A Virtual Reality and Scientific Visualization Laboratory for Undergraduates in Computer Science.*

Roger W. Webster, Ph.D.
Department of Computer Science
Millersville University
Millersville, PA USA 17551
webster@cs.millersv.edu
http://cs.millersv.edu
http://zansiii.millersv.edu

Steven Wayde
PRC, Incorporated
One Bala Cynwyd Plaza
Suite 324
Bala Cynwyd, PA 18004
wayde@philly.nosc.mil

Abstract

A great deal of research being performed in computer science and in undergraduate and graduate programs in the sciences, centers around collaborative scientific projects and the visualization of the results. With so much activity in the scientific community today comprised of collaborative projects which include and require computer scientists, a logical step at this time is to expose students to the hardware, special devices, and software techniques of virtual reality, virtual world modeling, and scientific visualization. This paper describes a virtual world modeling laboratory for undergraduates in Computer Science. Specific hardware, devices, software, project specifications, and laboratory experiences will be described.

I. Introduction.

Virtual reality and scientific visualization is comprised of high-performance 3D computer graphics for scientific modeling to simulate, train, and experiment in the natural/physical sciences. As we compete with the Japanese and other industrialized nations developing the sophisticated computer visualization systems, it would be beneficial for computer science students to have the opportunity to develop software for scientific projects utilizing virtual environments.

Virtual Reality (VR) has undergone three basic stages of development. The first stage began with military and warfare simulations under the auspices of DOD and NASA [1]. The second stage, VR games at arcades and entertainment development, brought this technology to the public forefront. The third and current stage, which is still in its infancy, sights VR becoming used more and more in the business arena and scientific community. The current and possible applications for scientific modeling and virtual environments include: air traffic control simulations, architectural design, aircraft design, acoustical evaluation (sound proofing and room acoustics), computer aided design, education (virtual science laboratories, cost-effective access to sophisticated laboratory environments, virtual planetariums), entertainment (wide range of immersive games), legal/police (re-enactment of accidents and crimes), medical applications (surgery, molecular docking), scientific

---

*This project partially funded by the National Science Foundation under grant number: DUE-9350841 and by Faculty Grants Committee of Millersville University.

visualization (aerodynamic simulations, computational fluid dynamics), telepresence and robotics, flight simulation, virtual manufacturing [2].

It is crucial for today's computer science and science undergraduates to be exposed to these areas of work, and experiment with them in a laboratory in order to develop the skills necessary to compete in a sophisticated, global job market. We are attempting to provide these opportunities for students with this laboratory system.

II. Laboratory Hardware and Software.

Hardware.

The laboratory has SUN workstations and Intel/Pentium based PCs. The graphical engine is the SUN SPARC10-ZX computer workstation with dual 90 Mhz processors, 96 MB memory, 2 GB Hard Disk drive, a ZX Raster Manager card for graphics Z buffering, a 21" Stereo color monitor, and CDROM drive. The Polhemus Corporation's Fastrak II real-time 6 degree of freedom magnetic tracking system is used for a data glove and head mounted display. The system has three sensors (transmitters) and one receiver. The first sensor is mounted on the helmet for 3D immersive applications, the other two sensors are used for joystick and data glove tracking. The Fastrak II system operates at an update rate of 120 Hz for one receiver and 40Hz for three. We use two receivers in most applications, but some applications require three sensors. The data latency, which measures how fast the unit can provide real-time 3D positional information to the computer, is 4 msec unfiltered. Other sensor systems typically have 50 to 200 msecs data latency. The Fastrak system has more than adequately provided high accuracy and real-time response in our time sensitive/critical applications.

A Virtual Research Corporation VR4 Head Mounted Display (hmd) unit is used for immersive virtual reality simulations (see Figure 1). The VR4 uses 1.3" active matrix liquid crystal displays (LCD) that have a 60 degree diagonal field of view at full overlap. The resolution for each eye is 742 x 230. The hmd takes NTSC signals (640x480) as input, which means that the hmd loses one half of the vertical resolution from the original NTSC signal. The interpupilary distance (IPD) is 52mm to 74mm. The VR4 head mounted display weighs only .935 kg (33 ounces) and has a rugged outer shell that makes it ideal for undergraduate laboratory use. The hmd also has audio headphones with excellent 3D audio spatialization. Comparisons of commercially available hmd's can be found in [3], [4], [5].

