Rapid Deployment of VRML Worlds for Military Planning Purposes

Lee A. Belfore, II
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529, U.S.A.
lbelfore@odu.edu

Alton Chavis
ACC/CEVP
129 Andrews St. Suite 102
Langley AFB, VA 23665
alton.chavis@langley.af.mil

Walt Baskin[1]
Sparta, Inc.
3630 George Washington Memorial Pkwy
Yorktown, VA 23693
walt.baskin@rossalyn.sparta.com

Kent Stevens
Loyola Enterprises
303 Butler Farm Road, Suite 106
Hampton, VA 23666
kent.stevens@langley.af.mil

Keywords: VRML, GIS, Rapid deployment, Terrain Data Management

Abstract

Broad dissemination of planning information presents problems to military planners due to the complexity of the information to be displayed and the unknown performance capabilities of computing platforms. The ubiquitous Internet provides a solution for information delivery. In addition, the Virtual Reality Modeling Language (VRML) supports the representation of information both in traditional and novel ways. Military planning information often describes places in the world where a military operation is underway or a base is planned. Such information includes of GIS information, and the location and placement of various structures. In this paper, we present an overview of the architecture and the process for creating these virtual worlds. In particular, we will discuss methods for creating and integrating terrain data, imagery, and other features into a virtual world. From an interactive perspective, the worlds feature a number of tools that aid in navigation and exploring the world, such as menus, and animated fly-throughs. In addition, we will discuss some of the tools that have been integrated into the various worlds to enhance their utility. Through an example, we show how the visualizations can be used for various planning applications through the control of feature visibility.

INTRODUCTION

The Internet has become a medium for communicating a rich variety of information not constrained by traditional boundaries. The form in which the information is presented is quite novel when compared to traditional delivery (i.e. paper/hard-copy) but the information providers are still learning how best to deliver the information and in what form. Furthermore, the delivery platform, web servers and clients, provide extraordinary flexibly because they are also programming platforms. Technologists are continuing to develop new programming platforms on which content can be delivered. Web pages include Javascript programming to increase the capability and flexibility of the web page presentation. Indeed, as the technology is applied, it evolves as evidenced by the variety of platforms for information delivery. In this work, we present an application that incorporates 3D content described in the Virtual Reality Modeling Language (VRML) [1].

The success of a particular approach can be measured in terms of the value added in its deployment. Virtual reality offers great potential for delivering information. Indeed, compelling content can be created in VRML with a moderate effort. Furthermore, the effort necessary to produce high quality content typically requires more time than is available. Some applications focus on presentation of the content only and contain no additional collateral value. In contrast, we describe a method for rapid deployment of compelling worlds that provide value beyond the rendering.

This paper is divided into five sections including an introduction, an overview of the application architecture, a discussion of how GIS data is integrated, a presentation of an example deployment and a summary.

[1]Work was conducted while Walt Baskin was employed by Loyola Enterprises.

APPLICATION ARCHITECTURE

The application architecture consists of several components that are integrated in various ways depending on the specific deployment desired. Because the application is GIS based, any object in the world must be referenced to a geographic coordinate system. Figure 1 presents a high level view of the virtual world architecture that supports rapid deployment. Each component is discussed in more detail.

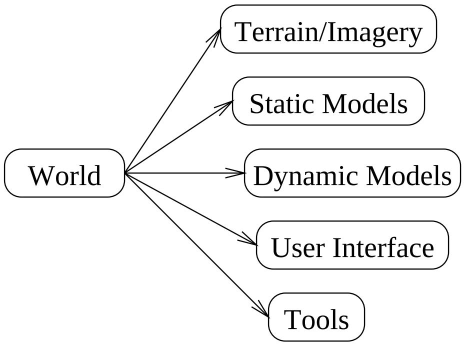


Figure 1. Components of Application Architecture to Support Rapid Deployment

Terrain and Imagery

The backdrops for the world consist of terrain and an image that is draped over the terrain. In cases where the deployment covers a region that is largely flat, an image on a plane is sufficient. VRML provides two primitives (nodes) that can represent terrains: ElevationGrid and IndexedFaceSet. The ElevationGrid requires that the terrain elevation be sampled on a rectangular grid. The IndexedFaceSet, on the other hand, provides a more general capability but is somewhat more complex. In cases where the terrain includes both flat and complex terrains, IndexedFaceSets can be employed because detail can be concentrated where necessary. Issues related to management of terrain and imagery are described in the next major section.

