An Autostereoscopic Display

Ken Perlin, Salvatore Paxia, Joel S. Kollin

Media Research Laboratory*, Dept. of Computer Science, New York University

ABSTRACT

We present a display device which solves a long-standing problem: to give a true stereoscopic view of simulated objects, without artifacts, to a single unencumbered observer, while allowing the observer to freely change position and head rotation.

Based on a novel combination of temporal and spatial multiplexing, this technique will enable artifact-free stereo to become a standard feature of display screens, without requiring the use of special eyewear. The availability of this technology may significantly impact CAD and CHI applications, as well as entertainment graphics. The underlying algorithms and system architecture are described, as well as hardware and software aspects of the implementation.

Keywords

graphics hardware, hardware systems, object tracking, optics, user interface hardware, virtual reality.

1 INTRODUCTION

1.1 Prior and Related Work

Computer graphics, even when rendered in high quality, still appears flat when displayed on a flat monitor. Various approaches toward creating true stereoscopy have been proposed so that the objects we simulate will look as though they are really in front of us [Okoshi, Lipton]. These fall into various categories.

The most common form of stereo display uses shuttered or passively polarized eyewear, in which the observer wears eyewear that blocks one of two displayed images from each eye. Examples include passively polarized glasses, and rapidly alternating shuttered glasses [Lipton85]. These techniques have become workhorses for professional uses, such as molecular modeling and some subfields of CAD. But they have not found wide acceptance for three dimensional viewing among most students, educators, graphic designers, CAD users (such as engineers and architects), or consumers (such as computer games players). Studies have shown that observers tend to dislike wearing any invasive equipment over their eyes, or wearing anything that impairs their general ambient visual acuity [Drascic]. This consideration has motivated a number of *non-invasive* approaches to stereoscopic display that do not require the observer to don special eyewear.

A graphical display is termed *autostereoscopic* when all of the work of stereo separation is done by the display [Eichenlaub98], so that the observer need not wear special eyewear. A number of researchers have developed displays which present a different image to each eye, so long as the observer remains fixed at a particular location in space. Most of these are variations on the *parallax barrier* method, in which a fine vertical grating or lenticular lens array is placed in front of a display screen. If the observer's eyes remain fixed at a particular location in space, then one eye can see only the even display pixels through the grating or lens array, and the other eye can see only the odd display pixels. This set of techniques has two notable drawbacks: *(i)* the observer must remain in a fixed position, and *(ii)* each eye sees only half the horizontal screen resolution.

Holographic and pseudo-holographic displays output a partial light-field, computing many different views simultaneously. This has the potential to allow many observers to see the same object simultaneously, but of course it requires far greater computation than is required by two-view stereo for a single observer. Generally only a 3D lightfield is generated, reproducing only horizontal, not vertical parallax.

A display which creates a light field by holographic light-wave interference was constructed at MIT by [Benton]. The result was of very low resolution, but it showed the eventual feasibility of such an approach. Discrete light-field displays created by [Moore], and the recent work by Eichenlaub [Eichenlaub99], produce up to 24 discrete viewing zones, each with a different computed or pre-stored image. As each of the observer's eyes transitions from zone to zone, the image appears to jump to the next zone. A sense of depth due to stereo disparity is perceived by any observer whose two eyes are in two different zones.

Direct volumetric displays have been created by a number of researchers, such as [Downing], [Williams] and [Woodgate]. One commercial example of such a display is [Actuality]. A volumetric display does not create a true lightfield, since volume elements do not block each other. The effect is of a volumetric collection of glowing points of light, visible from any point of view as a glowing ghostlike image.

Autostereoscopic displays that adjust in a coarse way as the observer moves have been demonstrated by [Woodgate]. The Dresden display [Schwerdtner] mechanically moves a parallax barrier side-to-side and slightly forward/back, in response to the observer's position. Because of the mechanical nature of this adjustment, there is significant "settling time" (and therefore latency) between the time the observer moves and the time the screen has adjusted to follow. In both of these displays, accuracy is limited by the need to adjust some component at sub-pixel sizes.

*Mailing Address: 719 Broadway, 12th Floor, New York, NY 10003

URL: http://www.mrl.nyu.edu/perlin/

1.2 Goals

The goals of our research have been to present a single observer with an artifact-free autostereoscopic view of simulated or remotely transmitted three dimensional scenes. The observer should be able to move or rotate their head freely in three dimensions, while always perceiving proper stereo separation. The subjective experience should simply be that the monitor is displaying a three dimensional object. In order to be of practical benefit, we sought a solution that could be widely adopted without great expense. We also wanted a solution that would not suffer from the factor-of-two loss of horizontal resolution which is endemic to parallax barrier systems.

These goals imposed certain design constraints. The user responsive adjustment could not contain mechanically moving parts, since that would introduce unacceptable latency. The mechanism could not rely on very high cost components. We also wanted the device to be able to migrate to a flat screen technology. Because we made certain simplifying design decisions for our first prototype, our initial test system displays only monochromatic images. However, this is not an inherent limitation of the technique.

1.3 Significance

The significance of this work is in that it enables a graphic display to assume many of the properties of a true three dimensional object. An unencumbered observer can walk up to an object and look at it from an arbitrary distance and angle, and the object will remain in a consistent spatial position. For many practical purposes, the graphic display subjectively *becomes* a three dimensional object. When combined with haptic response, this object could be manipulated in many of the ways that a real object can. Ubiquitous non-invasive stereo displays hold the promise of fundamentally changing the graphical user interface, allowing CAD program designers, creators of educational materials, and authors of Web interfaces (to cite only some application domains) to create interfaces which allow users to interact within a true three dimensional space.

