3D-Stereoscopic X Windows

Scott A. Safier
M. W. Siegel

The Robotics Institute, Carnegie Mellon University
Pittsburgh PA 15213

ABSTRACT

All known technologies for displaying 3D-stereoscopic images are more or less incompatible with the X Window System. Applications that seek to be portable must support the 3D-display paradigms of multiple hardware implementations of 3D-stereoscopy. We have succeeded in modifying the functionality of X to construct generic tools for displaying 3D-stereoscopic imagery. Our approach allows for experimentation with visualization tools and techniques for interacting with these synthetic worlds. Our methodology inherits the extensibility and portability of X. We have demonstrated its applicability in two display hardware paradigms that are specifically discussed.

Keywords: Stereoscopic displays, X Window System, computer graphics

1. INTRODUCTION

The X Window System (X) has become an industry standard adopted by many workstation vendors. Technologies have become available for realistic 3D-stereoscopic (binocular) viewing of video and computer-generated imagery. Developers of applications who want to use 3D-stereoscopic displays must confront merging these technologies with X.

X provides tools for managing windows that are assumed to have two dimensions; it does not take into account the possibility of a three-dimensional screen. Windows cannot be positioned at a depth, nor can the attributes of a 3D world be specified. Additionally, all the graphical user-interface tools for X, such as Motif, inherit X's two-dimensional world.

There are a variety of viable technologies used for displaying 3D-stereoscopic images but they are generally more or less incompatible with X. Stereographics' technology requires the left and right fields of the image to be stacked vertically about the center of the screen, and inserts an extra vertical sync-pulse; it is aesthetically desirable to make other 2D X windows invisible when using this technology. Toshiba™ has a system that interlaces the left and right fields of a 3D-stereoscopic image. Liquid crystal displays offer the possibility of subpixel polarization to encode the left and right fields of 3D-stereoscopic images. Applications that are intended to be portable must support the paradigms of these hardware platforms and others.

Our software is implemented on a Silicon Graphics workstation that is stereo-ready. We are using two different display techniques in our work. SGI's hardware implements Stereographics' mode and provides SGI compatible hardware to drive Stereographics Crystal Eyes goggles. We are also using the machine's double buffering capability to drive the stereo-in-a-window technique described in [1]. Toshiba goggles are gated by the hardware's parallel port and are synchronized with the buffer swapping of the images.

We have been working on using the existing functionality of X to construct generic tools for displaying 3D-stereoscopic imagery. Our approach allows for experimentation with visualization techniques and techniques for interacting with synthetic worlds. The implementations of these generic tools necessarily need to differ betweem the hardware modes used to display the 3D-stereoscopic scenes, but this effect is minimized through a uniform application interface.

We have begun to explore what augmentation the X Window Systems needs to support 3D-stereoscopic displays. We take the position that 3D-stereoscopy is an attribute of a window, just as color, perspective drawing or having a border is an attribute of a window. Although 3D-ness is not a native characteristic of X windows, we are pushing existing technology to understand the problems associated with building 3D-stereoscopic software applications.

Our first step in this exploration uses the X Toolkit Intrinsics package (Xt) to create a user-interface component (widget) for a 3D-stereoscopic window. Xt uses an event-driven paradigm, where programmed actions occur in response to particular input from a user or specific conditions that occur in an application. Toolkits such as Motif and Open Look™ are built on top of Xt.

We have designed a two-tiered hierarchy of widgets that permit rendering and interacting with 3D synthetic worlds within an X and Motif environment. First, we construct a widget for the particular hardware platform. This is augmented with a binocular viewing system that renders geometrically correct 3D-stereo scenes. These are described in detail in the rest of this paper, along with several applications that they are used in.

Some hardware modes for displaying 3D-stereoscopic scenes have the problem that 2D and 3D windows cannot comfortably co-exist on the same screen. They require that a second type of widget be designed for these 3D-stereoscopic displays. This widget, known as a *geometry-manager* widget, would automatically position normally 2D windows correctly within a 3D-stereoscopic display, allowing applications to use existing toolkits, like Motif. We have only begun to explore the issues associated with this second step.

