The Determination of Wheelchair User Proficiency And Environmental Accessibility Through Virtual Simulation

J. Edward Swan II[1]
Don Stredney[2]
Wayne Carlson[1]
Beth Blostein[1]

1: The Advanced Computing Center for the Arts and Design
1224 Kinnear Road
Columbus, Ohio 43212
{waynec, swan, blostein}@cgrg.ohio-state.edu

2: Ohio Supercomputer Center
1224 Kinnear Road
Columbus, Ohio 43211
don@osc.edu

Abstract

In this paper we describe a system which allows a power wheelchair user to drive through a virtual architectural environment. The system addresses the current lack of methods for evaluating a disabled person in order to match them with a suitable power wheelchair control mechanism. First we describe the system itself, including its hardware and software components and its user interface. Then we discuss both using the system to evaluate user proficiency and using the system as an architectural design tool, and we briefly describe our experience with a disabled person using the system. We conclude with a brief discussion of future plans.

1.0 Introduction

There are several hundred thousand power wheelchairs in use today, with approximately 23,000 new power wheelchairs purchased every year [Invacare 93]. Two recent developments promise greater accessibility to power wheelchair users: 1) recent advances in microcomputer technology have made increasingly sophisticated power wheelchair control mechanisms possible, and 2) the Americans with Disabilities Act of 1990 requires handicapped accessibility for (almost) all public structures. Yet despite this growth in the opportunities afforded by power wheelchairs, currently the evaluation of user proficiency and the suitability of a given wheelchair is largely guesswork, and user training is limited to practice with a (possibly unsuitable) wheelchair.

We are developing a system which uses virtual reality technology to evaluate user proficiency. Our system allows power wheelchair users to drive through simulated architectural environments. The system consists of an instrumented power wheelchair connected to a high-performance graphics workstation; it simulates the actual speed and maneuverability of the particular wheelchair within an architectural database. The display generates realistic interiors containing multiple light sources and surface textures, and is viewed in stereo through lightweight polarized glasses. The system maintains a hierarchical data structure which detects collisions between the virtual wheelchair and the environment.

This system is a tool for three groups of people: 1) for *health care professionals* it provides evaluations of power wheelchair users, 2) for *power wheelchair users* it provides more appropriate device fitting and training with wheelchair control mechanisms, and 3) for *architects and designers* it provides structure visualization that can both improve the handicapped accessibility of building designs and test a structure for ADA compliance.
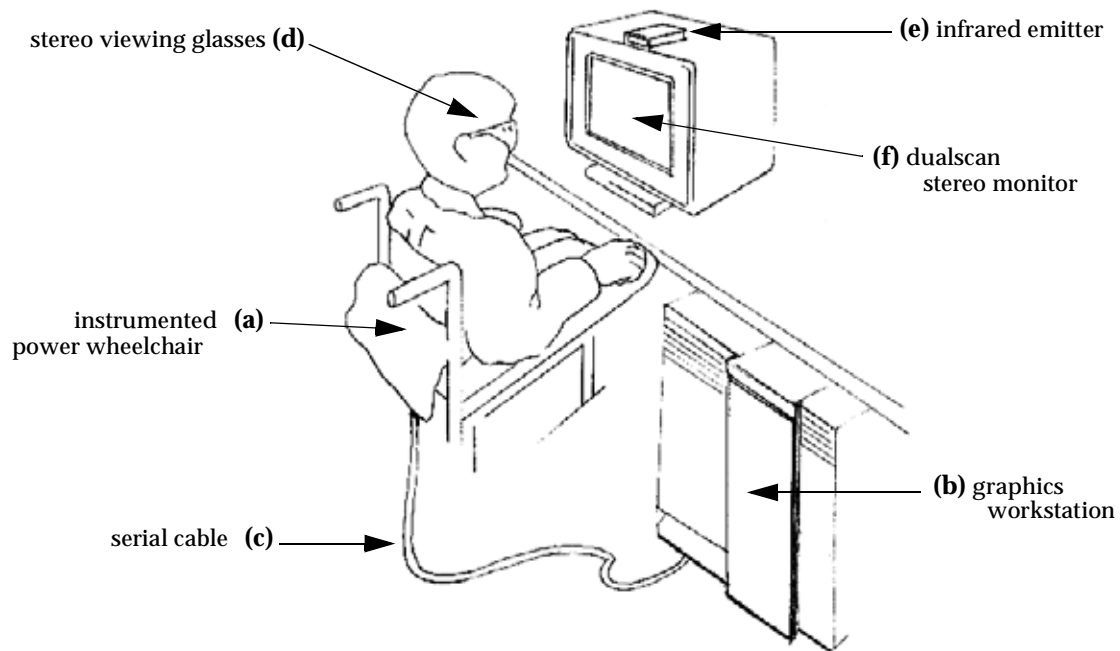
Figure 1: System configuration showing major hardware components.

