closer parts tend to obscure the parts behind,
- occular accommodation – focusing to different depths,
- parallax – scene change with motion of viewer or display,
- monocular cues, e.g. perspective, texture gradient, shading, and relative sizes.

Holography is a three-dimensional display technology that presents the viewer with a "life-like" three-dimensional image with all the depth cues, and most importantly parallax. However, hologram generation has traditionally been restricted to photographic processes, simply because of the massive amount of information necessary to record or calculate the image and depth information about the object of interest. Initial attempts at computer generation of holograms used photo-reduction processes to create the hologram on photographic film so it could be replayed for viewing [2].

In 1989, researchers at the Massachusetts Institute of Technology (MIT, USA) Media Laboratory Spatial Imaging Group announced an initial holographic video – *holo-video* –display that was capable of reconstructing a small computer generated hologram. Because it was computer generated, and because the display used electro-optic modulator devices that could be dynamically refreshed, the display could be changed or reproduced from a different viewpoint [3]. The reconstructed hologram exhibited all the visual depth cues that are expected from photographically produced holograms including horizontal parallax, but excluding vertical parallax. This exclusion was the first step in reducing the computational complexity of the display process, which at the time required a massively parallel computer to calculate the holographic fringes. Since this initial prototype, significant improvements have been made to both the optical reproduction process and the fringe calculation methods. Recent systems are capable of full colour display of projected or virtual images within a 150 x 75 x 160 mm space [4].

The uniqueness of this process when compared to traditional computer generated holographic attempts is the real time interaction by the user with the display. A host computer constructs the three-dimensional model of the scene to be displayed as would normally occur on a graphics workstation, or more recently home computer or video game systems. However, instead of one view of the scene being calculated and drawn to a two-dimensional display, the information is used to construct the fringes of the hologram. This computation process is lengthy and intense because it must simulate the diffractive processes that would occur naturally in the optical production of a holographic plate. Provided this rendering calculation can be achieved in sufficient time, the user can interact with the computer software causing a change of the three-dimensional model, with a new hologram being rendered.

In this work, we describe our contributions to such a holo-video display. This involves two distinct contributions. The first of these is the development of the optical system which will display the hologram, while the second involves the integration of this true three-dimensional display with existing computer graphics software. Rather than re-invent the wheel in terms of the user-computer interaction we have decided to utilise the Quickdraw3D framework [5], which provides all the modelling, storage, and user-interaction at the system level to applications on many platforms. Thus we are concentrating on producing a "renderer" plug-in for Quickdraw3D, which will integrate this display with the application environment. In this way, existing application software that uses Quickdraw3D for visualisation and interaction with three-dimensional datasets, can automatically make use of our display. An impression of the display, is shown in Figure 1., with the user interacting with the computer system and Quickdraw3D, while viewing a "live" holographic image.

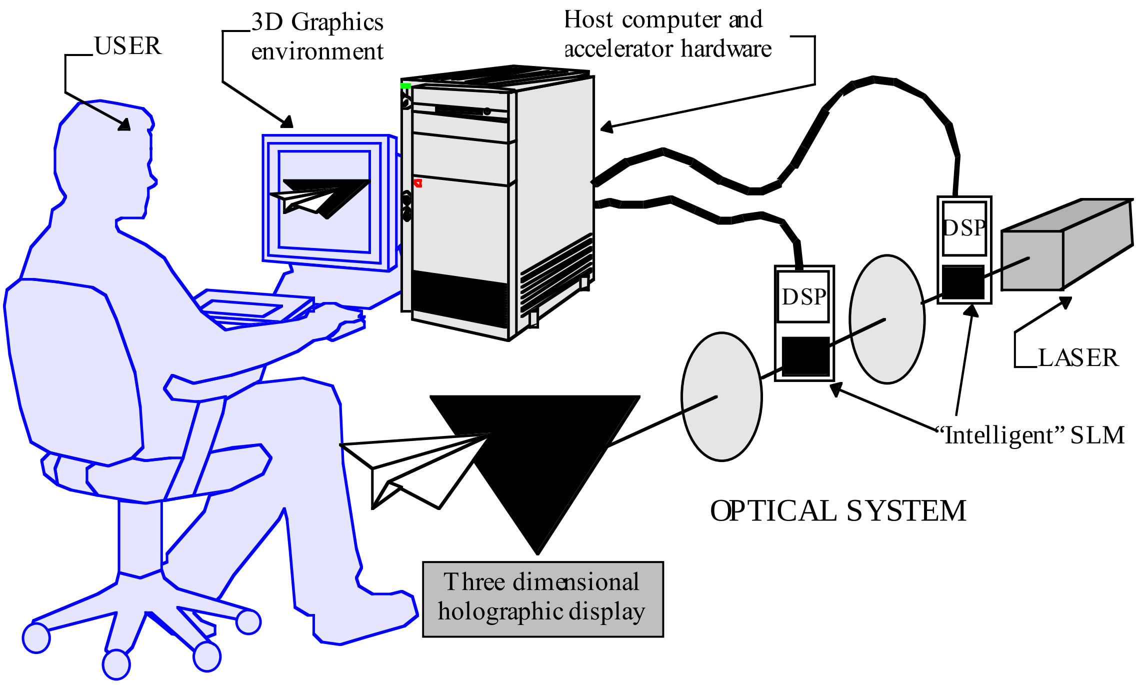
USER
3D Graphics environment
Host computer and accelerator hardware
DSP
DSP
LASER
"Intelligent" SLM
OPTICAL SYSTEM
Three dimensional holographic display

