Double buffering technique for binocular imaging in a window

Jeffrey S. McVeigh[1], Victor S. Grinberg[2] and M. W. Siegel[2]

[1]Department of Electrical and Computer Engineering
[2]Robotics Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT

Binocular digital imaging is a rapidly developing branch of digital imaging. Any such system must have some means that allows each eye to see only the image intended for it. We describe a time-division multiplexing technique that we have developed for Silicon Graphics Inc. (SGI™) workstations. We utilize the "double buffering" hardware feature of the SGI™ graphics system for binocular image rendering. Our technique allows for multiple, re-sizable, full-resolution stereoscopic and monoscopic windows to be displayed simultaneously. We describe corresponding software developed to exploit this hardware. This software contains user-controllable options for specifying the most comfortable zero-disparity plane and effective inter-ocular separation. Several perceptual experiments indicate that most viewers perceive 3-D comfortably with this system. We also discuss speed and architecture requirements of the graphics and processor hardware to provide flickerless stereoscopic animation and video with our technique.

Keywords: binocular imaging, double buffer, time-division multiplexing, display windows, human factors

1. INTRODUCTION

The recent development of digital imaging systems will facilitate the practical implementation of the long-known concept of binocular imaging[11]. Efficient processing and coding of binocular imagery are possible in the digital domain. Current research efforts attempt to compress stereoscopic video streams into only slightly more bandwidth than that required for monoscopic streams[13, 14] and to enhance the realism provided by stereoscopic imagery[3, 9]. Binocular imaging has applications in education, medicine, and entertainment, to name a few. These systems are crucial whenever improved depth resolution or increased realism are desired. Regardless of the application or processing performed on the stereoscopic data, any system whose end-user is a person must have some method of presenting the correct image to only the eye intended for it.

A simple and economical method of directing the correct image to only the eye intended for it is known as time-division multiplexing (TDM)[1, 4, 5, 6, 7]. The presentation of binocular imagery using the TDM technique is illustrated in Fig. 1. The left and right-eye images (stereoscopic image pair) are alternately displayed once every vertical refresh period (T_r). The viewer wears eye-wear that acts as an optical shutter. While the left-eye image is displayed the shutter in front of the left eye is "open" and allows light to pass through it. During this time the right-eye shutter is "closed" and light transmission is retarded. The shutters then reverse operation when the right-eye image is displayed. The main considerations of time-division multiplexing

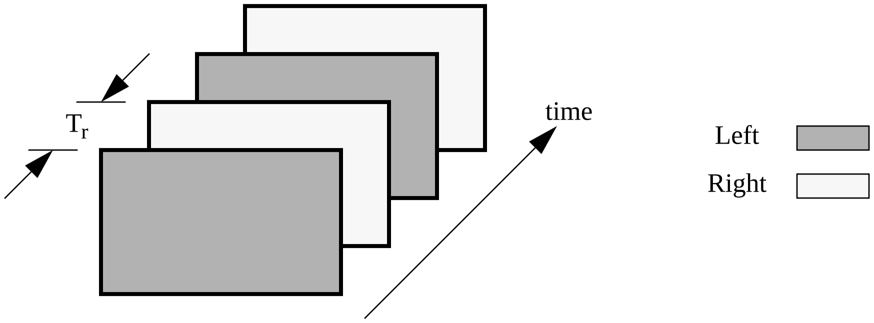

Figure 1. Time-division multiplexing technique for presenting the correct view to each eye.

schemes are the method used to rapidly alternate between the left and right-eye images and the rate at which the images are refreshed.

A common method to alternate between the two views on interlaced displays is to have each image occupy one of the two fields. A field consists of either the odd or even numbered scan-lines of a complete frame. On interlaced displays one field is followed by the other field at a rate equal to twice the frame rate of the display, while the vertical resolution of each field is one-half that of a frame. Binocular imagery is displayed with this method by merely line interlacing the left and right-eye images, with the eye-wear synchronized to the field rate. However, since common interlaced displays have a field rate of approximately 60 Hz, the image presented to each eye is only refreshed once every 1/30 second, and objectionable flicker can be perceived. This coupled with the reduction in vertical resolution cause this method to be sub-optimal.