2 PRINCIPLE

COMBINE SPATIAL MULTIPLEXING WITH TEMPORAL MULTIPLEXING

2.1 High level approach

We addressed our design goals by creating a modified parallax barrier that combines spatial multiplexing and temporal multiplexing. Since no fixed parallax barrier geometry could accommodate arbitrary observer position and orientation, we create a dynamically varying parallax barrier, one that continually changes the width and positions of its stripes as the observer moves. The use of a virtual dynamic parallax barrier is reminiscent of work by [Moore] and [Eichenlaub99], but to very different ends - instead of using a fixed dynamic pattern to create a fixed set of viewpoints, our goal is to create a result which is continually exact for one moving user.

As we shall show, each dynamic stripe needs to be highly variable in its width, in order to accommodate many different positions and orientations of the observer. For this reason, we make the dynamic stripes rather large, and use a correspondingly large gap between the display screen and the light-blocking parallax barrier. Because the stripes are large enough to be easily visible, we need to make them somehow unnoticeable. To do this, we rapidly animated them in a lateral direction. The observer then cannot perceive the individual stripes, just as a passenger in a car speeding alongside a picket fence cannot see the individual fence posts.

This large-stripe approach requires each stripe to be composed from some number of very slender *microstripes*, each of which is an individually switchable liquid crystal display element. To sum up: we use a dynamic parallax barrier consisting of very large stripes, which are made out of many slender ones, and we move these large stripes so rapidly across the image that the observer cannot perceive them.

2.2 Three phases

In a perfect world, a temporally multiplexed system could be made from just two alternating phases. Parallax barrier systems depend on the distance E between an observer’s two eyes (generally about 2.5 inches). Suppose that a display screen D inches away from the observer showed alternating stripes of a left and a right image. Suppose also that a light-blocking shutter were placed G inches in front of this display screen in a "picket fence" stripe pattern. If the width of each shutter stripe were chosen as E*G/D, and the width of each image stripe as E*G/(D-G), then during phase 1 the observer’s left eye would be able to see half of one image through the clear stripes, and the observer’s right eye would be able to see half of the other image through the clear stripes [Figure 1a]. If the light-blocking shutter were then flipped, and the display screen pattern simultaneously changed, then the observer would see the remainder of each respective image [Figure 1b]. If this flipping were done fast enough, then the observer would perceive two complete independent images, each visible only to one eye. The problem with this scenario is that the observer would need to be in precisely the correct position; the slightest deviation to the left or right would result in the wrong eye seeing a sliver of the wrong image.

For this reason, we animate the stripes in three phases. During each phase, the light-blocking shutter lets through only one third of each stripe. After each phase the stripe pattern is shifted laterally. Over the course of three phases, the observer’s left eye sees one entire image, and the observer’s eye sees a different entire image. The use of three phases guarantees that there is room for error in the observer’s lateral position [Figures 2a,2b,2c].

2.3 Varying distance

The observer can be at a wide range of distances, since we can always vary the stripe width so as to equal E*G/D, as described above. [Figure 3a] shows the observer relatively far; [Figure 3b] shows the observer much closer. Microstripe resolution puts a practical upper limit on the observer distance, since the stripes become narrower as the observer’s distance to the screen increases.

This upper limit increases linearly both with the gap between the display and shutter, and with the shutter resolution. In practice, we have set these so as to be able to handle an observer up to about five feet away.

2.4 Head rotation

In previous autostereoscopic techniques based on parallax barriers, all stripes were required to be of equal width. This presents a problem if the observer’s head is facing off to the side. This will often be true when the observer has other displays or paperwork in his field of view, or is engaged in conversation with a colleague. In this case, one of the observer's eyes will be perhaps an inch or so closer to the screen than the other. When this happens, it no longer suffices for the barrier stripes to be all of equal width. Rather, in this case the stripes should vary in width in a perspective-linear pattern [Figure 4].

Our dynamically varying stripe generation handles this case accurately. Given any two eye positions, we compute and display the proper perspective linear stripe pattern. The mathematics to support this are developed in the next section.

2.5 Positioning the stripes

In this section we develop the mathematics needed to properly place the stripes. To make the light blocking work properly, we need to interleave the left and right images on the display and also to create a corresponding set of opaque/clear stripes on the optical shutter. To compute where the stripes should go, we use a system of crossed lines:

Starting from the *right* eye and the left-most point on the display, draw a straight line, and see where it crosses the shutter. Then draw a line from the *left* eye through this point on the shutter, and see where this new line hits the display. This process is continued, always starting with this next point over on the display, to produce an effective pattern of left/right image display stripes and light-blocking shutter stripes for that pair of eye positions.

Starting at one side of the display, we cross the lines on the shutter as follows:

1. Draw a line from $\mathbf{x_n}$ on the display, through the shutter, to the right eye;
2. Draw a line from the left eye, through the shutter, to $\mathbf{x_{n+1}}$ on the display;
3. Iterate

[Figures 5a, 5b] show how we construct a sequence of stripe positions from two eye positions (shown as a green and red dot, respectively), a display surface (shown as the bottom of the two horizontal lines) and a shutter surface (shown as the top of the two horizontal lines). Starting from the left side of the display screen, we calculate the line of sight through the shutter to the right eye. Then we compute the line of sight from the left eye, through this point, down onto the display screen. [Figure 5a] shows this process after one iteration; [Figure 5b] shows the same process after three iterations. In these figures, the positions at which the shutter needs to be transparent are circled in gray.