Ultimately, we believe that future window systems must consider the requirements of 3D-stereoscopic applications from their inception; but these requirements are yet to be fully articulated and understood. When they are, protocols for efficiently transmitting graphical data to 3D-stereoscopic displays can be developed. As the use of 3D-stereoscopic displays becomes more common, user-interface toolkits can be developed for commonly used interaction paradigms analogous to menus and push-buttons for 2D-displays

2. IMPLEMENTING A 3D-STEREOSCOPIC GRAPHICS TOOL

2.1 The Stereo Window Widget

The widget most closely linked to each particular hardware mode is the corresponding *stereo window widget* instantiation. It creates an X/Motif window that has both a left and a right field. An application defined function generates either the left or the right image whenever an event to redisplay the contents of the window is processed. Application program interfaces to this widget are generic and are as independent as possible of the display hardware. An application can support multiple hardware modes simply by compiling it in each case with the appropriate hardware-specific library.

As we guessed, there are some general incompatibilities between X and 3D-stereoscopic displays. The first problem we encountered was the pointer[1] provided by X. This 2D pointer is incompatible with 3D-stereoscopic viewing because it appears in only one field. There has been much research on pointers and their interaction with 3D synthetic worlds [2,3], but all concrete results are application specific. We did not try to solve all the problems associated with designing an application independent 3D cursor either. Instead, we create a shadow pointer that tracks the motion of the real pointer in both fields of the stereo window. We create several subroutines that permit the application to control the pointer, including one that computes a disparity for the shadow pointer to give it depth.

[1] The term *pointer* comes from the X literature and refers to the cursor on the screen.

2.1.1 Stereographics

Stereographics Inc. provides a technology for viewing 3D stereoscopic images that requires they be displayed on the screen simultaneously by stacking them vertically. SGI's hardware implements Stereographics' mode by inserting an extra vertical sync-pulse; this has the effect of doubling the image height but halving the number of vertical lines each eye sees. For images to have the appropriate aspect ratio they must be anamorphically doubled horizontally by the program that generates the graphic. Stereographics provides SGI compatible hardware to drive LCD goggles that are synchronized with the toggling of the images on the screen.

A 3D-stereoscopic application that uses Stereographics hardware must consider this display technology when designing its graphical user interface. The 2D windows become distractions when the screen is split because they are stretched and appear in only one of the viewer's eyes. When using this hardware mode, it is thus desirable to hide all the other windows not associated with the 3D-stereoscopic application. The easy way to accomplish this is for the application to take over the full screen.

The stereo window widget for this hardware mode creates two subwindows of a designated size. The subwindows are aligned in each field to appear at a given position on the screen. The subwindows are linked so that they behave as a single widget within the user interface.

Creating the stereo window pairs is only part of the solution for this hardware mode. Like X , widgets from toolkits such as Motif or Open Look™ were designed for a 2D display. If an application uses widgets from other toolkits, such as Motif, these widgets must be positioned with care. Some widgets, like those that implement push buttons, can be duplicated to appear properly in both the left and right fields of the display; others can't be implemented properly in this paradigm: they can be seen flat in the right eye, or the left eye, but not in both.

There may also be application-level problems associated with this hardware mode. Consider for example a program that displays precomputed binocular images. This application may use scrollbars to control the horizontal and vertical disparity. This behavior would be simple in a 2D application using Motif's scrolled window widget, but is difficult in a stereo application because there is no corresponding 3D scrolled window widget. Our implementation of the stereo window widget for the Stereographics hardware mode tries to facilitate this type of interaction by linking the enclosing widget to the stereo window widget. Although window positioning is automated, synchronizing the behavior of the left and right fields of the enclosing widget is left to the application. For example, the application that displays precomputed stereo pairs must set the position of the left field's scroll bars based upon the position of the scroll bars in the right field.

2.1.2 *Stereo-in-a-window*

Stereo-in-a-window is a display technique that we developed to use double buffering to display 3D-stereoscopic images on the same screen as 2D X windows. Display of the left and right image fields are synchronized with a set of liquid crystal goggles through either the serial or parallel port of the host machine. As the goggles are toggled, the image buffers are simultaneously swapped to display the appropriate image.

The implementation of the stereo window widget for this hardware mode must drive the goggles and synchronize them with the field currently being displayed. Optimally, field swapping occurs with each refresh of the monitor. This occurs at 72 Hz on our particular hardware platform. The timing facilities provided by X are used to generate an event at the specified time to swap the fields. We have found that the size of the window and the complexity of the images being rendered affect the maximum refresh speed, and that a swapping 72 times a second is seldom achieved on our current hardware. This leads to flicker with large or complex imagery; however, it is in principle easily overcome by upgrading the underlying computer power.