Static Models

The static models that are integrated into the world come from a variety of sources and requirements. Existing buildings and ground cover can be integrated as necessary to produce the desired detail. In some cases, custom construction of the models is necessary to meet the requirements for a particular case. In our work, several models may be reused with little or no modification in a broad class of applications. For example, the rules defining runway approach-departure surfaces are defined by standard parameters. Figure 2 shows a model used to visualize runway approach-departure and transitional surfaces. Models for buildings are frequently customized, although in many cases presentable models can be created with simple geometries painted with suitable images for the building sides.

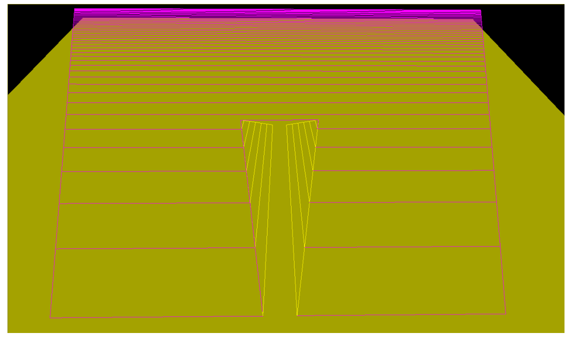


Figure 2. Airport Obstacle Avoidance Surfaces

Dynamic Models

Dynamic models have the capability of changing and moving within a world. Several examples have been deployed in various demonstration worlds. The models have included animated viewpoints intended to simulate what a person might see while traveling through the world. Components within the world can also be animated, such as aircraft hangar doors opening and closing. Two significant examples of dynamic models are described in more detail.

Tree Growth Model

Modeling tree growth can provide the ability to do future planning. In this example, a VRML EXTERNPROTO node has been developed to model tree growth near the approach end of a runway. In order ensure obstacle clearance, the trees must not pierce the approach-departure surface. To support the model, precise information about trees is required and must include the tree location, height, ground elevation, and approximate age. From this information, simple geometry can be used to determine whether the tree is a hazard. An image showing the tree growth model is shown in Figure 3. Trees growing near an airport runway pose a significant threat to approaching and departing aircraft. In our model, we have included the ability to view trees as they presently exist, and also extrapolate the growth patterns into the future. Because different species of trees grow at different rates, different growth profiles are included.

Aircraft Animation

A component of several applications is an aircraft animation. The aircraft animation has been designed to be a flexible component that can be included in any application. This flexibility has been achieved by packaging the animation components in a fully configurable VRML EXTERNPROTO node.

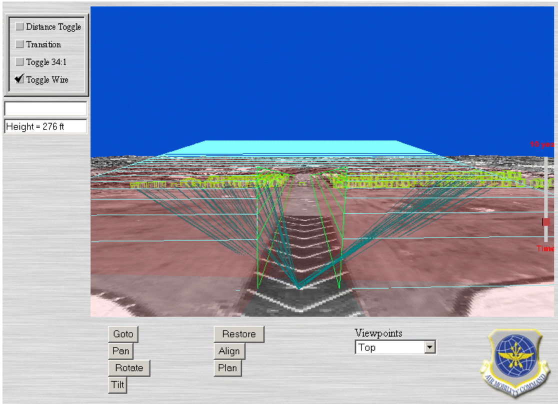


Figure 3. Tree Growth Modeling

The animation node includes the ability to define the path for the aircraft flight, pitch, roll, and speed changes. In addition, the aircraft model can be specified at instantiation time. The animation includes the special capabilities of changing the aircraft position with an external control, changing the speed, and including a text message keyed to different segments of the animation. Finally, in some applications it is necessary to exaggerate the terrain elevations by user request, requiring in a collateral change in the altitude scale. Figure 4 shows an image illustrating some of these capabilities.

