Multiple viewers in projection-based multi-screen immersive environments

by

Kristopher James Blom

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Major Professor: Carolina Cruz-Neira

Iowa State University

Ames, Iowa

2001

Copyright © Kristopher James Blom, 2001. All rights reserved.

Graduate College

Iowa State University

This is to certify that the Master's thesis of

Kristopher James Blom

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

The assistance of a number of individuals and corporations helped to make this research successful. The foremost is Gary Lindahl. His knowledge of the computer systems and VR hardware proved invaluable. Additionally his experience with attempts to make a multiple viewer system in the past provided insights into how to achieve the level of results we achieved. Mc Naughton Inc. must also be thanked for their assistance with the NuVision 60GX glasses and emitters. Without their assistance the complexity of this research would have risen to a much higher level of difficulty. Thanks also goes to Maxim IC which provided a number of chips for this research, and to the man who pointed me in the direction of the company, Brian Nowak. Additional thanks go to the people to lent technical expertise during the research, Dr. Julie Dickerson, Dr. Bill Black, and Sarah Kabala. A special thanks goes to Dr. John Mayfield from the Bioinformatics & Computational Biology Program at Iowa State University for his support which made it possible for me to complete this research.

ABSTRACT

Projection-based immersive virtual environments are very powerful tools to work on engineering and scientific problems involving teams of researchers. These environments enable the development of virtual representations of the problem domain in which multiple users can simultaneously share a virtual experience, providing a unique infrastructure for team work. However, current systems support only a single viewpoint, usually corresponding to a user wearing a head tracking device.

This thesis presents a system design which enables multiple simultaneous viewers in a multi-screen immersive environment. This system is designed to be flexible and easily scalable, preparing for future expansion to additional users. The design extends the time-multiplexed stereo viewing model commonly used to extend to multiple users. The research also addresses tools for creating the new multiple viewer virtual environments and extending existing applications to multiple viewer environments. An implementation of the system is presented in detail. This system enables two users viewing stereoscopic images or four users viewing monoscopic images to simultaneously view the correct images projected on the screen from their tracked viewpoint is presented.

1. INTRODUCTION

Projection-based virtual reality (VR) is a VR paradigm where the user is immersed in a computer generated environment projected onto one or more display surfaces. Multiple users can view the projected environment simultaneously lending teamwork capabilities to the environment. The images are displayed either in monoscopic or stereoscopic format. In the case of stereoscopic images, users wear a pair of filtering glasses to separate the images so that each eye only sees its corresponding image. Examples of projection based VR systems are: The Responsive Workbench [KRUE 95], a display in the format of a table top; The Powerwall [POWE 00], a large format tiled screen with multiple projectors producing a high-resolution image; The CAVE [CRUZ 93], a 4 projection surface cube structure; The C6 [C6 00], a six projection surface cube structure.

These projection systems allow multiple users to view a virtual environment simultaneously; however, the perspective view is computed correctly for only one user. This is because only one user is typically tracked. In this context the term "tracked" indicates that through sensors the position and orientation of the user is determined in three-dimensional space. The computer then generates correct perspective projections from the tracked user's viewpoint. Typically a stereoscopic pair of images is displayed, providing the user with an experienced world that appears to have depth. In addition to providing a stereoscopic display, projection-based environments allow multiple people to interactively share the experience. With the perspective being calculated for a single user, the remainder of the users are not receiving correct images due to the physical disparity of the user's position with that of the tracked user. This distorted view of the world can diminish the effectiveness of the viewer's experience ranging from disengagement for the environment to cyber-sickness [BURD 94]. This thesis presents research conducted to allow multiple people to be active participants within the virtual environment, where an active participant is defined as a viewer who has images calculated based on their position and orientation and have the computer reacting to their movement, creating an interactive environment.

Providing a method of delivering perspectively correct images to the users, enables us to address a number of situations in which having multiple active viewers could be beneficial. In the next paragraphs a number of these situations are outlined as motivation for this research.

1.1 Motivation

The ability to display virtual worlds to multiple people has created collaborative environments in which multiple people can view the same environment within the same physical space, referred to as **local collaboration**. These environments can be used to get people of diverse expertise's together to jointly explore the virtual experience and share their reactions to the environment. Examples include marketing personnel and engineers, or engineers and vehicle designers. In many of these situations the experts need information pertinent only to their particular field. In the case of vehicle design, the engineers want to see the structural design, while the designer is concerned with the exterior "look and feel." Under current systems with a single active participant, either all information is displayed at one time or the pieces of information have to be sequenced and all users see them even though they may not be interested in that piece. If all the information is displayed, the level of complexity of the environment increases drastically creating a crowded and convoluted environment. Switching between information levels solves the complexity issue, but does not allow the experts to see the information they need while the other expert is discussing it. It is clear that for local collaboration there is a strong demand for a method that allows more than one active participant. Allowing each user to see their own view of the virtual environment enables each user to see the additional information pertinent to them, while retaining a view of the common parts. In this way they could have a shared experience with pieces of the environment that are pertinent to all parties. [Oliver 97]

Remote collaborative environments, where two VR facilities in different locations share a virtual environment, remain in the realm of desired environments and are areas of active research. The design of such remote collaborative environments is hindered by the very nature of the environment being shared by two facilities potentially thousands of miles apart where no one person can see all parts of the collaboration. Furthermore, testing and debugging such applications can be very difficult. Support for the design of remote collaborative environments is a strong motivation for the development of our multiplexed system. With advances in technology, collaborative environments are now beginning to move to "main stream" interest. Research still needs to be done on effective communication and interaction paradigms for remotely shared virtual worlds. This research is not only technically challenging to implement, but it is also very challenging to test and debug. The challenge in testing and debugging is do to the very reason collaborative environments are desired, sharing a virtual environment in two physically separate environments. The distance between the sites and the lack of good communication among the remote participants makes it difficult to provide feedback at early design stages. With a system that supports multiple simultaneous viewpoints the testing and debugging issues related to remote collaboration can be simplified. We can simulate a collaborative experience in the same physical space allowing face-to-face discussion about how the information is presented to the different sites, how the interaction and data control methods work, and many other aspects of the design and usability of the application.

The projection based systems being addressed here are quite expensive to build and maintain. They are also rarely single user systems. These devices are generally needed by multiple people, often all simultaneously. While simulators provide users a base level for design, design in the actual environment is often required. Viewings by sponsors, potential funding groups, and public demonstrations often occupy a considerable portion of the time availability. Getting the maximum amount of usage out of such devices is often a priority. Projects are often delayed when the device needed is occupied in other uses. Pushing our multi-user technology to its limit it can provide a unique solution to increasing the availability of the systems. By allowing two stereo viewpoints to be viewed, the time available on the system is effectively doubled. In this way two developers could potentially work on their projects simultaneously, or allow developers to continue their work concurrently with demonstrations. As this multi-user technology scales upward the number of people who can use the VR system simulantiously increases, allowing each user to view their own virtual environment without disrupting any other user's environment.

1.2 Scope of Work

The work presented here addresses the development of a method which allows multiple users to be active participants in a projection-based virtual reality platform. A design approach for multiple user system is given and details of an implementation of the design for a two user stereoscopic environment are presented. This design accommodates multiple synchronized projections and provides the flexibility to support both stereoscopic and monoscopic viewpoints. Software integration is taken into consideration in the design process, allowing previously developed software to be used without modification.

The research presented here has been developed following these stages:

- Investigate the requirements of a time-multiplexed VR system.
- Define possible design approaches for achieving multiple active viewers and analyze their feasibility.
- Specify a design approach based on the feasibility analysis.
- Identify previous work in this area.
- Implement and test our design.
- Discussion of the results and identification of future research.

In the following chapter, background material on projection-based VR and the software libraries VR Juggler are presented. In sections 3.1 and 3.2, we investigate design approaches for achieving multiple active viewers in a VR system and analyze the feasibility of the possible designs. In section 3.3, previous work on multiple viewer systems is identified. We then proceed with a detailed explanation of our design in section 3.4. Chapters 4 and 5 describe the implementation of our design for two stereoscopic users or four monoscopic users, followed by the results of our implementation in chapter 6. In the final chapter, chapter 7, we identify future research to be conducted both in the area of design and implementation of a multiple user system and for research using such systems.

2. BACKGROUND

This research sits upon and depends on two technologies, projection-based VR system and the VR Juggler software libraries. This chapter presents information on these technologies since they are the underlying technology infrastructure for the research described.

2.1 Projection Based Virtual Reality

The two most popular technologies for virtual reality systems are head mounted displays (HMDs) and projection based systems. As the name implies, in a head mounted display system, displays are in some fashion fixed to the participant's head. These displays are typically in the form of small LCD or CRT displays mounted on a helmet. The displays are opaque allowing the participant to see only the virtual world, therefore allowing only visual cues from the virtual world increasing it's immersive effect. In projection based VR systems, the virtual world is presented to the user with projectors and display surfaces. In contrast to HMDs, the user is able to see both physical and virtual realities simultaneously. This allows the form of collaboration called **local collaboration**, described in the introduction. Since projection based virtual reality is the basis for this research, design specifics of projection based VR will be presented.

