A Platform for Gaze-Contingent Virtual Environments

Robert Danforth and **Andrew Duchowski** and **Robert Geist** and **Elizabeth McAliley**
Department of Computer Science, Clemson University
{*bobd* | *andrewd* | *rmg* | *emcalil*}*@cs.clemson.edu*

Abstract

This paper describes hardware and software requirements for the development of a gaze-contingent virtual reality system which incorporates several cues to presence. The user's gaze direction, as well as head position and orientation, are tracked to allow dynamic level-of-detail changes for rendering. Users can see themselves, rather than representations thereof, within blue-screened virtual environments, and limited vestibular feedback is provided through a motion simulator. The aesthetic appearance of environments is driven by advanced graphical techniques (i.e., radiosity) motivated by the goal of photorealistic representation of natural scenes. Taken together, the components identified in this paper describe a platform suitable for development of a variety of "smart" virtual environments.

Introduction

The extent to which subjective telepresence enhances human performance remains undetermined, largely due to difficulties in delivering and measuring varying degrees of subjective presence. Nevertheless, the conjecture of enhanced performance has been sufficiently compelling to motivate researchers to seek both techniques for providing subjective presence and techniques for measuring it (Schloerb 1995). Of particular interest are systems for operator training on tasks where mistakes could incur tremendous costs. Examples include tele-inspection of highly radioactive sites (Geist *et al.* 1997) or simulation of interplanetary surface exploration.

Real-time traversal of photo-realistic, virtual environments with augmented cues to presence, such as vestibular, haptic, and proprioceptive feedback, remains a challenging task. Although numerous researchers have provided solutions to component problems, to our knowledge an integration of component solutions to provide an operational training system that can meet such challenges has not yet occurred.

The purpose of this paper is to describe such a system, which has been under development at Clemson University for several years. User gaze directions, as well as head position and orientation, are tracked to allow dynamic level-of-detail changes for rendering. The dynamic 3D Point Of Regard (POR) is stored for post-immersive examination of the user's overt spatio-temporal focus of attention in the environment. Users can see themselves, rather than representations thereof, within the virtual environments, and limited vestibular feedback is provided through a motion simulator.

Hardware Platform

Our primary rendering engine is a dual-rack, dual-pipe, SGI Onyx2® InfiniteReality™ system with 8 raster managers and 8 MIPS® R10000™ processors, each with 4Mb secondary cache.[1] It is equipped with 3Gb of main memory and 0.5Gb of texture memory.

Multi-modal hardware components include a binocular ISCAN eye tracker mounted within a Virtual Research V8 (high resolution) Head Mounted Display (HMD). The V8 HMD offers 640×480 resolution per eye with separate left and right eye feeds. HMD position and orientation tracking is provided by an Ascension 6 Degree-Of-Freedom (6DOF) Flock Of Birds (FOB), a d.c. electromagnetic system with a 10ms latency. A 6DOF tracked, hand-held mouse provides the user with directional motion control. The HMD is equipped with headphones for audio localization.

An rs232-controlled motion simulator, powered by compressed air, provides brief bursts of acceleration toward specified attitudes. The simulator is a single-person seat which allows up to 60 degrees of pitch and roll control. The seat position is controlled by the host system through a serial port. A snapshot of a user (author McAliley) engaged in VE traversal is shown in figure 1.

Eye Tracking

Interest in gaze-contingent interface techniques has endured since early implementations of eye-slaved flight simulators and has since permeated several disciplines including human-computer interfaces, teleoperator environments, and visual communication modalities (Jacob 1990; Starker & Bolt 1990; Held & Durlach 1993). The functional benefits of eye tracking for human-computer, multi-modal interfaces, and the technical benefits for data compression, have been recognized, but the benefits have yet to be fully exploited in real-time traversal of virtual environments.

[1] Silicon Graphics, Onyx2, InfiniteReality, are registered trademarks of Silicon Graphics, Inc.