The principle advantage to the stereo-in-a-window technique is that the stereo window can co-exist with other 2D windows, such as those provided by Motif. Application programs need not take over the full screen to hide other windows, nor must the application duplicate widgets for the right and left fields. Likewise, there are no special requirements for taking advantage of the full range of widgets from other user-interface toolkits. Thus, this is the preferable mode in all respects except the requirement for a more powerful underlying computer; as computer power becomes ever cheaper, the cost of the computer power eventually becomes a non-issue.

2.1.3 Applications

The stereo window widget provides the basic functionality of a 3D-stereoscopic window for X. We have used this widget in several image processing applications. The first application we developed displays precomputed stereo pairs and the user to adjust the horizontal and vertical disparities. Our second application is a modification of the above that displays video image sequences. This application demonstrates the stereo look around capability described in [4]. These applications demonstrate the feasibility of mixing 3D-stereoscopic windows among 2D X windows.

2.2 Perspective Rendering for 3D Graphic Applications

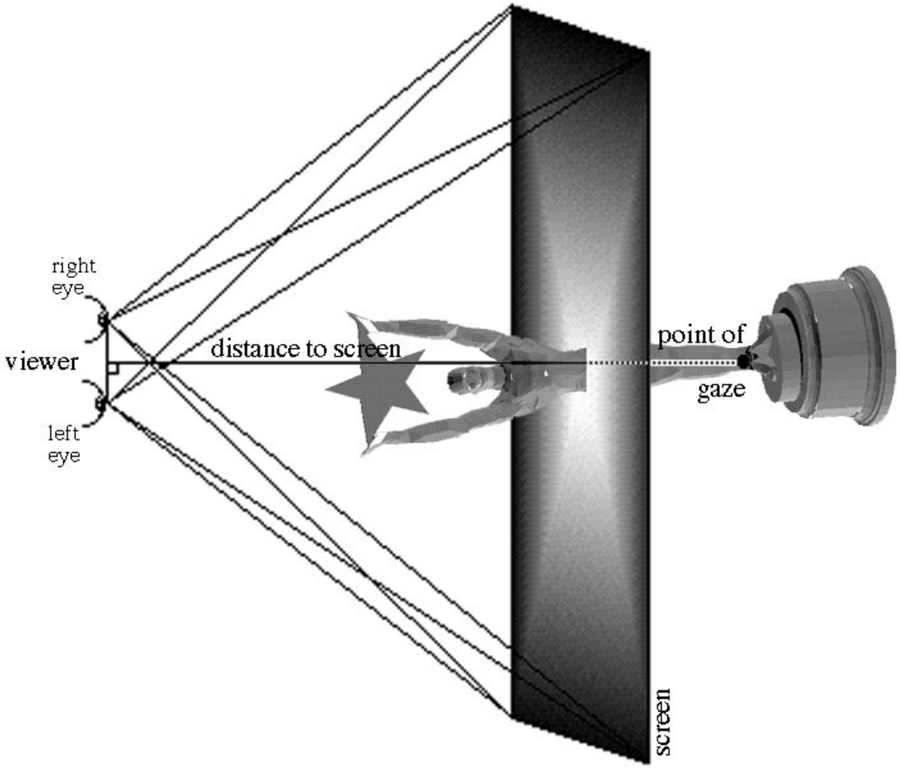


Figure 1: Binocular viewing system

The stereo window widget encapsulates the necessary functionality to drive a 3D-stereo display and provides applications with a capability to display 3D-stereoscopic images in an X window, but it lacks any data about the viewer of these images. In [5], we show that it is necessary to know the position of the viewer's eyes relative to the monitor in order to render geometrically correct stereo image-pairs. The correct stereo-image pairs are uniquely associated with the position of the viewer in this binocular viewing system. We have designed and implemented a *stereo 3D widget* that includes all the functionality of the stereo window widget and extends it to include data about the binocular viewing system. With this widget, applications can render and interact with 3D synthetic worlds.

All measurements in our system are in real units (e.g. millimeters) and we require the synthetic world to be rendered in a right-handed coordinate system. The monitor must also be calibrated so that the exact height and width of the drawable portion of the screen are known.

For simplicity, we make several restrictive assumptions about the binocular viewing system.

1. The distance between the user and the monitor is always equal to the distance from the synthetic eyes to the synthetic screen.
2. The user is assumed to be looking at the center of the screen. Any horizontal or vertical offsets are defined from the center of the screen.
3. The line that runs through the midpoint of each eyes is always parallel to the horizontal axis of the display hardware. The line that runs through the midpoint of each synthetic eye is always oriented as the horizontal of the synthetic screen.

