Rapid Modeling and Design in Virtual Environments

Abstract

We discuss the evolution and current status of investigations at our Graphics, Visualization, and Virtual Environments Laboratory. The progression of user interface technology has led us from display systems evaluation, to performance studies of graphics workstations, stereoscopic and head-mounted displays, user-input devices, and interactive techniques for virtual environments. We have been involved in prototyping applications for terrain visualization, situation awareness for command and control, maintenance procedures training, and interactive design. Our current emphasis is on developing techniques for rapid modeling of virtual environments, to support mission planning, rehearsal, and interactive design and visualization. We discuss in detail three examples of these applications, and we present a set of guidelines that we have found to be useful for quickly constructing effective virtual environments.

1 Introduction

Over the past 7 years, virtual environment technology (VET) has come to be recognized as a valuable, vital, and expanding field of study. During the same period, MITRE has evolved a Graphics, Visualization and Virtual Environment Laboratory. The goal of the laboratory is to build the infrastructure and experience necessary to support interactive visualization applications for the Air Force, its sister services, and government agencies. Virtual environment laboratories in the defense community generally emphasize simulation and training (IST, 1995; Stytz, Block, & Soltz, 1993; Zyda, Pratt, Falby, Barham, & Kelleher, 1993). Our efforts complement that work by focusing on design, command, and control systems. This effort began as evaluation and tradeoff studies of component technologies: displays, input devices, and software products. From there the effort has grown into development of a variety of prototypes for potential applications, both near-term and long-range (Breen, 1992).

Virtual environment technology now permits the creation of natural, often intuitive, user interfaces for synthetic models. Though in its infancy, the technology already provides quality workstation images, three-dimensional (3D) stereoscopic displays, realistic 3D sound output, as well as voice and postural input to synthetic environments. VET is appropriate today for many diverse applications. The market offers several modeling packages for object design, input devices for interactive applications, rendering systems that support “walk-through” functionality, and output devices harnessing the powers of the human eyes, ears, and touch. However, much of the time required to develop an application is spent in the modeling of objects. One of our goals is to reduce the time required to adapt virtual environments to new applications. We will describe our approach to this problem for applications in which modeling detail initially is not the primary requirement.

We will first review areas of investigation and laboratory facilities. Then, we will describe projects within the laboratory. Three examples from our current research will be discussed in detail, along with guidelines for rapid modeling.

2 Areas of Investigation

This section discusses the scope of work carried out at the laboratory. First we provide an overview of our work in evaluating technology in relation to our sponsor’s needs. Then we describe the resources avail-

Georges G. Grinstein¹
and
David A. Southard
The MITRE Corporation
Bedford, MA 01730-1420

¹ Also with the Institute for Visualization and Perception Research, University of Massachusetts at Lowell, Lowell, MA 01854.
able to projects in the laboratory. Finally, we review the projects in which we are involved, and which have evolved into virtual environment work.

2.1 Technology Evaluations

The laboratory is involved principally in technologies relating to graphics workstations, displays, and user-interface devices.

2.1.1 Graphics Workstations. The Department of Defense's increasing emphasis on utilization of commercial off-the-shelf (COTS) equipment led us to evaluate graphics workstations, and graphics standards, for military applications. Early studies evaluated computational and graphics performance of GKS and PHIGS for situation displays on multiple graphics workstations (Leger & Conway, 1989; Conway, Freiter, & Leger, 1990). Surprisingly, COTS workstations and standard-based graphics usually fell short of performance requirements. The trend uncovered by the two studies, however, revealed that 3D graphics, as well as the traditional 2D displays, would soon be feasible in military applications.

2.1.2 Displays. Our work evaluating cathode-ray tube (CRT) displays for contrast and resolution characteristics (O'Callaghan & Veron, 1989) eventually led to VET studies. We evaluated several types of stereoscopic displays, based on active screen-mounted and wearable liquid-crystal optical switches (Veron, Southard, Leger, & Conway, 1990a). These results prompted us to build our own large screen stereoscopic display, using dual projectors and passive polarizers. This approach eliminated ghosting due to phosphor decay and liquid-crystal switching.

When this technology is applied to terrain visualization (Veron, Hezel, & Southard, 1990a), certain viewing artifacts became evident. The artifacts were due to the software algorithms for stereoscopic projection. This led us to develop a new geometrical projection algorithm that used an accurate model of the viewing geometry, and takes into account human visual system constraints (Southard, 1992).

