PRESENCE: Teleoperators and Virtual Environments; First Special Issue on The Application of Virtual Environments to Architecture, Building, and Large Structure Design, Volume 4, Number 3, Summer 1995, pages 297–305,© The Massachusetts Institute of Technology.

The Determination of Environmental Accessibility and ADA Compliance Through Virtual Wheelchair Simulation

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

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

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

Abstract:

The widespread use of power wheelchairs has greatly increased the requirements for accessibility of buildings and other architectural structures to handicapped persons. In addition, recent advances in microcomputer technology have made possible increasingly sophisticated power wheelchair interfaces, such as halo, puff and sip, and muscle control mechanisms, which can provide mobility for an even larger portion of the handicapped population. Finally, the ADA (the Americans with Disabilities Act of 1990) requires handicapped accessibility for (almost) all public structures.

We have developed a virtual structure prototyping system that allows navigation by a person using a power wheelchair. The system is a tool for three groups of people: 1) 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; 2) for *wheelchair users*, it provides more appropriate device fitting and training with wheelchair control mechanisms; and 3) for *health care professionals*, it provides evaluations of wheelchair users. The system consists of an instrumented, joystick-driven power wheelchair connected to a high-performance graphics workstation; it simulates the actual speed and maneuverability of the particular wheelchair within a virtual

structure. 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.

In this paper we discuss 1) the system's user interface, 2) the system's hardware and software configuration, 3) the impact of the system on the architectural design process, and 4) future system additions. In the last section we also discuss some implications of virtual manipulation for enabling technology.

1 Introduction

With the passage of the Americans with Disabilities Act of 1990 (ADA 1990), the U.S. Congress opened a new world of accessibility for many people who had previously been excluded. This law prohibits barriers to working, traveling, learning, and playing for disabled individuals at most public facilities. However, at the same time that the law has increased access for this group of people, it has also placed a tremendous burden on architects and designers to propose structures that comply with the accessibility regulations. The Architectural and Transportation Barriers Compliance Board has helped with this dilemma by publishing the ADA Accessibility Guidelines (ADA 1991), and organizations such as the American Institute of Architects have published guides dealing with accessibility issues [!!! reference for this?]. Many planners and architects consider these as minimal construction and alteration guidelines, which need to be expanded upon to be adequate.

Trial and error has traditionally determined whether following construction guidelines provides adequate accessibility for handicapped people. What is needed is a means of testing structures to determine the level of handicapped accessibility, and whether the designs are in compliance with the ADA. This paper describes our effort to provide such a testing mechanism. It uses the computational power of a modern computer graphics workstation, together with an innovative interface, to create a virtual visual environment representing an architectural design. In this environment a wheelchair user can navigate and maneuver and identify accessibility problems.

Several hundred thousand power wheelchairs are in use today, with approximately 23,000 new power wheelchair purchases each year (Frost & Sullivan, 1991). These statistics lend credibility to our decision to create a human-computer interface based on power wheelchair technology. Our system consists of an instrumented, joystick-driven power wheelchair connected to a high-performance graphics workstation; it simulates the actual speed and maneuverability of the particular wheelchair within a virtual structure. In addition to simulating the architectural structure and testing for ADA compliance, the system provides training for wheelchair users and evaluations of users for health care professionals. It will also provide experimental data that could be used to expand and modify the ADA guidelines to accommodate emerging adaptive technologies.

An interdisciplinary group of specialists from computer science, architecture, medicine, private industry, and the state of Ohio are working to ensure that the system is consistent with the needs and expectations of the various constituents in the project. At the Ohio State University, researchers from the Advanced Computing Center for the Arts and Design and the Ohio Supercomputer Center are collaborating with experts from the OSU Gait Analysis Laboratory and the Division of Orthopedic Surgery, as well as with Invacare, Inc., the manufacturer of the power wheelchair used with the system. Once fully developed, the system will be integrated into a broader Rehabilitation Engineering Research Center at the Ohio State University Hospitals that investigates the quantification of physical performance. Subsequent sections of this paper provide an overview of the system design and details of the implementation of the wheelchair and visualization interfaces. Implications of this implementation are discussed, as are future uses of this technology.

2 Hardware Configuration

The hardware components of the system are displayed in Figure 1. The system consists of

- an instrumented, motorized wheelchair,
- a graphics workstation,
- a serial cable, and
- a pair of stereo viewing glasses.

We are using an Action™ joystick-driven wheelchair supplied by Invacare Corporation, a leading manufacturer of assistive devices. This wheelchair 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.