Software.

The operating system is Solaris 2.4 (AT&T system V Unix) with X-windows. Other software includes: Sense8 Corporation's WorldToolkit Virtual Reality software for Virtual World Modeling, the 3D Studio graphics modeller system used to build 3D models and textures on the Pentium systems, XIL, XGL. All programming code is written in SUN 'C' using the Sense8 WorldToolkit library calls.

Sense8's WorldToolkit (WTK 2.1) is a very powerful, easy to use, object oriented package of 'C' programming function calls for visual simulation and virtual reality applications. WorldToolkit's high-level application programmer's interface (API) allows students to prototype applications quickly and reconfigure them on the fly. WTK also supports many vendor's virtual reality devices and has device drivers built in to the package. In the laboratory all of our VR devices except the Nintendo gun and steering wheel have WTK device drivers already in the library. Students can easily make a function call to open, and use the VR devices. WTK has in its object oriented library more than 650 high level calls to configure, interact, and control, real-time virtual environment simulations. Even our most complex applications have been written in WTK and the students have been able to locate, understand, and use the function calls easily and quickly.

The WorldToolkit software is organized into 18 classes which include universes, graphical 3D objects, viewpoints, sensors, paths, and lights. Functions are included for device instantiation, display setup, loading objects from 3D studio (*.3ds files) and autocad (*.dxf files), dynamically creating geometry, specifying object behavior, controlling rendering, and handling events such object collisions (used to detect if arrows or bullets have hit something). WorldToolkit is designed according to the philosophy of OpenVR. This means we can export our WTK demonstration software to many other computer platforms.

III. Current Projects.

Virtual Backhoe.

In our undergraduate course entitled "CS373 - Graphics and Virtual Reality", students are developing a virtual reality simulation of a John Deere backhoe and front end loader. Using the 3D Studio software product the students have built a model of the backhoe. The 3D graphics model is imported into the WorldToolKit Virtual World Modeling software. The control software has been written such that when the user moves the physical control levers, the 3D graphics model of the backhoe moves the way a real backhoe works. The user who is wearing the head mounted display sees the computer graphics model of the backhoe (see Figure 2) as it would actually move its boom, arm, and bucket (up, down, dig, swing side to side). The real-time control software has modeled the behavior of the backhoe to react to the physical lever movements. This is accomplished with a 'C' program that checks the positions of the Polhemus sensors attached to each of the two levers. Thus, people may train on how to operate a backhoe without having an actual backhoe available.

The students in the course built a wooden pod in the shape of a John Deere backhoe cab. A female student in the class who grew up on a farm managed to get a set of actual backhoe levers from an old backhoe. The students then assembled the lever unit into the pod and painted the entire pod with "John Deere" yellow (see Figure 3). The students spent many long hours working on their virtual reality project because it was interesting, challenging, and fun.

In this project the students learned about object oriented programming, virtual reality programming, virtual world modeling, and real-time software development which manipulates complex 3D objects. The control software which moved the graphic model of the boom, arm, and bucket of the backhoe according to lever movements, had to account for collisions of the components with each other. For example, moving the bucket too close to the boom (digging yourself up), and thus crashing into the backhoe unit. The students developed a software envelope which protects each component (boom, arm, bucket, cab) from colliding with each other. The software utilized a WTK function call which checks if any two objects touch or intersect. Also, the control software was written with the boom, the arm and the bucket 3D objects in an object hierarchy. The WTK software allows a 3D object to be "attached" to another object. The order of attachment is then a hierarchy. Any movement to an object further up in the hierarchy then moves the objects below the hierarchy accordingly.