Projection based virtual reality systems use projected images to present the virtual environment to the user in the straightforward manner by projecting an image(s) onto a surface(s) to be viewed. A typical system is composed of a number of components: projector, display surface, computer, glasses, tracker, and input devices. Projectors are the method used to deliver the images to the user. The projectors used in VR systems are typically large projectors with separate red, green, and blue tubes. The display surface is typically a semi-translucent screen for rear projection. The screens can be either made of hard or flexible materials. The computer is responsible for the generation of the images for the projectors, which entails input from devices to create the interactive environment, any calculations to create the environment, and the creation of the image information. The image generation hardware can be considered separately even though it is intrinsically tied to the rest of the computer. The image generation hardware must be capable of generating signals of acceptable resolution at refresh rates higher than 60 Hz. Typically rates of 96 Hz to 120 Hz are used in order to circumvent visible flicker in the images. Since the user is typically presented with stereoscopic images, the hardware usually supports quad-buffering, which gives double buffering for each eye's image generation. The glasses are a system used to distribute the two images displayed to the correct eye of the user. The methods these glasses use are discussed further in chapter 3. A tracker is used in order to create the interactive environment for the user, by conveying the position and orientation of the user in three dimensions to the computer. The most commonly used tracking system in projection based VR systems in electro-magnetic tracking. Small magnetic coils are attached to "tracked" objects, i.e. the user and often a input device, with a wire connected to circuitry which can then detect the magnetic field sent out by a companion device placed in the environment to produce the magnetic field. Other technologies for tracking include vision based systems, ultrasonic based systems, and systems using reflected light. Input devices in VR systems allow users to move beyond the physical limitations of the system, i.e. how far they can walk. While numerous devices can be used, with only the basic requirement of having a method in which the computer gets information from them, the most common device is commonly referred to as a "wand." A wand is simply a device which is held in one hand, usually having digital and/or analog inputs, tracked so that the computer knows it's position and orientation, and originally had a shape reminiscent of one's fairy godmother's wand. Wands are now often made from remote controls, flight sticks, or computer mice.

Having described projection based systems in general, we now look at some variations. The projection based systems come in a multitude of configurations: a single display surface with a single projector, a single display surface with multiple projectors, or multiple display surfaces with multiple projectors. At this point it is worth noting our careful use of the term display surface. In projection based systems the display surface could be one of a variety of shapes and materials. The surface could be any wall which is projected onto or as noted previously a semi-translucent screen, either flexible or rigid. The shape of the surface could be flat, curved, hemispherical or any feasible shape. In the case of multiple surfaces they may be aligned at arbitrary angles to each other. Some configurations will be discussed in the coming paragraphs.

Early projection based systems had the projectors placed in front of a white opaque surface. An alternative to this arrangement was devised to remove occlusion of the projection by the viewer. This leads to the rear projection systems typically used, although front projection is still used in some systems where the viewer(s) can not occlude the images, such as in an IMAX theatre. Rear projection is accomplished by using semi-transparent materials between the user and the projector. These semi-transparent surfaces pass portions of the light to the user and provide surfaces for the calculations of the projection matrices for the graphics. Two classic examples of rear projection based VR systems are FakeSpace's Immersive Workbench™ and the CAVE™. The Immersive Workbench™ is a flat table top surface onto which a virtual environment is projected [KRUE 95]. The CAVE™ surround-screen system [CRUZ 92]positions users in a 10'x10'x10' cube where the front wall, side walls and floor are projected onto. The majority of today's projection based VR systems use time-multiplexed/shutter glasses for presenting images and then their respective decoding to the viewers. In early shutter glass designs the glasses were tethered, connected by a wire or multiple wires to a computer and to a power source. The need for the tethering was primarily due to the power consumption of the early LCD technologies. The power consumption of LCD technologies has decreased since, enabling the tether to be removed. The movement to untethered glasses requires a wireless method of synchronizing the glasses with projected stereo images. This is done with Infra-Red (IR) emitters and IR receivers housed within the glasses. A number of emitters are placed in the physical environment, such that the space occupied by the viewers is flooded with the signal. Even flooding the environment cannot remove all occlusion possibilities, such as the user of the glasses placing their hand in front of the IR detector in the glasses. This IR system intrinsically provides a mechanism for allowing multiple viewers in the environment, since all glasses in the space will pick up the IR signal removing the need for additional communication paths with the new glasses. All users with glasses then get to view the stereo environment.

While we have established how the images forming the environment are presented to the viewers, projection onto a surface, and briefly how the images are decoded. A major component for this environment to become virtual reality is still missing. As we have defined virtual reality we are still missing interactivity with the computer. In most of these systems this comes in two forms. The first and most important is the tracking of the user. Tracking in projection based VR enables the computer knows the location of a user. Using the user's location the software then performs calculations for the projected images so that they reflect the perspectively correct environment. The tracking is a continuous update, therefore allowing the computer to change the virtual environment to match the movement of the user. The second is interaction with the environment with an input device. Usually an input device provides the ability to interact with the environment and the ability to move throughout the environment beyond the bounds placed on their physical movement.

2.2 VRJuggler-Virtual Reality Libraries

In the previous section the hardware components of a projection-based virtual reality system were presented. The software that creates the virtual environment for the user must be built to integrate this complex set of components together. The software must use the data from the tracking system to determine the location of the user with respect to each of the projections and then calculate the perspectively correct images for each surface. The software must create the correct images for each eye when showing a stereo pair. This must be done for each of the projections in the system and be placed in the correct portion of the frame buffer. In our previous statements the creation of the imagery was assumed trivial, although, in reality it is a major endeavor in itself.

With these tasks in mind the creation of a virtual environment is a daunting task, even without considering what makes up the environment or how to provide the interactivity with the environment. For this reason, software development tools are needed. A number of VR software libraries exist today [BIER 98], where each library provides differing amounts of help for the programmer and differing areas in which they provide assistance. The desirable areas that one would want the software libraries to address are: performance of the libraries, cross-platform support, support for common VR hardware, support for distributed environments, rapid prototyping capabilities, interfaces to high and low level graphical interfaces, and extensibility [BIER 98]. Of the possible VR resources available, few meet this list of desirable attributes.

The VR Juggler libraries [JUGG 00] are an Open-Source platform for the creation of virtual environments created at the Virtual Reality Application Center at Iowa State University. Since this software is being created in house, it is freely distributed, provides needed support of multiple user environments, and support is close at hand, we chose VR Juggler for this research. The VR Juggler platform aims to ease the creation of the environment, present a flexible framework for the environment, and provide good performance.

VR Juggler is a object-orientated platform created using advanced software engineering principles that addresses most of the desirable VR software features mentioned previously. Either the SGI Performer libraries or lower level OpenGL libraries are used for the graphics, providing an interface for advanced low level programmers or for easier programming. By forming an abstraction layer over the computer hardware, the libraries create a good platform-independent development environment. This hardware abstraction layer is also used to abstract away from the specifics of the VR hardware being used, such as input devices, letting the programmer worry about the interaction methods instead of implementation details of the hardware being used. The object-orientated construction of the software allows VR Juggler to provide an extensible platform for new paradigms and hardware. In the coming paragraphs we give a slightly more in-depth look at what the VR Juggler libraries provide [BIER 01].

From a programmer's perspective, the VR Juggler libraries provide a relatively high level abstraction for coding virtual reality environments. These environments can be created with minimal VR system dependence allowing the programmer to concentrate on the environment being created. VR Juggler is built around an object orientated environment. As such, an application follows an object structure. The VR Juggler application has a central kernel which controls execution. The kernel controls various managers that perform specific functions or tasks. The functions and tasks include: the input manager (handles the input devices and tracker data), the display manager, the draw manager, and the network manager. The application is

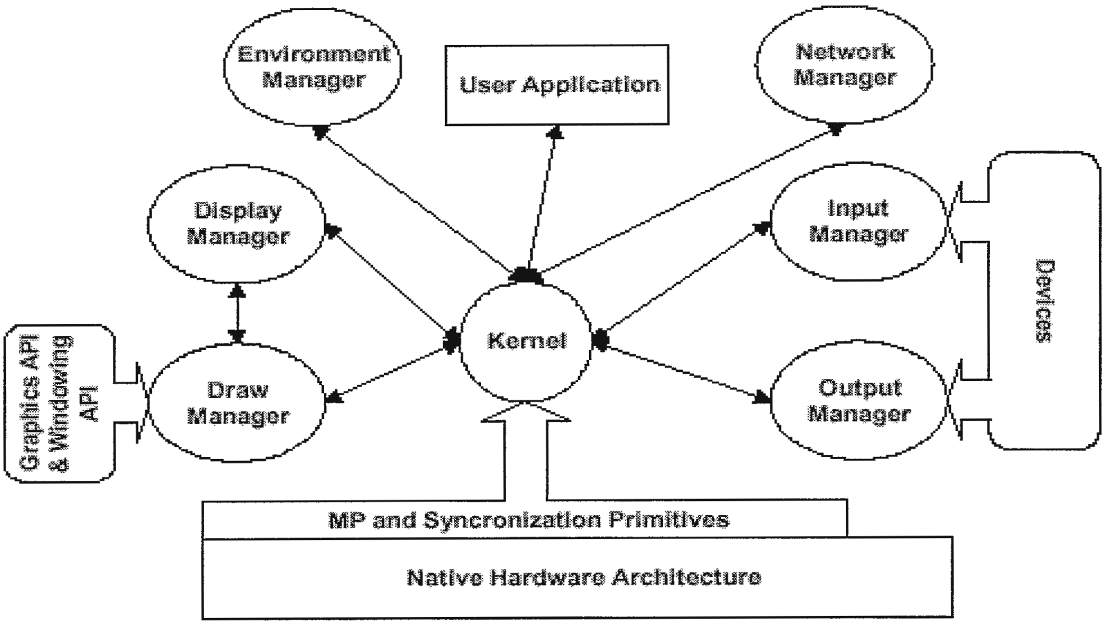


Figure 2.1 VR Juggler Object Relationships [JUST 98]

another object controlled by the kernel. *Figure 2.1* illustrates the relationship of the objects in VR Juggler.