Figure 1: The proposed display system for this project, illustrated here, involves research in two main areas; the rendering software (controlled by the host computer), and the optical system including the intelligent spatial light modulators (utilising a digital signal processor and electro-optic modulator). The optical system is altered under algorithmic control by the programmable diffractive elements written on the spatial light modulators.

THE HOLO-VIDEO DISPLAY

The research into holographic video displays has largely been spearheaded by the Spatial Imaging Group at the MIT Media lab, who in 1990 introduced a "holo-video" display[3,6,7] capable of full colour, real–time display of a horizontal parallax hologram. This has subsequently been improved to produce three dimensional imagery occupying a litre volume[4]. The system they used incorporates an acousto-optic modulator[1,7] to display the lines of holographic fringes calculated by computer, which are deflected vertically to construct a two-dimensional hologram frame when integrated over the vertical refresh time.

Two advances in technology have been responsible for the introduction of computer generated holographic video; the first is the increased computational capability of current computers, and the second is the development of spatial light modulators (SLM), such as the acousto-optic modulator used at MIT. A SLM is a device which provides spatial control of a light beam, either electronically or by another light beam. Such devices shape a light wavefront in such a way as to place information across its extent. While SLM have been available for some time, only recently have low cost, readily available devices with the desired modulating properties been manufactured. Useful SLM include the liquid crystal display (LCD) [8, 9], and the digital micro-mirror device (DMD) which employs small 10 μm square tiltable mirrors on top of a memory cell [2], both of which are illustrated in Figure 2. While other SLM are available they tend not to have alternative uses in the consumer electronics market, e.g., commercial video or data projectors, which is primarily responsible for the availability and reduced cost.