A bird's eye view of the two backhoe levers shows the lever movements and corresponding control software effects on the backhoe.

```
     a              e
     |              |
  c--o--d        g--o--h
     |              |
     b              f
 Left Lever     Right Lever
```

where:

Move Left Lever
'a' <--> 'b' ==> Raises/Lowers boom.
'c' <--> 'd' ==> Swings boom left/right.
Move Right Lever
'e' <--> 'f' ==> Raises/Lowers arm.
'g' <--> 'h' ==> Moves Bucket up/down.

Figure 1. Virtual Reality Head Mounted Display and Sensor.

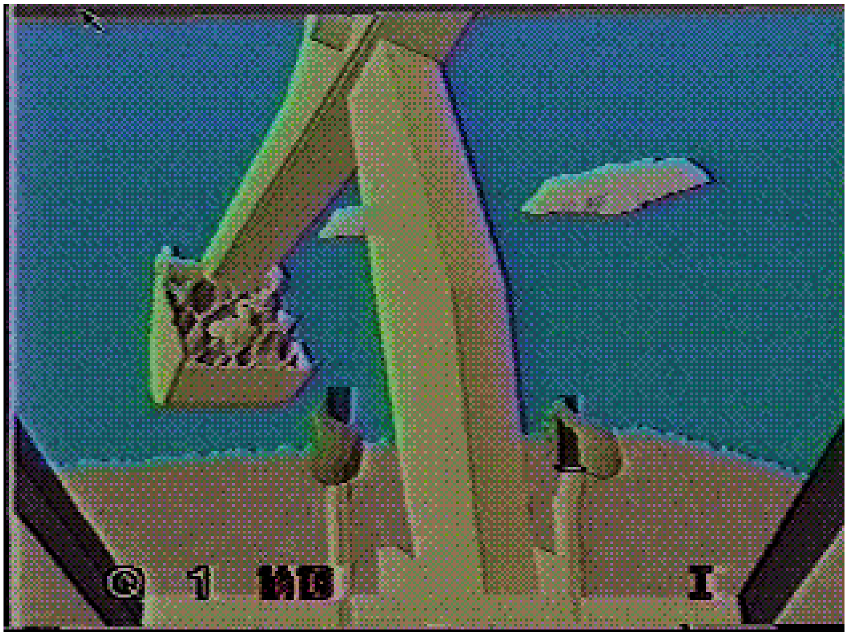

Figure 2. View from within Head Mounted Display of Virtual Backhoe System.

The goal of virtual environments is to make the simulation appear and feel as real as possible. As Kawalski in [6] notes "the three major components to a virtual environment are the visual, auditory, and haptic/kinesthetic environments. Each adds to the synthetic sensory experience that may one day present an interface to the human sensory system that is very close or indistinguishable from the real physical world." Thus, the auditory component of a virtual experience can increase the illusion and feeling of reality. The students in this virtual backhoe project taped the sounds of a real backhoe, digitized them, and used the audio in the software. Thus the user, while operating the backhoe, hears the sounds of an actual backhoe in the headphones. Well timed audio effects can enhance the virtual experience.

It is planned to have Physics students working with the Computer Science students to develop and program the physical characteristics of the model of digging snow, dirt, asphalt, and rock, with gravity and flow computations. Scientific modeling is so very important in today's current research environment, it appears to be beneficial to have undergraduate laboratory project activity such as this. Information and an mpeg video clip on the work-in-progress can be found on our WWW server at http://zansiii.millersv.edu/cs373.html (see VR Backhoe Project).

Virtual hang glider.

The virtual hang glider project is another project just under way. In this simulation, a user wearing the VR helmet can practice hang gliding without the fear of actually crashing. Three dimensional models of terrain, hills, valleys, and rivers have been built and/or downloaded off the Internet. A hang gliding bar was built from pvc piping and hung from the ceiling. A Polhemus 3-space tracking sensor was attached to the hang gliding bar. The students have written real-time control software which models the behavior of the hang glider to react to the bar movements just like a real hang glider. The control software monitors the Polhemus sensor mounted on the hang glider bar. When the user pushes the bar forward the user's view in the helmet shows the user flying up. When the user pulls the bar backward the user's view in the helmet shows the user flying downward. The user can also bank right and left by pushing the bar right and left as it works in a real hang glider.