A survey of head-mounted display technology determined significant limitations in the state-of-the-art for immersive displays (Hezel & Veron, 1993; Veron et al., 1994a). Liquid-crystal display (LCD) systems, for example, suffer from poor resolution. CRT-based systems offer much higher resolution, but lacked color, and had restricted fields of view. For many applications, we still prefer large-screen stereoscopic displays.

2.2 Resources

The laboratory contains a number of resources for investigating virtual environment technology and developing applications. The primary computing platform is a Silicon Graphics Inc. (SGI) Onyx/2 RE², configured with the multichannel option (MCO). This workstation is used for system development, prototyping, and demonstrations. The secondary platform is a SGI Power Series 4D340/RE, also configured with MCO, a Calcomp digitizing table, and a Canon CJ-10 color scanner/printer. This workstation is used for model development, and as a secondary platform to explore collaborative virtual environments.

The multichannel option allows us a great deal of flexibility in choosing display devices. Until recently we used a large-screen stereoscopic front-projection display, which we constructed from dual Electrohome 4100 video projectors and passive linear-polarizing sheets. The passive linear polarizers afforded two advantages: ghosting due to long phosphor decay as exhibited with time-multiplexed active liquid-crystal polarizers is eliminated, and low-cost polarizing glasses are widely available for stereoscopic presentations to large groups. For most of our work we found it convenient to drive these displays using a time-multiplexed 1280 × 492 stereoscopic video mode. However, we could double the resolution and brightness of the stereoscopic display by using the 1280 × 1024 multichannel mode. The MCO also provides the capability of recording and displaying stereoscopic material using the VREX projection system and normal NTSC video equipment, as well as various head-mounted displays.

We recently replaced our dual-projector installation with an Electrohome Marquee-8000 "retro" rear-projec-
tion unit. This new system uses a time-multiplexed 1280 × 492 120Hz video mode in conjunction with StereoGraphics CrystalEyes active shutter eyewear for stereoscopic viewing. The brightness and fast phosphor-decay characteristics of the projector unit mitigate ghosting experienced with previous active-shutter systems. The rear-projection system substantially increases the brightness and off-axis viewing quality, permitting group demonstrations even in office ambient lighting.

Over the years, we have tested many of the user-interface devices that are available. Our current selection of devices has proved to be useful and reliable. The Spaceball is used for viewpoint control. For whole-hand input, we use the CyberGlove. In many cases, however, we find that a glove can be replaced by the Logitech 3D mouse. We often use two-handed input, in various combinations, such as left and right gloves, glove and 3D mouse, Spaceball and 3D mouse, and so on. Our prototype systems require only a limited number of posture commands: point, designate, and grab. Most users find it easier to issue these commands using the mouse buttons than with hand postures. The acoustical tracking technology used by Logitech is quite accurate and cost-effective. For more complex tasks requiring tracking of multiple objects, we use the Ascension Flock of Birds. Verbal commands are effected by the Voice Navigator system for the Macintosh, via a serial-line connection to the SGI workstation.

2.3 Projects

This section discusses the evolution of various virtual environment-related prototypes that have been built at the laboratory. Each prototype progresses to include more sophisticated concepts of presence and interaction.

2.3.1 Terrain Visualization. Our first prototype was a 3D terrain visualization, built in 1989, using a Silicon Graphics IRIS 4D80/GTX workstation (Veron et al. 1990a; Veron, Southard, Leger & Conway, 1990b; Southard, 1991a). This platform allowed smooth shading of terrain scenes; however, texture and antialiasing were not supported at that time. Some techniques were developed to improve the efficiency and accuracy of terrain models (Southard 1991b). The scenes contained from 10,000 to 15,000 polygons per scene. Most of these polygons were devoted to portrayal of the terrain surface. The scenes could be rendered stereoscopically at 3 to 4 frames per second. The user could move his viewpoint and "fly" anywhere in the scene, which covered an 10,000 km² area. At that time, comparable real-time scenes were being rendered only by computer image generators, such as used in flight simulators, albeit at higher frame rates. Even so, the images generated were impressionistic at best. With the rendering rates possible at that time, the interactive element was limited.