The programmer first creates a kernel for the application by an instantiation of **VJKernel**. The new application is then derived from the **VJApp** class and registered with the instantiation of **VJKernel**. Although this is a trivialization of the programming, it describes the initial part of creating a VR Juggler program. In the inherited class from **VJApp,** many of the components that constitute a virtual reality program are specified. The basic control loop for an application is shown in *Figure 2.2*, where *app* is the name of the derived class. The programmer simply overwrites the functions in **VRApp** and registers a draw function with the kernel. In an OpenGL based program, an additional *contextinit* function is used for establishing OpenGL contexts for each of the displays when multiple displays are used.

The hardware abstraction VR Juggler provides is very important for this research. As has been stated previously, this freedom from the hardware allows the programmer to focus on the development of the environment. A VR Juggler program can be written on a computer running a provided simulator, in a projection-based environment, or on a head-mounted display. In addition, the program then becomes more extensible as new VR systems become available.

```
app.init();
app.apiInit();
 while (drawing) {
    app.preDraw();
    draw();              // Draw the graphics
    app.postDraw();
    sync();              // Synchronize the Juggler system
    app.postSync();
    UpdateTrackers();
 }
app.exit();
```

Figure 2.2 VR Juggler Control Loop [Bier 01]

The simulator environment provides a convenient way for the programmer to develop the application without having to acquire resource time for the actual VR devices. The developer can almost completely create their environments within the simulator environment.

With VR Juggler developers have a platform which allows them to rapidly create virtual environments without having to deal with many of the specifics created in using the multiple viewer system. One time configuration files created can be used for any VR Juggler program. Any legacy program in VR Juggler can be immediately used in multi-viewer mode by simply loading the configuration files created for the system. In all VR Juggler creates the ability for the multi-user system to be smoothly integrated into the working environment.

3. MULTIPLEXED VIEWER CONCEPT

This chapter discusses the design of a multiple viewer system. In the first section, we present the current methods available for presenting stereo images for a single viewpoint. The second section addresses the scalability of these technologies to match the needs of a multiple viewer system. Previous and ongoing research in this area is discussed in the third section. The final section is a discussion of the design chosen for this research.

3.1 Stereoscopic Viewing Methodologies

This section reviews stereoscopic viewing technologies for projection systems. For stereoscopic viewing, two images are shown to the user, one intended for the left eye and the other for the right eye. In all methods the goal is to separate the two images without the viewer's eye seeing the other eye's image. There are three popular approaches to deliver stereoscopic images: polarization, anaglyphs, and time-multiplexed/shutter glasses systems.

3.1.1 Polarization

Polarization is a technique in which the light from the projector is polarized, and polarlized lenses are then used to filter out the images appropriate for each eye. Polarized light is aligned such that it oscillates in a particular manner. Polarization techniques have two components. The first is the polarizer, which aligns the light or polarizes the light. The second is a medium which passes only light polarized in a particular manner. The polarizer is an optical lens placed at the projector. The second medium is typically referred to as a lens and is used directly by the user over their eyes in the form of glasses.

Two basic polarization methods are used, linear and circular polarizations, see *Figure 3.1*. Linear polarization occurs when all of the light oscillates in a single direction. If light from two sources are linearly polarized 90 degrees with respect to each other then a filter can discriminate between them. In the VR field these are usually denoted as horizontal and vertical

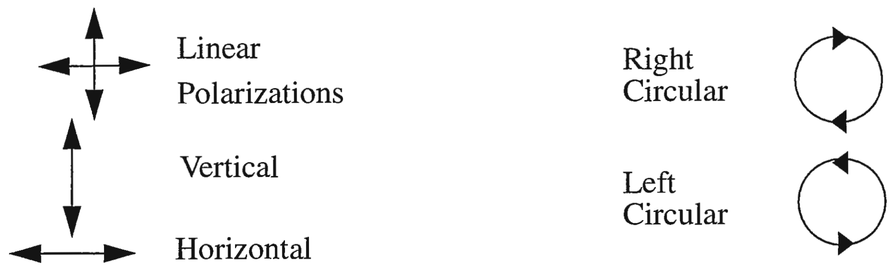


Figure 3.1 Polarization of Light

polarizations, and are typically implemented in the manner the names suggest. Circular polarization is slightly different; the light does not oscillate in a single direction, but rather the oscillation rotates around all 360 degrees. The polarization is denoted by the rotational direction, giving right and left polarizations. To make it easier to conceptualize circular polarization it is helpful to liken it to a clock. Over time the hand is aligned in all 360 degrees, and has a rotational direction associated with it, clockwise versus counter clockwise. In a similar manner light is polarized circularly, left and right. As with the linear case, filters can be used in order to pass one of the directions of circular polarization without passing the opposite circular polarization [PEDR 93].

There are a number of advantages to polarization methods along with limitations. Polarization of light can be done by placing special optics in front of the projector. For the size of the projector output these are rather large optics, but can still be produced. Polarization systems have the advantage in being “passive-glasses” systems [MCAL 93]. There is only a stationary filter on the user’s glasses, producing glasses which are lightweight and inexpensive. Polarization techniques require that the optical path before the filters, which separate the images, be free of any mediums which may alter the polarization characteristics. In many cases projection based VR systems use mirrors to fold the images to conserve space [CRUZ 93]. These mirrors are often large, often not actually made of silvered glass, and may not preserve concise optical paths. The surface onto which the image is projected is also a concern for the same reason as the mirrors, a possible distortion of the optical path. This effect can be readily seen, particularly with linear polarization techniques. As an example, in the context of using polarization for de-multiplexing images we are placing the filters on a users head. For linear polarization to work correctly requires the alignment of the filters to be precise or else the filters will allow both images to pass. Recall that the second filters are the lenses in the user's glasses. Only when the user's head is aligned properly will the other eye's image be blocked out. If mis-alignment is present it creates ghosting, seeing the other eye's image in addition to the correct image, or in the extreme case the user reverses the stereo pair by turning their head to the side making what was the horizontal filter become a vertical filter and vice-versa. Circular polarization is intrinsically more tolerant to the users movement, although distortion in the optical path will create similar effects.

3.1.2 Anaglyph

A second passive-glasses system is the anaglyph method. This method uses complimentary color encoding, red blue filters [MCAL 93]. Because of the simplicity of anaglyphs they have become a popular gimmick, even found on the back of cereal boxes and have been used in movie formats. This method has the major advantage of cost; No additional pieces are needed for the projector and the glasses for this method cost only a few cents. Anaglyphs, however, are incapable of displaying multi-color images, and therefore anaglyphs are not considered in the remainder of this thesis.

3.1.3 Time-multiplexed / Shutter Glasses

While the two methods previously described place both images on the projection surface simultaneously, the third method uses a different paradigm, field-sequential or time multiplexed images. In this method images are placed on the screen in an alternating fashion and electro-optical means are used to separate the images. The images must be shown at a minimum rate of 30 frames per second per eye, yielding a rate of 60 Hz for images for both eyes. The reason for this frame rate necessity is that a human can perceive the image updates if the rate is slower. This slow rate is rarely used, however, as some viewers will be able to pick up on the image refresh, seeing what is commonly referred to as flicker. Generating images for time-multiplexed stereo is a challenge because of the refresh rates required for flicker free viewing, a challenge for both the image generation hardware and for the projection systems. In the original CAVE, at a rate of 120 Hz, the original projectors used a green phosphor, which had too long a persistence and created ghosting effects. High end projectors have since been produced with special green tubes, which have a faster green phosphor decay, but they are quite costly [Cruz 93]. The de-multiplexing system consists of glasses comprised of Liquid Crystal Display (LCD) lenses, which can be made opaque or translucent by applying an electrical field to the crystalline structure. The glasses are therefore heavier and significantly more expensive than the glasses for the two passive-glasses systems. The glasses also require a method of synchronization with the projections. This is either accomplished with a tether, a wire run to the glasses, or by Infra-Red(IR) communications. IR is the common method in VR environment for obvious reasons, although they do create the potential for occlusion from the IR signal causing the glasses to fail. The advantage of this system is that, with the exception of the IR occlusion problem, they work well, allowing little ghosting, while the user is free to view the projections from any angle or even while doing a headstand.