Another method for alternating between the left and right-eye images is the "split-screen" technique[6]. This technique utilizes the ability to "trick" some displays into refreshing at twice their nominal rate. If operated at the standard refresh rate, the binocular images would be displayed in a split-screen fashion, with the left-eye image displayed above the right-eye image, each squeezed vertically by a factor of two. Thus again each image has one-half the vertical resolution of the display. By inserting a vertical sync pulse between the two images the refresh rate is doubled. The left-eye image is stretched vertically and occupies the entire display for the new refresh period. During the next refresh period the right-eye image is likewise stretched and displayed. The eye-wear is synchronized to this new refresh rate. Each eye observes an image refreshed at 60 Hz and perceivable flicker is eliminated. This method is employed on Silicon Graphics™ workstations, which have become the machine of choice for researchers in the field of stereoscopic visualization of computer graphics.

Since the refresh rate is altered and the upper and lower portions of the display are overlapped, it is difficult to display monoscopic information while using the split-screen technique. To display monoscopic information an identical copy of the image must be written to the top and bottom halves of the display. The significance of the difficulty of displaying monoscopic and stereoscopic imagery simultaneously is evident when we consider the current trend in display systems. Consumers desire multi-purpose display systems capable of performing multiple, independent tasks simultaneously. The most obvious example of this trend is the popularity of window-based systems. To adequately handle common monoscopic objects such as menus, scroll-bars, and pointers while using the split-screen technique a 3D-windowing system must be developed (see the work by Safier and Siegel for an in-depth discussion of this topic[12]). We desire a binocular imaging method based on the windowing paradigm that stipulates independence between the functionality of concurrently running processes.

In this paper we demonstrate a simple, cost-effective method for presenting the correct view to only the eye intended for it. Our technique allows for multiple, re-sizable, full-resolution stereoscopic (along with monoscopic) windows to be display simultaneously. We utilize the "double buffering" hardware feature of the SGI™ graphics system for binocular image rendering. In this implementation, one frame buffer is dedicated to the right-eye image, the other to the left-eye image. Software triggers an external logic level to reciprocally toggle – synchronously with the buffer state – the left and right-eye shutters of a pair of LCD goggles.

The remainder of the paper is organized as follows: Section 2 discusses the double buffering hardware including its original purpose and our alterations; Section 3 describes software tools developed to exploit this hardware; Section 4 treats implementation issues regarding processor and graphics hardware; and Section 5 provides concluding remarks.

2. DOUBLE BUFFERING TECHNIQUE

Double buffering is a hardware feature of graphics workstations intended for the smooth presentation of animation and video sequences[8]. It is often desirable to render or write the next frame completely before displaying it. If a system does not possess double buffering capability a displayed frame may consist of portions of the current frame and also the previous frame. The occurrence of this discontinuity obviously depends on the speed of the rendering hardware and the complexity of the current frame. These dependencies can be alleviated with the use of a double buffer.

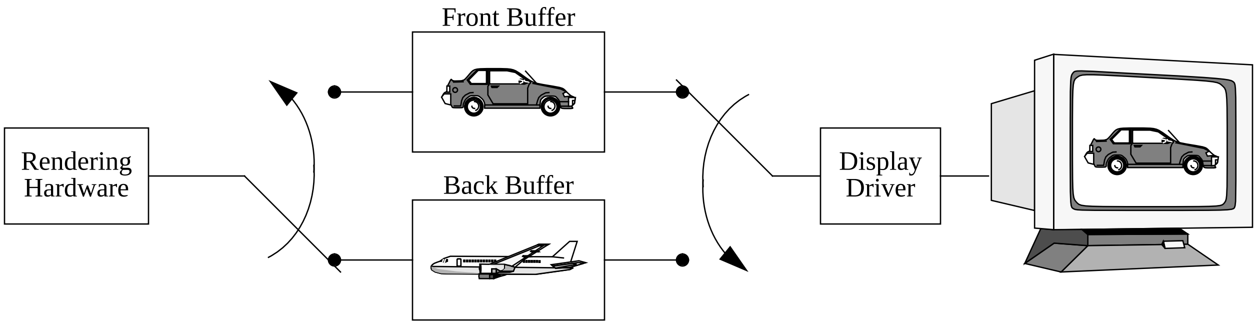

Figure 2. Double buffer operation

The basic structure of a double buffer and the switching between the two buffers are depicted in Fig. 2. In the double buffer configuration a program addresses frame memory as if it were two, distinct buffers. The display driver reads from only the "front" buffer and this information is displayed on the monitor. At the same time, the rendering hardware writes into the "back" buffer. Once the next frame has been completely written to the back buffer the two buffers switch roles. To avoid frame discontinuity, the swapping of buffers is normally delayed until the next vertical retrace interval.