Figure 1: User Interface

In our system, a dedicated PC calculates the POR in real-time (60Hz) from left and right video images of the user's pupils and IR corneal reflections (first Purkinje images). Figure 2 shows a user wearing the eye tracking HMD. Eye

Figure 2: Eye Tracking HMD

images captured by the cameras can be seen in two video monitors near the lower right of the figure. Presently, it appears that the binocular eye tracker coupled with an HMD capable of vergence measurement in VR is the first of its kind to be assembled in the United States. Although binocular eye trackers integrated with HMDs have previously been proposed (Ohshima, Yamamoto, & Tamura 1996), no reports of their actual construction or operation have been found.

The calculation of vergence depends on only the relative positions of the two eyes in the horizontal axis. The parameters of interest here are the three-dimensional virtual coordinates, (x_g, y_g, z_g), which can be determined from traditional stereo geometry calculations. Figure 3 illustrates the basic binocular geometry. Helmet tracking determines both helmet position and the (orthogonal) directional and up vectors, which determine viewer-local coordinates shown in

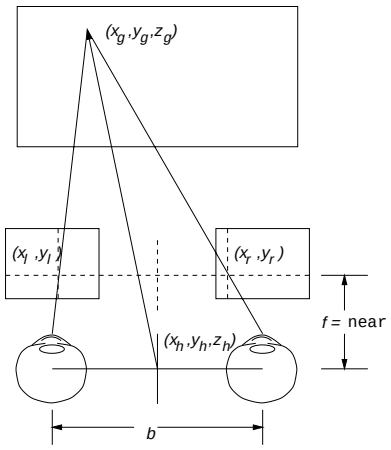

Figure 3: Basic Binocular Geometry

the diagram. The helmet position is the origin, the helmet directional vector is the optical (viewer-local z) axis, and the helmet up vector is the viewer-local y axis.

Given instantaneous, eye tracked, viewer-local coordinates (x_l, y_l) and (x_r, y_r) in the left and right image planes, at focal distance f along the viewer-local z axis, we can determine viewer-local coordinates of the gaze point, (x_g, y_g, z_g) as:

$$x_g = s(x_l + x_r)/2 \qquad y_g = s(y_l + y_r)/2 \qquad z_g = sf \qquad (1)$$

where $s = b/(x_l - x_r + b)$. Adding these to the helmet position as offsets along the viewer-local axes, we have the gaze point in virtual world coordinates.

The derived three-dimensional gaze point serves as either a real-time or a post-immersion diagnostic indicator of the user's overt focus of attention. The collection of gaze points taken over the course of immersion, the user's *scanpath*, serves as a diagnostic tool for post-immersive reasoning about the user's actions in the environment. Figure 4 shows a user's 3D scanpath in a simple virtual environment.[2] In this case the user's task was simply to wander about the room and inspect the "artwork" hanging on the virtual walls. In more task-specific environments, e.g., training, scanpath information can be used to compare experts to novices and thereby evaluate the effects of training. As a real-time interface modality, the point of gaze addresses imprecision and ambiguity of the user's viewpoint in a virtual environment by explicitly providing the 3D location of the user's point of regard. In our gaze-contingent system, we are working towards using the real-time gaze vector to dynamically alter the level of detail of surfaces with intricate geometry, e.g., a virtual terrain.

Rendering

While the Onyx2 platform and its OpenGL programming interface provide excellent graphics performance, they do not

[2]Currently the scanpath coordinates are closely correlated with head location. We are working towards the specification of focal and disparity parameters to give us clearer depth information of the gaze point, dissociating the head position from the point of gaze.