These second and third assumptions can be eliminated with the use of an instrument that returns both the position and the orientation of the viewer's head. All the commercial head trackers that we know about only provide position information. They do not return information about the orientation of the eyes. For these head trackers, an assumption about the orientation of the eyes relative to the screen must still be made.

Figure 1 graphically depicts the binocular viewing system. The position of the viewer's eyes are located relative to a screen of projection and these form an eye-screen system that is used to view the synthetic world. Objects in a world cast two central projections upon the screen, one for each eye. The viewer is oriented toward a point in the synthetic world, called the *point of gaze*. The eye-screen system is positioned in the synthetic world such that there is a one-to-one correspondence with the real world.

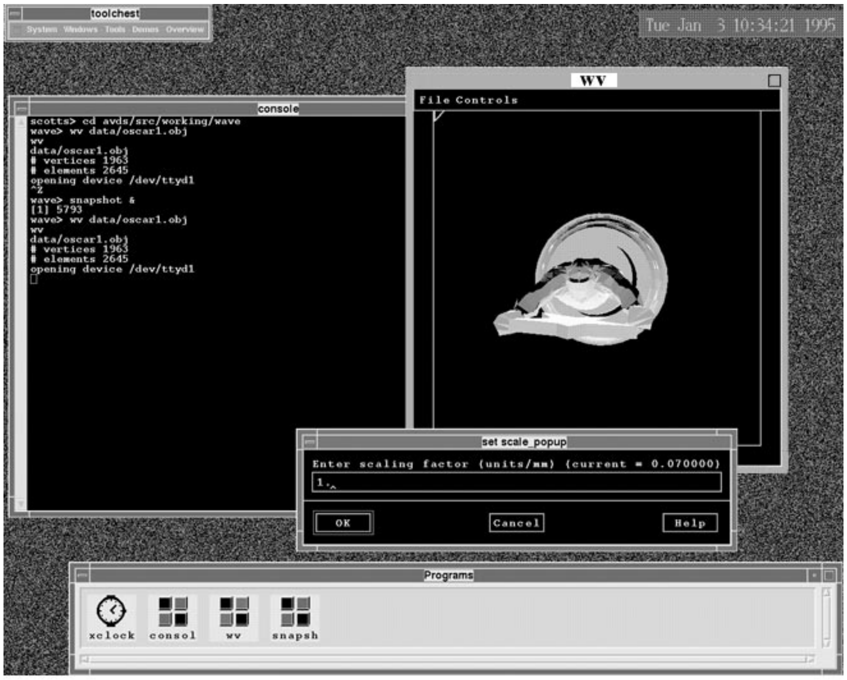
We have provided two methods of defining the binocular viewing system in the stereo 3D widget. Both require that the distance from the eyes to the screen be known, as well as the orientation of the screen. Both also define a point in the synthetic world known as the point of gaze. The eye-screen system is oriented such that the line between the midpoint of the eyes and the point of gaze intersects the screen. Either the position of the right and left eyes in the synthetic world can be specified exactly, or, in the other case, the midpoint between the eyes and the pupilary distance can be used to calculate the positions of each eye.

Several coordinate systems can be used to describe the binocular viewing system. We made a design decision to represent the coordinates of elements in our binocular viewing system in the same coordinate system as the synthetic world. This system is the most natural for the application since the coordinate system of the synthetic world is defined by the application. However, this coordinate system is not necessarily the most natural choice for other purposes, such as reasoning about the eye-screen system or computations involving the real world. There is however a unique mapping between each of these coordinate systems that can be computed with little difficulty.

Besides the binocular viewing system, the stereo 3D widget provides other functionality to the application. It is possible to scale the synthetic world so that objects appear proportionally larger or smaller. Navigation in the synthetic world occurs by moving the eye-screen system. We provide the application with several functions to translate the eye-screen system, or rotate it about the point of gaze or the eye midpoint. Lighting sources in the synthetic world need only be specified once. The stereo 3D widget maintains a lighting data base that assures that the appropriate model is employed every time the scene is drawn. We are also currently experimenting with synthetic optical systems, such binoculars and stereo microscopes [6]. These optical systems are defined in terms of the focal length of the device, the distance between the principal planes, and the indices of refraction in object and image spaces. Lastly, there is a method of training to help a user overcome the convergence-accommodation conflict. When the user is being trained, the disparity is set to zero, and increase linearly with time until it has reached its intended value.