A small follow-on effort upgraded the workstation processor to the IRIS 4D240/GTX model, and added some fog and haze models, time-of-day lighting models, and insets of LANDSAT imagery (Southard, 1991c). The latter was only partially successful, as a true texturing capability was not yet available. The image insets were modeled as polygonal meshes. The image resolution of LANDSAT was not detailed enough for satisfactory visualization of low-level flight. However, once again the trends indicated that workstations would be able to render detailed terrain models within a few years.

2.3.2 Maintenance Trainer. The next initiative, during the summer of 1991, went under the title Virtual Intellectively Realized Task-oriented User Environment, or VIRTUE prototype (Mitchell & Segal, 1992). The goal of the project was to simulate a single maintenance procedure, such as might be required for a satellite repair in space. The task required unfastening an access panel secured with four latches, removing the access panel, thus exposing a cavity, grabbing a circuit card, placing the circuit card into the cavity, then, replacing and fastening the access cover. VIRTUE allowed each object to determine which gestures it recognized, and how to respond to each one. This project presaged the use of virtual reality for repair of the Hubble Space Telescope, which was later fulfilled by workers at Lockheed (Hancock, 1993a,b).

2.3.3 Object-Oriented Architecture. Our third VR prototyping effort, in 1992, focused on developing a robust object-oriented software architecture that would support intelligent object behaviors, highly inter-
The main concepts developed with this prototype are an object-oriented system architecture and user interaction models. A unique feature of this application is the integration of a knowledge-base, which handles interpretation and response to the user’s queries, and directs the behaviors of the simulated vehicles.

At a recent Air Force Association conference we demonstrated a large screen rear-projection stereoscopic display system, using active liquid-crystal shutter glasses from StereoGraphics, as part of a proposed virtual reality command and control system (Cooper, 1994). An advantage to this setup is that it is visible under normal ambient lighting, which makes virtual environment applications in command centers and offices possible.

2.3.5 Battle Simulation. Another aspect of this application is the “what if” scenario. In this case, it is desirable to set up simulations with several different initial conditions, then to view a simulation at faster-than-real-time speeds. Conversely, if too many things are happening at once, the user may wish to slow down the scenario, so that all the object interactions can be observed. The current-generation prototype visualizes a simulated war-gaming exercise, using symbolic representations for aggregate military units (Erbacher & Southard, 1995).

2.3.6 Power-Plant Control Panel Emulation. A full scale replica of an electrical power utility control room panel has been developed to train plant operators (Grinstein 1993; Rayson, Masterman, & Fray, 1994). The completed facility will be reconfigurable for different power plants, and supports multiple users, multiple hands, full scale, high resolution, color instrument replicas, no device tethering, and real-time interaction support. Funding for this and continuing similar activities originated with the Electric Power Research Institute.

3 Applications

In working with our sponsors, we saw that virtual environments could be applied to high-priority, short-duration projects, such as training and rehearsal sce-
scenarios, if the environments could be designed quickly enough. However, in many cases meticulous off-line modeling of objects is required before the virtual environment can be entered. We felt there was a need to develop techniques for rapidly modeling complex environments, so that people could see results within the virtual environment quickly. We also began to recognize a need for VET not only as visualization, but also part of the design process itself. Virtual environments should be used for more than simply visualizing and exploring models. They should be interactive enough to allow the user to make design modifications while inside the virtual environment.

For these reasons, we are pursuing two avenues of investigation: rapid modeling and interactive design within virtual environments.

### 3.1 Rapid Modeling

Though one might assume that models ought to have enough fidelity to approximate reality throughout, in practice few applications require high fidelity in every aspect. Many parts of a representation can fully meet their requirements with relatively low resolution. For example, with personnel rescue training scenarios, which require rehearsal within 24 to 72 hr, the user needs high resolution stairways, ducts, and similar environmental objects, but can accommodate lower resolution building exteriors. Likewise, in scene and building design, criteria can be explored to permit greater coarseness of representation in areas where detailed decisions have not yet been made. In both examples, the goal is to increase understanding in a short period of time.

### 3.2 The Model Shop Environment

We have implemented a concept for interactive design called the virtual Model Shop. The Model Shop consists of a set of tools and procedures for rapid modeling, together with interactive design software, termed the Virtual Interactive Planning system, or VIPs (Wingfield, 1995). VIPs has proved to be a productive environment for experimentation with 3D user interfaces. The system features a large-screen stereoscopic display, spatially located stereophonic sound, with input from voice commands, gloves, trackers, a 3D mouse, and a Spaceball.