The user wearing the head mounted display lies prone on a hammock to add to the effect of the simulation. The user must fly over two mountains and a river to get to the destination which is a landing pad. The user must also avoid telephone poles, mountains, and also keep from getting caught in a down draft of wind (see Figure 4). After being immersed in the virtual hang glider for a while, the user must then land on the landing pad or he/she will crash into a brick wall at the end of the landing pad. Information and an mpeg video clip on the work-in-progress is on our world wide web site at (see VR Hang-Gliding Project).

Virtual environment games.

Virtual environment games have been used as a software platform for students to practice the software engineering techniques endemic in real-time simulation systems. One game, which the students have called the "Star Wars Trench" game, has the user flying down the trench in an xwing style space vehicle, to shoot tie-fighters and the imperial fighter space craft manned by Darth Vader. This game requires very high speed graphics rendering capability and real-time sensor responsiveness.

In order to simulate the cockpit of the xwing jet, the students have connected the Nintendo steering wheel (see Figure 5) to the SUN sparcstation by designing their own signal converter printed circuit board. The board changes the Nintendo signals to be compatible with the SUN sparc station parallel port which normally connects to a printer (see Figure 6). With the signals converted and rewired into the parallel port, a small task written in 'C' can continuously check the paper-empty error condition (the pin is asserted high by the signal converter when the steering wheel shoot button is pressed). When the paper-empty error is detected, an interprocess communication (IPC) routine sends a message to the virtual realty application that the gun or steering wheel has shot. A diagram of the circuitry is shown in Figure 7.

Figure 3. The Virtual Backhoe Pod.

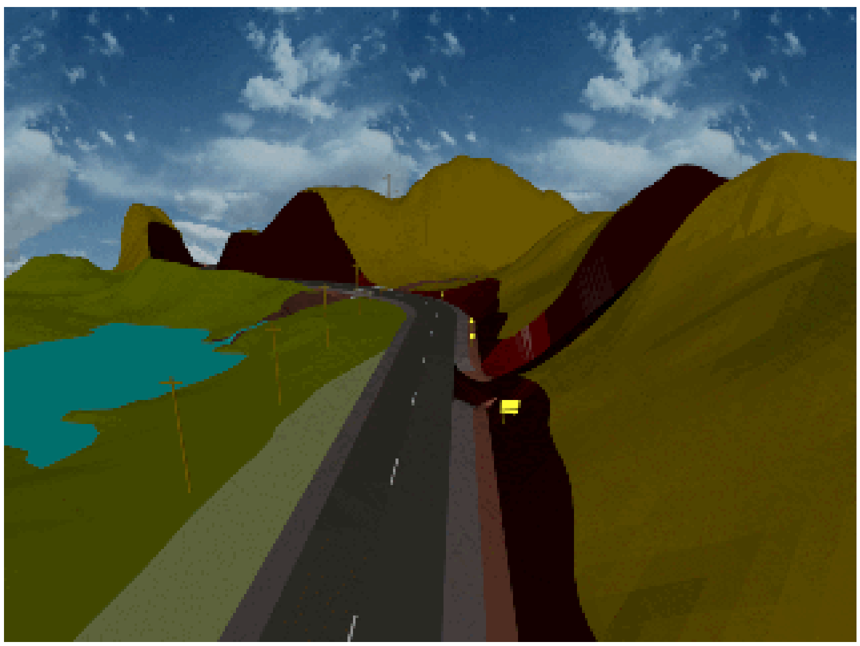

Figure 4. View from Head Mounted Display of Virtual Hang Gliding Training System.