3 User Interface

From the handicapped user's point of view, the system's user interface is simple. 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 moves with the same speed and turning radius as it would in the physical world.

The scene can be any architectural space, either the interior of a building or an outdoor setting. The system can handle the inclusion of common methods of changing levels or floors such as ramps or elevators.

3.1 Wheelchair Control Mechanisms

The system easily supports chairs with different control mechanisms. Invacare's power wheelchairs are fitted with a variety of controls that support users with different levels of physical disability. Each control specifies the wheelchair's speed and direction of travel. Among the controls produced by Invacare are

- a hand-operated *joystick,*
- a *chest muscle actuator,* which detects flexing and relaxing of the operator's chest muscles,
- a *halo,* which detects the 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.

Wheelchairs with any of these control mechanisms can be instrumented with a serial computer interface and used with our system. 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 simulator, and
2) it supplies the simulator 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, the system can be used interchangeably with any properly instrumented power wheelchair. All the Action Power™ wheelchairs come equipped with a similar serial computer interface.

3.2 Stereo Viewing

Stereopsis is an important perceptual attribute of any virtual environment; perceptual scientists have long known that stereopsis is important in perception and navigation (Gordon, 1989). And Drascic (1993) discusses the importance of stereo viewing whenever precise spatial discrimination tasks are required. By providing stereo viewing our system improves the dimensional quality of images that would otherwise contain only monocular depth cues such as occlusion, texture gradients, and shadows.

Our system uses stereo viewing glasses and display hardware, which are available as off-the-shelf components. The hardware alternately displays the scene rendered from the viewpoint of each of the observer's eyes. As the screen displays the scene as viewed 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.

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,
- it can accurately track the position and orientation of the user's eyes,
- 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 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 1992).

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 (1992) gives a detailed comparison of these and other display techniques available for virtual environments.

4 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 primary processing from the time the program is invoked until the time the event loop is entered. In this phase, the user selects the particular architectural scenario, as well as the exact wheelchair model. Next, the system establishes communication with the wheelchair through the serial port. It then loads the scene data into an octree data structure, which assists with both culling and intersection testing (described in Section 4.1 below). Next, the system loads the spatially sorted data into the workstation's hardware display list. Finally, it configures the window system with a callback that periodically returns control to the program.

Figure 3b shows the processing that occurs each time the window system invokes the program's callback. If a new message has arrived from the chair, the system reads the speed of each wheel from the serial port buffer. It uses this data, along with the elapsed time since the arrival of the last message from the chair, to calculate a new eye point and center of interest for the rendered scene (described in Section 4.2 below). Next, the system tests for intersections between the wheelchair and the rest of the environment. If there are no intersections with any objects in the scene, 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, it renders the scene from the new eye point looking towards the new center of interest.

If an intersection is detected, the eye and center of interest are not updated (otherwise the user would drive through the intersected object). Instead, the system issues a beep and jars the scene slightly to indicate a collision[!!! Don: info on visual feedback].

4.1 Spatial Subdivision into Octree Data Structure

As shown in Figure 3b, each time the user's eye 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 loads the scene data into an N-objects octree, which recursively subdivides the scene until each octree node contains not more than N objects (Figure 4). This ensures that only those polygons in close proximity to the chair are tested for collision. An N-objects octree has other benefits for intersection testing, such as quickly locating neighboring nodes (Samet 1990) and a spatially efficient subdivision of the scene; details may be found in Swan (1993).

The octree data structure also speeds the rendering pipeline. Before traversing the display list, the rendering hardware traverses the octree hierarchy and culls those nodes that do not intersect the viewing frustum. Since the octree structure is hierarchical, if a node lies completely in or out of the viewing frustum it is not subdivided and processed further; all of its sub-nodes and polygons are either marked as in the frustum (and subject to further processing in the rendering pipeline) or out of the frustum (and not processed further). Since a user's view of an architectural environment typically encompasses only a small fraction of the total number of polygons, this step greatly re-

PRESENCE: Teleoperators and Virtual Environments; First Special Issue on The Application of Virtual Environments to Architecture, Building, and Large Structure Design, Volume 4, Number 3, Summer 1995, pages 297–305,© The Massachusetts Institute of Technology.

duces the number of polygons rendered with each screen update. Thus, the octree subdivision is critical for real-time performance.