We now describe the mathematical details for this process. To place the stripes properly on the display screen, assume the two eye positions are: $\mathbf{p}=(p_x,p_y)$ and $\mathbf{q}=(q_x,q_y)$, that the display screen is on the line y=0, and that the shutter is on the line y=1. Given a location (x,0) on the display screen, we find the line-of-sight location $f_p(x)$ on the shutter that lies between display screen location (x,0) and eye position $\mathbf{p}$ by linear interpolation:

$$f_p(x) = p_x\, p_y^{-1} + x\,(1 - p_y^{-1})$$

Given a location $(\mathbf{x},1)$ on the shutter, we can find the corresponding line-of-sight location on the display screen by inverting the above equation:

$$f_p^{-1}(\mathbf{x}) = (\mathbf{x} - p_x\, p_y^{-1}) \,/\, (1 - p_y^{-1})$$

Therefore, given a location x_n on the display screen that is visible through a clear stripe on the shutter from both $\mathbf{p}$ and $\mathbf{q}$, the next such location is given first by finding the location on the shutter $f_p(x_n)$ in the line-of-sight from $\mathbf{p}$, and then finding the corresponding location on the display screen which is in the line-of-sight from $\mathbf{q}$:

$$x_{n+1} = f_q^{-1}(f_p(x_n))$$

which expands out to:

$$(p_x\, p_y^{-1} + x\,(1 - p_y^{-1}) - q_x\, q_y^{-1}) \,/\, (1 - q_y^{-1})$$

This can be expressed as a linear equation $x_{n+1} = A\, x_n + B$, where:

$$A = x\,(1 - p_y^{-1}) \,/\, (1 - q_y^{-1})$$
$$B = (p_x\, p_y^{-1} - q_x\, q_y^{-1}) \,/\, (1 - q_y^{-1})$$

The nth location in the sequence of stripe locations on the display screen can be calculated by iterating $x_{n+1} = A\, x_n + B$:

$$x_0 = 0 \quad x_1 = B \quad x_2 = AB + B$$
$$x_3 = A^2B + AB + B$$
$$x_n = B\,(A^{n-1} + \ldots + A + 1)$$

In the above sequence, the even terms locate the centers of those portions of the image visible from the right eye, and the odd terms locate the centers of those portions of the image visible from the left eye. The openings in the shutter are centered at

$$f^{-1}{}_q(x_0),\ f^{-1}{}_q(x_2), \text{ etc.}$$

3 IMPLEMENTATION

Various physical arrangements could be used to implement this technique. For our first implementation, we used an approach that would allow us the greatest flexibility and ability to conduct tests. For the display screen, we used a Digital Light Processor (DLP) micro-mirror projector from Texas Instruments [TexasInstr], because DLP projectors handle R,G,B sequentially. This allowed us to use color to encode the three time-sequential phases. We used a Ferroelectric Liquid Crystal (FLC) element from [Displaytech] to shutter the start/stop time of each temporal phase.

For the light-blocking shutter, we had a custom pi-cell liquid crystal screen built to our specifications by [LXD], which we drove from power ICs mounted on a custom-made Printed Circuit Board (PCB). To control the sub-frame timings, we used a Field Programmable Gate Array (FPGA) from [Xilinx]. These were all driven from a Pentium II PC, running OpenGL in Windows NT.

3.1 Architecture

As flowcharted in [Figure 8] the steps to display a frame are:

(1) An eye tracker locates the observer's eyes, and sends this information to the CPU.

(2) The main CPU uses the eye tracker info to render two 3D scenes: one as seen from each eye.

(3) The main CPU also uses the eye tracker info to compute, for each of three phases, the proper left/right alternation pattern. These are interleaved into three successive time phases as red, green, and blue, respectively.

(4) The main CPU also uses the eye info to compute the three phases of stripe on the light shutter. These are encoded into three one-dimensional bit-maps, each indicating an on-off pattern for the shutter micro-stripes at one of the three phases. These bit-maps are shipped to the FPGA.

(5) The FPGA sends the three bit-patterns to the pi-cell light shutter in rotating sequence, every 1/180 second. The timing for this is controlled by the DLP projector, which produces a signal every time its color wheel advances.

(6) The DLP projector displays the three image phases in succession. The color wheel on the projector is removed, so that each of the red, green, and blue components displays as a gray scale image.

(7) The FLC element is modulated by the FPGA to block the light from the DLP projector lens in a 180 Hz square wave pattern. This allows finer control over timing.

(8) A rear projection screen (RPS) diffuses the image from the DLP projector.

(9) The pi-cell light shutter positioned in front of the RPS displays a different horizontally varying on-off pattern every 1/180 second.

Steps (5) through (9) above are part of the ‘‘real-time subsystem” which is monitored by the FPGA. These parts of the process are monitored continuously by the FPGA to synchronize all the events which must occur simultaneously 180 times per second.

Creating the three phased images

We use OpenGL to encode the red/green/blue sub-images which the DLP projector will turn into time sequential phases. To do this, we first render the compute separate left and right images in OpenGL, into off-screen buffers, as show in [Figures 6a,6b].

Then we slice each of these into their component image stripes, and reconstruct into three interleaved images that will be displayed in rapid sequence, as red, green, and blue components, as shown in [Figures 7a,7b,7c], respectively.

If this image were simply displayed on an unenhanced monitor, it would appear as in [Figure 9]. When filtered through the light-blocking shutter, each of the observer’s eyes will reconstruct a complete image from a single viewpoint. If the DLP projector’s color wheel were engaged, then the left and right eyes would see [Figure 10a] and [Figure 10b], respectively. With the color wheel removed, each of the observer’s eyes simply sees the correct stereo component image of [Figure 6a] and [Figure 6b], respectively.