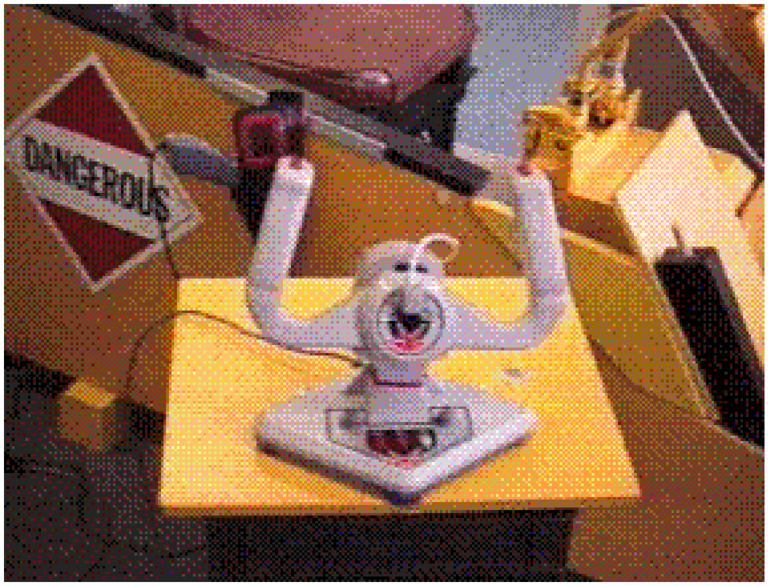

Figure 5. Nintendo Steering Wheel connected to the SUN parallel port.

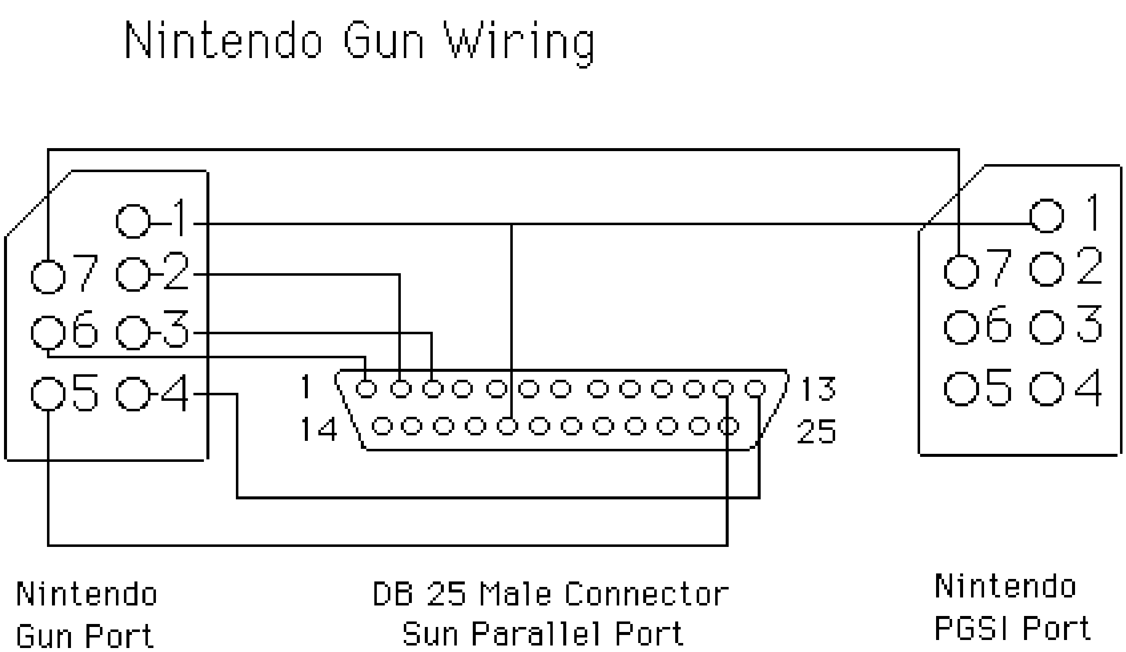

Figure 6. Nintendo Steering Wheel and SUN parallel port pin outs.