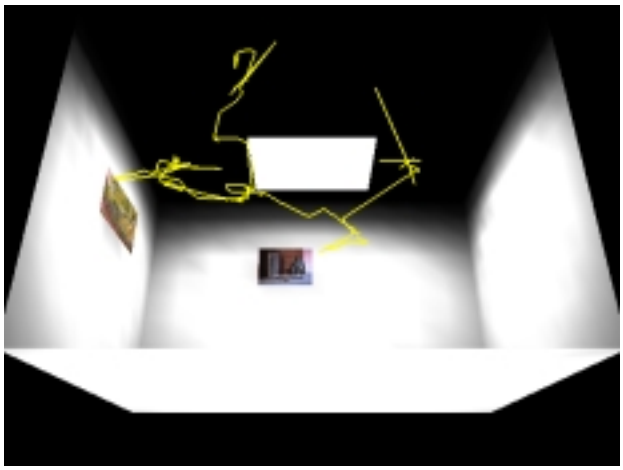

Figure 4: 3D Scanpath in a Virtual Environment

support a global illumination model, and such is essential if we are to effect photo-realistic virtual environments. The constraint of real-time display precludes sophisticated treatment of specular information, but it allows global illumination effects through the ambient and diffuse components. A suitable rendering equation, in terms of radiance, is thus of the form

$$c = B/\pi + k(H \cdot N)^m \tag{2}$$

where B denotes radiosity (exiting irradiance), computed through a classical formulation of environment patch interaction, and the remainder is a layered, first-order specular component, where N denotes surface normal and H is a unit vector halfway between the vector pointing to the light source(s) and that pointing to the viewer's eye position.

Radiosity Radiosity-based illumination is a well-studied topic. An excellent treatment may be found in (Cohen & Wallace 1993). As first observed by Heckbert and Winget (Heckbert & Winget 1991), the classical radiosity equation can be regarded as simply a finite element solution of Kajiya's rendering equation (Kajiya 1986) for the special case of Lambertian surfaces. Every object in an environment is specified in terms of discrete patches. The radiosity (exiting irradiance) of patch i, B_i, is given by:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j \frac{A_j}{A_i} F_{ji} \tag{3}$$

where E_i = emission of patch i, F_{ji} = form factor from j to i, A_i = area of patch i, ρ_i = reflectivity (bi-hemispherical reflectance) of patch i, and n = number of discrete patches. A matrix series formulation,

$$B = (I - diag(\rho)F)^{-1}E = \sum_{k=0}^{\infty} (diag(\rho)F)^k E \tag{4}$$

is particularly useful for an implementation based on a finite number of lighting passes.

Radiosity-based techniques are essential for photo-realistic representations of interior environments where higher-order illumination effects are significant. Radiosity techniques provide a view-independent solution for ambient and diffuse illumination, and this allows real-time environment traversal. A photograph of a real test environment and a virtual replica thereof (from (Geist *et al.* 1997)) are shown in figure 5.[3] For exterior environments, where there

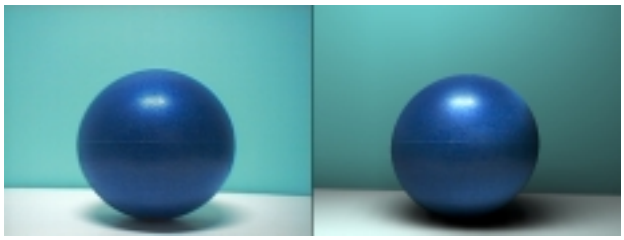

Figure 5: Real (left) and Synthetic (right) Test Environments

is limited need for higher-order illumination effects, texture mapping alone may suffice.

Texture Mapping For interior surfaces that are to be texture mapped, the radiosity computation must reflect the texture's approximate reflectivity to other patches. We use per-patch averages of the textures for this computation. At the end of the radiosity preprocessing phase, when we attach the textures to the surfaces, we must then modify the radiated patch colors so that the application of the texture blend function will yield the correct result. This amounts to replacing patch exiting irradiance with incident irradiance, $(B_i - E_i)/\rho_i$.