In modern hardware with modern window-based operating systems each process window possesses double buffering capability. The buffers for each window may be swapped independently of other windows. Also, windows with single and double buffers may be displayed simultaneously. The swapping of the buffers is under software control, with the swap period equal to integer multiples of the refresh period of the display.

Double buffering capability provides a means of rapidly alternating between two images. This is exactly the goal of time-division multiplexing techniques for binocular imaging. In our technique we dedicate one buffer to the left-eye image, and the

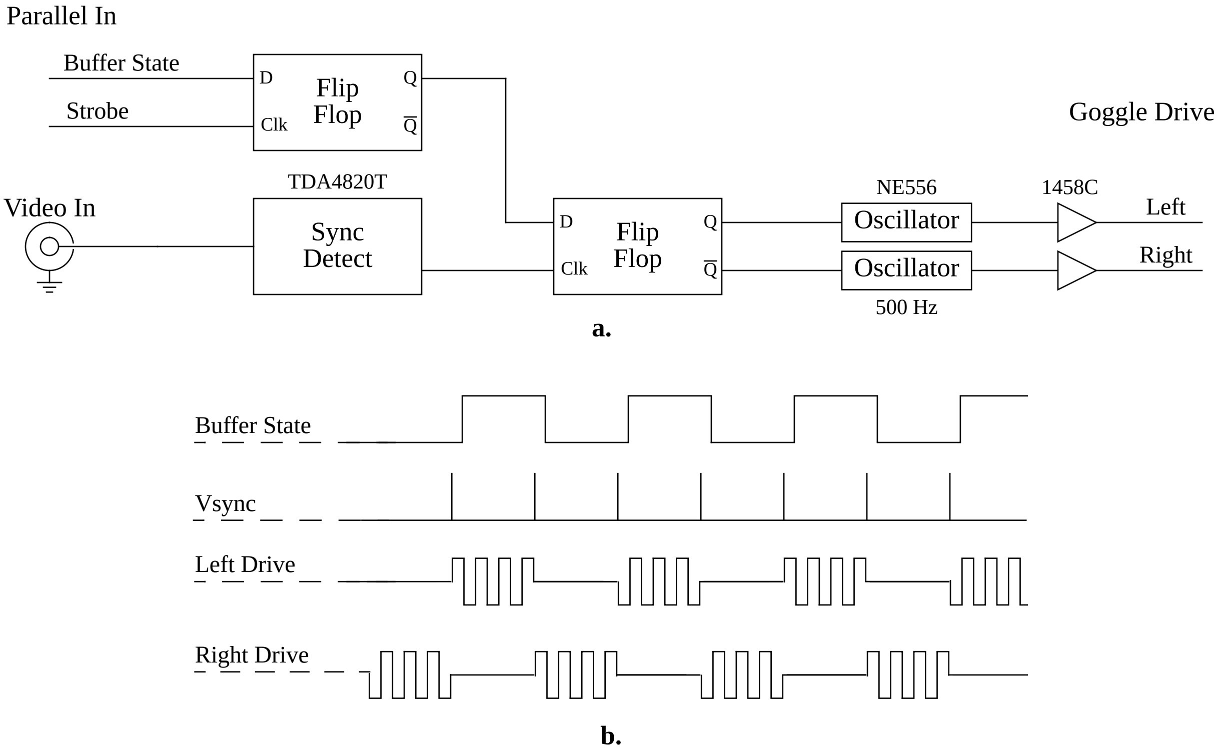

Figure 3. Goggle drive. a). logic diagram, b). sample signal levels

other to the right-eye image. During initialization we write the left-eye image to the back buffer. The buffers are then swapped and the right-eye image is written to the new back buffer. The buffers are then continuously swapped, at a rate whose maximum is equal to the refresh rate.