2.0 System Description

2.1 Hardware Configuration

The hardware components of the system are displayed in Figure 1. The system consists of a) an instrumented power wheelchair, b) a graphics workstation, c) a serial cable, d) a pair of stereo viewing glasses, e) an infrared emitter (which synchronizes the stereo viewing glasses), and f) a dualscan stereo monitor (which alternately displays the scene from the viewpoint of each eye).

We are currently using an Action Power™ joystick-controlled wheelchair supplied by Invacare Corporation, a leading manufacturer of assistive devices. This model is instrumented with a computer interface that communicates through a standard serial cable. Our workstation is a Silicon Graphics IRIS Crimson VGXT™, with a 150 MHz CPU and 16 MB of internal memory. The monitor provides stereo vision through a pair of CrystalEyes™ polarized stereo viewing glasses, manufactured by Stereo Graphics.

2.2 User Interface

From the handicapped user's point of view the system's user interface is very simple. First the wheelchair is connected to the workstation with a serial cable, and then the user dons polarized stereo viewing glasses, selects an architectural scene, and drives through the virtual environment. A screen from a typical environment is shown in Figure 2. The system accurately simulates the dynamics of the particular wheelchair in the particular environment: the chair has the same speed and turning radius as it would in the physical world.

2.2.1 Wheelchair Control Mechanisms

The system easily supports chairs with different control mechanisms. The Action Power™ line of wheelchairs are fitted with a variety of controls that support users with different levels of disability. Among the controls produced by Invacare are a hand-operated *joystick*, a head-operated *chin control*, a *chest muscle actuator* which detects flexing and relaxing of the operator's chest muscles, a *halo* which detects the
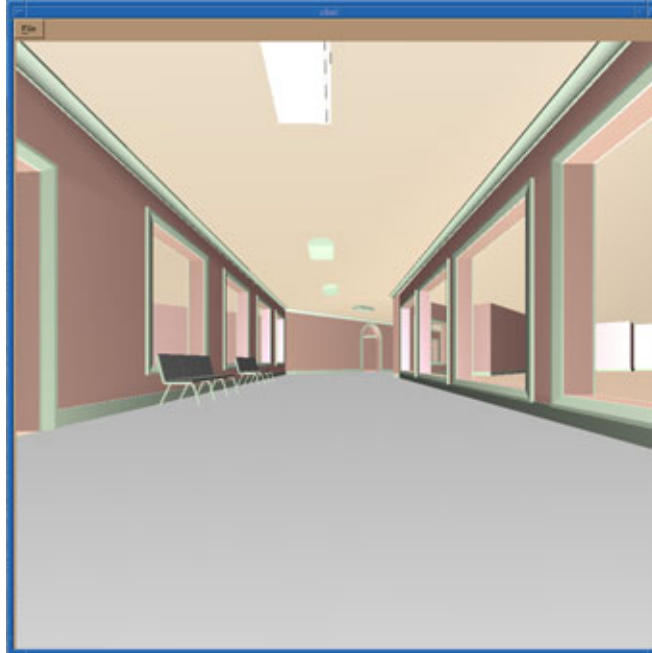
Figure 2: Typical interior scene rendered by the system.

tilt of the operator's head, and a *sip & puff* device, which consists of a straw that detects positive or negative air pressure from the operator's mouth.

When connected to a computer the wheelchair interface does two things: 1) it turns off the motors, so the chair does not actually move while connected to the system, and 2) it supplies the system with the speed and direction of rotation of each wheelchair wheel. Since the system always receives the same information regardless of a particular wheelchair's control mechanism, it can be used interchangeably with any properly instrumented power wheelchair. All the Action Power™ wheelchairs come equipped with a similar interface.

2.2.2 Stereo Viewing

Stereopsis is an important perceptual attribute of any virtual environment; perceptual scientists have long known that stereopsis is important in navigation [Gordon 89]. And [Drascic 93] discusses the importance of stereo presentation whenever precise spatial discrimination tasks are required. By providing stereo viewing our system improves the depth quality of its images.