Control Yoke Converter Circuit

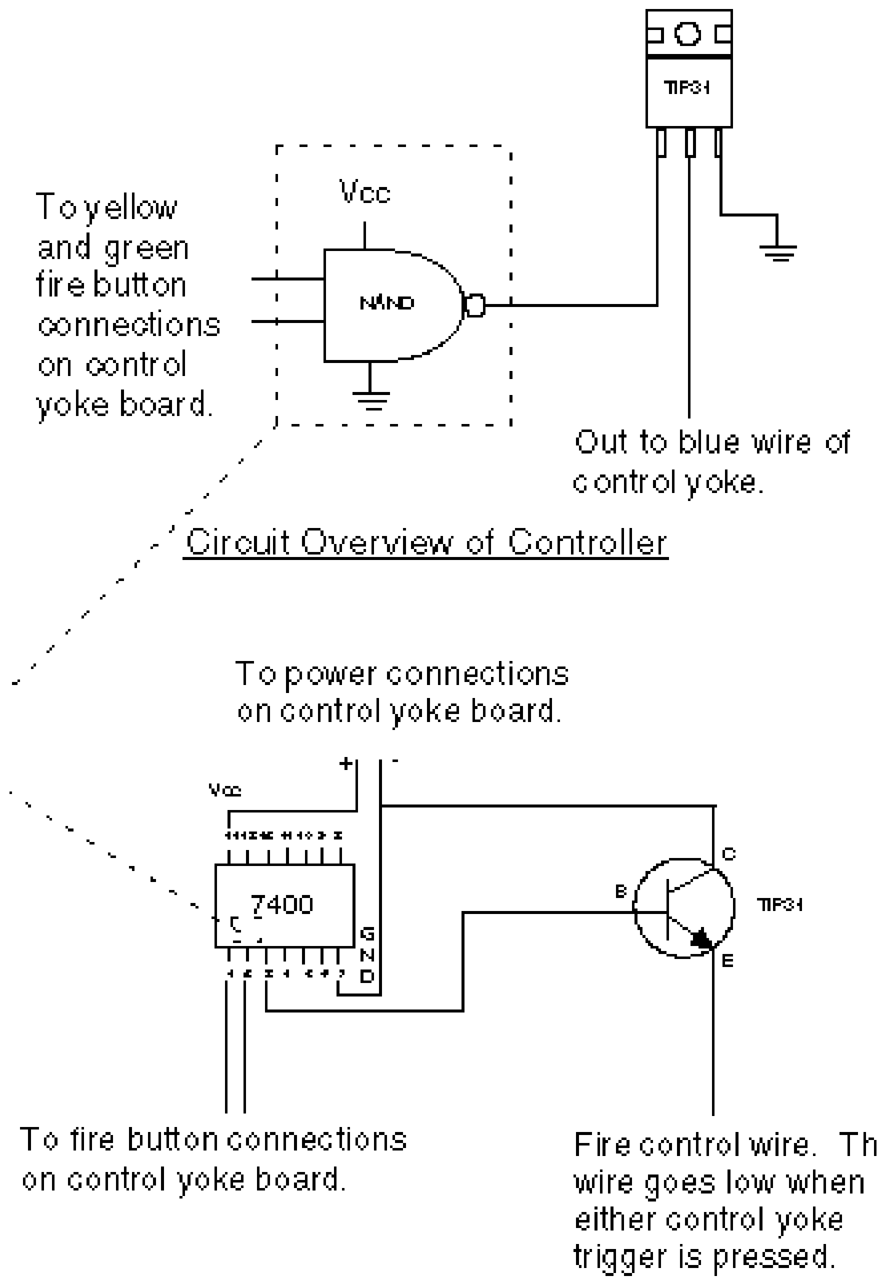

Figure 7. Schematic of signal converter - designed and built by Win Heagy

.

IV. Stimulating Interest in Engineering in Younger Students.

Demonstrations of the equipment and the projects are performed regularly in the laboratory. These presentations to area schools, businesses, and community groups bring recognition to the University and to the lab. More importantly, these presentations stimulate interest in science and engineering among young students. Presentations have been given to many diverse community groups: high school students, middle school and elementary school children, a rural partnership program for tenth and eleventh graders, and to the public in various University programs.

There is also a special three week research program for gifted high school students as part of the University Summer Science Training Program (SSTP). These young students spend three weeks, eight hours per day, five days per week with a University professor engaged in a small research project in Computer Science and Engineering. This is a tremendous opportunity for high school students to prepare for college.