The swapping of the buffers does not occur automatically, but instead must be initiated by the application program with SGI's Graphics Library™ `swapbuffer` command. It is therefore possible for the swapping to not occur every vertical refresh. This delay may be attributed to concurrently running processes or to the binocular imaging program itself (see Section 4 for a discussion of the effect of this delay on the perceived quality of the stereoscopic information). This, coupled with the fact that the exact instant when the first binocular image was displayed is unknown, requires additional hardware to discern the current state of the buffers (i.e., the vertical sync signal alone cannot be used to indicate which eye-image is displayed). The logic diagram of the additional hardware constructed and sample signal levels are shown in Fig. 3. Once the program calls the `swapbuffer` command the buffers are swapped during the next vertical refresh. After each `swapbuffer` call the application's program must write the next state of the double buffer to the parallel port; our convention is that a high logic level indicates that the left-eye image will be in the front buffer during next frame period. This state is clocked by the parallel port strobe and is stored in the first flip-flop. We perform sync detection on the video signal to extract the vertical sync pulse (vsync), which clocks the second flip-flop. From the sample signals depicted it can be observed that the vsync signal comes slightly before the transition of the buffer state signal. The buffer state is thus written to the parallel port *after* the `swapbuffer` command to avoid a race-condition; if the state were written before the `swapbuffer` call the timing relationship between the outputs of the first flip-flop and the sync detector could be unstable. The outputs of the second flip-flop gate a pair of 500 Hz oscillators, only one of which is thus active at a time. The oscillators drive the left and right lenses of a pair of active LCD shuttering goggles[1], where an active oscillator drives the corresponding lenses to the closed state.

Each binocular window is of full-resolution using this double buffering technique. Multiple binocular windows may be displayed simultaneously and moved anywhere on the screen surface. In our display program we have included a simple scaling routine of the stereoscopic image pair to allow for window re-sizing. The use of binocular windows does not affect the function of any other windows present on the display, and these other windows can be viewed naturally without removing the shuttering goggles.

We have implemented this technique on a monitor capable of a refresh rate of 72 Hz. Several informal perceptual experiments indicate that most viewers perceive 3D-stereoscopy comfortably with this system. The system was also demonstrated to novice and experienced viewers of binocular imagery; depth was perceived comfortably by all viewers with normal stereopsis, with only a few objections to perceptible flicker. We speculate that the relative lack of flicker at 36 Hz per eye versus the well-known flicker at 30 Hz per eye may be due to the small window size relative to the spatial resolution of the flicker perceptor.

3. SOFTWARE TOOLS

We have developed user-controllable options for our binocular imaging software to fully exploit this technique and to provide viewers with comfortable stereoscopy. These options include allowing a viewer to specify the location of the zero-disparity plane and the effective inter-ocular separation. We will begin with a brief digression into possible causes of discomfort when viewing stereoscopic images to explain the origin of these options.

The presence of discomfort or fatigue experienced while viewing stereoscopic images has been reported in the literature[6, 10, 11, 17]. The most likely cause of this discomfort is due to the violation of relationships among cues that are tightly coupled in the real-world. Here we will assume that the images are refreshed at a rate fast enough to eliminate any perceivable flicker and its associated visual artifacts. Some discomfort may then be attributed to the breakdown of the accommodation/convergence relationship and conflicts between interposition and disparity depth cues.

[1] We used the goggles sold with the Toshiba™ stereoscopic camcorder in our experiments. The drive circuitry, to the right of the second flip-flop in Fig. 3, might be somewhat different for other goggles.

When a viewer observes an object in the real world the eyes converge to a specific point and focus (accommodate) to the same point, reducing retinal disparity. When viewing a stereoscopic image pair on a screen the eyes converge as if portions of the image are located at different distances, but they will remain focused on the plane of the screen. The breakdown of the accommodation/convergence relationship when viewing binocular imagery on a screen is an often discussed source of eye-strain.

If an object in a binocular image appears to lie between the screen surface and the viewer, and it is cut-off by the screen or window surround, a conflict of depth cues will arise. The cue of interposition indicates that the object must lie behind the screen surface, while the object's disparity indicates that its should lie in front of the screen.