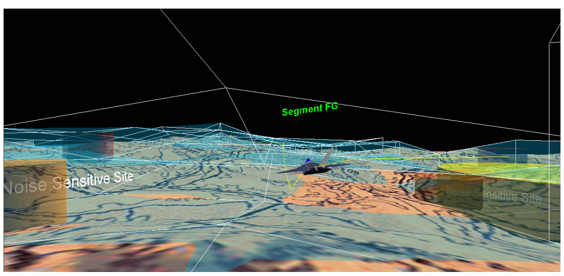


Figure 4. Aircraft Animation

User Interface

The user interface includes several components that aid in navigating through a world, interacting with the world, and identifying important features and locations within the world. In several deployed worlds, a collection of components have been created that improve the quality of the experience. Several components of the user interface are described in the following paragraphs. Depending on the requirements of the deployment, one or all of these components can be integrated.

Menu

The menu provides the user direct control over major modes, appearance and properties of the world. The menu may be integrated into the visualization in one of three ways. First, the menu may be visible only at a specific position within the visualization. Second, the menu may be incorporated directly within the world as a head's up display, always following the user's avatar. Third, the menu can be external to the world, residing in a frame within the browser. The choice of the approach depends on the effect desired.

Modes

Certain components of a world are sensitive to user activity and can result in world mode changes either by sensing the user's location or by detecting mouse events over sensitive components within the world. The mode changes may result in different options that are made available or different aspects of the visualization appearing.

Viewpoints

Viewpoints are VRML nodes that define camera positions. Placing the viewpoints at strategic points in the world facilitates the ability to extract useful experiences from the world.

Displays

In order to provide continual feedback to the operator, messages are provided to the user at various times. In most cases, the display is integrated with the menu in the head's up display. In addition, messages are occasionally directly located with the related object. For example, the aircraft flight animation can include a text scroll of preprogrammed messages that occur at various points in the animation. In addition, using the tools described above, information can be generated to appear in a convenient head's up display

Tools

Tools integrated in the visualization enhance the quality of the presentation and the usefulness of the virtual world.

Distance and Area Tools

The distance and area tools can be used to measure features within the world. The distance tool enables measurement between two points on a terrain by calculating the distance of a user's mouse click-drag on the terrain. The initial and final points appear connected by a line. The distance appears either on the distance tool geometry or can be passed to a text display. This tool can be used to verify the geographic scale matches the world scale. An area tool has also been devised that allows the operator to identify vertices of an enclosing polygon, show the enclosed area, and also calculate the area within the area.

Navigation Tool

Exploring a virtual world from a computer display can be somewhat disorienting due to factors related the awkwardness of the plug-in navigation controls and the limitations of the display. A cue can be provided by displaying a thumbnail of the geography with a marker indicating the current location.

INTEGRATION OF GIS DATA

GIS data provides the context for a particular world. Three issues must be considered when integrating GIS data into VRML worlds. First, all solid models in a VRML world are tied to a three dimensional Cartesian coordinate system. Second, proper knowledge of technological issues improves the quality of the VRML world experience. Third, VRML worlds cannot directly handle the large data sets typical in terrain data. These issues have been addressed by other researchers and are discussed in more detail below.

Coordinate Systems

Because VRML supports only Cartesian coordinates, using other coordinate systems require projections onto the Cartesian coordinate systems. One may use projected information in one of two ways. First, the projections are applied prior to integrating within the VRML world. Indeed, the generic export tools provided with ArcInfo and ERDAS do not export the geospatial coordinate transforms. Each of the VRML scene graph objects must be manually transformed (rotated, translated, scaled) in order to register correctly in coordinate space. This is necessary in that various tools that have been developed (distance measuring, area, and future planned tools) require a geospatial metric related to SI measurement units in order to properly function. Second, geocoordinates can be used in the VRML world, but converted in script methods for proper appearance. The GeoVRML group [2] provides general treatment of geocoordinate referencing and terrain imagery swapping. We have chosen not to use GeoVRML because our worlds are more limited in scope enabling us to use simpler programming constructs to enhance the experience and improve performance. The source of the GIS data is from ESRI compatible data formats, DEMs, and etc.