For exterior environments, where radiosity computations are often unnecessary, nominal textures must often be modified to include appropriate shadowing. Raytracing simple elevation maps produces shadow maps that can easily be blended with nominal textures prior to application. Mipmapping (multi-level texture mapping) is essential to avoid dynamic aliasing.

Glint Mapping An elementary specular component can be added to surfaces exhibiting specular reflection by a technique we term *glint mapping*. It is similar, in spirit, to environment mapping (Blinn & Newell 1976). Specifically, to each surface/light-source pair we associate a glint map, which is a simple texture map showing a radially isotropic cosine brightness function with maximum brightness at the center, coordinates (0,0). Upon user motion, we find, for each surface vertex, the unit vector, H, halfway between the vector to the light source and that to the new eye position. Each vertex is stored with a fixed pair of orthogonal surface tangent vectors, and the dot product of H with these tangent vectors provides the texture coordinates used in attaching the glint map to the surface. Note that if these dot products are both zero, then H is aligned with the normal to the surface, and so we should have maximum specular addition, which is exactly what we find at texture coordinates (0,0).

[3] There are visible differences in these images; as explained in (Geist *et al.* 1997), the virtual environment was created automatically from limited radiometric information.

Level of Detail For environments containing significant topological detail, such as virtual terrains, rendering with multiple levels of detail, where the level is based on user position and gaze direction, is essential to provide an acceptable combination of surface detail and frame rate. Recent work in this area has been extensive. Particularly impressive is Hoppe's view-dependent progressive mesh framework (Hoppe 1998), where spatial continuity is maintained through structure design, and temporal continuity is maintained by *geomorphs*.

Our approach is comparatively simple, but still reasonably effective. A surface with significant topological detail is represented as a quadrilateral mesh, which is divided into fixed-size (number of vertices) sub-blocks. Rendering for level-of-detail is then carried out on a per-sub-block basis. From a fully-detailed surface, lower levels of resolution are constructed by removing half of the vertices in each direction and assigning new vertex values. The new values are averages of the higher resolution values. Resolution level is chosen per sub-block, and it is based on viewer distance. The resolution level is not discrete; it is interpolated between the pre-computed discrete levels to avoid "popping" effects.

Techniques for incorporating additional attenuation of resolution, based on gaze, are under development. Here we must measure distance from the central gaze ray of the view volume. Peripheral vision is particularly sensitive to motion, so this additional attenuation of resolution must be limited to high-level surface detail.

In figure 6 we show a snapshot from a traversal of a synthetic Martian terrain. In figure 7 we show the corresponding

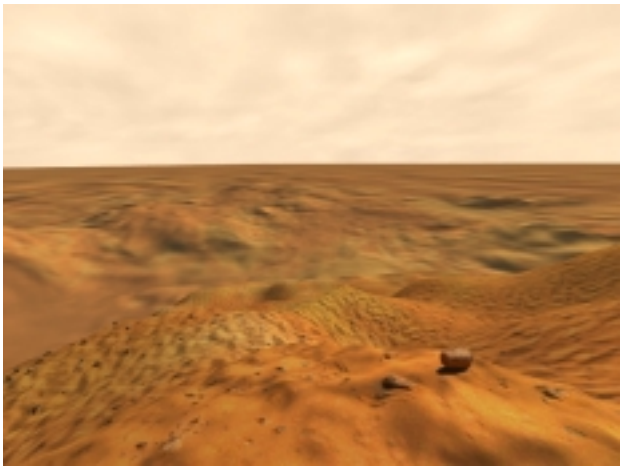

Figure 6: Synthetic Martian Terrain

wireframe image. The rocks were rendered by bill-boarding, i.e., images of rocks from the Pathfinder mission to Mars (see: <*http://mars.jpl.nasa.gov*>) were rendered onto 2D transparent planes that rotate to maintain an orientation orthogonal to the viewer.