The user-interface employs an electronic catalog. The catalog has “pages” that can be selected to show various collections of objects. The user is able to select the appropriate objects from the catalog, and to “pull” them into the scene being designed. The objects have properties that allow them to be manipulated and customized by the user. The sizes can be adjusted, and multiple objects can be “snapped” together to form complex constructions.

Figure 1 shows VIPs used in a prototype application for designing a Marine mobile command post. The tent's top and two sides have been removed to simplify editing. A communications module has been pulled from the catalog and is about to be placed in the tent. An information panel at the top keeps track of electrical power requirements and the total weight of the equipment inside the tent. The tent automatically updates the information panel as the user experiments with different layouts and combinations of equipment. This feature allows the user to determine, for example, power, air conditioning, and structural requirements for the installation site. The prototype illustrates how the interactive design environment could be coupled to software such
as databases, spreadsheets, and schedulers, which supply amplifying information to the user.

4 Case Studies

We have built several prototypes to explore design issues. An environment is built in two major steps: building the models and integrating them into the virtual environment. VIPs allows us to experiment with building a complex environment from simpler components. Therefore, providing a complete set of components is on the critical path to any application. We will use three example applications to illustrate our rapid modeling approach. These projects are typical of a large class of problems that can use rapid modeling.

- **Design of a surgical operating room.** With an eye toward supporting ARPA’s Operating Room of the Future project, as well as other health care design issues, we completed a rapid conceptual design of a virtual operating room. This project demonstrates the conversion of detailed CAD models into a set of models suitable for interactive virtual environments.

- **Design of a public place.** As a proof-of-concept exercise, we undertook the task of rapidly modeling a large public building, so that people could easily compare a virtual environment with a familiar real one (cf. Fredrickson, 1994; Brooks et al., 1992). We chose a local shopping center, the Burlington Mall, as a convenient example. This exercise determined how fast we could develop an interactive real-time walkthrough, giving the user a high sense of presence. The goal was to concentrate the modeling effort on details that would enhance the sense of presence, and not on time-consuming details that would not be noticed.

- **Crisis mission planning.** We recently participated, along with the Defense Mapping Agency, in an exercise that tested the ability to create a simulation-based 3D visualization of a hypothetical crisis location. The objective was to provide a mission drive-through from a ground-level viewpoint, and to identify the limitations of the terrain data production and modeling technologies, all within a 72-hr time span. MITRE’s role was to provide models and an interactive 3D visualization of a selected terrain area rapidly within 24 hr.

4.1 Operating Room of the Future

In this application we used the MultiGen modeling tool. To develop a scene in VIPs, models of the component objects must be created and imported, typically as a MultiGen “Flight” format file. The initial model, supplied by Harvard’s Graduate School of Design, represented a monolithic operating room, complete with tables, monitors, lights, and other details. This model was provided as an AutoCAD file. Using MultiGen’s CAD option, we were able to read AutoCAD files and convert them to a Flight file. To create an interactive operating room in the Model Shop, we had to separate the tables, monitors, lights, and other objects into individual files. Then VIPs is able to integrate the new models with existing models from the VIPs object catalog.

4.1.1 Data Conversion. The original CAD model contained over 7000 polygons. This complexity exceeds what typically can be animated at interactive frame rates. Using a series of simplification techniques, we were able to reduce the polygon count to approximately 600, while maintaining the model’s general appearance.

The first simplification was to reduce the number of vertices in polygons used to model curved or circular shapes. The simplified polygons were then triangulated, because triangles can be rendered more quickly than complex polygons. Objects optimized in this manner include the doorknobs and lights. Similarly, cylinders used to model swivel armatures, each of which originally had 24 sides, were replaced with 10-sided cylinders, using smooth shading to produce the same visual effect (cf. Blinn, 1989).

We then reduced excessive detail. A telephone, for example, had fully modeled three-dimensional buttons. Since the display screen does not normally have the resolution to render three-dimensional effects for very small objects, the buttons were replaced by simple rectangular polygons on the surface of the phone. Another way we
reduced the complexity was to use MultiGen’s combine faces command to combine contiguous coplanar polygons that had similar color. We also deleted redundant overlapping coplanar polygons.

Next, we removed repeated objects. Separate files were created for the room’s monitors and lights, permitting the user to bring up the catalog and instance as many monitors or lights as needed.