Managing Technological Issues to View GIS Data

Certain technological issues must be carefully managed to create a well behaved VRML world. The source data is taken from industry standard GIS tools such as ArcInfo and ERDAS. GIS tools provide the ability to manipulate GIS information and also to limit the scope of the data for integration into the world. Furthermore, these tools have the ability to export information into VRML. As noted above, VRML uses single precision numbers but GIS systems use double precision, which must be taken into account when the data is exported. Furthermore, matching the sizes of imagery with the computer architecture can result in improved performance.

Georeferenced coordinates require double precision numbers for accurate representation. VRML numbers provide insufficient precision to accurately render georeferenced data. We handle this by defining an origin in the world being constructed and then using all georeferenced coordinates relative to this new origin origin. As a result, the translated coordinates retain sufficient precision to accurately represent the world.

Imagery is used as a “canvas” that gives context to the VRML objects that are placed in the scene graph, and provides a way to embed information that is useful to the user. Due to the bandwidth limitations present throughout the distributed network environment, a process of tiling and decimating the resolution of the images has been developed to optimize transfer of the scene graph over the network. Experience has shown that tiling the images in a $2^n \times 2^n$ format is optimal, since this size tends to be an ideal match with graphics and memory architectures. Images are also reduced in resolution, or decimated, in order to reduce the file sizes. This decimation is conducted outside the immediate areas of interest, and helps in performance and navigation through the scene graph for computers with minimal graphics memory. Scripts have been developed in ERDAS to automate the tiling process, however, image decimation is still largely accomplished using manual methods.

Terrain Data

Terrain data is derived from a variety of sources including USGS DEM, NIMA DTED, and terrain byproducts of stereopair imagery orthorectification. Terrain from USGS and NIMA Level 1 is provided with 90 Meter postings and can be integrated into a scene graph in one of two ways. First, the data can be incorporated as a terrain grid, which provides relatively high fidelity to the actual terrain in a large scale area. Second, for smaller scale areas which extend out over hundreds or thousands of miles, the terrain must be “loosened” or resampled to reduce the number of postings. In reducing the number of postings, navigation performance through the scene graph is enhanced. However, in reducing the number of postings, problems may be introduced in that features that may occur between the postings may be excluded. In navigating a scene graph created for a grid with regular postings, any imagery draped over the grid appears to be “synthetic” in that stark angles and shadows appear from various viewpoints. Another method of modeling terrain is by using a Triangular Irregular Network (TIN) mesh. This has the advantage of modeling the terrain according to the slope and aspect of the terrain, and maintains a greater fidelity to the terrain features. As slope and aspect change, smaller triangles are calculated to represent the changes in the terrain and all features are represented to the limits of the terrain data. Imagery that is texture mapped over a TIN has a more realistic and natural appearance when navigating the scene graph, and shadows appear as softer, smoother and not synthetic. Scene graph navigation performance using a TIN is dependent on how many TIN elements are used in the terrain model. Applications such as Lightwave 3D, 3D Studio MAX have features that can remove TIN elements to sparse the TIN mesh to the degree desired.

The world designer must decide what information is vital in any given situation and create a suitable scene graph. In these cases, mixing resolutions for different parts of the world can provide a reasonable result. One method for accomplishing this is the use of quad trees to subdivide the terrain being modeled [3]. Additional resolution levels can be provided in critical focus points. Figure 5 presents a scene graph for a terrain. Quad tree tiles are managed by a level of detail (LOD) primitive built into VRML. LOD nodes allow the image resolution to change automatically as a function of the viewing avatar's position. An alternative approach is to tile the terrain for a world, and for selected tiles, provide higher resolution as appropriate. Figure 6 gives and example of integrated mixed resolution (1 and 2 meter) imagery.