4.2 Calculation of Wheelchair Movement

The calculation of the wheelchair movement is a basic dynamics problem; it can be solved with standard methods (Shames 1960). The geometry of the solution is shown in Figure 5. Figure 5a 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 are 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. The wheels 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 • d_2, then the calculation of the new e and c is more complicated, as shown in Figure 5b. 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 θ.

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:

$$r = -\frac{d_1 w}{(d_1 - d_2)},$$

and then determine the turning angle θ:

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

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.

4.3 Architectural Data Processing

In our system, we begin with traditional architectural drawings (i.e. plan, section, and elevation) as a guide for modeling the building in the computer. We use the Macintosh software package Form-Z™ for this stage of the process. There are several reasons why we settled on Form-Z: 1) it is particularly flexible in its ability to build and modify the elements required for the development of an architectural space; 2) numerical inputs allow for accuracy; and 3) the ability to select individual elements of an object and then manipulate them, which makes modifications easy. Next, the individual architectural elements are saved as DXF files and then converted to a local format for use with the wheelchair simulator.

Treating each element as a separate file allows us to assign unique visual parameters such as color, material, texture, and reflective properties to each element. For example, all the floors in a building might visually simulate a gray carpet and the paneling a brown woodgrain.

5 Architectural and Design Considerations

5.1 The Use of Structure Visualization in Architectural Design

Computer visualization tools are beginning to revolutionize the way architects design structures. However, as discussed in Reynolds (1993, pp. 105–111), architects currently design almost exclusively through the use of 2D plans and elevations. Although axonometric views are sometimes used because of their ease of construction, these are aerial views and therefore have limited relevance. Thus, for the most part, an architect relies on intuition to determine the spatial correctness of a building, and where this fails, design guidelines or building codes. However, it is common for both to be insufficient.

Perspective views may be produced for client presentations, but typically this is done only after the project is finalized. Furthermore, properly drawn perspectives are time consuming and expensive to construct. Despite these drawbacks, perspective views allow the involvement of the many parties involved in an architectural project and not just those trained in reading 2D construction documents.

The usefulness of computer visualization is that it allows many 3D views to be set up quickly and easily. Furthermore it allows the study of color and texture in relation to form, space, and lights — an ability that hand-drawn perspectives lack. Finally, it allows the architect and the client to see the effect of navigating through a building.

5.2 The Use of Visualization and Simulation for Handicapped Accessibility

Design standards can assist the builder or architect in designing handicapped accessible spaces (Raschko & Boetticher 1982; Packard & Kliment 1989; ADA 1990); the ADA is an example of a mandatory design standard. However, guidelines and standards refer to discrete spaces and cannot possibly account for all of the permutations a building goes through under the hands of an architect or builder (Vanier, 1993). Even when architects think they have interpreted the standards relevant to their building correctly, and this building passes inspection, handicapped persons can still have difficulty navigating through the space.

Should architects limit themselves to tested, handicapped-accessible prototypical spaces, or should they continue attempts to interpret and apply building codes to achieve handicapped accessibility (and ADA compliance)? Artistic expression of architectural form and space would be lost if the responsible architect felt forced to use a predetermined library of handicapped accessible spaces, although in many instances this is exactly what transpires.

Furthermore, handicapped accessibility is often a last-minute accessory added to a completed structure; a common example is a seemingly cohesive building with a wheelchair ramp tacked on the front. Because architects will now, through the ADA, be forced to deal with this problem before construction, there is the opportunity for the aesthetic integration of ramps and other accessibility structures. Some buildings, such as the Wexner Center on the OSU campus (Peter Eisenman) and the Guggenheim (Frank Lloyed Wright) have integrated ramps into their designs. In both cases, the ramps provide an aesthetic and rewarding experience for all visitors (although the original intent of the architects mentioned may not have been to provide handicapped accessibility).

For home renovation, some of the "tacked on" elements may be impossible to avoid, since an essential element of residential design is a pragmatic and efficient use of space. Yet an unfortunate outcome of renovating a building for handicapped accessibility is considering only the pragmatics and not the aesthetic experience. The designer or architect cannot simply lift the plan of a handicapped accessible bathroom from *Architectural Graphic Standards* (Packard & Kliment, 1989) or insert unappealing ramps where there once were stairs. The relationship of all the elements as a whole must be considered.

Our visualization system addresses these concerns. It enables the designer to work directly with the client in developing the spatial relationships that are most effective. Furthermore, since the accessibility of new designs may be tested, it gives architects and designers a tool in addition to handicapped-accessible predefined organizations. Both the aesthetic value of a building as a work of art and the pragmatics of handicapped accessibility can be explored.