Our system provides stereopsis through stereo viewing glasses and display hardware. The hardware alternately displays the scene rendered from the viewpoint of each of the observer's eyes. As the screen displays the scene from one eye, the glasses darken over the opposite eye. An infrared emitter on top of the monitor keeps the glasses in sync with the display hardware (Figure 1e); since the glasses are also battery powered they require no tether to the system.

In the future we plan to augment the stereo viewing glasses with another device that tracks the position of a small reflective spot. This spot can be affixed to the glasses, and provides the system with the distance of the user's eyes from the screen. This allows the system to more accurately compute the two viewpoints, and increases the sense of stereo depth.

Another popular method for providing stereopsis in virtual environments is a head mounted display (HMD); as HMD technology improves we will consider integrating one into our system. An HMD offers several advantages over stereo viewing glasses:

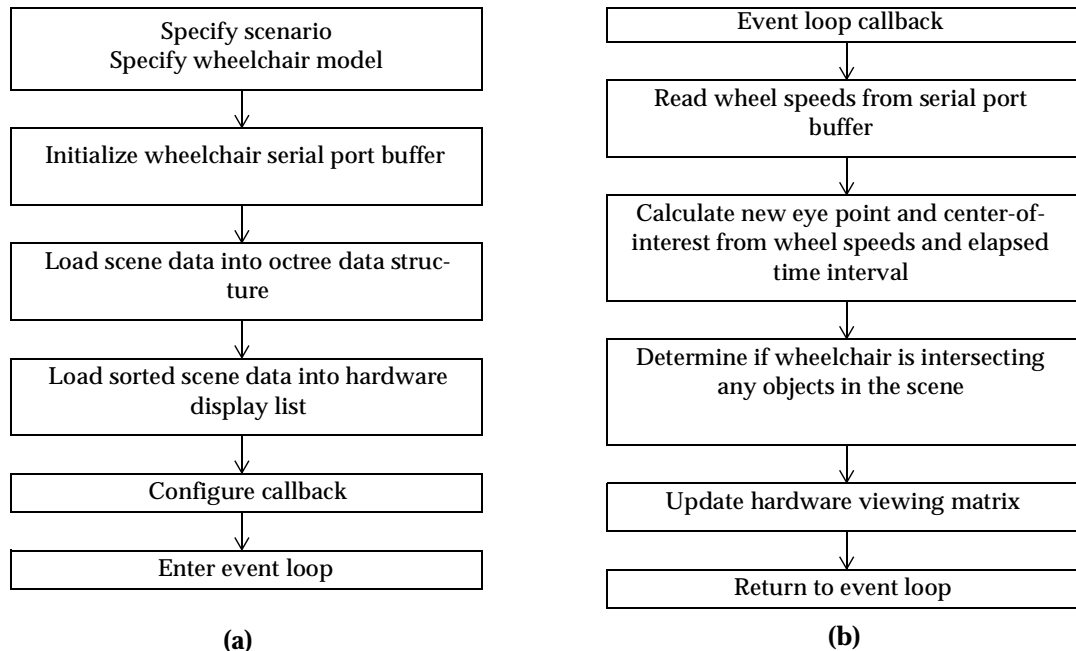- it provides a greater sense of immersion,

Specify scenario Specify wheelchair model		Event loop callback
Initialize wheelchair serial port buffer		Read wheel speeds from serial port buffer
Load scene data into octree data structure		Calculate new eye point and center-of-interest from wheel speeds and elapsed time interval
Load sorted scene data into hardware display list		Determine if wheelchair is intersecting any objects in the scene
Configure callback		Update hardware viewing matrix
Enter event loop		Return to event loop

(a)   **(b)**

Figure 3: Software architecture. **(a)** Initialization actions when the program is invoked. **(b)** Program actions each time the callback is invoked.

- it can accurately track the position and orientation of the user's head,

- it allows the user to look from side to side without turning the chair, and

- it can provide peripheral vision to help users orient themselves within a scene.

Despite these advantages, we have currently chosen to utilize the stereo glasses for their advantages over HMDs:

- stereo glasses are inexpensive and available off-the-shelf,

- current HMD models are expensive and cumbersome, especially those which support peripheral vision, and

- designing a computational model for an HMD that correctly simulates the display's optical properties is difficult and rarely achieved [Robinett & Rolland 92].

In contrast the stereo glasses are lightweight, are easy to don and remove, and provide an unencumbered method for stereo viewing without obtrusive equipment and tethers. [McKenna & Zeltzer 92] gives a detailed comparison of these and other display techniques available for virtual environments.