3.2 Feasibility Analysis of Techniques

Increasing the number of views also proportionally increases the number of images that must be displayed; therefore, an analysis of the scalability of the current stereo methods to our multi-user paradigm is given. The passive-glasses schemes are not scalable beyond the current two independent views (stereoscopic pairs). In the case of polarization, there are only two orthogonal polarizations, left and right circular, and vertical and horizontal linear polarization. Linear and circular polarizations are not mutually independent meaning that one cannot be applied on top of the other, therefore, combining the two types of polarizations can not be used to scale to higher numbers of images that can be discriminated between. In the case of the time-multiplexed shuttered glasses the scalability of the method seems possible. The graphics hardware and projection technologies are the limiting factor to the scalability at some level. The glasses are scalable, but they are limited by the speed at which the LCDs can switch without showing bleeding. In the case of the passive glasses systems the images are placed simultaneously on the screen, however, in are projected to the screen in a time multiplexed manner. Alignment of a scaling number of projectors would have to be done for the passive glasses systems if they were used. For the time-multiplexed method to scale to multiple users the frequency of refresh for the images will have to increase proportionately, linearly to the number of images being projected.

The concept of combining the time-multiplexed and polarization methods is appealing. However, it is not possible to combine them since the based technology used for the LCD lenses is polarization. In the workspace provided the user is encouraged to view the environment from every angle, where linear polarization is not sufficient for such viewing. Circular polarization suffers from difficulty to achieve the polarization effect on the multiple projectors per screen accurately enough to eliminate the bleed through problem. An additional consideration is the extra work of alignment of the projectors for the passive systems. Since multiple images and therefore projectors would be required simultaneously on each display surface, they must be aligned and calibrated together. Although the current ceiling on the scalability of the time-multiplexed method is expected to be low at current times, the possibility for the underlying technologies to improve in the future seems likely, and it was expected that the current technologies would be able to support the extension to two stereoscopic or four monoscopic viewers. In the end, the non-scalable nature of the combination polarization/time-multiplexed method led us to decide to proceed with the time-multiplexed/shutter glasses method.

3.3 Previous Work

At this time there is only one published research paper in this area, the Duo Viewer by FakeSpace Inc. [AGRA 97]. This research is being done under an Office of Naval Research (ONR) contract, topic number N97-125. Under this contract FakeSpace is to design and build a stereoscopic viewing system for their CRT based Immersive Workbench. The Immersive Workbench is a single screen rear projected system in the form of a table, where the tabletop is the projection surface. The DuoViewer used a method similar to the one which this research presents, time-multiplexed images using shutter glasses to view the images. Although the DuoViewer does show promise we feel a more extensible solution is feasible. The design of the DuoViewer was not scalable to multiple projections. Additionally the DuoViewer had significant image artifacts, namely image ghosting. A recent contract awarded by ONR has a first phase of improving the DuoViewer's imagery, indicating that the ghosting problems were significant enough to merit a need for it's improvement. Our research addresses issues left unresolved by the FakeSpace research: multiple projection surfaces and integration into programming environments. They developed their own hardware and specialized software. All applications to be run had to be rewritten using the specialized software, making it difficult to incorporate the technology into existing applications.

A second organization, the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago (UIC), has also done some research into this area. The research done at EVL is a system for use in their CAVE™. The approach taken at EVL is similar to that of FakeSpaces' research, time-multiplexed. Since the development is for a CAVE™, the EVL research additionally addresses the multiple projection surfaces extension. Custom hardware and custom glasses have been developed for the project. Due to the difficulty of creating custom glasses hardware, EVL's system has suffered similarly to the FakeSpace implementation in the area of ghosting, creating artifacts from other users' images. The EVL solution is for two users seeing stereoscopic images, but does not address scaling to additional users or allow flexibility in allowing monoscopic viewing.[LIND 00]

3.4 Design Overview

As discussed in the background material, the single user stereoscopic environment uses two images time-multiplexed together, each image correlating to the users left or right eye. In order to have more than one user seeing stereoscopic images a left and right eye view for each user has to be calculated. Extending the time-multiplexed concept for a multiple user system, all of these images need to be interlaced, which in the context of this thesis reflects time-multiplexing in some fashion.

There are two basic interlacing patterns to accommodate more than one user in a time-multiplexed manner. The first is to display both images for each user while the other users do not see the images, **interlace by user**. The second is to alternate users showing only one eye for the user at a time, **interlace by frame**. Since the scope of the implementation in this research requires only two users in stereo currently, the examples given to explain these conceptual ideas assume a two user stereoscopic or four user monoscopic setup. The patterns are extensible to higher numbers of users.

The first method we analyze is interlace by user, where both eyes are shown for a user before changing to another user. Assuming that the users are User A and User B the pattern is: User A right, User A left, User B right, User B left. This can be seen as the third line of *Figure 3.2.* The second method would be to switch between users every frame. This would yield a pattern as such: User A right, User B right, User A left, User B left. This format can be seen graphically as the fourth line of *Figure 3.2.* We will use the interlace by user method, the main reason being to simplify the software structure without drastically increasing the hardware complexity. The reason for this can be seen by inspecting the figure closely. In our interlace by frame example we have ignored that fact that User A does not have their left eye being displayed when it is required, nor does User B have their right eye open being displayed when it is needed. This can be fixed in software, but is a much more difficult solution than our simple hardware to handle interlace by user. Although a single design was settled upon, the design across all the components is flexible so that modifications can be made to alter the pattern if a new pattern is perceived to be a more viable design at a future time.

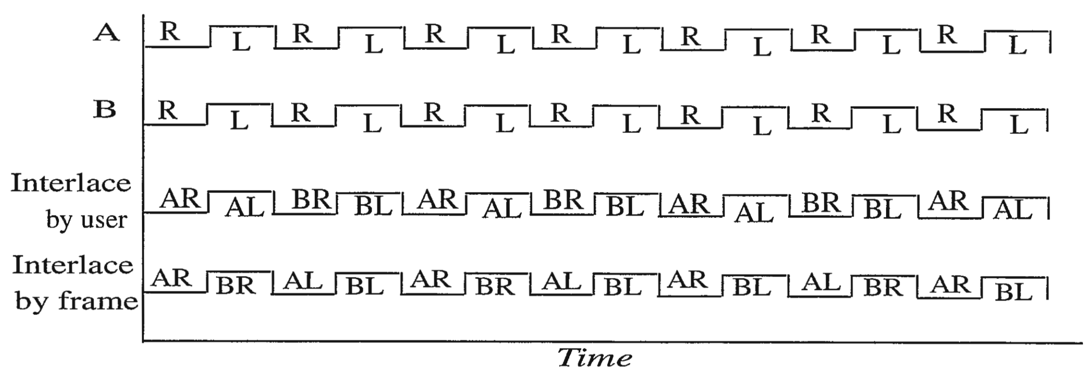


Figure 3.2: Video Multiplexing Pattern for Two Users, A & B

Having decided upon using the time-multiplexed/shutter glasses and interlace by user methods, a number of components are needed to interlace the images and decode the images. The first and most crucial component is hardware to do the actual interlacing of the images, which will be referred to as the **switcher**. The switcher interlaces two channels of images into one, following the interlace by user pattern shown in *Figure 3.2.* The switcher is placed between the image generation hardware and the projection system. The second component is the combination of the **glasses** and the **emitter**, which synchronize the shuttering of the glasses. The emitters and glasses used for this research are commodity parts, controlled by micro-controllers which can be reprogrammed to other functionalities. The flexibility inherent to the micro-controller allows the glasses and emitter hardware to remain unmodified in our design. By using these commodity parts and not modifying them on a hardware level some potential areas for fault in our design are removed and we insure that our peers will be able to reproduce our results. The third and final component is the software. The virtual reality software has to support a multi-user environment. The VR software must produce all the proper viewpoints for the multiple projections. The software must also be able to perform the extra interaction for multiple people.

Each of these three component pieces are described in the following two chapters. In the next chapter the hardware design will be described in detail: how the multiplexing is accomplished for a single projector, how multiple projectors can be supported with the hardware, and the hardware design for synchronizing the emitters to the interlacing scheme. In the fifth chapter the firmware and software designs will be described. This includes both the glasses and emitter firmware programming and also the VR Juggler software and configuration to allow for the multiple user environment.

4. HARDWARE DESIGN

The hardware design has three distinctive portions. The first and foremost is the actual multiplexing of the images for a single projection. Once the multiplexing for a single projection is addressed, the need for multiple projections to be multiplexed must be resolved. This second part is important because in a multi-surface or a multi-projector surface the individual projections need to be synchronized together. The third hardware piece addresses synchronization of the display hardware with the emitter and glasses. In the following sections each part is described separately. In the first section we discuss the multiplexing hardware. The multiplexing hardware then becomes the basis on which the other two hardware components rely, addressing the chaining for multiple projections in the second section, and finally the synchronization with the emitters in the third section.

4.1 Video Multiplexing

As was previously discussed, there are two potential methods for interlacing the video streams. The first is interlacing by user, which is then interlaced by eye as in the traditional stereo method. The second is to interlace the users by frame, that is switching the user each frame. *Figure 4.1* shows both these patterns. As discussed in the previous chapter, for reasons of simplifying the programming a hardware design using the first of the interlacing methods is desirable.