V. Conclusion.

The authors are extremely pleased with the use of the laboratory so far. Students are very anxious to use the systems and spend many long hours programming, developing demonstration software, experimenting and exploring the techniques of virtual reality and scientific visualization.

In summary, the benefits of this VR laboratory are as follows:

o Many students gain hands on experience with software that is good preparation for graduate school and also in high demand in industry.

o The lab has increased student involvement in independent study with faculty members.

o It has facilitated student interest in pursuing more scholarly activities such as student publications, going on to graduate school, and seeking employment in research oriented institutions.

o It has stimulated interest in the co-operative education program on campus.

o Demonstrations to area schools stimulates interest in science and engineering to young students.

The laboratory is exposing students to the hardware, special devices, and software techniques of virtual reality, virtual world modeling, and scientific visualization. As we compete with other industrialized nations developing the sophisticated computer visualization systems, it appears to be beneficial for computer science students to have the opportunity to develop software for scientific projects utilizing virtual environments.

VI. Future Work.

Future work includes a project to visualize meteorological systems. In Meteorology the inherent complexity of evolving three-dimensional atmospheric circulation systems has always been difficult to visualize and describe. Scientific visualization and virtual world modeling offers new options for the development and display of atmospheric systems simulations changing in space and time. With the help of computer science students and this laboratory, students in Meteorology will be able to virtually experience the 3-D flow characteristics of systems such as thunderstorms and jet streams, for example, from both Lagrangian and Eulerian points of view.

Acknowledgments

This work was partially funded by the National Science Foundation under grant number: DUE-9350841 and by the Faculty Grants Committee of Millersville University. The authors would like to thank the following undergraduate research students: Win Heagy, Wendy Mechler, Jennifer Hare, Tim Merkt, Chris Scott, Brent Shelly, Stephan Colisimo, Adam Plowcha, Deb Mohler, Forrest Schenk, Mike Reinhart, Brian Donnelly, and Jim Rutkowski.

References

[1] Martin Cheek, "Business Waking up to Virtual Reality", IEEE Computer, December 1994, pg. 8-9.

[2] Roy Kawalski, The Science of Virtual Reality and Virtual Environments, Addison-Wesley, Reading, Massachusetts, 1993, pg. 14.

[3] Linda Jacobson, Virtual Reality, Sams Publishing, Prentice Hall, Indianapolis, IN, 1994, pps. 75-105.

[4] Jonathon Merril, Virtual Reality Special Report, Miller Freeman Publishing, San Francisco, CA, January 1996, pg. 58-66.

[5] John Vince, Virtual Reality Systems, Addison-Wesley Publishing, Reading, Massachusetts, 1996, pg. 288-295.

[6] Roy Kawalski, The Science of Virtual Reality and Virtual Environments, Addison-Wesley, Reading, Massachusetts, 1993, pps. 5-7.

Dr. Roger W. Webster

Dr. Webster earned the Ph.D. degree in Computer Science from the School of Engineering at Temple University in Philadelphia, Pennsylvania. From 1979 to 1982 he worked at the Hewlett-Packard Corporation's Medical Systems Division in Waltham, Massachusetts as a software engineer. Since 1983 he has been a Professor of Computer Science and Director of the Intelligent Machines laboratory at Millersville University in Millersville, Pennsylvania. Dr. Webster has been awarded five grants in six years in the area of real-time systems, robot vision, and artificial intelligence. He has published over 20 articles in robot vision, real-time systems, computer graphics and virtual reality, and artificial intelligence. Dr. Webster holds memberships in the IEEE Computer Society, the Association for Computing Machinery (ACM), the American Association for Artificial Intelligence

(AAAI), the Internet Society, and the American Society for Engineering Education (ASEE). His research interests are: Virtual Reality, Real-Time Systems Engineering, Robot Vision, and Artificial Intelligence.

Steven Wayde

Steven Wayde is an Assistant Computer Analyst for the PRC Corporation in Bala Cynwyd, Pennsylvania . He designs and implements Human Computer Interfaces (HCI) for the Joint Maritime Command Information System (JMCIS) to access the Naval Intelligence Processing System The work is being done using Motif, Sybase, and the C programming language on unix systems.