2.2.1 Applications

We have developed several applications that use there stereo 3D widget inside an X windows graphical user interface. These applications demonstrate the ability to mix 3D widgets with conventional 2D Motif widgets. Figure 2 demonstrates this capability in a stereo-in-a-window application that displays a geometric model of a statue (window WV). It is surrounded by several normal 2D X windows that occupy the workstation's desktop including a subwindow of the application that is querying the user for a scaling factor for the image.

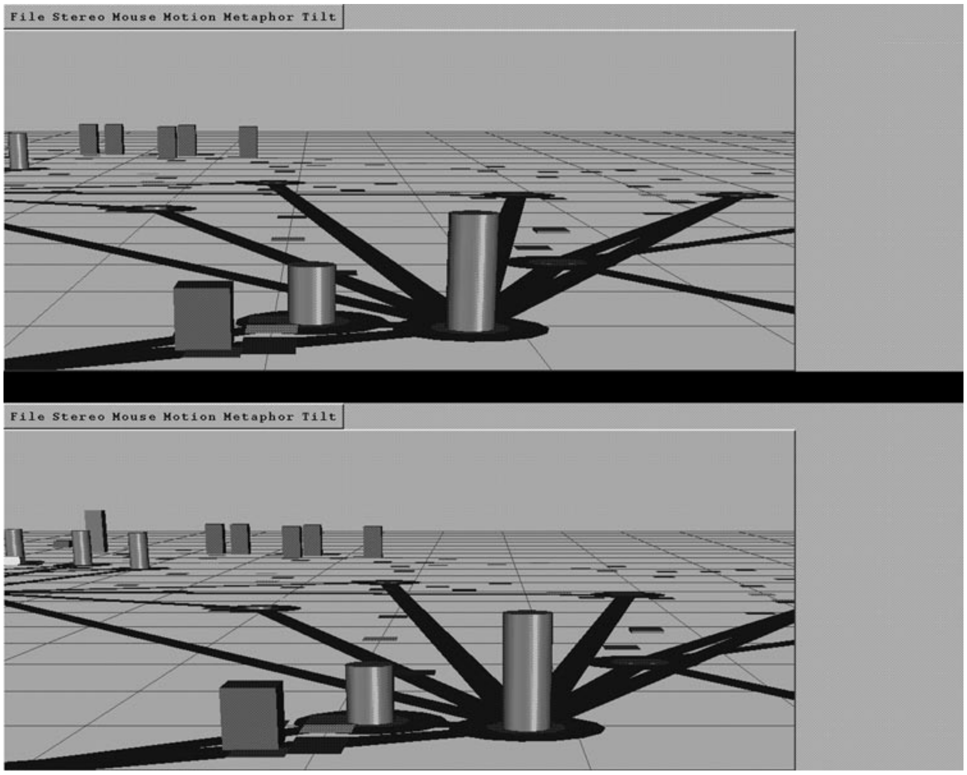


```
scotts> cd avds/src/working/wave
wave> wv data/oscar1.obj
wv
data/oscar1.obj
# vertices 1963
# elements 2645
opening device /dev/ttyd1
^Z
wave> snapshot &
[1] 5793
wave> wv data/oscar1.obj
wv
data/oscar1.obj
# vertices 1963
# elements 2645
opening device /dev/ttyd1
```

Enter scaling factor (units/mm) (current = 0.070000)

1.

Figure 2: Stereo-in-a-window application



File Stereo Mouse Motion Metaphor Tilt
File Stereo Mouse Motion Metaphor Tilt

Figure 3: 3D-stereoscopic visualization of logistics database

Figure 3 is an interface to a military logisticians database that is implemented using the Stereographics hardware mode. The interface is configured for a 3D display that uses Stereographics hardware, so all interaction with the interface occurs in the top half the screen. The menu bar in the lower half of the screen was created so it would be visible in both eyes. Because all the interaction occurs in the top half of the screen, the pull down menus from these menu bars only appear there, and therefore only in one eye. This leads to some discomfort which is probably inevitable when using this mode.

The graphic window in this figure uses geometric objects to represent objects in the data base. Supply points are cylinders, and are connected to other supply points by transportation routes, represented as rectangular roads. Units are represented as rectangular blocks. These objects are positioned on a 2D grid corresponding to actual map positions. Height is used to represent the rate at which material is being consumed. Using this interface, trained logisticians can examine a supply database and identify planning errors, such as units that have not been assigned supply points or areas where consumption exceeds supply, and make corrections. They can also navigate through the database, and examine it at different scales and from different perspectives.