Timing requirements

There are two types of timing we need to address for this display: frame time, and shutter switching time.

In order to prevent eyestrain due to movement latency, we ideally want to maintain a frame refresh rate of at least 60 Hz, with a latency within 1/60 second between the moment the observer’s head moves and the moment the correct image is seen. This consideration drove the timing design goals for the display: to be able to respond within the 1/60 interval from one screen refresh to the next. Within this time window, we make standard assumptions: that there is a known and fixed small latency to compute a frame, and that a Kalman filter [Grewal] can extrapolate from recent eye-tracking samples to predict reasonable eye positions at the moment of the next display refresh. If the user’s head is moving, then the host computer should ideally compute the left and right images and merge them within this 1/60 second window.

The real-time subsystem maintains a more stringent schedule: a synchronous 180 Hz cycle. The pattern on the light-shutter needs to switch at the same moment that the DLP projector begins its red, green, or blue component. This timing task is handled by the FPGA, which reads a signal produced by the projector every time it the color wheel cycles (about once every 1/180 second) and responds by cycling the light shutter pattern. To help tune the on/off timing, the FPGA modulates a ferro-electric optical switch which is mounted in front of the projector lens.

The main CPU is not involved at all in this fine-grained timing. The only tasks required of the CPU are to produce left/right images, to interleave them to create a red/green/blue composite, and to put the result into an on-screen frame buffer, ideally (but not critically) at 60 frames per second.

3.2 The Parts

The essential components we used to implement this process are shown in the photograph [Figure 11] below. In this section, each is described in some detail.

FPGA

Every 1/180 of a second (three times per frame, from the observer’s point of view), we need to update the light shutter with a different phase pattern of on/off stripes. To do this quickly enough, we built an ISA interface board with a non volatile Xilinx 95C108 PLD and a reconfigurable Xilinx XC4005E FPGA. The PLD is used to generate the ISA Bus Chip Select signals and to reprogram the FPGA. The XC4005E is large enough to contain six 256 bit Dual Ported RAMs (to double buffer the shutter masks needed for our three phases), the ISA Bus logic, and all the hardware needed to process the DLP signals and drive the pi-cell. When loaded with the three desired patterns from the main CPU, this chip continually monitors the color wheel signals from the DLP projector. Each time it detects a change from red to green, green to blue, or blue to red, it sends the proper signals to the Supertex HV57708 high voltage Serial to parallel converters mounted on the Pi-cell, switching each of the light shutter’s 256 microstripes on or off.

Pi-cell

A standard twisted nematic liquid crystal display (such as is widely used in notebook computers) does not have the switching speed we need; requiring about 20 msec to relax from its on state to its off state after charge has been removed. Instead, we use a pi-cell, which is a form of liquid crystal material in which the crystals twist by 180^o (hence the name) rather than that 90^o twist used for twisted nematic LC displays.

Pi-cells have not been widely used partly because they tend to be bistable - they tend to snap to either one polarization or another This makes it difficult to use them for gray scale modulation. On the other hand, they will relax after a charge has been removed far more rapidly than will twisted nematic - a pi-cell display can be driven to create a reasonable square wave at 200 Hz. This is precisely the characteristic we need - an on-off light blocking device that can be rapidly switched. Cost would be comparable to that of twisted nematic LC displays, if produced at comparable quantities.

[Figure 12a] and [Figure 12b] show the pi-cell device that was manufactured for us by [LXD]. The image to the left shows the size of the screen, the close-up image to the right shows the individual microstripes and edge connectors. The active area is 14"x12", and the microstripes run vertically, 20 per inch. The microstripe density could easily have exceeded 100 per inch, but the density chosen required us to drive only 256 microstripes, and was sufficient for a first prototype. Edge connectors for the even microstripes run along the bottom; edge connectors for the odd microstripes run along the top. We used four power chips to maintain the required 40 volts, each with 64 pin-outs. Two chips drive the 128 even microstripes from a PCB on the top of the shutter, the other two drive the 128 odd microstripes from a PCB along the bottom. To turn a microstripe transparent, we drive it with a 5 volt square wave at 180 Hz. To turn a microstripe opaque, we drive it with a 40 volt square wave at 180 Hz.

Ferro-electric optical switch

A ferro-electric liquid crystal (FLC) will switch even faster than will a pi-cell, since it has a natural bias that allows it to be actively driven from the on-state to the off-state and back again. A ferro-electric element can be switched in 70 microseconds. Unfortunately ferro-electric elements are very delicate and expensive to manufacture at large scales, and would therefore be impractical to use as our light shutter. However, at small sizes they are quite practical and robust to work with. We use a small ferro-electric switch over the projector lens, manufactured by Displaytech [Displaytech], to provide a sharper cut-off between the three phases of the shutter sequence. We periodically close this element between the respective red, green, and blue phases of the DLP projector's cycle. While the FLC is closed, we effect the pi-cell microstripes transitions (which require about 1.2 ms).

User tracking

After surveying a number of different non-invasive eye tracking technologies available, we settled on the use of retroreflective camera based tracking. Because the back of the human eyeball is spherical, the eye will return light directly back to its source.

A system based on this principle sends a small infrared light from the direction of a camera during only the even video fields. The difference image between the even and odd video fields will show only two glowing spots, locating the observer's left and right eyes, respectively. By placing two such light/camera mechanisms side-by-side, and switching them on during opposite fields (left light on during the even fields, and right light on during the odd fields), the system is able to simultaneously capture two parallax displaced images of the glowing eye spots. The lateral shift between the respective eye spots in these two images is measured, to calculate the distance of each eye.