4.1.2 Texturing. Perhaps the greatest opportunity for simplification is to substitute textures for complicated geometry. Textures add realism to a model without increasing the number of polygons. For example, the original operating room model had two air vents, each louver modeled in splendid detail. Each of the vents was replaced with a single polygon textured with an appropriate image; thus 80 faces were replaced by one. Textures also were used for the clock, walls, and floor tiles.

Finally, to use the operating room model interactively, we had to give the user the power to see into the room and place objects in it. We had two options: either modify the model so the user can issue a command and cause the ceiling and any two walls to disappear, or delete those surfaces from the model. Given the deadline of 2 days, we chose the latter.

4.1.3 Time Required. Our simplification of the AutoCAD model in MultiGen took one person 2 days, plus a little more than half a day to incorporate the new Flight models into the VIPs environment. Figure 2 shows the operating room in the VIPs environment. The scene shows a light that has been instanced from the catalog, and interactively placed into the room. The light swivels, and the user can visualize different lighting situations. In Figure 3, the user is placing monitor displays. The user is able to experiment with different room arrangements, and visibility of the monitors from different parts of the room.

4.2 Burlington Mall

The purpose of the Burlington Mall project was to show how a detailed model could be created quickly, using readily available resources. We were able to create this model without detailed plans, such as blueprints. For this demonstration, our “walkthroughs” use perfly, a program provided by Silicon Graphics to illustrate the use of the IRIS Performer library, a high-performance rendering toolkit.
4.2.1 Modeling the Surroundings. To provide an environment for the mall model, we scanned a National High-Altitude Photography (NHAP) infrared image, available from the U.S. Geological Survey, of a 1.25-square mile area surrounding Burlington, Massachusetts, which encompasses the real shopping mall. Infrared imagery normally shows vegetation in shades of red, so we simulated a visual image by exchanging the red and green channels. The image was scanned at 400 dots per inch, then cut into four tiles, each $256 \times 256$ pixels. All five images were sharpened, and the color balance adjusted using software from GBA Inc. It is important at this stage to make sure that all images are enhanced consistently, so that they will match-up later.

The aerial images are used as textures for the ground surface. The first texture is applied to a 1.25-square-mile polygon as a backdrop. The next four textures are applied to four subfaces of that polygon to provide higher resolution as the viewpoint moves closer to the ground. For purposes of our prototype, the terrain is modeled as a flat surface, with the texture providing some illusion of three dimensionality.

Levels of detail (LODs) provide a mechanism for simplifying objects in the distance, which do not require fully-detailed rendering. This technique helps to maintain a desired steady frame rate, by keeping the rendering load as light as possible. As the viewpoint approaches the mall, the user sees only the lowest level of detail, namely, the aerial photograph, which contains an image of the real mall. As one gets closer, the exterior of the mall switches in first, followed by the roads, the parking lot, and the interior of the mall. The switching distance is entered into the Flight file using MultiGen. A way of determining the switching distances is to zoom in on a LOD until the object appears unsatisfactory, then create a higher resolution LOD that switches into view just before that distance is reached. Since the switching distances are calculated based on the geometric center of each LOD model, we made copies of a large transparent polygon for each LOD so that their centers would be the same.

Figure 4 shows the merged aerial image and mall model. As the viewpoint approaches the building, additional detail fades in. Figure 5 shows a view of the mall entrance.

4.2.2 Acquiring Model-Specific Textures. Generic textures, collections of which are available from commercial vendors, were applied to the roads, parking lots, exterior walls, and roof. An experimental technique for this project was the acquisition of model-specific textures, such as storefronts, from video sources. Video allowed us to capture a large area with a sweep of a cam-
camera, instead of having to take numerous photographs. We shot the videotape on a morning when the mall is opened before the shops themselves open. We stood directly in front of each store while taping. A brief side-to-side sweep creates a visual record that helps to place each viewpoint's location in context, which helps in the modeling later. Permission should be obtained from the appropriate security office prior to taping, as such behavior might be viewed suspiciously.

To digitize frames from a videotape, we used the camcorder to play the video into a PC equipped with a VideoBlaster frame grabber. Video technology saves time here, as well, because there is no need to wait for film processing, printing, and scanning of the images. The frames were saved in a 24-bit Targa format. Each frame occupies about 1.3 megabytes. This PC unfortunately did not have a network connection, so we transferred the selected images to our modeling machine via floppy disk. A network connection would certainly have streamlined the processes.