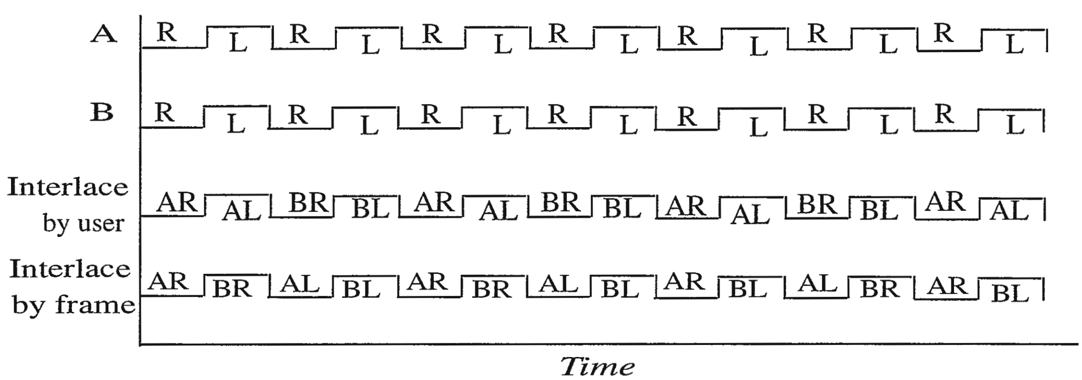


Figure 4.1: Video Multiplexing Pattern for Two Users, A & B

Recalling that the image generation hardware used in projection based virtual environments produces a synchronization (sync) signal which can be used to indicate the left and right eye of the user, *Figure 4.1* signal *A*. In *Figure 4.2,* the first line is this sync signal. Analyzing the sync signal generated by the computer and the desired pattern, it is noted that the user timing is simply a time doubled version of the original signal. This time doubled signal can be done with a standard time doubling circuitry using a D-type Flip-Flop. The connections for the chip are then made as shown in *Figure 4.3*. The truth table for the flip-flop is also included in the figure.

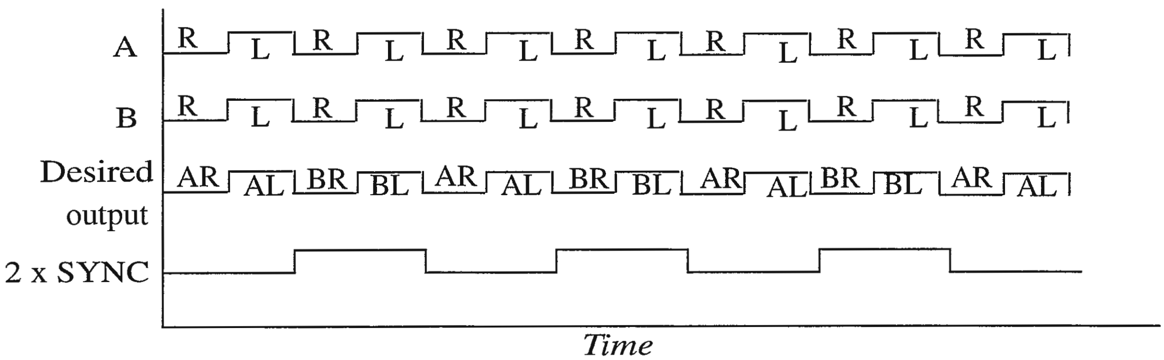
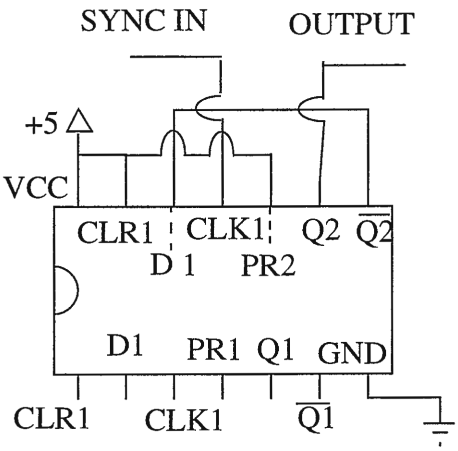


Figure 4.2: Signal for Video Switching

The time double signal can then be used for interlacing the images. For the actual interlacing a commercial chip designed for high speed switching of image signals is used. This chip has a single input for determining which of the two input signals to output, which can be connected directly to the output of the time doubling circuitry

4.2 Chaining for Multiple Projectors

Up to now a single multiplexed signal has been achieved from two signals, which drives a single projection. In order to expand this to multiple projectors, unfortunately, is not as simple as building another switcher for each of the projectors. If this simplistic approach is taken, guaranteeing that each of the multiplexing devices is in concert with the others is an issue. It is not possible to definitively guarantee that the cycle will start at the same point for any two of the switchers. Since this is difficult to conceptualize we have represented it graphically in *Figure 4.4*. This leads to the time doubled signals produced for switching to be inverted between



CLR	CLK	PR	$\overline{D}$	Q	$\overline{Q}$
H	↑	H	H	H	L
H	↑	H	L	L	H
H	L	H	X	Q_0	$\overline{Q}_0$

H = HIGH Logic Level   L = LOW Logic Level
X = Either LOW or HIGH Logic Level
↑ = Positive Going Transition
Q_0 = Logic level of Q prior to this state

Figure 4.3: Time Doubling Circuitry[FAIR 00]

the multiple multiplexing devices. Referring to *Figure 4.4*, the first line is the sync signal from the graphics generation hardware, labeled "Sync." The lines labeled "2 x Sync [P|Q]" are the time doubled signals of the two possible starting points. The problem is that the two signals are exactly 180 degrees out of phase. In other words when one switcher is showing User A the other will be showing User B. Due to the nature of our time doubling these two signals can be verified to be the only possible arrangements, as all others are simply periodic multiples of one of the two signals.

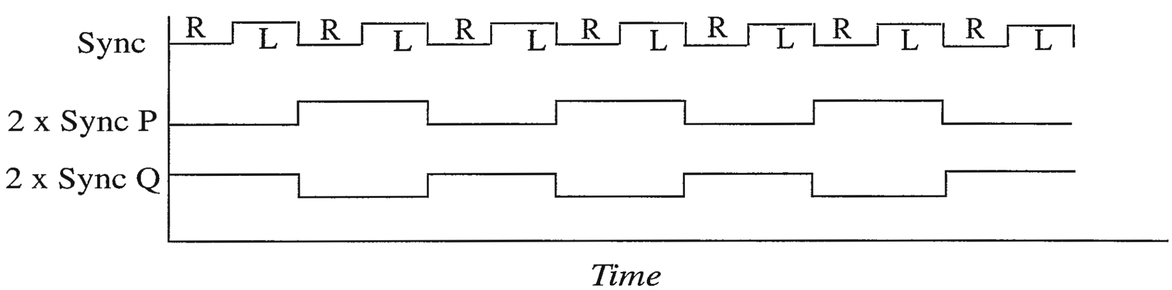


Figure 4.4: Video Multiplexing Pattern For Two Users

To correct this potential pitfall the signal must simply be synchronized. This is accomplished by having only one switcher control the multiplexing. This switcher is then the **master switcher**, giving signals to the **slave switchers**. A simple connection between them is all that is required and just chaining them together is quite appropriate, see *Figure 4.5*. This design has a possible pitfall, however, in that the flip-flop is not designed to drive multiple TTL sources. To counteract this the addition of a driver chip to allow chaining of multiple multiplexing slaves together is required. As the number of slave switchers, which correlates directly to the number of projections, gets higher, additional drivers may need to be added to continue the multiplexing signal to the higher number of slaves. This would act as a signal repeater and be in a slave switcher at some point down the chain.

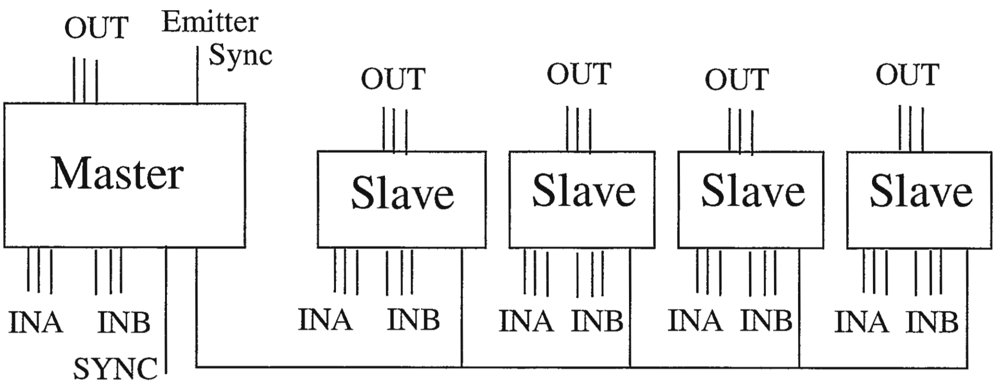


Figure 4.5: Switcher Chaining

4.3 Synchronization with Emitters

With the addition of the master and slave switchers to control synchronization between the multiplexers, the final hardware challenge arises. The original design goals were to design this system using off-the-shelf components as much as possible without having to modify the hardware of those components. In the case of the emitters a significant drawback was uncovered. These emitters take in a only one input signal. This signal is the sync signal produced by the graphics generation hardware under traditional single user stereo viewing. Two signals are identified as being needed for the emitter, the sync signal and a signal indicating which user's images are currently being displayed.