The result is two (x,y,z) triplets, one for each eye, at every video frame. A Kalman filter [Grewal] is used to smooth out these results and to interpolate eye position during the intermediate fields. A number of groups are planning commercial deployment of retroreflective-based tracking in some form, including IBM [Flickner]. For calibration tests we used the DynaSite from Origin Systems [Origin], which requires the user to wear a retroreflective dot, but does not block the user's line of sight.

The user tracking provides as a pair of 3D points, one for each eye. As noted above, this information is used in three ways. *(i)* Each of these points is used by OpenGL as the eye point from which to render the virtual scene into an offscreen buffer; *(ii)* The proper succession lateral locations for left/right image interleaving is calculated, which is used to convert the left/right offscreen images into the three temporally phased images; *(iii)* The proper positions for the light shutter transitions are calculated. This information is converted to three one dimensional bit-maps, each indicating an on-off pattern for the shutter micro-stripes at one of the three phases. This information is sent to the FPGA, which then sends the proper pattern to the light shutter every 1/180 second, synchronously with the three phases of the DLP projector.

3.3 Experience

The goals of this current research version of the system were *(i)* low latency and *(ii)* absence of artifacts. In this section we discuss how well our experience matched those goals.

The most important question to answer is: ‘‘does it work?” The answer is yes. As we expected, the experience is most compelling when objects appear to lie near the distance of the display screen, so that stereo disparity is reasonably close to focus (which is always in the plane of the projection screen). When the system is properly tuned, the experience is compelling; as an observer looks around an object, it appears to float within the viewing volume. The observer can look around the object, and can position himself or herself at various distances from the screen as well. Special eyewear is not required.

The system always kept up with the renderer. Our software-implemented renderer did not achieve a consistent 60 frames per second, but rather something closer to 30 frames per second. In practice this meant that if the observer darted his/her head about too quickly, the tracker could not properly feed the display subsystem when the user moved his/her head rapidly.

The more critical issue is that of position-error based artifacts. Not surprisingly, we have found that it is crucial for the system to be calibrated accurately, so that it has a correct internal model of the observer's position. If the tracker believes the observer is too near or far away, then it will produce the wrong size of stripes, which will appear to the observer as vertical stripe artifacts (due to the wrong eye seeing the wrong image) near the sides of the screen. If the tracker believes the observer is displaced to the left or right, then this striping pattern will cover the entire display. We found in practice that a careful one-time calibration removed all such artifacts. This emphasizes the need for good eye position tracking.

One artifact we observed, which is exhibited by all polarization-based stereoscopic displays, is a small amount of ghosting - a faint trace of the wrong image is seen by each eye. This ghosting becomes noticeable when a bright object is placed against a black background. This is at least partly due to imperfections in the polarization; some light is scattered in the optical shutter and therefore becomes wrongly polarized. Some of this ghosting may also be due to imperfections in the timing, so that some light from the wrong phase gets through while the pi-cell shutter is still settling. In ongoing work, we plan to test this hypothesis by systematically varying the timing of the ferroelectric switch.

4 ONGOING WORK

We are designing an alternate version of this display that will work in full color with current stereo-ready CRT monitors. This will require a more sophisticated light-blocking shutter, since CRT monitors use a progressive scan, rather than displaying an entire image at once. For this reason, this version of the shutter will have separately addressable multiple bands from top to bottom, triggered at different times within the CRT monitor's scan cycle. This version would be in full color, since it will create phase differences by exploiting the time variation between different portions of the full-color CRT's vertical scan, instead of relying on sequential R,G,B to produce time phases.

In parallel, we are working with manufacturers of rapidly switchable flat-panel displays, to create a flat panel version. This version would be in full color, since it would not rely on sequential R,G,B. One of our goals for this flat-panel based version is a hand-held "gameboy" or "pokémon" size platform, for personal autostereoscopic displays. The costs of the pi-cell light shutter and its associated control electronics is roughly proportional to display area, which leads us to believe that portable hand-held autostereoscopic displays can be a practical low cost platform. This configuration will also depend on the success of ongoing work in the development of low cost eye position tracking for handheld platforms.

One of our current projects will use this display platform for teleconferencing. With a truly non-invasive stereoscopic display, two people having a video conversation can perceive the other as though looking across a table. Each person's image is transmitted to the other via a video camera that also captures depth [Kanade95]. At the recipient end, movements of the observer's head are tracked, and the transmitted depth-enhanced image is interpolated to create a proper view from the observer's left and right eyes, as in [Chen]. Head movements by each participant reinforce the sense of presence and solidity of the other, and proper eye contact is always maintained.

We plan to implement an API for game developers, so that users of accelerator boards for two-person games can make use of the on-board two-view hardware support provided in those boards to simultaneously accelerate left and right views in our display. We are also investigating variants of this system for two observers.

ACKNOWLEDGEMENTS

This work was supported by the NYU Center for Advanced Technology, NY3D Inc., and the Interval Research. The authors would also like to thank Clilly Castiglia, Chris Poultney, Jay Konopka, Michael Wahrman Dennis Zorin, and everyone else on the 12th floor who provided their time, effort and moral support.

References

Actuality Systems: http://actuality-systems.com/

S. Benton, T.E. Slowe, A.B. Kropp, and S.L. Smith, Micropolarizer-based Multiple-Viewer Autostereoscopic display. SPIE Proceedings Volume 3639: Stereoscopic Displays and Virtual Reality Systems VI, (SPIE January 1999) paper 3639-10.