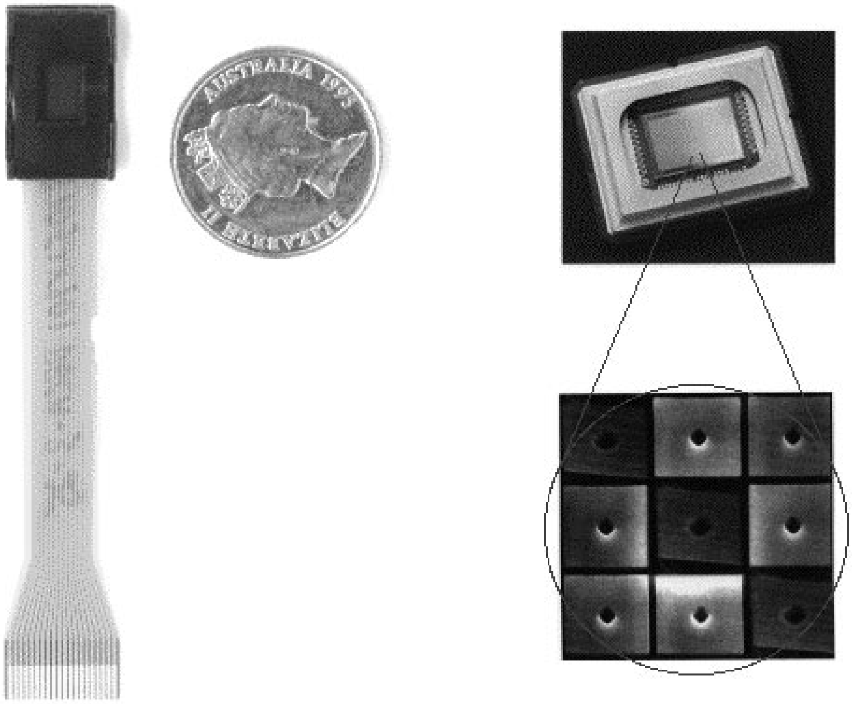

Figure 2: Spatial Light Modulation can be achieved using a Liquid Crystal Display (left), or by a Digital Micro-Mirror Device (right). The smaller the pixel size of these SLM, the better, as is evident in the comparison of the 320 by 240 pixel LCD (from Kopin Corporation), to a five cent piece, and the magnified image of the tiltable mirrors of the DMD (from Texas Instruments).

Holo-video displays function by displaying information on the SLM to either change the function of the optical system, or take the place of the photographic medium used in still holography [10]. For example by displaying the image shown in Figure 3 on an SLM, the light is altered in the same way as if a lens were used instead of the SLM. However, by displaying a slightly different “lens” on the SLM, we obtain the advantage of an adjustable imaging system. Such images are known as “programmable diffractive elements” [8,11,12] and like have been used to advantage in other types of non-holographic three-dimensional displays such as those employing the “varifocal” technique [13].

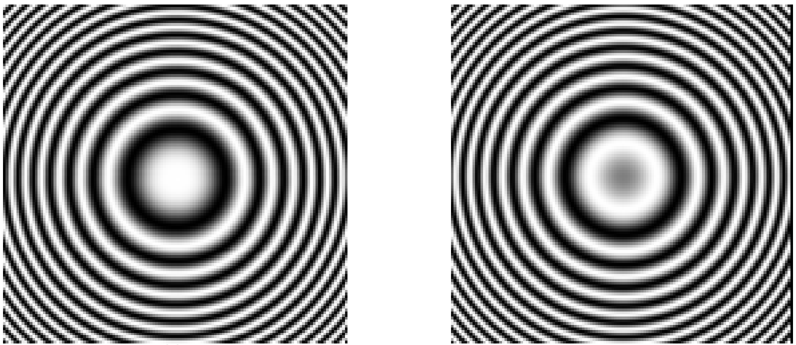

Figure 3: Programmable diffractive elements may be displayed on the SLM to change the function of the optical system. Alternatively, the holographic fringes themselves may be displayed on the SLM.

RENDERING WITH QUICKDRAW3D

The shear amount of data in a hologram is the biggest hurdle to overcome in this display. Ref. [3] suggests that a hologram of dimension 100 mm by 100 mm, giving a viewing angle of 30°, contains 25 GB for a single frame. Rendered at 8 bit resolution for 60 frames/sec, we need 12 TB/s of data to be displayed. Clearly, this is impossible with today's technology, even with such vector processing hardware as is found in consumer games machines, G3 Powerbooks, and "Altivec" style PowerPCs. However, we can adopt the practices in the three-dimensional graphics community, and simplify the model, and consequently the hologram, or use "tricks" to reduce the requirements on the SLM, and the computation required of the host computer.

One standard practice is to remove parallax in the vertical direction, providing the sense of depth by this method on side-to-side motion only. Similarly, we can use a stereogram approach, rendering different views to be shown in different directions, at least for low-fidelity occasions when the user is changing the model frequently. Other practises have also been proposed to minimise the computation time [14,15,16]. Holographic fringes may be generated for high-fidelity viewing over a longer time period. A similar, scalable, approach to three dimensional display is proposed for the ICVision[17] system, where adjacent pixels are directed to alternate viewing areas using micro diffraction patterns on each pixel site – however, this does not provide for full holographic display.