Feedback

The most obvious, and yet most overlooked, cue to presence is the simple ability to see oneself within the environment

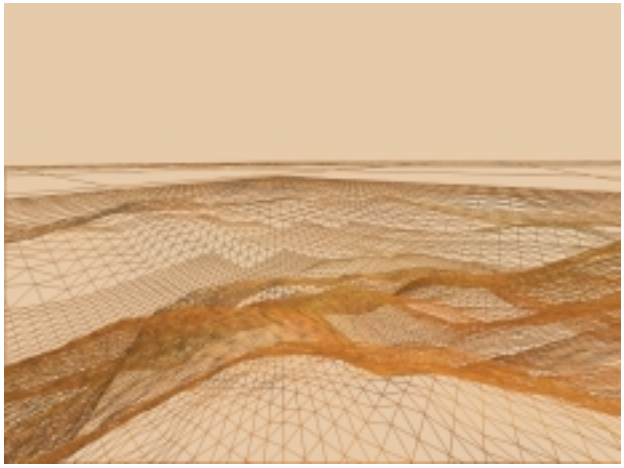

Figure 7: Wireframe for Synthetic Terrain

(proprioceptive feedback). Those systems that do provide some image of the user typically render a stylized arm or hand over which the user has some control, possibly through magnetic or infrared tracking of the user's real arm or hand.

In (Slater 1997), Slater notes that immersion requires a self-representation in the VE—the Virtual Body (VB). The VB is both part of the perceived environment, and represents the being that is doing the perceiving. Formal experiments to test this type of proprioceptive requirement for immersion have not yet been conducted or are just beginning. Nevertheless, we conjecture that the inclusion of a natural image of one's self will lead to a significant performance enhancement on a number of tasks.

In an award-winning paper (Van Pernis 1999), Van Pernis described a simple technique to achieve this inclusion of a self-image. A forward-facing camera is mounted on the HMD and is attached to a video capture card. The user's physical environment is then draped in blue-screen material and a chroma-key extraction is performed on non-blue pixels from the camera image stream. The extracted stream is then alpha-blended with the contents of the frame buffers containing virtual images. Thus the user literally sees himself or herself standing or sitting in the virtual environment. No registration of the "digital self" in the VE is performed. However, when a Flock Of Birds receiver is held or otherwise attached to the user's hand, the user is able to naturally manipulate virtual objects. The interaction precision is limited by the relatively coarse granularity provided by the single FOB receiver.

Although as yet we have made no attempt to provide for virtual occlusion (e.g. moving a real hand behind an opaque virtual object) the effects are fairly dramatic. A snapshot of a real user's hand engaged in sculpting a virtual NURBS surface is shown in figure 8.

Of all cues to presence, vestibular and haptic feedback are probably the most difficult to achieve. Although mechanisms have been built, their success has been limited to extremely narrow problem domains, e.g., rudder manipulation on aircraft. Nevertheless, within our virtual environments

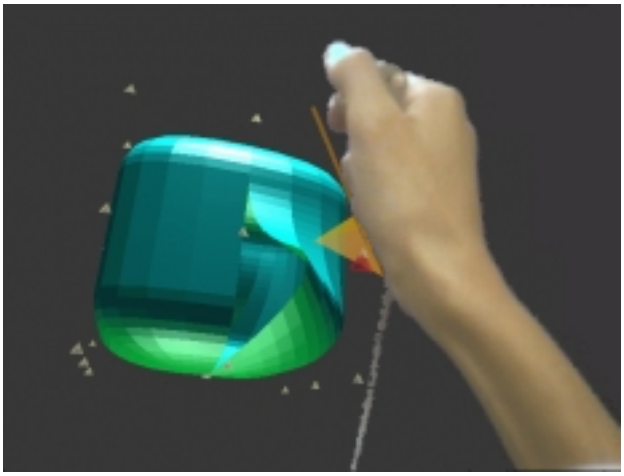

Figure 8: A Real Hand in a Virtual Environment (Van Pernis 1999)

we face the problem of meta-motion: users are restricted, by magnetic field radius and helmet cord length, to a physical radius of approximately 6 feet. Our original solution to this problem was to provide a second, hand-held, tracked device that could be used to indicate desired direction/speed for virtual flight within the virtual environment. The effects of physical motion during virtual flight were additive, but virtual flight with feet firmly planted on the laboratory floor was not a realistic experience for most users.