The violation of the accommodation/convergence relationship can be minimized if all points within a stereoscopic image pair lie on or behind the screen surface. If this condition is met the conflict between interposition and disparity depth cues will be completely resolved. All points can be forced to lie on or behind the screen surface by adjusting the location of the zero-disparity plane to coincide with the point closest to the viewer; implementation requires simply horizontal image translation. The zero-disparity plane is the plane on which points lie on the screen surface. To adjust the zero-disparity plane towards the viewer, the left-eye image is shifted to the left with respect to the right-eye image. This causes points that previously appeared to lie in front of the screen to lie on or behind the screen surface. The reverse operation is performed to move the plane away from the viewer. Our binocular imaging software allows the viewer to control the location of the zero-disparity plane with either key-strokes or by adjusting a scroll-bar. As the plane is adjusted, cropping is automatically performed on the edges of the images, eliminating the vertical strips where overlap between the two images no longer exists.

Discomfort experienced when viewing binocular imagery is not completely the result of the breakdown of the above mentioned relationships. Hence, merely shifting the zero-plane disparity does not totally remove this discomfort. The authors observe that some viewers prefer a greater sense of depth, which is provided by a relatively large effective inter-ocular separation, while others enjoy a reduction in the disparity range, achieved by a smaller effective inter-ocular separation. The effective inter-ocular separation can be varied if intermediate views of the stereoscopic image pair are available. An intermediate view is the view that would be obtained by a sensing device located between, and on a straight-line connecting the two sensing devices that recorded the original binocular image.

Our software allows the user to control the effective inter-ocular separation by displaying appropriate intermediate views. If only a single left and right-eye image pair are available to the display application this option is disabled. The generation of intermediate views is quite straightforward for computer-generated images where a three-dimensional model of a scene is rendered. In graphics, the views can be generated by altering a few parameters and then by rendering the new images. The more interesting case occurs when a complete three-dimensional model of the scene is unavailable and must be estimated, as is the case for real stereoscopic images (see the references on the synthesis of intermediate views from real image data for a more complete discussion[2, 9, 15]). If these intermediate views are available the viewer can again use key-strokes or a scroll-bar to alter the effective inter-ocular separation. Intermediate views can also be used to provide "look-around" capability[9, 16]. In our software we have implemented simulated motion parallax by tracking the viewer's horizontal location. The appropriate left and right-eye intermediate views are then displayed for the position obtained from a head-tracking device.

A sample binocular image displayed in a window with our double buffering technique is shown in Fig. 4. The left and right-eye images are overlapped, as would be observed on the display without the use of goggles. The horizontal scroll-bar controls the location of the zero-disparity plane. The plane location has been adjusted for this stereopair to force the large foreground tree to lie on the screen surface and the rest of the scene to lie behind the screen surface. The effective inter-ocular separation is controlled by the vertical scroll-bar. With the vertical scroll-bar set at its highest position the extreme left and right-eye images are displayed. Conversely, the same images are presented to each eye when the vertical scroll-bar is at its lowest position, and no stereoscopy can be obtained.

4. QUADRUPLE BUFFER

Throughout this discussion of our binocular imaging technique in a window we have concentrated on the presentation of still stereoscopic information. For static images the only process that must be performed by the imaging software is the swap-