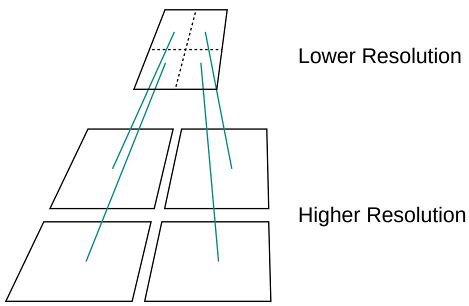
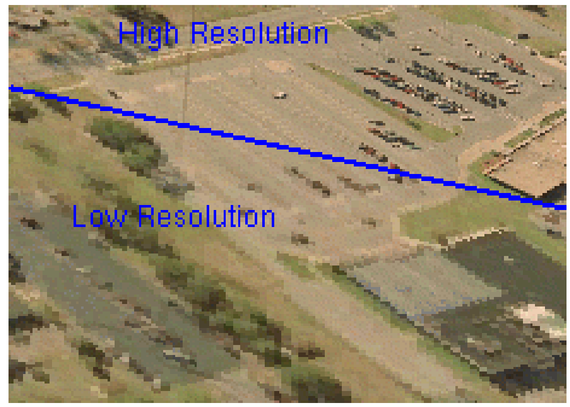


Figure 5. Quad Tree Structure Used For Some Terrain Models

EXAMPLE DEPLOYMENT

In this section, an example deployment is presented. The visualization shows an example deployment for airbase planning. Several of the capabilities of the visualization are described below. The virtual world demo contains many aspects described above tuned to a specific deployment. Figure 7 shows the opening view of the airbase. The imagery includes planned modifications to an existing. A special perspective can be gained by an animation of a flying aircraft that includes a viewpoint. Figures 8 and 9, respectively show a viewpoints on the approach and from a guard tower. Aspects of the world can have additional information accessible through anchors that give access to web pages. Figure 10 shows the hospital as well as a web page associated with the hospital. Any information that can be formated and displayed as a web page may be provided.

(a) Original Image



(b) Area Magnified

Figure 6. Mixed Resolution Imagery

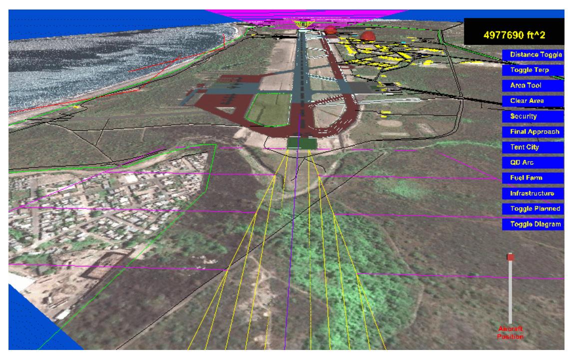


Figure 8. Aircraft on Approach

SUMMARY

In this paper, we have outlined an approach to construct VRML demonstration worlds rapidly. By appropriate derivation and conversion of GIS information into VRML, compelling worlds can be created that present realistic and useful visualizations that can be employed in military planning applications. Furthermore, since VRML is a standardized Internet technology operating within browsers and plug-ins available at no cost, the worlds can be deployed easily and inex-