2.3 User Interface Devices

One of the advantages of X windows is its ability to utilize a variety of I/O devices. Most of these I/O devices assume a 2D computer display and a 2D interaction with things displayed on that screen. Standard keyboards are based on typewriters that print to a sheet of paper. Standard computer mice move a pointer around a 2D electronic desktop. Interactions with a 3D-stereoscopic world require 3D input devices. These input devices can be used to manipulate the 3D synthetic world, and return 3D information about the real world.

Consider the problem of generating correct stereo images for a viewer. As we describe above, the correct images depend on the location of the viewer. A head tracking device that returns the position and orientation of the viewer's eyes permits dynamic updating of the parameters for the binocular viewing system. This ability permits the viewer to see the geometrically correct stereo images at all times. As the user's head moves, the 3D-stereoscopic image is updated for realistic and natural viewing of the objects in the synthetic world. Information about specific head tracker devices is encapsulated in the stereo 3D widget, allowing applications to generate the correct stereo images automatically.

Interacting with 3D synthetic worlds is an area of much research. To support development of many alternatives that can be fairly evaluated, we avoid dictating what the best methods of interaction are. Instead, we provide hooks for 3D input devices that the application can access. One input device we have been experimenting with ourselves is a six-degree of freedom mouse. Conventional computer mice are limited to specifying a position two dimensions. The six-degree of freedom mouse allows for specifying both a position and an orientation in a three dimensional space. We have used this device to control the position and orientation of the eye-screen system to navigate through the synthetic world.

3. CONCLUSION

We have discussed two user-interface components (widgets) that we have designed and implemented in an X/Motif environment using two different 3D-stereoscopic display techniques. We have discussed several generic problems, such as rendering, cursors, pointing devices, and head tracking, and discussed our solutions to them. As displays, particularly flat panel displays, become less expensive and more capable, software to drive them in 3D-mode and protocols for applications in 3D will have to be further developed. These are the first steps.

4. ACKNOWLEDGMENTS

We acknowledge the contributions of the members of our group for their insights into our work: Tom Ault, Victor Grinberg, Alan Guisewite, Priyan Gunatilake, Angel Jordon, Jeff McVeigh, Gregg Podnar, and Sriram Sethuraman. We also acknowledge the members of the Sage group for their synergetic effort with the logistics application: Mei C. Chuah, Joe Mattis, and Steve Roth.

This work is supported by the Advanced Research Projects Agency under ARPA Grant No. MDA 972-92-J-1010.

5. REFERENCES

1. J. S. McVeigh, V. S. Grinberg, and M. W. Siegel, "Double Buffering Technique for Binocular Imaging in a Window", Stereoscopic Displays and Applications VI, *Proceedings of the 1995 IS&T/SPIE Conference on ElectronicImaging: Science & Technology*, San Jose, California, USA, SPIE, 5-10 Feb.,1994, in press.

2. S. Zhai, W. Buxton, and P. Milgram, "The 'Silk Cursor': Investigating Transparency for 3D Target Acquisition", *CHI '94 Proceedings*, pp. 459-464+483, April, 1994

3. S. D. Kahn, G. A. Chappell, A. Smellie, M. Shantz, and S. Teig, "Knurls: Effective 3D Intra-Molecular Manipulation with a 2D Device", *CHI '94 Proceedings*, pp. 291-292, April, 1994

4. J. S. McVeigh, M. W. Siegel, and A. G. Jordon, "Technique for Synthesizing Intermediate Views from a Stereoscopic Image Pair", *Signal Processing: Image Communication*, submitted

5. V. S. Grinberg, G. W. Podnar, M. W. Siegel, ``Geometry of BinocularImaging'', Stereoscopic Displays and Applications V, *Proceedings of the 1994 IS&T/SPIE Conference on Electronic Imaging: Science & Technology*, San Jose,California, USA, SPIE, 8-10 Feb., 1994, pp. 56-65.

6. V. S. Grinberg, G. W. Podnar, M. W. Siegel, ``Geometry of Binocular Imaging II: The Augmented Eye'', Stereoscopic Displays and Applications VI, *Proceedings of the 1995 IS&T/SPIE Conference on ElectronicImaging: Science & Technology*, San Jose, California, USA, SPIE, 5-10 Feb.,1994, in press.

Reference: S. A. Safier and M. W. Siegel, ``3D-Stereoscopic X Windows'', Stereoscopic Displays and Applications VI, *Proceedings of the 1995 IS&T/SPIE Conference on ElectronicImaging: Science & Technology*, San Jose, California, USA, SPIE, 5-10 Feb.,1994, in press.