PRESENCE: Teleoperators and Virtual Environments; First Special Issue on The Application of Virtual Environments to Architecture, Building, and Large Structure Design, Volume 4, Number 3, Summer 1995, pages 297–305,© The Massachusetts Institute of Technology.

6 Implications for Assistive Environments

Along with these architectural and design considerations, the system has additional implications of virtual manipulation for the interaction of the disabled with enabling technology:

Performance Measurement. The system can measure the performance of a wheelchair user in a proposed setting. From our collaboration with the OSU Gait Analysis Laboratory, the Division of Orthopedic Surgery, and Children's Hospital in Columbus, we plan to develop a clinical protocol that matches disabled patients to the appropriate type of power wheelchair controller. Once patients have been fitted with new wheelchairs, the system can serve as a training tool to help them learn to operate the chair's controller.

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 discrete phrase recognition as an interface to assistive designs, such as opening doors and delivering commands to elevators. Voice-activated computer systems can certainly play a role in engaging assistive devices and as an interface for the disabled to interact with the environment and effect change.

Level of Detail Management. As the simulation becomes finer and finer grained, the level of detail of questions that can be answered increases. Eventually, our intention is for the system to support the representation of the environment at varying levels of detail. This would have two benefits. First, level of detail management can improve rendering speed. For example, a detailed model of a lamp (i.e., perhaps composed of ~500 polygons) could be rendered when the eye point is near, while a simplified lamp model (i.e., composed of ~50 polygons) could be rendered when the lamp is far away and subtends only a few pixels. Crow (1982) describes such a system, while Funkhouser and Séquin (1993) discusses an elegant model for level of detail management in an interactive virtual environment.

A second benefit of multiple levels of detail is finer-grained assessments of environmental accessibility. For example, a high level-of-detail representation of a door might include the means by which the door must be opened (i.e., a door handle or pushplate), 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 level of detail management is a prerequisite for the use of simulation to test this type of fine-grained environmental accessibility.

Acknowledgments

We would like to acknowledge Invacare Inc. for the Action™ power wheelchair, SGI for technical help and numerous equipment loans, Cynthia Hayes for her initial work in constructing the architectural data set, and Autodessys, Inc. for the donation of Form Z™. For advice, direction, and ideas we would like to thank Dr. Sheldon Simon from the Division of Orthopedic Surgery, Dr. Rosiland Batley of Children's Hospital, Kathleen Davey of the Center for Instructional Research, Judy Harris from the Ohio Technology Related Assistance Information Network, and J. B. Richey, Dave Williams, Hymie Pogir, Ted Wakefield, and Michael Devlin from Invacare. For careful editing we thank Barbara Woodall.

References

ADA Accessibility Guidelines for Buildings and Facilities (1991), Federal Register, V 56, # 144, July 26, 1991.

Americans with Disabilities Act of 1990 (1990), Public Law 101-336, July 26, 1990.

Crow, F.C. (1982), "A More Flexible Image Generation Environment", *Computer Graphics Proceedings SIGGRAPH '82,* Annual Conference Series 1982, pp. 9–18.

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

Frost & Sullivan (1991), *The U.S. market for Rehabilitation Equipment and Supplies*, Summer 1991.

Funkhouser, T. A., and Séquin, C. H. (1993), "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, Ian E. (1989), *Theories of Visual Perception*, John Wiley & Sons.

Mäntylä, M. (1988), *An Introduction to Solid Modeling,* Computer Science Press: Rockville, Md.

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

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

Raschko, and Boetticher, B. (1982), *Housing Interiors for the Disabled and Elderly*, Van Nostrand Reinhold Co.

Reynolds, R. A. (1993), *Computing for Architects*, Butterworth-Heinemann Ltd.: Oxford, Boston.

Robinett, W., and Rolland, J. P. (1992), "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, H. (1990), *Applications of Spatial Data Structures*, Addison-Wesley.

Shames, I. H. (1960), *Engineering Mechanics and Dynamics*, Prentice-Hall.