Once the frames were moved to the workstation, the images were cropped, resized—usually to 256 × 256 pixels—sharpened, color-balanced, and imported into MultiGen. We used the MultiGen texture editing tools to warp the textures when it seemed the camera was at an angle to the subject: This provided us with the capability of straightening out perspective views of building and storefronts.

4.2.3 Modeling the Building. Since we did not have the blueprints for the facility, the mall’s exterior dimensions were estimated from the images. Interior dimensions were estimated by counting the square-foot tiles on the floor and walls. Measurements can also be made using an ultrasonic range finder. In the future, it might be possible to alter the camcorder to obtain distance to the subject from the automatic focus apparatus. Approximate dimensions were also taken from the directory of stores “You Are Here” map available to the general public.

Standard architectural features are represented, wherever possible, in the simplest way. Columns and plants, for example, are represented by billboards. Billboards are a rendering trick, supported by MultiGen and IRIS Performer, that turns the model always to face the viewpoint during rendering. Thus, objects that exhibit some cylindrical symmetry can be represented by a single polygon that always faces the viewer. This technique is highly effective in monoscopic perspective views; however, it is much less effective with stereoscopic displays, where the added perception of depth reveals these features as obviously flat. High-fidelity model views were provided in central locations and near hall intersections.

Storefronts are modeled as single, textured rectangles marked off at each end with a pillar. Any storefronts that were missed on the video were approximated by borrowing the texture of a similar shop, with an overlay polygon featuring the store’s name.

Our RealityEngine configuration can support only up to 1024 × 1024 pixels of texture while rendering a scene. If more textures are required within the field-of-view, the machine will swap-in textures from disk. Such swapping slows performance drastically. Therefore, the storefronts are modeled using two LODs: a generic storefront for the lower LOD, and the actual storefront texture for the higher LOD. Thus the viewer can look down a hallway of “many” storefronts with no performance penalty.

The last objects to be added to the model were the minor furniture: waste containers, plants, and individual vendor carrels and kiosks along the main walkway. Some objects are beyond present capability; for example, although the mall uses numerous mirrors, the multiple reflections they create cannot yet be rendered in real time.

Figure 6 shows a view on the ground floor of the mall. An individual can be seen in the scene. These “shoppers” can be scripted to walk through the mall to give one a sense of traffic patterns (Fig. 7). Figure 8 shows a view from the second floor of the mall, over an open atrium.

4.2.4 Time Required. Acquisition of the data (mall images, aerial photographs, and related materials) took one person 1.5 days. Transfer and data fusion took 1 day. The integration of these fused images and the morphing of the textures took 2 days. Final integration and touchups for the full interactive walkthrough took 1 day, for a total model production time of 5.5 days.
total, the model design took about 60 hr and integration about 20 hr.

We plan to refine this exercise with the goal of reducing the total time to 3 days. In a production environment, we imagine that an individual gathering the source data could be equipped with a suitcase-sized toolkit containing a differential GPS terminal, a video camera, an ultrasonic range finder, and a memory device to which all the correlated data could be downloaded. The ultimate goal is to enable an individual to walk through a building several times and download the data to a system that could automatically create the building model.

4.3 Crisis Mission Planning

The purpose of the ground mission planning exercise was to develop from scratch, and within 48 hr, a three-dimensional representation of a selected mission area. The Defense Mapping Agency’s Vector Product Format (VPF) VMap Level 2 and Digital Terrain Elevation Data (DTED) terrain databases were selected as source data for the exercise. Because currently available modeling tools did not support VMap input, we wrote a conversion utility that translates VMap databases into an older format, namely Digitized Feature Analysis Data (DFAD), which is presently supported by modeling tools.

4.3.1 Key Features. For mission planning over land both permanent features such as roads, rivers, and buildings, as well as moving features such as people, vehicles, and aircraft must be modeled. The permanent features we modeled included transportation features (roads, train tracks, bridges), hydrological features (streams, rivers, lakes, ocean), and structures and key point features (buildings, windmills, and microwave
towers). The features were modeled using textures from previous activities we had performed as well as from newly acquired photographs. The dynamic features modeled included trucks, personnel carriers, and amphibious vehicles.