As shown in figure 1, we now use an rs232-controlled motion simulator that is powered by compressed-air and provides brief bursts of acceleration toward specified attitudes. In addition to providing the simulated acceleration, the mechanism removes the laboratory floor as a point of reference for the user, and we conjecture that this alone contributes significantly to immersion.

Conclusions

We have described an operational platform for real-time traversal of photo-realistic virtual environments. The platform is based on high-end graphics engines and an electromagnetically tracked, binocular helmet equipped with infrared eye tracking capability. Rendering software includes an integrated approach to radiosity, texture mapping, and glint mapping to realize an illumination model of the form given in equation (2). Tracking software delivers helmet position and orientation in real-time, which can be used directly to provide updated images to the screens in the binocular HMD.

User gaze direction is tracked in real-time, and we are in the process of integrating this tracking information with the helmet tracking, as described in section 3, to provide gaze-contingent level of detail in rendering.

Users can see themselves in virtual environments by use of a forward-facing camera and a blue-drape of the physical environment, and motion is simulated by use of a compressed-air powered, single-user seat.

Unfortunately, although we have developed an extensive capability to deliver sensory cues to effect immersion and presence, there is little evidence to point to which cues are necessary or sufficient to accomplish this task. Controlled studies of human performance in the presence of differing collections of such cues are crucial to the future development of VR technology, and such studies are long overdue.

Acknowledgements

This work was supported in part by the MRI Program of the National Science Foundation under award CDA-9724271 and by the U.S. Dept. of Energy under award DE-FG07-96ER14728.

References

Blinn, J., and Newell, M. 1976. Texture and reflection in computer-generated images. *CACM* 19:542–547.

Cohen, M. F., and Wallace, J. R. 1993. *Radiosity and realistic image synthesis*. Cambridge, MA: Academic Press Professional.

Geist, R.; Schalkoff, R.; Stinson, T.; and Gurbuz, S. 1997. Autonomous virtualization of real environments for telepresence applications. *PRESENCE: Teleoperators and Virtual Environments* 6(6):645–657.

Heckbert, P. S., and Winget, J. M. 1991. Finite element methods for global illumination. Technical Report UCP/CSD 91/643, Computer Science Division (EECS), University of California, Berkeley.

Held, R., and Durlach, N. 1993. Telepresence, time delay and adaptation. In Ellis, S. R.; Kaiser, M.; and Grunwald, A. J., eds., *Pictorial Communication in Virtual and Real Environments*. London: Taylor & Francis, Ltd. 232–246.

Hoppe, H. 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *Proc. IEEE Visualization 1998*, 35–42.

Jacob, R. J. 1990. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*, 11–18. ACM Press.

Kajiya, J. T. 1986. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proc.)* 20:4:143–150.

Ohshima, T.; Yamamoto, H.; and Tamura, H. 1996. Gaze-Directed Adaptive Rendering for Interacting with Virtual Space. In *Proceedings of VRAIS'96*, 103–110. IEEE.

Schloerb, D. 1995. A quantitative measure of telepresence. *Presence* 4:64–80.

Slater, M. 1997. A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments. *Presence* 6(6):603–616.

Starker, I., and Bolt, R. A. 1990. A Gaze-Responsive Self-Disclosing Display. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*, 3–9. ACM Press.

Van Pernis, A. P. 1999. Surface construction from within a virtual environment. In *Proc. of the* 37^{th} *Annual ACM Southeast Conf.*