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

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

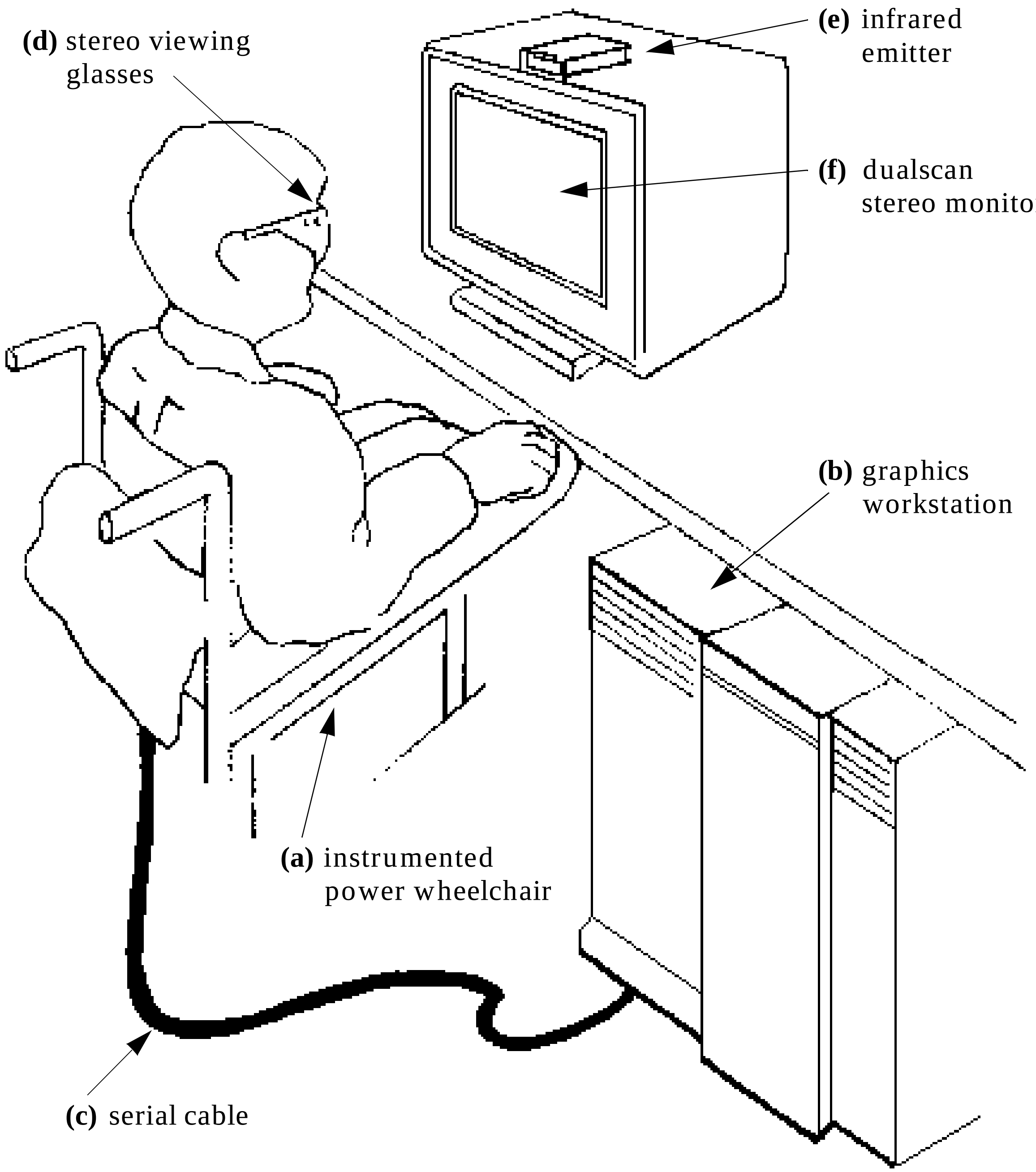

Figure 1: System configuration showing major hardware components. **(a)** Instrumented, power wheelchair, **(b)** graphics workstation, **(c)** serial cable, **(d)** stereo viewing glasses, **(e)** infrared emitter (synchronizes stereo viewing glasses), **(f)** dualscan stereo monitor (alternately displays the scene from each eye point with each refresh cycle).