2.3 Software Description

The software component of the system is implemented on top of X Windows™ and Motif™ for the user interface, and SGI's IRIS™ Performer for rendering and scene database management. Figure 3 describes the general software architecture. Figure 3a shows the system's initial processing from the time the program is invoked until the time the event loop is entered. In this phase the user selects the particular scenario and wheelchair model that will be used. Next, communication is established with the wheelchair through the serial port. The scene data is then loaded into an octree data structure, which provides intersection testing (described in Section 2.3.1 below). Next, the spatially sorted data is loaded into the workstation's hardware display list. Finally, X Windows is configured with a callback that periodically returns control to the system.
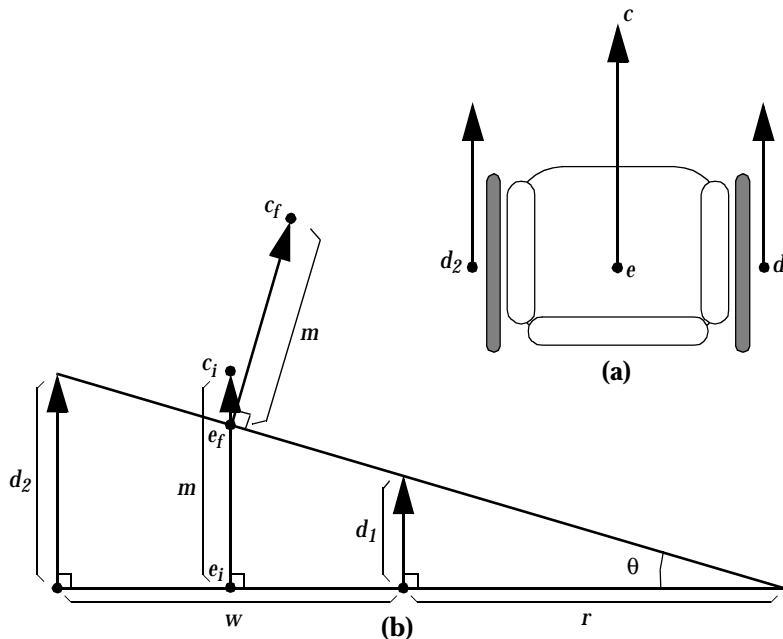
Figure 4: Geometry of wheelchair movement. **(a)** Top-down view of wheelchair showing eye point e, center-of-interest c, velocity of right-hand wheel d_1, and velocity of left-hand wheel d_2. **(b)** Calculation from initial eye point e_i and center-of-interest c_i to final eye point e_f and center-of-interest c_f

Figure 3b shows the processing that occurs each time X Windows invokes the system's callback. If a new message has arrived from the chair, the speed of each wheel is read from the serial port buffer. This data, along with the elapsed time since the arrival of the last message, is used to calculate a new eye point and center-of-interest for the rendered scene (described in Section 2.3.2 below). Next the system tests for intersections between the wheelchair and the rest of the environment. If there are no intersections the system updates the hardware viewing matrix with the new eye point and center-of-interest, and returns control to the event loop. The next time the rendering hardware traverses the display list, the scene is rendered from the new eye point looking towards the new center-of-interest.

If an intersection is detected, the eye point and center-of-interest are not updated; otherwise the user would drive through the intersected object. Currently the system issues a beep and jars the scene slightly to indicate a collision.

2.3.1 Spatial Subdivision into Octree Data Structure

As shown in Figure 3b, each time the user's eye point and center-of-interest are updated, the system performs intersection testing between the chair and the environment. This prevents the user from running through walls or other objects. A naive implementation detects an intersection by testing the chair against every polygon in the scene, yet the complexity of this operation would prevent the system from running in real time for even simple scenarios. Instead the system uses an N-objects octree data structure to reduce the number of polygons tested and hence speed collision detection. An octree is a common data structure in computer graphics; it is described in standard textbooks such as [Foley et al. 90]. An N-objects octree is a modification which recursively subdivides the scene until each octree node contains not more than N objects [Samet 90]. Details of the subdivision and collision detection algorithm may be found in [Swan 93].

2.3.2 Calculation of Wheelchair Movement

The calculation of the wheelchair movement is a basic dynamics problem; it can be solved with techniques from standard textbooks such as [Shames 60]. The geometry of the solution is shown in Figure 4. Figure4a

shows a view of the wheelchair from above; the wheelchair is traveling towards the top of the page. The position and direction of the wheelchair is represented by the eye point e and the center-of-interest c. Ten times per second the wheelchair sends the velocity of both wheels to the workstation, which travel the distances d_1 and d_2 during this time interval. If both wheels have traveled the same distance ($d_1 = d_2$), then the calculation of the new e and c is easy — we simply move them forward along the vector ec.