It is not possible to transmit two separate signals to the emitter's processor, since there is only a single input line to the emitter. The programmable processor in this emitter however makes the design task significantly easier. The flexibility of the processor can be exploited in this instance. Our basic design follows: an additional signal is added to the sync signal, which is detected by the emitter processor after it determines which eye to shutter. The designed

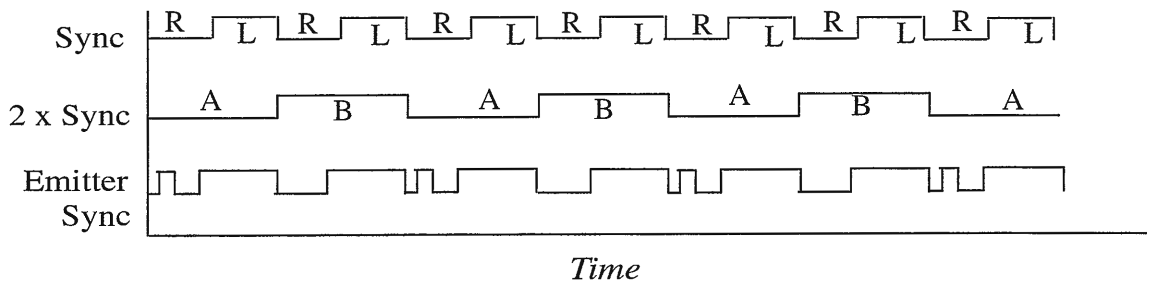


Figure 4.6: Building the Emitter Synchronization Signal

approach does allow for one viewer to receive the incorrect pair of images the first time that the stereo signal is activated, although this will not be perceivable to the users. After this 50% chance of error the pattern is fixed and will be correct until the stereo signal is removed. *Figure 4.6* shows the signal.

To generate this signal requires two steps. The first step is the generation of the additional signal, which specifies the user being displayed. To this end a one shot multi-vibrator can be used. The first trigger of the multi-vibrator is when the sync signal goes low and the time doubled signal goes low also. For this we set the delay to a large amount, 80 μ sec. This large delay is to guarantee that the emitter can process the previous information and be ready to check. After this delay the line is brought high. A delay of 40 μ sec is then produced to allow plenty of space for assuring that the emitter can synchronize. A more accurate depiction of this can be seen in *Figure 4.7*. In *Figure 4.6*, the extra signal was exaggerated, although even in *figure 4.7* the timing is not scaled properly for it would not be viewable to actual scale. This

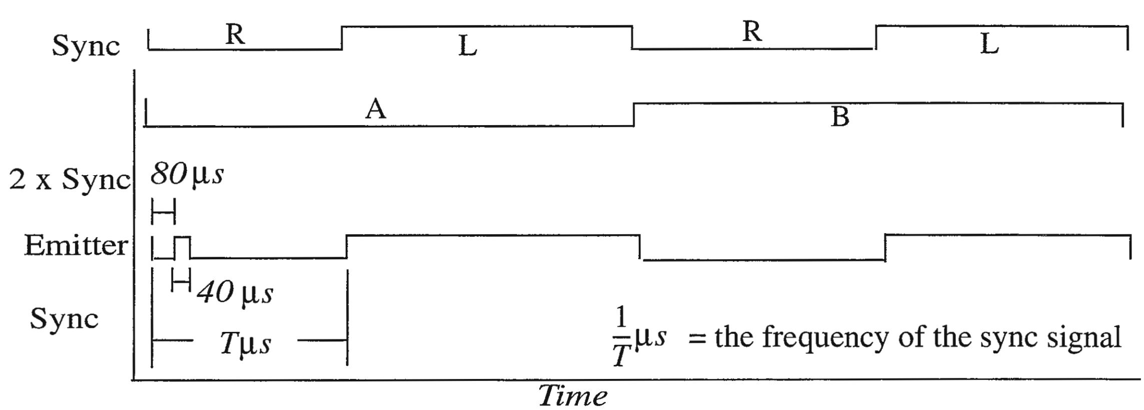


Figure 4.7: Emitter Synchronization Signals

can be done without worrying about the signals overlapping. The highest frequency at which projectors can be driven at this time is 180 Hz. With that the 120 μ sec until the line is brought low again is well within the limits set forward.

In this chapter the design of the hardware for the multiple viewer system was presented. First the method of multiplexing the images together was presented, followed by a method for chaining the switchers developed in the first portion in order to achieve a multiple display surface system. The final section describes the additional hardware necessary to synchronize the emitters and glasses with the switchers. In the coming chapter the focus moves from hardware to the firmware of the glasses and emitters and to the software for creation of the virtual environment.

5. FIRMWARE AND SOFTWARE DESIGN

The hardware design for the multiplexing of the video signals, as described in the previous chapter, forms the core of the multiple user system presented. The next challenge of this research is the firmware and software designs presented in this chapter. These designs are on either side of the switchers, on de-multiplexing the images and the other creating the images for input into the switchers. The firmware addresses the issue of de-multiplexing the images displayed in a manner consistent with the algorithm described in Section 3.2. Our software design must create the proper imagery in the proper places for the display system, handle creating correct images for each user, and support the extra interaction of having multiple users.

The algorithm for the de-multiplexing of the signals is an extension of the original projection based design. In the two user stereo case each user sees their respective left and right images in the correlating eye only. The four user monoscopic case requires each user seeing one frame out of four frames with both eyes. The de-multiplexing method requires two parts: 1) The glasses are modified to block out the images that are not to be viewed by the user, in synchronization with the multiplexing hardware. With the commercial glasses used in this project, synchronization can be achieved without modification to the circuitry of the glasses, but rather changes to the low level code, which drives the glasses and emitters. 2) The software is based on VR Juggler, which was described in the background Section 2.2. VR Juggler does most of the work necessary for the hardware arrangement chosen.

5.1 Firmware- The Glasses and Emitter

The multiplexed signal is created by the switcher design described in Section 4.3. This multiplexed signal must be de-multiplexed for viewing by the user. As has been noted before, the basic design extends the technology used in most projection based virtual reality, additionally constraining the design to use off-the-shelf parts where possible. The glasses and emitter combination used for this project are the NuVision 60GX glasses and the emitter pair [MCNA 98]. The glasses and emitters used were modified slightly to allow for easily reprogramming the systems. The flexibility inherent in the embedded processing units is enough to confine all

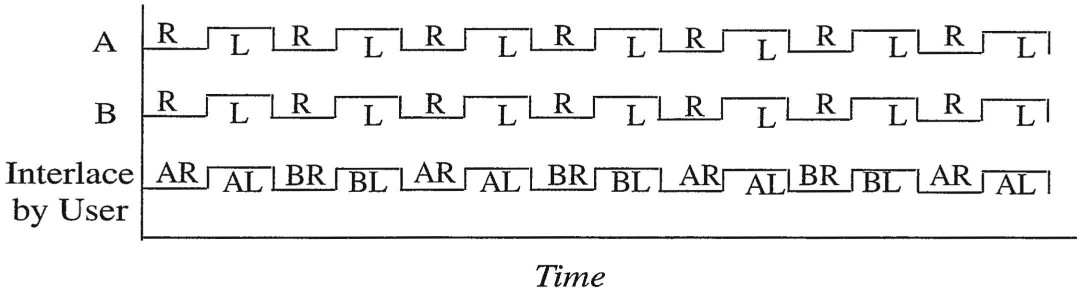


Figure 5.1: Video Multiplexing Pattern for Two Users, A & B

of the modifications required for the different encoding schemes to be done with firmware modifications only. Although the glasses used for this research were modified physically, those modifications are needed only for development and the glasses and emitters in a production system would not require physical modifications.

Although the multiplexing scheme has been discussed in the previous two chapters, a brief review of the scheme follows. Recalling that the hardware image generators produce a stereo synchronization signal which correlates to the vertical sync of the signal as it switches between the left and right eye images. The now familiar *Figure 5.1* shows the periodic pattern of the synchronization for labeled *A* and *B*. The multiplexing of the signals *A* and *B* is accomplished by interlacing the users; User A's right and left eye images are shown then User B's right and left images are shown. This pattern is graphically represented as the signal labeled "Interlace by user" in *Figure 5.1*.