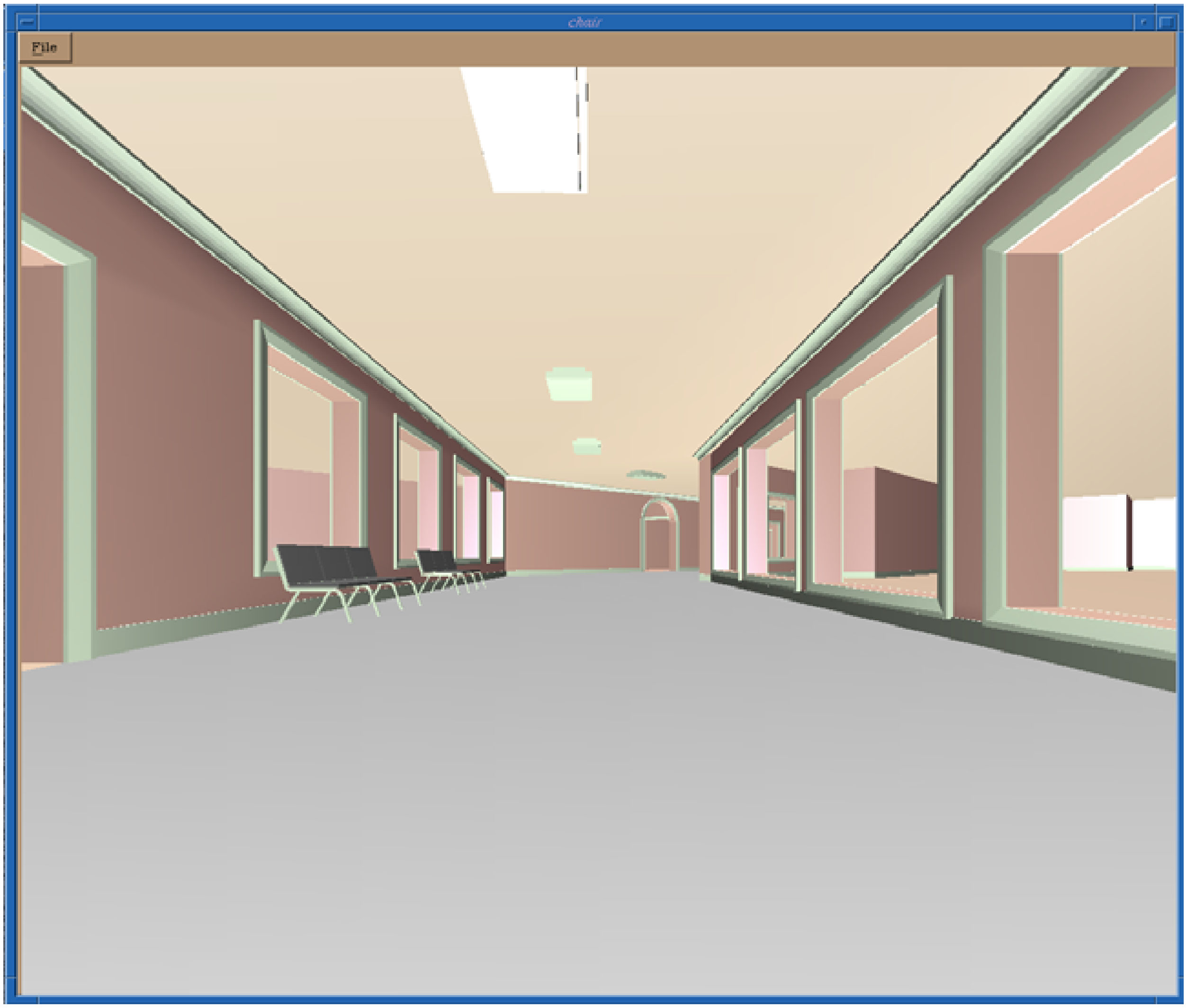

Figure 2: Typical interior scene rendered by the system.