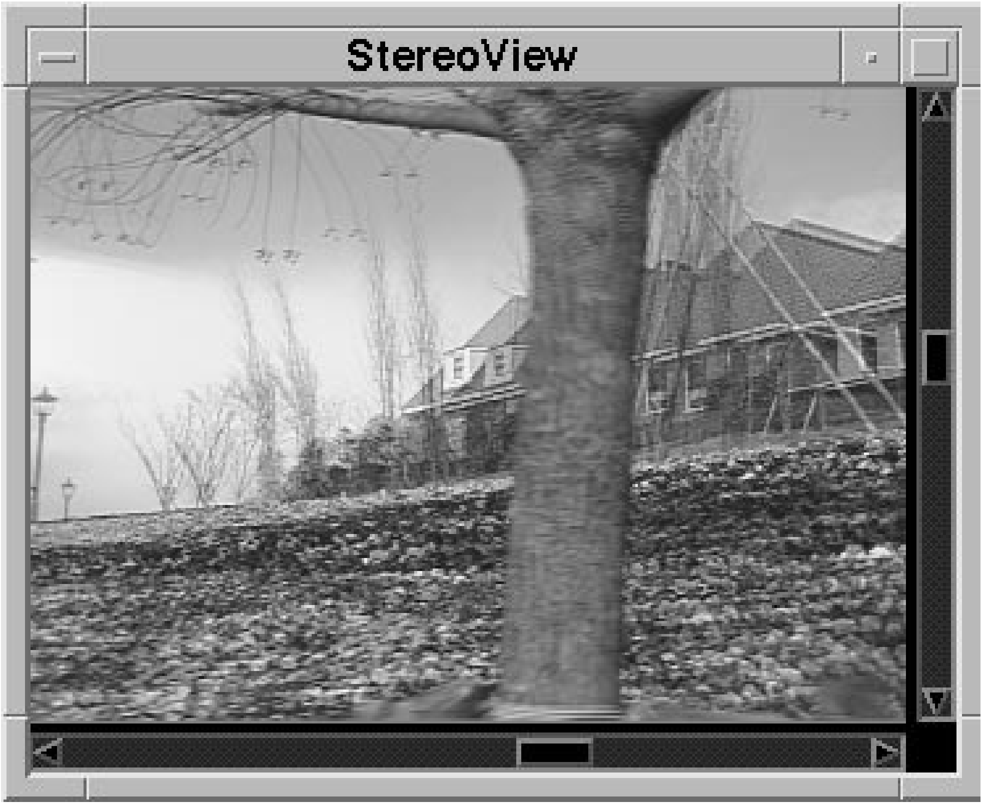

Figure 4. Sample binocular image pair time-multiplexed in a window with user-controllable display options

ping of the front and back buffers once every vertical refresh. More interesting scenarios occur when the binocular image is from a sequence of image pairs that are to be displayed in a continuous manner. Now the software may be required to read images from disk (possibly involving decompression) for video sequence or the images must be rendered for computer-generated animation. The images must then be written to the back buffer. Depending on the complexity of the images and the speed of the hardware, these operations may take more than one frame period to complete. The additional hardware constructed to drive the shuttering goggles (Fig. 3) will keep the eye-wear synchronized with the displayed images. However, if the buffers are not swapped every refresh period perceptible flicker and visual artifacts from an uneven duty cycle will occur.

We have implemented our windowed binocular imaging software on an Indigo XZ, SGI™ workstation with one 4400 processor. This hardware is capable of performing a bit-map operation from RAM to the back buffer in one refresh period for video frames containing approximately 1.3×10^5 pixels. Our benchmarks for animated computer-generated images are provided in Table 1. Therefore, an increase in processor and graphics engine hardware by an order of magnitude will allow for the smooth presentation of full-screen (1280×1024 pixels) binocular animation and video with this technique.

Operation	Time/pixel (nsec)
Clear screen	8
Clear z-buffer	7
Draw polygon	30

TABLE 1. Rendering hardware benchmarks

In the discussion of double buffering the original purpose of this capability was described as the ability to display smooth monoscopic animation and video. The double buffer alleviates the dependence on processor and rendering hardware to achieve this goal. Likewise, to alleviate these dependencies for the smooth presentation of stereoscopic animation and video we require a quadruple buffer configuration. Each double buffer pair is dedicated to either the left or right-eye image stream. Swapping between the double buffer pairs should occur exactly once every vertical refresh, without requiring a software trigger. The two front buffers continuously swap while the next frame for the left and right-eyes are written to the two back buffers. Systems with quadruple buffering capability exist, and, pending funding, we hope to implement this technique on such a system shortly.

5. CONCLUSION

We have presented a simple method for the presentation of binocular imagery using a time-division multiplexing technique. We utilized the double buffering capability of Silicon Graphics Inc.™ workstations to rapidly alternate between the two eye-images. One frame buffer was dedicated to the left-eye image, the other to the right-eye image. We also described goggle drive hardware that was constructed to synchronize the eye-wear with the buffer state. While the use of a double buffer for binocular imaging has been suggested in the literature[4], to our knowledge this work is the first to allow for multiple stereoscopic and monoscopic images to be displayed simultaneously in a windowing system. The independence of this technique with respect to concurrently running processes enhances its usefulness for both practical and experimental purposes.