The rendering process for a three-dimensional holo-video display is therefore, complicated. While the model or scene still originates from three-dimensional graphics software running on the computer, the rendering process encompasses software algorithms, fast computational hardware (possibly external to the host computer), and the use of spatial light modulators to control a custom optical system. However, these problems must all be hidden from the application software and the user, if this type of display is to be easily integrated into an existing graphics visualisation environment. It is for this reason we have adopted Quickdraw3D. We are able to write a software renderer for use with Quickdraw3D that encapsulates all these concerns and requirements. In addition we are able to abstract the optical system, its SLM, and the external compute hardware in the RAVE architecture, should we so desire. (Such a move would allow Quicktime movies containing pre-computed holograms to display as well as those requiring on-the-fly rendering.) Using this renderer we (and other applications) can access the user interaction, storage, and modelling features of Quickdraw3D, without extra programming. Just think, we could all be playing with "Gerbils" in true 3D!

THE RENDERER

Traditional renderers are responsible for taking a set of drawing commands, primitives, or objects, and changing them using such entities as coordinate transformation matrices, shaders, lights, transparency attributes, etc, for drawing into a two dimensional window defined by its physical location on a screen, and by the camera or viewing direction and aspect. This process involves much number-crunching, which is easily performed in vector processing hardware. The renderer is therefore responsible for clipping the display to that which would be reasonably seen – a *frustum*, and for removing those parts of the model that are occluded by others. This is illustrated in Figure 4. The rendering of a stereogram is a similar approach, where many passes are made by the renderer, each from a different viewing direction, so as to form the images that will be directed to that viewing position by the stereogram display. (The "stereo glasses" feature of "Gerbils" is such an example, where blue and red encoded disparate images are superimposed to give the wearer of suitable glasses the perception of depth.)

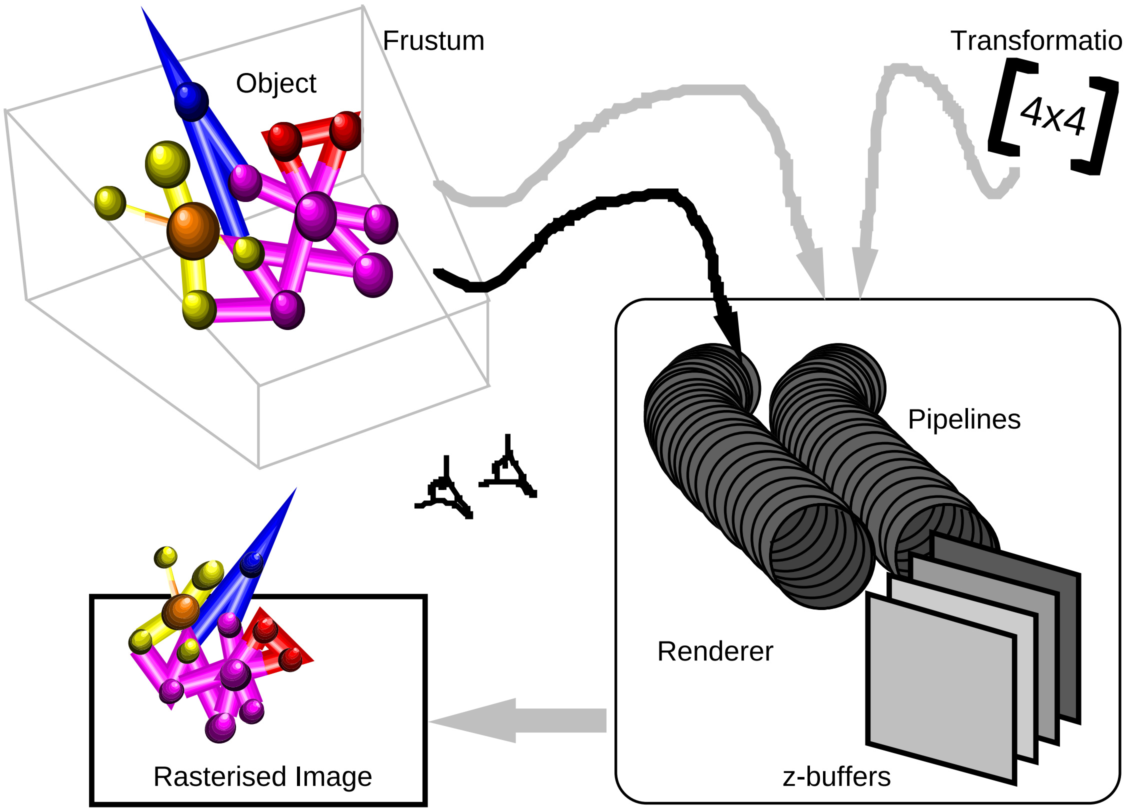

Figure 4: The renderer is traditionally responsible for manipulating a three-dimensional model or scene for display in a two dimensional port.