S. Benton. The Second Generation of the MIT Holographic Video System. In: J. Tsujiuchi, J. Hamasaki, and M. Wada, eds. +Proc. of the TAO First International Symposium on Three Dimensional Image Communication Technologies. Tokyo, 6-7 December 1993. Telecommunications Advancement Organization of Japan, Tokyo, 1993, pp. S-3-1-1 to -6.

R. Börner. Three Autostereoscopic 1.25m Diagonal Rear Projection Systems with Tracking Features. IDW´97, Proc. of 4th Int'l Display Workshop, Nagoya, Japan, Nov. 1997, p.835-838

S. Chen and L. Williams. View Interpolation for Image Synthesis. Computer Graphics (SIGGRAPH 93 Conference Proc.) p.279-288.

Displaytech: http://www.displaytech.com/shutters.html

Elizabeth Downing et.al. A Three-Color, Solid-State, Three-Dimensional Display. Science 273,5279 (Aug. 30, 1996), pp. 1185-118.

D. Drascic, J. Grodski. Defence Teleoperation and Stereoscopic Video. Proc SPIE Vol. 1915, Stereoscopic Displays and Applications IV, pages 58-69, San Jose, California, Feb 1993.

J. Eichenlaub. Multiperspective Look-around Autostereoscopic Projection Display using an ICFLCD. Proc. SPIE Vol. 3639, p. 110-121, Stereoscopic Displays and Virtual Reality Systems VI, John O. Merritt; Mark T. Bolas; Scott S. Fisher; Eds.

J. Eichenlaub, Lightweight Compact 2D/3D Autostereoscopic LCD Backlight for Games, Monitor, and Notebook Applications. Proc. SPIE Vol. 3295, p. 180-185, in Stereoscopic Displays and Virtual Reality Systems V, Mark T. Bolas; Scott S. Fisher; John O. Merritt; Eds. April 1998.

M. Flickner: http://www.almaden.ibm.com/cs/blueeyes/find.html

M. Grewal, A. Andrews, Kalman Filtering: Theory and Practice, Prentice Hall, 1993.

T. Kanade, et al. Development of a Video Rate Stereo Machine. Proc. of International Robotics and Systems Conference (IROS-95), Pittsburgh, PA, August 7-9, 1995.

L. Lipton, et. al., U.S. Patent #4,523,226, Stereoscopic Television System, June 11,1985

L. Lipton, and J. Halnon. Universal Electronic Stereoscopic Display. Stereoscopic Displays and Virtual Reality Systems III, Vol. 2653, pp. 219-223, SPIE, 1996

LXD: http://www.lxdinc.com/

J.R. Moore, N.A. Dodgson, A.R.L. Travis and S.R. Lang. Time-Multiplexed Color Autostereoscopic Display. Proc. SPIE 2653, SPIE Symposium on Stereoscopic Displays and Applications VII, San Jose, California, Jan 28-Feb 2, 1996, pp. 10-19.

Okoshi, T. Three-Dimensional Imaging Techniques. Academic Press, New York 1976. ISBN 0-12-525250-1.

Origin Systems: http://www.orin.com/3dtrack/dyst.htm

A. Schwerdtner and H. Heidrich. Dresden 3D display (D4D). SPIE Vol. 3295, p. 203-210, Stereoscopic Displays and Virtual Reality Systems V, Mark T. Bolas; Scott S. Fisher; John O. Merritt; Eds.

P. St.-Hillaire, M. Lucente, J.D. Sutter, R. Pappu, C.J.Sparrell, and S. Benton. Scaling up the MIT Holographic Video System. Proc. of the Fifth International Symposium on Display Holography (Lake Forest College, July 18-22, 1994), SPIE, Bellingham, WA, 1995.

Texas Instruments: http://www.ti.com/dlp

R. Williams. Volumetric Three Dimensional Display Technology in D. McAllister (Ed.) Stereo Computer Graphics and other True 3D Technologies, 1993

G. J. Woodgate, D. Ezra, et.al. Observer-tracking Autostereoscopic 3D display systems. Proc. SPIE Vol. 3012, p.187-198, Stereoscopic Displays and Virtual Reality Systems IV, Scott S. Fisher; John O. Merritt; Mark T. Bolas; Eds.