We have illustrated user-controllable options implemented in our binocular imaging software. The goals of these options are to reduce discomfort often experienced while viewing stereoscopic information and to increase the realism through simulated motion parallax. These options include the adjustment of the zero-disparity plane location and the effective inter-ocular separation. Finally, we discussed the need for quadruple buffering capability to allow for smooth binocular animation and video with this technique.

6. ACKNOWLEDGEMENT

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

7. REFERENCES

1. P. Bos and T. Haven, “Field-sequential stereoscopic viewing systems using passive glasses,” *Proc. of the SID*, Vol. 30, No. 1, pp. 39-43, 1989.

2. C. Cafforio, F. Rocca and S. Tubaro, “Motion compensated image interpolation,” *IEEE Trans. on Communications*, Vol. 38, No. 2, pp. 215-222, February 1990.

3. V. S. Grinberg, G. Podnar and M. W. Siegel, “Geometry of binocular imaging,” *Proc. SPIE Inter. Conf. on Stereoscopic Displays and Applications*, pp. 56-65, SPIE, San Jose, CA, February 6-10, 1994.

4. L. F. Hodges and D. F. McAllister, “Stereo and alternating-pair techniques for display of computer-generated images,” *IEEE CG&A*, pp. 38-45, September 1985.

5. H. Isono and M. Yauda, “Flicker-free field-sequential stereoscopic TV system and measurement of human depth perception,” *SMPTE Journal*, Vol. 99, No. 2, pp. 136-141, February 1990.

6. L. Lipton, The CrystalEyes Handbook, StereoGraphics™ Corporation, San Rafael, California, 1991.

7. J. A. Mays, “A high-resolution monoscopic/stereoscopic display,” *Proc. of the SID*, Vol. 28, No. 4, pp. 409-414, 1987.

8. P. McLendon, Silicon Graphics Inc. Graphics Library Programming Guide, Vol. 1, Silicon Graphics Inc.™, Mountain View, CA, 1992.

9. J. S. McVeigh, M. W. Siegel and A. G. Jordan, “Technique for synthesizing intermediate views from a stereoscopic image pair,” *Signal Processing: Image Communications*, submitted.

10. T. Miyashita and T. Uchida, “Cause of fatigue and its improvement in stereoscopic displays,” *Proc. of the SID*, Vol. 31, No. 3, pp. 249-254, 1990.

11. S. Pastoor, “3D-television: A survey of recent research results on subjective requirements,” *Signal Processing: Image Communication*, Vol. 4, No. 1, pp. 21-32, 1991.

12. S. A. Safier and M. W. Siegel, “Three-dimensional stereoscopic X windows,” *Proc. SPIE Inter. Conf. on Stereoscopic Displays and Applications*, elsewhere within these proceedings, SPIE, San Jose, CA, February 5-10, 1995.

13. A. Schertz, “Source coding of stereoscopic television pictures,” *Proc. IEE Inter. Conf. on Image Processing and its Applications*, pp. 462-464, IEE, Maastricht, Netherlands, April 7-9, 1992.

14. S. Sethuraman, M. W. Siegel, and A. G. Jordan, “A multiresolution framework for stereoscopic image sequence compression,” *Proc. IEEE Inter. Conf. on Image Processing*, Vol. 2, pp. 361-365, November 13-16, 1994.

15. R. Skerjanc and J. Liu, “A three camera approach for calculating disparity and synthesizing intermediate pictures,” *Signal Processing: Image Communications*, Vol. 4, No. 1, pp. 55-64, 1991.

16. C. D. Wickens, “Three-dimensional stereoscopic display implementations: Guidelines derived from human visual capabilities,” *Proc. SPIE Inter. Conf. on Stereoscopic Displays and Applications*, pp. 2-11, SPIE, Santa Clara, CA, February 12-14, 1990.

17. T. Yamazaki, K. Kamijo and S. Fukuzumi, “Quantitative evaluation of visual fatigue encountered in viewing stereoscopic 3D displays: Near-point distance and visual evoked potential study,” *Proc. of the SID*, Vol. 31, No. 3, pp. 245-247, 1990.