A renderer suited to holographic calculations is slightly more complicated, depending on the algorithm adopted. It may for example, accept all objects to be rendered, and process them into planes for subsequent processing into a hologram. This process is known as "z-buffering", and is useful for such displays as the varifocal display, or for the occlusion processing required in standard graphics. However, a second pipeline stage is necessary for the hologram generation algorithm we have developed. We progressively take the back frame, and calculate the diffraction pattern it would produce at the next plane. This then gets modified by the next frame, and the process is repeated forwards through the set of z-buffers. This then results in a two dimensional representation of the model, which is reasonably simple to manipulate into a hologram for display on the SLM. One could of course imagine cascading these processes across DSP or similar compute engines, rather than attempting to process the entire work on the host computer.

PROGRAMMING

Finally we give a brief overview of how to actually program a renderer for use with Quickdraw3D, based on our experience with the Quickdraw3D sample code, "Simple Renderer" and "Sample Renderer". These produce, in our case, a shared library for use with Quickdraw3D on MacOS, or in the case of the Windows version of Quickdraw3D, a dynamically linked library.

The renderer is dynamically linked to Quickdraw3D through a set of *metahandlers*, which in response to a query from Quickdraw3D, return a set of functions to which rendering information and objects to be rendered are to be passed. For example, the main metahandler will handle the “Start Frame”, “End Frame”, “Start Pass”, “End Pass”, and “Cancel” calls to control the status of the renderer operation. A typical renderer might use these functions to prepare or finish of the rendering process. The user may also configure a particular renderer if the “Modal Configure”, and “Get/Set Configuration Data” requests are handled. But, perhaps the most important metahandlers, are the Matrix, Style, Attribute, and Geometry metahandlers. The latter in turn is queried for a routine to process each of the geometry primitives contained in Quickdraw3D, while the others provide a way for Quickdraw3D to change the rendering parameters such as coordinate transformations, etc.

The sample code provided by Apple employs a standard methodology for the geometry rendering. The rendering parameters are used to configure graphics *pipelines*, one for each type of geometry that is supported. (Quickdraw3D will break more complex geometries down to those supported by your renderer.) These pipelines then process the geometries to a form suitable for *rasterising*. The rasterising process is dependant on the type and depth of the screen to which we are rendering, but now turns the transformed, processed, and clipped primitives emerging from the pipelines into pixels in the destination.

The process in the case of the holographic display is easily illustrated. The pipelines in our case transform the primitive geometries by the provided matrices, but do not necessarily clip to the standard frustum, as we expect to view the hologram from a wide set of angles. The geometries are then mapped into our renderer’s z-buffer planes. The rasterising process in our case, then proceeds with the algorithm described above on the stored z-plane contents, producing the two dimensional diffraction calculated image, which finally is transformed into a set of holographic fringes or similar for display on the SLM of our holo-video unit. The reason for placing the z-buffering in the rasterisation process is simply to exploit parallel processing hardware that may be external to the host-computer. The pipelines themselves may exploit RAVE and the newer PowerPC additions such as Altivec in the future.

CONCLUSION

We have described our work in brief on the development of a true three-dimensional display device. This follows work by others in the area of holographic video display, and offers all the depth cues expected of a realistic three-dimensional scene. We have shown how we can transparently integrate this display into the Quickdraw3D graphics architecture in the form of a plug-in renderer, which hides the complexity of the device and algorithms from the user. Quickdraw3D was chosen because of its size, expansibility, cross-platform portability, system level integration such that all applications can use it, and because of its ability to handle the storage and manipulation of three-dimensional models and scenes.

The internal working of our display under development has been briefly described, including our preliminary algorithms, and the specialist SLM used in the optical system. While the details of the system and the actual algorithm are not clearly defined at present, we have illustrated how this development can be integrated with Quickdraw3D, and have given an overview of how others may develop renderers for use with Quickdraw3D.

REFERENCES

[1] M. Lucente, T.A.Galyean, “Rendering Interactive Holographic Images”, ”, in *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, New York, 1995, pp. 387-394.