If $d_1 \nmid d_2$ then the calculation of the new e and c is more complicated, as shown in Figure 4b. Here the initial eye and center-of-interest positions are e_i and c_i, and the final positions are e_f and c_f; m is the distance between e_i and c_i. The distance the two wheels have traveled are again d_1 and d_2, and for this example $d_2 > d_1$. The width of the wheelchair base is w. We must now find the radius of curvature for the right-hand wheel r. Once we know this we can use simple trigonometry to find the turning angle q.

Since the wheels are both attached to the wheelchair, they have the same angular velocity, given by

$$\frac{d_1}{r} = \frac{d_2}{w+r} \ .$$

We solve this for r:

$$r = \frac{d_1 w}{(d_1 Ð d_2)} \ ,$$

and then determine the turning angle θ:

$$\theta = \operatorname{atan} \frac{d_2}{w+r} \ .$$

From this angle we can calculate the final eye position e_f. From e_f we construct a perpendicular vector of distance m to find the new center-of-interest c_f.

A disadvantage of this scheme is that the user's eyes and the chair direction always move together, which means a user can only orient themselves in a scene by turning the chair from side to side. If we integrate a head-mounted display into the system, we can enrich the dynamic model to allow the user to look from side to side without moving the chair.

3.0 Using the System

3.1 Evaluating User Proficiency

Our primary motivation for developing this system is the evaluation of user proficiency with a power wheelchair control mechanism. A memo we received from Invacare elegantly states the need:

> Presently, evaluation to determine whether or not a person can use a power drive wheel-chair is essentially nonexistent. The problem lies in the fact that all to often the selection of the chair and/or control mechanism is based on guesswork which can be very costly if the system does not enable the consumer to achieve the goal of independent mobility. At the same time, the lack of affordable simulators means that user training is limited to "trial and error" which can result in significant damage to expensive equipment and/or injury to the user [Invacare 93].

In collaboration with the Ohio State University (OSU) Gait Analysis Laboratory, the OSU Division of Orthopedic Surgery, and the OSU Children's Hospital, we plan to develop a clinical protocol that matches disabled patients to the appropriate type of power wheelchair controller. Once fully developed the system will be integrated into a broader Rehabilitation Engineering Research Center at the OSU Hospitals that deals with the quantification of physical performance.

Figure 5: Photograph of Joey using the system.

We expect this to have at least two benefits for wheelchair users: 1) They can be more quickly fitted with the appropriate type of controller, which should result in more affordable devices. And 2), they can train with the system to develop confidence and improve their responses to a new type of controller.

3.2 Architectural Considerations

Design standards such as [Raschko & Boetticher 82; Graphic Standards 89; ADA 91] can assist an architect in designing handicapped accessible spaces; the ADA is an example of a mandatory design standard. However, these standards refer to discrete spaces and cannot account for all of the permutations a building goes through under the hands of an architect or builder [Vanier 93]. Even when an architect thinks they have interpreted the standards relevant to their building correctly, and this building passes inspection, it can still be discovered that handicapped persons have difficulty navigating through the space. Our system addresses these shortcomings inherent in design standards.

Thus the system provides an architect with several capabilities: 1) It allows them to study the interaction between the different handicapped-accessible elements available from design standards, which can help in the design of an aesthetically pleasing structure. 2) Both the architect and the client can comprehend the visualization (as opposed to traditional 2D architectural plans, which are readily comprehended only by those with architectural training); the system allows the client to be an active partner in the design process. And 3) the architect can determine how easily a structure may be navigated with a power wheelchair, and thus test the structure for ADA compliance.

3.2.1 Subject Experience

We have some limited experience observing a handicapped person using the system: Joey, a 16-year-old with Cerebral Palsy, operated a prototype version (see Figure 5). We made two observations from this experience: 1) Joey was very uncomfortable in the power wheelchair which we have used to develop the system, as Joey's own chair contains customized padding. The best solution would have been to disconnect the joystick controller from our chair and temporary install it on Joey's chair, but we could not do this easily. This indicates the need to test controllers installed on users' own chairs. 2) Collision

detection was not yet implemented in the system, and when Joey discovered he could drive through walls he became quite excited. This demonstrates how VR applications can posses an engaging, game-like aspect that enhances their effectiveness as teaching tools.