Xilinx: http://www.xilinx.com/

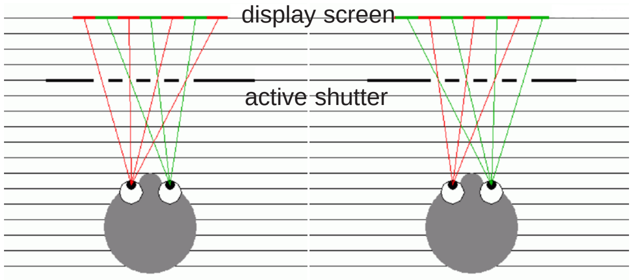

figure 1a

figure 1b

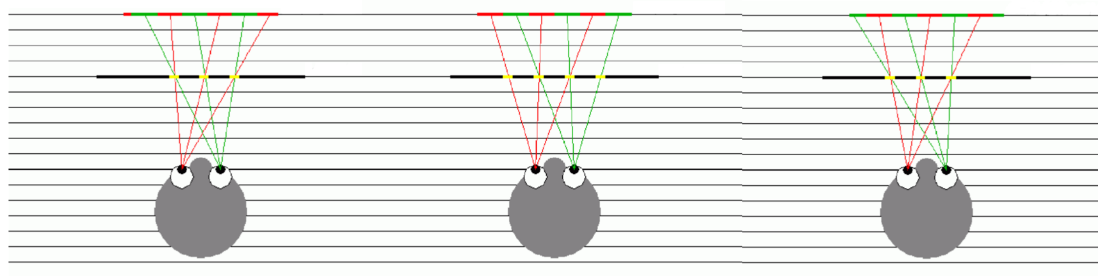

figure 2a

figure 2b

figure 2c

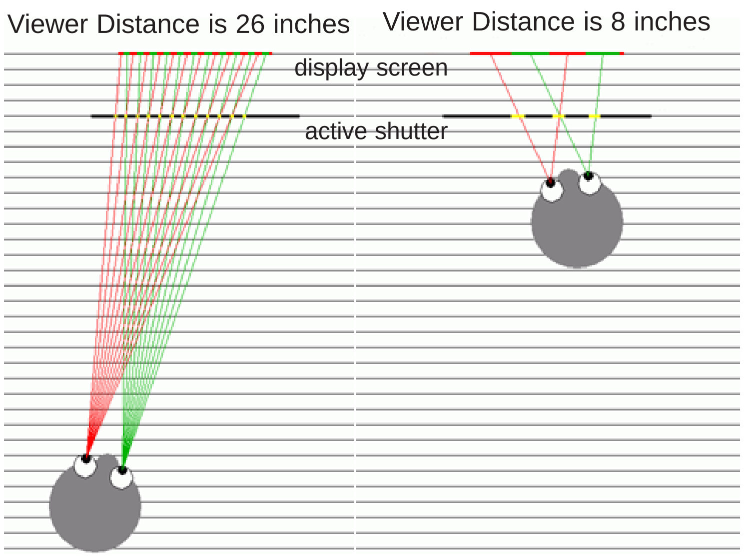

figure 3a

figure 3b

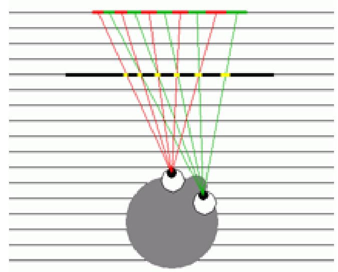

figure 4

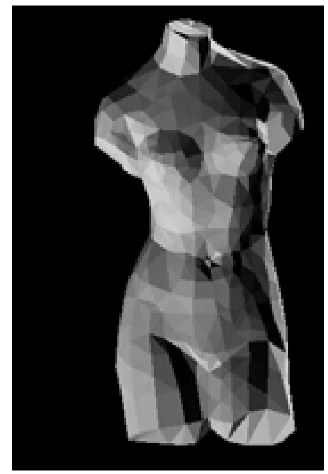

figure 6a

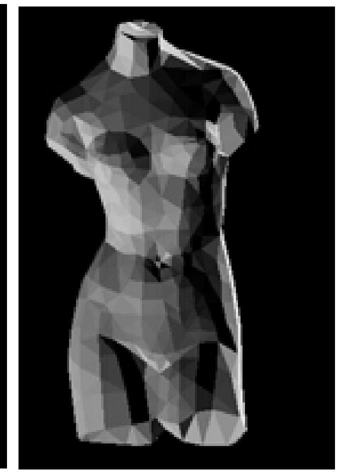

figure 6b

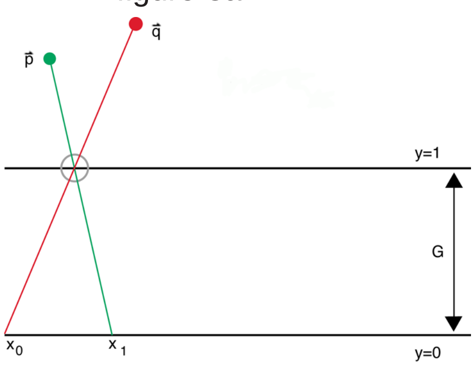

figure 5a

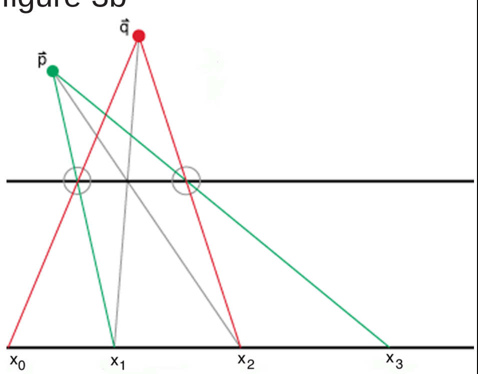