[2] P. Hariharan, Optical Holography: principles, techniques, and applications, Cambridge University Press, 1984.

[3] P. St Hiliare, S.A.Benton etal, “Electronic Display System for Computational Holography” *SPIE Proceedings*, vol. 1212, *Practical Holography IV,* 1990, S.A. Benton editor.

[4] P. St Hiliare, S. A. Benton, etal, “Color images with the MIT holographic video display”, in *Practical Holography VI*, Proc. SPIE, vol. 1667, 1992, S.A. Benton editor, pp. 73-84.

[5] 3D Graphics Programming with Quickdraw3D, Apple Computer, 1997.

[6] S. A. Benton, “Experiments in holographic video”, in *Practical Holography IV*, Proc. SPIE, no.1667, 1992.

[7] P. St Hiliare, S. A. Benton, etal., “Real-time holographic display: Improvements using a multichannel acousto-optic modulator and holographic optical elements:, in *Practical Holography V*, Proc. SPIE, v. 1461, 1991, S.A. Benton editor.

[8] Lambert, A. J. “Optical Image Processing and its Application”, Ph.D. thesis in Electrical Engineering, U.N.S.W., 1996.

[9] Lambert, A. and Cochrane, G., “A study of liquid crystal devices for spatial light modulation in optical computing,” in *IVCNZ'95 Image and Vision Computing New Zealand*, Lincoln, Canterbury, New Zealand, August 1995, pp. 149-154.

[10] Hashimoto, N. and Morokawa, S., “Real-time electroholographic system using liquid crystal television spatial light modulators”, *J. Electronic Imaging*, v. 2(2), April 1993, pp. 93-99.

[11] Lambert, A., Cochrane, G., and Fraser, D., “Optical image processing using Liquid Crystal Displays,” in *DICTA'95 - Digital Image Computing: Techniques and Applications*, Australian Pattern Recognition Society, Brisbane, Australia, December 1995, pp. 354-359.

[12] Davis, J.A., “Programmable diffractive optical elements using spatial light modulators” *in Liquid Crystal Materials, Devices, and Applications II*, vol. 1911, Efron,U. and Wand, M.D., SPIE, pp.214-255, February 1993.

[13] M. A. A. Neil, etal., “Spatial light modulator based three-dimensional multiplanar display”, in Proc. of SPIE, v. 3012, pp. 337-341, 1997.

[14] M. C. Lucente, “Interactive computation of holograms using a lookup table”, J. Electronic Imaging, v. 2, no. 1, pp. 28-34, Jan 1993.

[15] M. C. Lucente, “Computational holographic bandwidth compression”, IBM Systems Journal , v. 35, nos. 3&4, pp. 349-65, October 1997.

[16] M. C. Lucente, “Holographic bandwidth compression using spatial subsampling”, Optical Engineering, v. 35, no. 6, pp. 1529-37, June 1996.

[17] Kulick, Jeffrey, H. etal., “ICVision - a VLSI-based diffractive display for real-time display of holographic stereograms,” in *Practical Holography VIII*, Proc. of SPIE, v. 2176, 1994.

ABOUT THE AUTHOR Dr. Andrew Lambert

Dr. Lambert has been programming Macintosh applications since 1988, principally in the areas of image and signal processing, and has developed many in-house applications based on Fourier signal processing. He obtained his B.Sc.(Hons) degree from the University of Otago, New Zealand in 1987, and his Ph.D. in Electrical Engineering from The University of New South Wales in 1997. His research areas include digital signal processing, optical image processing, atmospheric imaging, and three-dimensional holographic display. Dr. Lambert is a Lecturer in the School of Electrical Engineering at University College, Canberra. The author has received three AUDF grants, one of which, “Three dimensional, interactive holographic video display”, is the project from where this paper originates.

TARGET AUDIENCE Graphics (Quickdraw3D and Quicktime), Visualisation, and Display Technology.

CONTACT DETAILS

School of Electrical Engineering,
University College,
The University of New South Wales,
Australian Defence Force Academy,
Northcott Drive,
Canberra, ACT 2600.

Ph:	(02) 62688351
Fax:	(02) 62688443
e-mail:	a-lambert@adfa.oz.au
web:	http://www.ee.adfa.oz.au