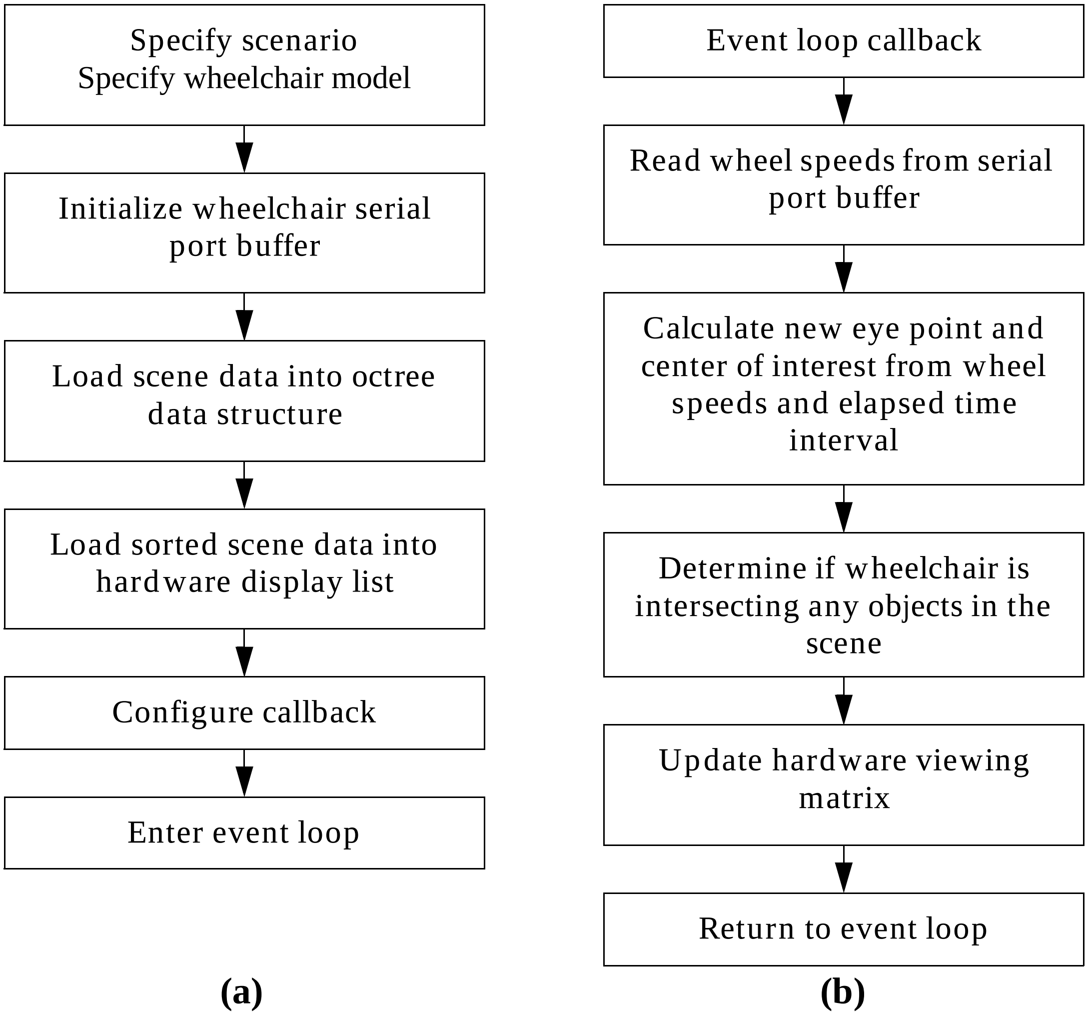

Figure 3: Software architecture. **(a)** Initialization actions when the program is invoked. **(b)** Program actions each time the callback is invoked.