In section 4.3 we presented a signal for synchronization between the switchers and the emitter. The signal generated by the hardware is shown representatively in *Figure 5.2*. The processing of this signal for the emitter is then the next step. The emitter and glasses both use a flexible micro-processor, Microchip's PIC16C554 processor [MICR 98]. The PIC processor is programmed in assembly language and provides many convenient functionalities. For the purpose of detecting the pulse for the user synchronization, a very simplistic approach can be taken. The time of the delay was chosen in proportion to the time that the processor is active in state determination and in communication with the glasses. The time it takes for this process is approximately 100 μ sec. After the 100 μ sec delay, the processor needs only to check the

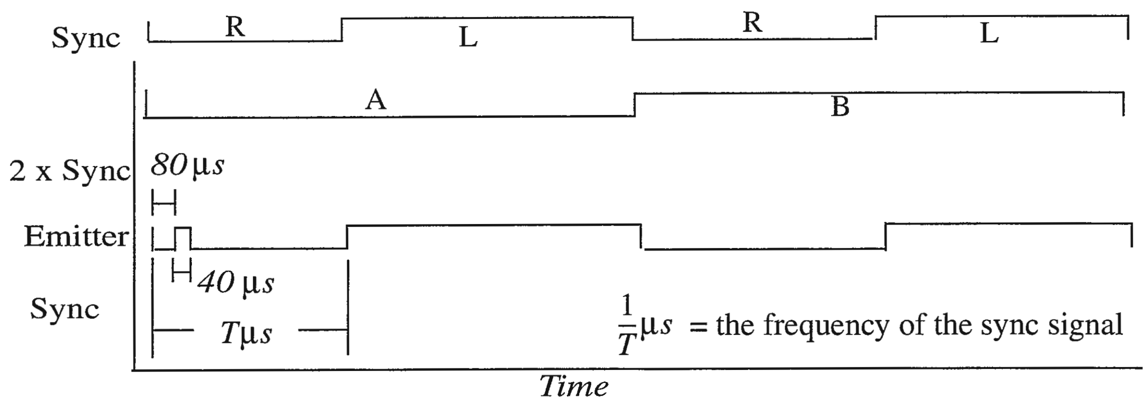


Figure 5.2: Emitter Synchronization Signals

input line to see if it is logically high. Referring again to *Figure 5.2* we can see that the 100 μ sec delay fits nicely into the center of our 40 μ sec pulse, giving 20 μ sec worth of leeway.

When the stereo sync signal from the graphics hardware is first applied, the emitter must guess which user to display until it receives the user sync signal, i.e. the pulse added to the signal. This means that the glasses could potentially be showing the wrong user the images, however, this can only happen for the first two frames, after which the system will receive the pulse and determine which user should actually be viewing the images. The user information is then updated internally and the correct user starts receiving their images. The pattern is then consistent, guaranteeing the correct user viewing the environment until the system is reset again, i.e. the loss of the stereo sync signal.

The second part of the firmware revision is for the actual de-multiplexing of the images. This is done via the shuttering of the LCD lenses. The glasses are sequenced via the communication with the emitters; the emitter signals when to change the LCD shuttering. The pattern of the glasses shuttering is four steps for both the two user stereo and four user mono setups. In the two user stereo system, the glasses have three states: both lenses closed (when the other user images are on the screen), left eye lens open, right eye lens open. For the four user mono setup the two states are both lenses open and both lenses closed. The emitter sends a sequence of pulses to delineate the four steps, correlating to the four images. *Figure 5.3* shows the states for the two user stereo setup and for the four user mono setup.

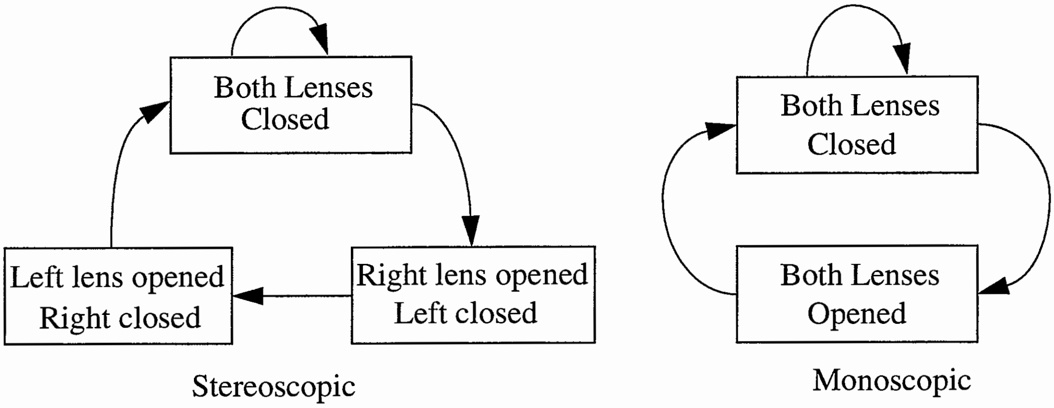


Figure 5.3: State Diagrams for Glasses Transitions

The four steps then map to the individual states for each set of glasses. The important thing to note from the start here is that the individual glasses' programming is what determines which state progression is used and where it aligns with the emitter sequence. An explanatory example of how the states progress is giving in *Figure 5.4*. The sequencing steps from the emitter will be denoted as steps ***d***, ***e***, ***f***, and ***g***. The sequencing will follow that order-

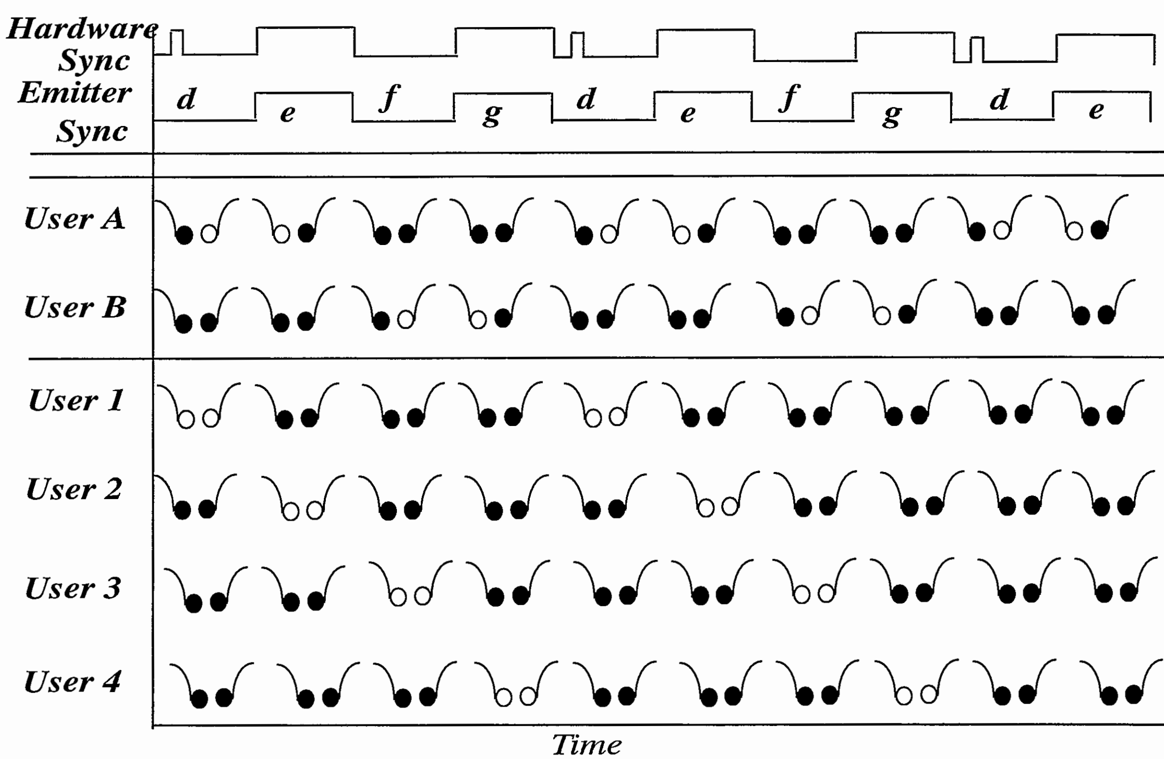


Figure 5.4: Glasses Sequencing Example

ing. Combining the sequencing determined for the images, refer to chapter 3 if needed, and the sequencing signal of the emitter. Looking at *Figure 5.4* the impact of this is seen. In the figure User A is designated to be the first user and hence User A's images occur during the ***d*** and ***e*** parts of the signal. The ***d*** part is the right eye image and the ***e*** part is the left eye image.

Having determined the emitter's synchronization the glasses must receive a signal from the emitter to actually do the shuttering. The pattern of this shuttering has already been determined and shown in the sequencing example, *Figure 5.4*. The glasses are each programmed to be a state machine, where state correlates to a signal from the emitter. When the glasses get a signal from the emitters they decode that message into one of the steps, ***d***, ***e***, ***f***, or ***g***. Each set of glasses has a state machine which is established based on this sequencing. Referencing the example again, if User A's glasses receive a signal from the emitter which indicates ***e*** it will open the left lens and close the right, knowing that in the state before the right has to have been open. User B's glasses will loop back to the same state it was in, which was a closed state. In the four user monoscopic case, one set of glasses will have both eyes open and the other glasses will be in the closed state for both eyes.

5.2 Integration into VR Juggler

The final step required for our design is to have the correct images being produced and placed on the correct screens. As is typical in virtual reality environments a software library is used to facilitate these actions. The software library presented in Chapter 2, VR Juggler, has been selected because it includes a number of important abilities required for the development of software for this project, namely the support for multiple active users.