figure 5b

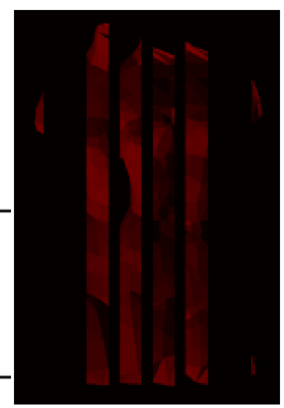

figure 7a

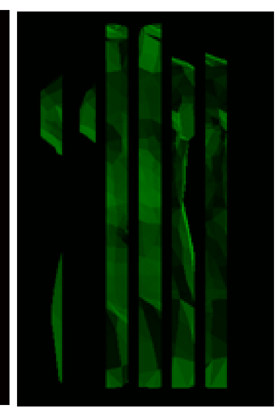

figure 7b

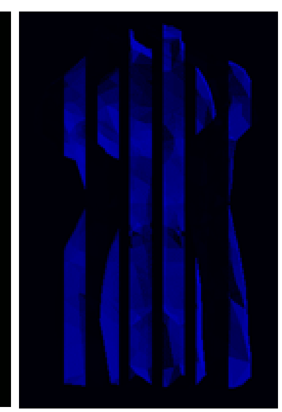

figure 7c

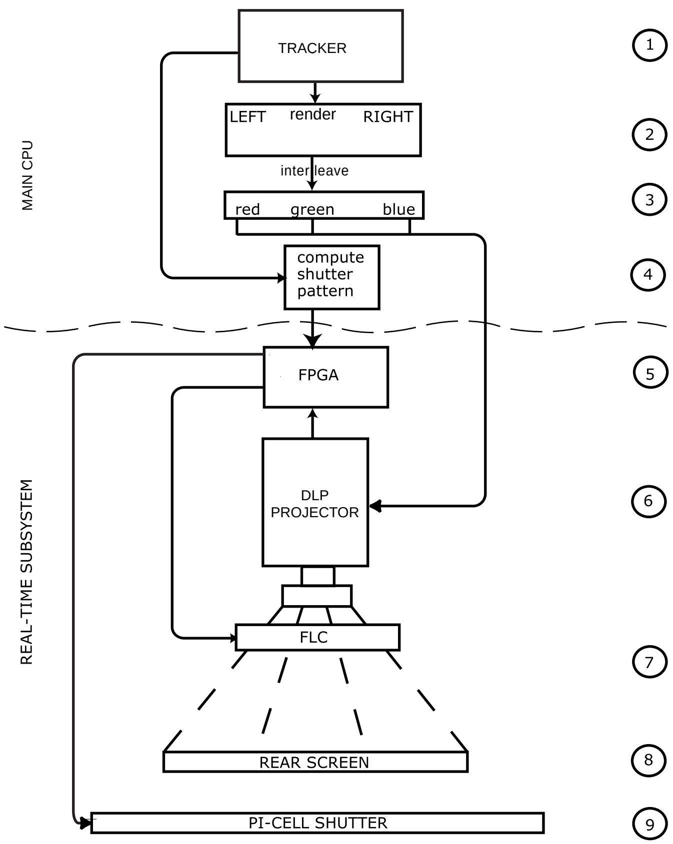

figure 8

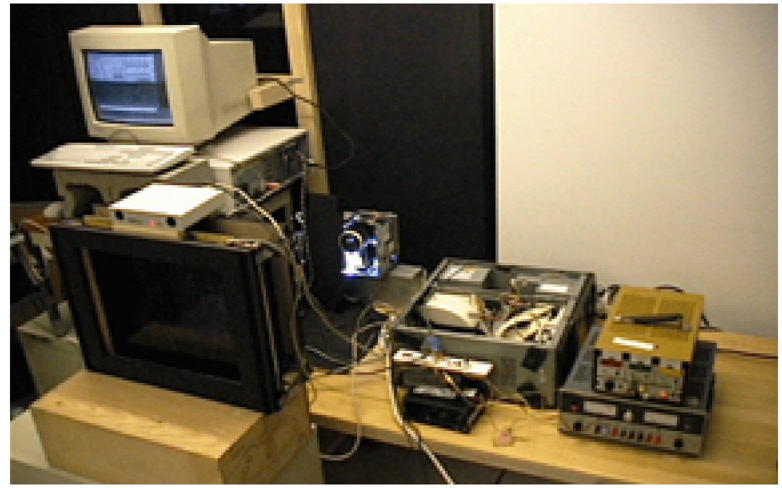

figure 11

figure 12a

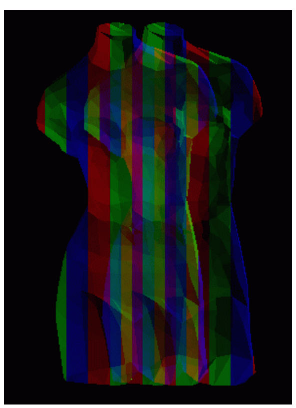

figure 9

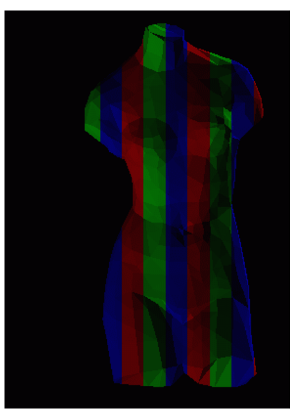

figure 10a

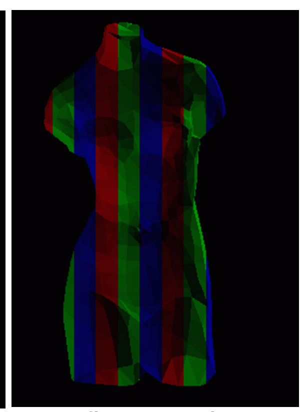

figure 10b

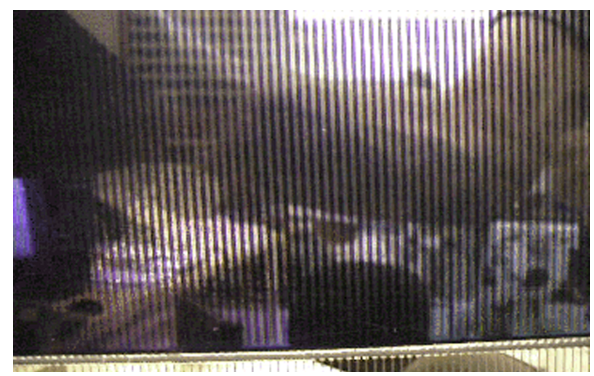

figure 12b