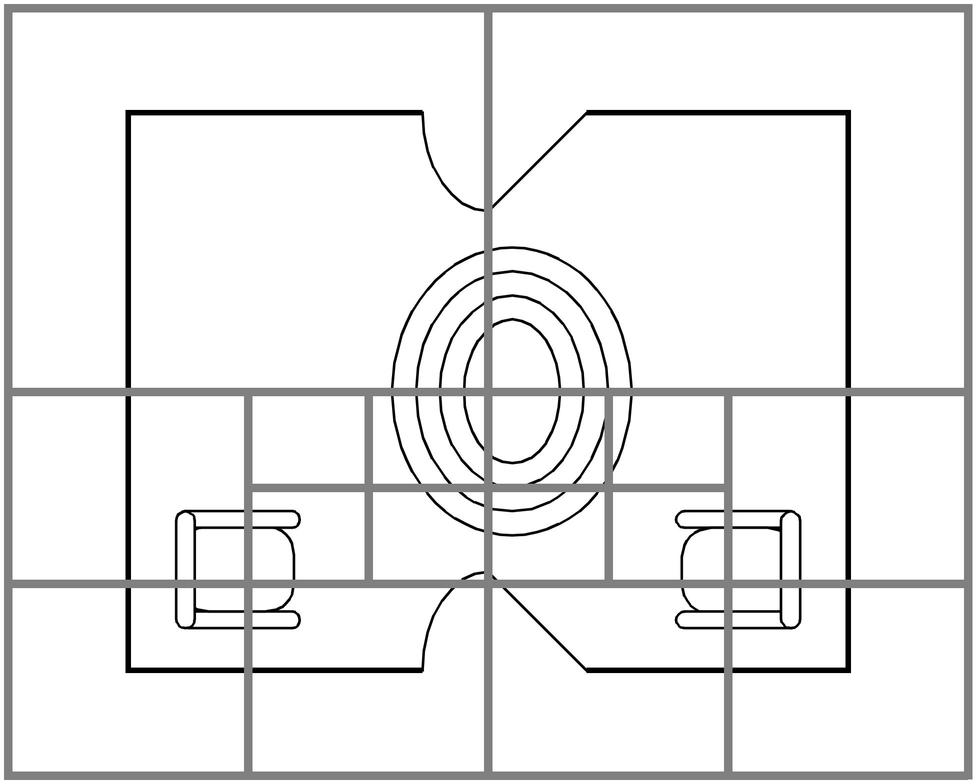

Figure 4: A N-objects quadtree subdivision of an architectural floor plan; a quadtree is the two-dimensional analogue of a three-dimensional octree. Shown is a top-down schematic of a room containing two chairs and an oval rug. Each closed shape above is considered a polygon. For this example N = 5; subdivision continues until each node contains not more than 5 polygons. For more information see Mäntylä (1988) and Samet (1990).

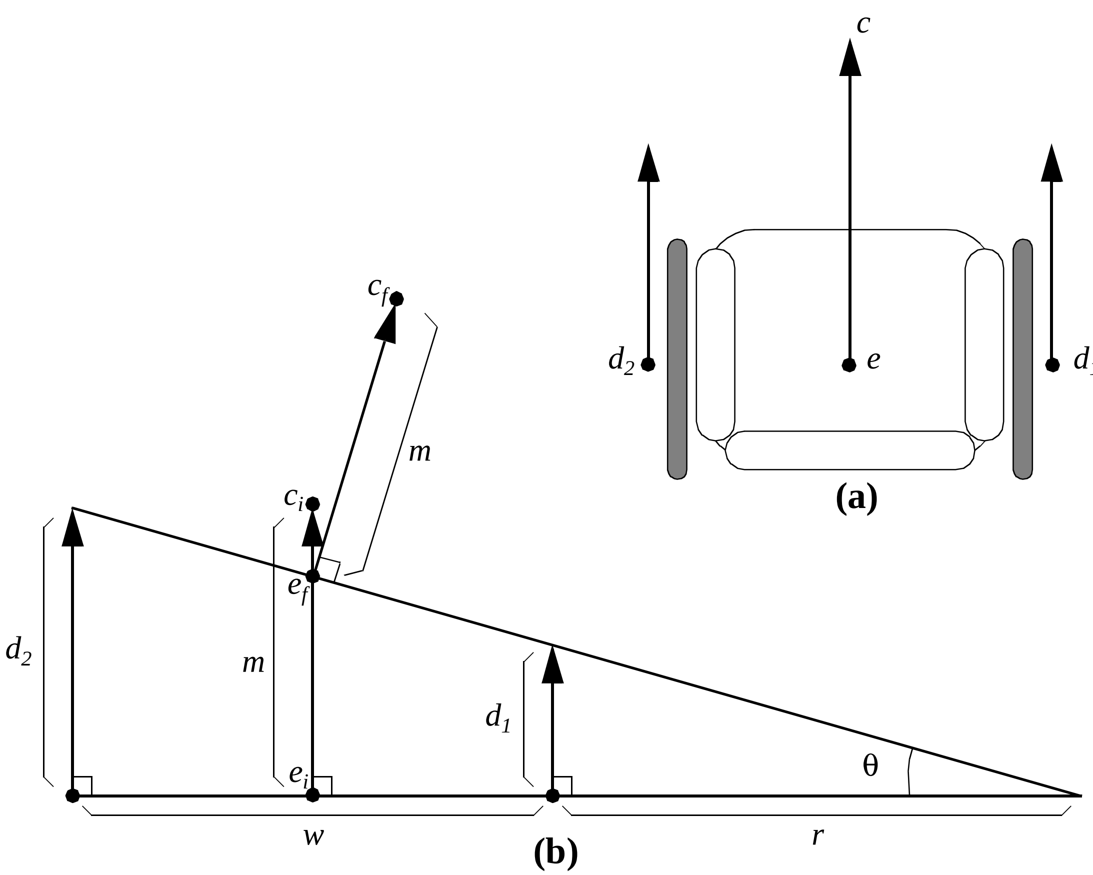

Figure 5: Wheelchair dynamics. **(a)** Top-down view of wheelchair showing eye point *e*, center of interest *c*, velocity of right-hand wheel d_1, and velocity of left-hand wheel d_2. **(b)** Calculation from initial eye point e_i and center of interest c_i to final eye point e_f and center of interest c_f.