4.0 Future Work

Enhanced Visualization. We intend to speed the image generation and provide more visual detail by using level-of-detail (LOD) management, where the environment is represented in varying levels of detail. For example, a detailed model of a lamp (i.e. composed of ~500 polygons) could be rendered when close to the eye point, while a simplified lamp model (i.e. composed of ~50 polygons) could be rendered when the lamp is far away. [Funkhouser and Séquin 93] discusses an elegant model for LOD management in an interactive virtual environment.

Fine-Grained Assessment of Environmental Accessibility.LOD management will also allow us to ask finer-grained questions about accessibility. For example, a high level of detail door might include the means by which the door must be opened (i.e. a door handle or push-plate), and a gloved interface might test whether the door is handicapped accessible. Yet clearly a virtual environment cannot represent every object to this level of detail and retain real-time performance, so LOD management is a prerequisite for testing fine-grained environmental accessibility.

Better Model Interaction. Currently the user can interact with a model by running into walls. We intend to enhance this with additional objects that allow interaction, such as elevators and ramps which take the user to other levels, smaller objects with dynamics properties (such as a lamp that might be knocked over), and moving objects such as other people and vehicles. (A motivating example is a busy airport terminal, full of hurrying passengers and baggage carts).

Additional Assistive Interfaces. The system can also function as a testbed for the exploration of additional assistive interfaces. Future plans include using a gloved interface to allow users to manipulate manual devices such as door handles, and investigating the use of voice recognition as an interface to assistive designs such as opening doors and delivering commands to elevators.

Enhanced User Feedback. The system could be enhanced to provide mechanical feedback to user actions. For example, the chair could be mounted on a platform that jars with a wall is encountered, or tilts when the user navigates a ramp.

Support for Manual Wheelchairs. Together with the above idea, hardware could be constructed to capture the wheel rotation of a manual chair. This would extend the system to evaluate manual wheelchair users. There are approximately 100,000 prescription manual wheelchairs sold annually in North America [Invacare 93], which is about four times the number of power wheelchairs sold.

5.0 Acknowledgments

References

[ADA 91] *ADA Accessibility Guidelines for Buildings and Facilities*, Federal Register, V 56, # 144, July 26, 1991.

[Drascic 93] Drascic, D., "Stereoscopic Vision and Augmented Reality", *Scientific Computing and Automation*, June 1993, 9(7), pp. 31–34.

[Foley et al. 90] Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F., *Computer Graphics, Principles and Practice*, 2nd ed., Addison-Wesley, 1990.

[Funkhouser & Séquin 93] Funkhouser, T. A., and Séquin, C. H., "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments", *Computer Graphics Proceedings SIGGRAPH '93*, Annual Conference Series 1993, pp. 247–254.

[Gordon 89] Gordon, Ian E., *Theories of Visual Perception*, John Wiley & Sons, 1989.

[Graphic Standards 89] *Architectural Graphic Standards*, R. T. Packard, AIA, and S. A. Kliment, FAIA, eds., John Wiley & Sons, 1989.

[Invacare 93] *Invacare Corporation*, Memo to Dr. Sheldon R. Simon, January 15, 1993.

[McKenna & Zeltzer 92] McKenna, M., and Zeltzer, D., "Three Dimensional Visual Display Systems for Virtual Environments", *Presence: Teleoperators and Virtual Environments*, MIT Press, Fall 1992, 1(4), pp. 421–458.

[Raschko & Boetticher 82] Raschko, and Boetticher, B., *Housing Interiors for the Disabled and Elderly*, Van Nostrand Reinhold Co, 1982.

[Robinett & Rolland 92] Robinett, W., and Rolland, J. P., "A Computational Model for the Stereoscopic Optics of a Head-Mounted Display", *Presence: Teleoperators and Virtual Environments*, MIT Press, Winter 1992, 1(1), pp. 45–62.

[Samet 90] Samet, H., *Applications of Spatial Data Structures*, Addison-Wesley, 1990.

[Shames 60] Shames, I. H., *Engineering Mechanics and Dynamics*, Prentice-Hall, 1960.

[Swan 93] Swan, J.E. II, "Octree-based Collision Detection with Fast Neighbor Finding", *OSU/ACCAD Technical Report*, OSU/ACCAD-12/93-TR7, December 1993.

[Vanier 93] Vanier, D. J., "Minicode Generator: A Methodology to Extract Generic Building Codes", in *CAAD Futures '93*, Flemming, U. and VanWyk, S., eds., Elsevier Sciences Publishers, 1993, pp. 225–226.