### 4.3.2 Functionality

Our interactive terrain visualization environment is currently constructed using Paradigm’s Vega and LynX, together with Silicon Graphics’ IRIS Performer library. We have added functionality, which enables us to load computer-based battle simulation playback files, which in turn drive vehicles through the virtual terrain environment. We have termed this application a “virtual VCR,” because it allows us to play back simulation events at normal, fast-forward, reverse, and fast-reverse speeds. Unlike a real VCR, however, we are able interactively to move the viewpoint to any position or to attach the viewpoint to a vehicle, and to query the status of simulated objects. This application supports crisis mission planning by allowing decision makers to preview planned actions, and to refine planned actions quickly when necessary.

### 4.3.3 Integration Issues

Time critical modeling requires well-defined interfaces. The lack of these led to most of the problems we encountered.

* Data completeness. Databases intended for mapping or radar modeling are really only a starting point for visual simulation, since they are silent on many important issues, such as the location and size of doors and other architectural features on buildings.

* Coordinate transformations. All three of the COTS packages we used assumed different values for the earth’s radius in their coordinate transformations, yet only one allowed the user to adjust this parameter. This resulted in misregistration between vehicles, point feature locations, and the terrain, which had to be resolved using special-case adjustments in the run-time application.

* Degree of automation. Current tools are not able to automatically generate compound features, such as a road going over a bridge, or tunnels going through a mountain. Manual intervention is still required to handle many common situations.

* Performance. Detailed terrain modeling continues to challenge modeling capacity and run-time performance, due to its nearly unbounded scope and the constant desire for the highest accuracy possible for ground-level visualization. In this respect it is important to structure the visualization database for both rapid culling and level-of-detail management.

### 4.3.4 Results

The VPF translation utility is able to convert the mapping data to DFAD format in a matter of a few minutes. Conversion of the DFAD data into a visualization database required approximately 26 hr of modeling time for a 5 × 7 square mile urban area, which included detailed road network, buildings, bridges, a river, and vegetation areas. The road network contained over 300 road segments, in addition to over 70 railroad segments, and over 2000 instantiated point features. Vehicles were programmed to follow the road network on a tour of the city. The run-time visualization application allowed the user to follow the vehicle tour, or to roam freely throughout the city.

### 5 Conclusion

We first summarize our major points, then provide our outlook for future virtual environment activities.

#### 5.1 Summary

Modeling for interactive and real-time rendering systems, such as virtual environments, has a set of challenges quite distinct from other types of modeling. Below is a set of guidelines that we follow. Although undoubtedly many modelers have employed these ideas before, we attempt here to codify these practices, so that not everyone will have to learn these guidelines on their own.

1. Use smooth shading to simulate curvature with simple geometry.
2. Reuse geometry by instancing repetitive objects.
3. Substitute textures for geometry wherever possible.
4. Prefer generic textures to specific textures.
5. Use levels-of-detail modeling (LODs) to reduce background complexity.

Even with these guidelines, some experience is required for their most effective application. It is also important to have at hand a complete set of image processing and format conversion tools for creating textures.

5.2 Future Activities

We are pursuing two main directions. First, we are working to increase the fidelity and capabilities of the VIP's environment. Second, we are identifying means to automate the model generation process. This entails acquiring data from multiple imagery sources—photographs, video, aerial, and satellite imagery—and migrating the models into the Model Shop. One major improvement would be a way to derive model geometry directly from imagery. Another would be in the development of design standards and format interchange standards that incorporate object constraints and behavior. Other projects on the horizon include

- **Modeling and design.** We would like to see the entire modeling process take place within a virtual environment. This would allow construction of complex 3D objects from primitive objects in a very intuitive way. The challenge is to find user interaction styles that are intuitive, precise, and that can be used for long periods of time without fatigue.
- **Traffic simulations.** Virtual environments offer an excellent way to visualize the dynamics of traffic patterns at busy or unsafe intersections. City planners and town agencies could use this capability to design safer streets.
- **Terrain visualization.** A continuing challenge is the visualization of actual terrain, which is extremely complex at all levels of detail. We will investigate alternative data representations that can support seamless, continuous levels of detail from foreground to background.

Acknowledgments

The Virtual Interactive Planning system was conceived and constructed by Michael A. Wingfield, with component models by Jeffrey R. Leger. The Burlington Mall and operating room models were acquired and developed by Andrew Cheng. Terrain visualization applications programmed by Robert F. Erbacher. This work was supported by The MITRE Corporation.

References