VR Juggler already has built into it support of a multiple user virtual environment. This support is namely the ability to have multiple "heads" configured for the environment. In VR Juggler each user has a head, indicating the user's position. Each head automatically has projections calculated based on its position and the position of the screens in the VR system. The same virtual environment is used for both heads, with the users being located at different positions in the environment. Additional users can be added to any compiled virtual environment through only a configuration change, i.e. configuration files loaded when the program is executed. This allows movement for the user as far as tracking and the physical constraints of the environment allow them to move. If the world is translated, rotated, scaled, etc. by a user via the usual input methods, any user's view will also be affected by this. VR Juggler provides the capability to give each user independent control over the world. To make the users have independent control the programming must be changed. The change is not extremely significant and can be done by any programmer at a high enough competency to be creating a virtual environment. The basic method is to have multiple input devices, which in some manner form the input for the transformations, with individual devices for each user. The object orientated approach taken by VR Juggler allows this to be done fairly simply; An input object is created for each input device and separate user data is created for each user. VR Juggler automatically does the calls to the draw function for each user, when the user data associated with that user is valid. The cubes example in the VR Juggler distribution shows how to perform this in software [JUGG 00].

The main work which has to be done in order to have the multiple viewer setup function is the configuration files for VR Juggler. As was stated in the background VR Juggler loads a number of configuration files to establish the parameters of the environment, removing the environment dependence from the compiled binary. VR Juggler uses projection surfaces to define the physical environment of the VR system. Each physical projector is correlated with a projection surface, which is directly representative of the surface that is using. The signal for each projector is produced by a piece of graphics hardware that has an associated portion of the screen which it is drawing. VR Juggler must therefore coordinate the position and size of the graphics window for each surface with the proper place in the frame buffer(i.e. the portion that the graphics hardware connected to the projector uses). So we must change the configuration files to include twice the number of display surfaces, where each is a duplicate of one in the "normal" system. These new display surfaces are then associated with the correct pipes and then with the correct "head."

6. RESULTS

Four switchers were built for use with the C2, a CAVE™ like VR system at ISU. One switcher is a master switcher and the remaining three slave switchers being chained together. The glasses and emitters were reprogrammed to provide two separate stereoscopic views. Multiple pairs of each reprogrammed glass were created in order to provide a more thorough evaluation.

The setup was run at a variety of refresh rates and resolutions, as allowed by the graphics hardware and projectors. We ran the system with refresh rates and resolutions of: 1024x1024 at 96Hz, 1024x 1024 at 120 Hz, 1024x768 at 140 Hz, and 800x600 at 160 Hz.

Although initial results were encouraging, the results were not to the level desired. Running the system at the all of the above rates caused significant bleed through problems. The glasses were reprogrammed to counteract the persistence problem. The persistence remains noticeable particularly in the blue and green colors, while the red tubes persistence was small enough to not be noticed. The persistence problems are similar to those found in the original CAVE research [CRUZ 93]. In the original CAVE™ research the refresh rate of the green tube was found to be too slow. At this increased speed and with distinct images found in a multiple viewer system, both the blue and green tubes have too long of persistence. Testing has been performed on original glasses of both NuVision 60GX and Stereo Graphics' CrystalEyes III glasses. We found that the bleed through was actually reduced in our reprogrammed glasses from the original glasses. One possible reason for this is that the two images in a stereoscopic environment are similar enough that the mind can remove the ghosting from the images to for the stereoscopic pair without the viewer noticing.

After the additional modifications to the glasses encoding to account for the persistence problems the system was tested again. On the extreme ends of the refresh rates, 96 Hz and 160 Hz, the same amounts of bleed through were exhibited. Analysis of the bleed through problem leads us to believe at this time that it is due to the LCDs, and no longer the persistence problem. Although the remaining bleed through is present in all three colors, it is a significantly reduced bleed through than the one caused by low-persistence phosphor. In starkly contrasting environments, one user's images black and the other's images white, the bleed-through is unpleasant but not significant enough to make the multi-user system unusable.

7. CONCLUSIONS

The design presented in this thesis meets all of the requirements of the design specification, providing the flexibility and scalability desired. Additionally, the implementation of the design was successful within the scope of this research. The design chosen provides a method which is scalable as the underlying technologies improve to allow for more users. The design also provides flexibility for research to allow other configurations, such as monoscopic displays with multiple people. Throughout the course of the research development the flexibility of the design increases productivity and significantly shortened the time to implement the scheme. Because of the nature of the equipment required for time-multiplexed viewing, the glasses and emitters, it is possible to use off the shelf parts with only software modifications to extend their usage to additional users.

The implementation of the design presented successfully displays the ability to have two active users viewing stereoscopic environments. The implementation is anchored by switchers for four display surfaces. The system functions with a small amount of bleed through such that the system operates at a level which can be used for research into areas which will rely on this technology. The choice of software library platforms also proved advantageous. The configuration files for VR Juggler setup the proper displays and interaction for the two user stereo system. Because of this platform all previously written software can be used with the multiple viewer system with only a change in configuration files.

8. FUTURE WORK

A number of future areas for work were identified in the process of this research. A large area for future research is the exploitation of this research. The potential uses and benefits of this research was beyond the scope of this research, although several potential uses were identified. The most useful ways to exploit this technology remains an open for future research.

The foremost is the ability for projectors to produce images at higher speeds without ghosting effects due to decay times and/or persistence problems. This is the problem that originally occurred with the original CAVE research done by Cruz-Neira at UIC, where the green tube had too long of a persistence. This problem was overcome by the introduction of P43 coated green tubes [Cruz 93]. With the frame rate increases for this research and even at the traditional speeds of single stereoscopic views a ghosting problem was noted on both the green and blue tubes. As the speed requirements of scaling to additional viewer expands this will continue to be an issue. Either new display technologies capable of higher refresh rates will have to be designed or faster tubes will have to be developed

The graphic generation hardware is another area which is now lagging behind. For the speeds at which this technology can perform the graphics generation hardware can not produce high enough quality images. Secondly the ability to hardware genlock the graphics pipes is only available on a few high end graphics machines. This ability needs to be implemented in lower end graphics devices, particularly at the commodity part level. If this were to be done it would introduce VR to new arenas and create new possibilities for multiple view systems, both in application space and in new capabilities for adding more viewers.

BIBLIOGRAPHY

[AGRA 97]Agrawala, M., Beers, A., McDowall, I. Frohlich, B. Bolas, M. and Hanrahan, P. "The Two-User Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space." Proceedings of SIGGRAPH '97. (1997): 327-332.

[BIER 99]Bierbaum, A., Just, C. "Software Tools for Virtual Reality Application Development" SIGGRAPH '97. Course 14.

[BIER 01]Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., Cruz-Neira, C. "VR Juggler: A Virtual Platform for Virtual Reality Application Development", Proceedings of IEEE VR 2001. (2001)

[BURD 94]Burdea, B., Coiffet, P. Virtual Reality Technology. John Wiley & Sons. New York, New York. 1994.

[C6 00]The C6 web site. Virtual Reality Applications Center, Iowa State University. <http://www.vrac.iastate.edu/new/c6page.html> 10 October 2000.

[CRUZ 92]Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R., and Hart, J.C."The CAVE: Audio Visual Experience Automatic Virtual Envrionments." Communications of the ACM. 35(6) (1992): 64-72.

[CRUZ 93]Cruz-Neira, C., Sandin, D.J., DeFanti, T.A. "Surround-Screen Projection Based Virtual Reality: The Design and Implementation of the CAVE." Proceedings of SIGGRAPH '93. (1993): 135-142.

[FAIR 00]Fairchild Semiconductors. "DM74LS74A Data Sheet." March 2000. South Portland, Maine.

[JUGG 00]The VR Juggler web site. Virtual Reality Applications Center, Iowa State University. <www.vrjuggler.org> 10 October 2000.

[JUST 98]Just, C., Bierbaum, A., Baker, A., Cruz-Neira, C. “VR Juggler: A Framework for Virtual Reality Development”. Proceeding of Immersive Projective Techonology(IPT) ‘98. (1998):

[KRUE 95]Kruger, W., Bohn, C.-A., Frohlich, B., Schuth, H., Strauss, W., Wesche, G. “The Responsive Workbench: a virtual work environment”. IEEE Computer, 28(7) (1995): 42 -48.

[LIND 99]Lindahl, G., Personal Communicaton. Aug. 00

[MCAL 93]McCallister ed. Stereo Computer Graphics: and Other True 3D Technologies. Princeton Univ. Press. Princeton, New Jersey. 1993.

[MCNA 98]MacNaughton Inc. “Nuvision 60GX User’s Guide.” 1998.

[MICR 98]Microchip Inc. “PIC16c55X Data Sheet”. Document DS40143C. 1998. Chandler, Arizona.

[PEDR 93]Pedrotti, F. Pedrotti, L. Introduction to Optics. 2nd Edition. Prentice-Hall. New Jersy. 1993.

[POWE 00]PowerWall. University of Minnesota. <http://www.lcse.umn.edu/research/powerwall/powerwall.html> 10 October 2000.

[STAN 97]Stanney, K. and Kennedy, R. "The Psychometrics of Cybersickness," Communications of the ACM, 40(8) (1997): 66-68.