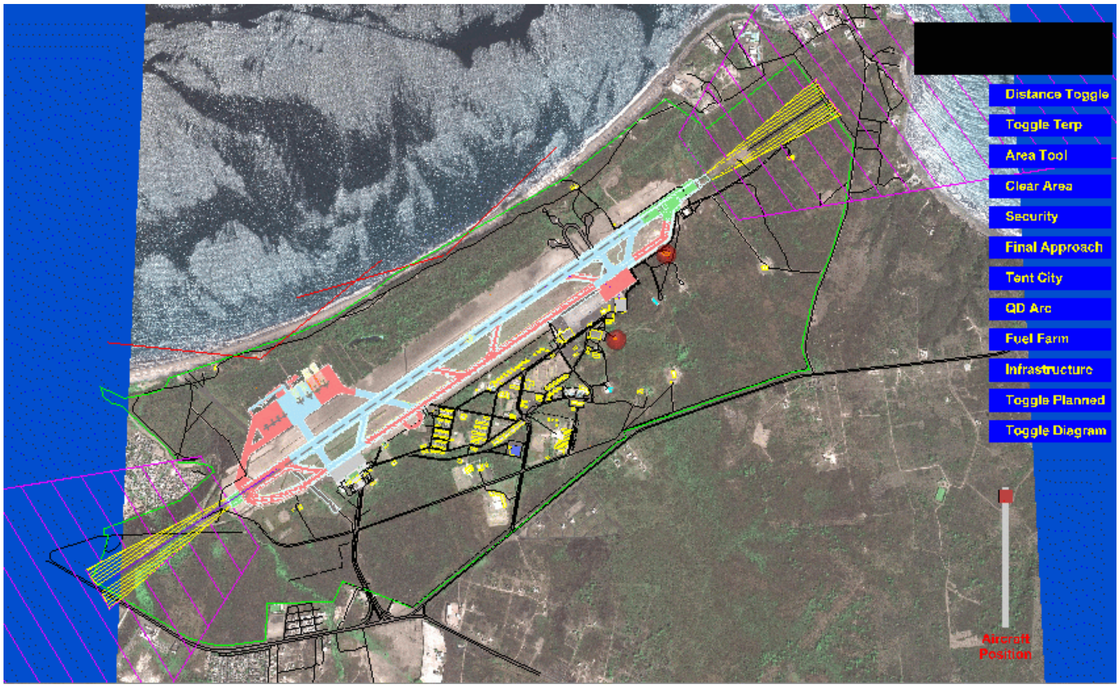


Figure 7. Airforce Base with Overlay Highlighting Planned Modifications

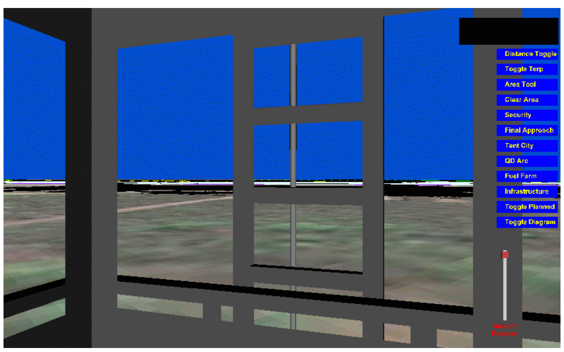


Figure 9. View from Guard Tower

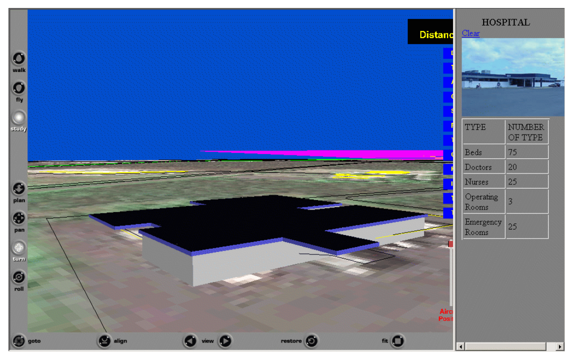


Figure 10. Hospital and Associated Information

pensively over the Internet.

REFERENCES

[1] The Web3D Consortium, Incorporated, “The Virtual Reality Modeling Language”, http://www.web3d.org/Specifications/VRML97/index.html, 1998.

[2] Martin Reddy, Lee Iverson, and Yvan G. Leclerc, “Under the hood of GeoVRML 1.0”, in *Proceedings of the Fifth Symposium on the Virtual Reality Modeling Language VRML2000*, Monterey, CA, February 2000, pp. 23–38.

[3] Martin Reddy, Yvan Leclerc, Lee Iverson, and Nat Bletter, “TerraVision II: Visualizing massive terrain databases in VRML”, *IEEE Computer Graphics & Applications*, vol. 19, no. 2, pp. 30–38, March/April 1999.