An Open-Source Virtual Object Model for 3D GIS

Barry Bitters
Senior Research Scientist
Department of Environmental Studies
University of West Florida
11000 University Parkway
Pensacola, Florida 32544
ph. [850] 936-1060; FAX [850] 936-1060
mailto:bbitters@uwf.edu

Abstract

In the newest version of ArcGIS®, the 3D Analyst extension will feature an implementation of a new 3D visualization software – ArcGlobe™. What is unique about this software tool is that it allows the storage of limitless amounts (assuming limitless disk storage) of geospatial data and then the ability to interact with that data in a virtual environment. Further, ArcGlobe allows the representation of stored feature data using generic or specific 3D models in both OpenFlight™ and 3D Studio™ format.

With the addition of the capability to import and display OpenFlight 3-dimensional models into ArcGlobe, ESRI has entered the arena of traditional real-time simulation technology; a technology that has been used for many years by NASA and the Department of Defense. Real-time simulation technology is a mature simulation environment and is a viable means to create immersive displays of GIS data in conjunction with realistic virtual landscapes. However, for this functionality to be a viable and "user friendly" capability, additional software must be developed to allow the implementation of rapid and user-transparent exploitation of OpenFlight and 3DS formats and associated capabilities.

We have tested a prototype version of a non-proprietary, virtual object model and its associated library of 3-D objects. The object model and its associated 3-D model library provide translation and display mechanisms to port GIS data into the ArcGlobe simulation environment. This paper will provide background on the history of real-time simulation, our efforts to promulgate this technology into the GIS arena, and our current efforts to deliver a robust virtual object model and its associated 3-D model library.

Introduction

Environmental Systems Research Institute (ESRI®) has recently introduced a new capability in the new release of ArcGIS 3D Analyst 9.0 – a capability that will allow users to manage and visualize extremely large sets of three-dimensional geographic data. This capability will be an integral part of ArcGIS 3D Analyst 9.0 in a new desktop application called ArcGlobe™. ArcGlobe provides the capability to seamlessly interact with any geographic information as data layers on a three-dimensional globe.

Using extremely efficient and highly optimized data retrieval and display techniques, ArcGlobe provides seamless and rapid access to nearly unlimited volumes of geographic information. Using intelligent scale-dependent paging, ArcGlobe allows efficient data loading in and out of memory as required. When viewed from a distance, data is generalized, but as the observer moves in closer, full detail is exposed. This intelligent data handling capability eliminates display freezing and reduces the use of system resources. This means that ArcGlobe can simultaneously display raster imagery, vector feature, and elevation data sets representing data from a global scale to a local scale with performance rivaling traditional twodimensional mapping capabilities.

As an extension to the ArcGIS Desktop products, ArcGIS 3D Analyst allows users to use the traditional GIS analysis tools available in ArcView®, ArcEditor™, and ArcInfo™ to perform geo-processing tasks in a 3D environment. Using standard interactive mapping tools, users can also pan, query, and analyze data at any scale, or they can zoom right into their local area and view very high-resolution spatial data, such as parcels or detailed aerial photographs, of their area of interest. Further, ArcGlobe incorporates a conceptual model from real-time simulation technologies that allows the display of representative 3D models for each instance of features stored within vector databases.

For all of this to be possible, a basic understanding of the principles behind real-time simulation are essential. The generation of interactive and realistic 3D visual displays is not a straight forward process. It does not involve simply adding imagery, elevation and feature data into an ArcMap display and then turning on the ArcGlobe functionality and automatically interacting with the virtual environment. First, a significant effort must be expended to insure that all data sets are correlated with each other and that all data sets are compatible for use within 3D environment. Second, all data must be segregated based on predefined rules for the handling of data within the third dimension. Finally, the software, hardware, and firmware environments must be optimized to insure that efficient processing will result in an aesthetic and fully functional display. This paper will address all of these issues by initially discussing the concept of the real-time simulation database, then addressing the repercussions of real-time simulation on the ArcGlobe concept, and finally discussing operational aspects of database creation and the efficient use of visual databases in ArcGlobe.

The Virtual Environment Database

The creation of virtual environment databases for use in real-time simulation systems is not new. In the early 1970s, NASA created synthetic natural environments for use in early real-time simulation systems. These early systems were not digital or analog systems, but were based on detailed terrain models developed by teams of model makers. A movie camera mounted on a precision gantry was maneuvered over the terrain models to generate visual cockpit displays. At about the same time, the U.S. Department of Defense (DOD) was experimenting with a gantry mounted camera moving over specially prepared plastic relief maps, also as a source of visual cockpit displays. The conceptual basis of this form of virtual environment database development was based purely on manual creation of physical terrain models that represented the geographic area of interest. Generally, resolution and levels of detail were fixed and were dependent on time and fiscal constraints in the operating budget.

Today, the conceptual basis of virtual environment database development is not so straightforward. Often, hardware and software limitations dictate how databases are developed, when in fact database generation procedures should be designed primarily to capture the essence of a specific natural or cultural landscape. Hardware and software plays an important part in determining what volume of data can be displayed on an image generation display.

In recent years though, formal conceptual design of architectures for the development of Virtual environment database has been performed. Birkel (1999) defined a formal conceptual model for synthetic natural environments (Figure 1). In this model, he defined an Environmental Ground Truth element and a Military Systems Models element. An integral part of the Environmental Ground truth element, the Environmental State, is that data used to describe and categorize the natural and cultural landscape – the ultimate end product of the database generation process.

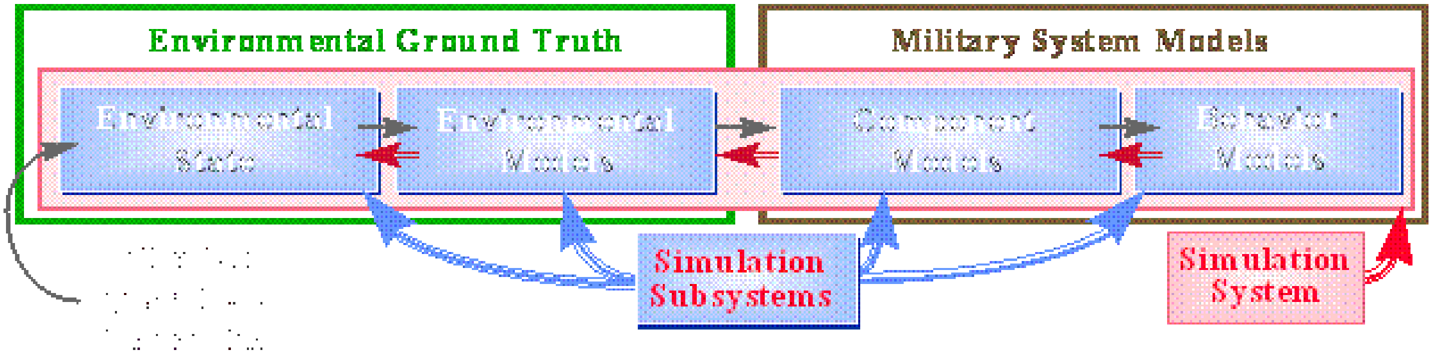
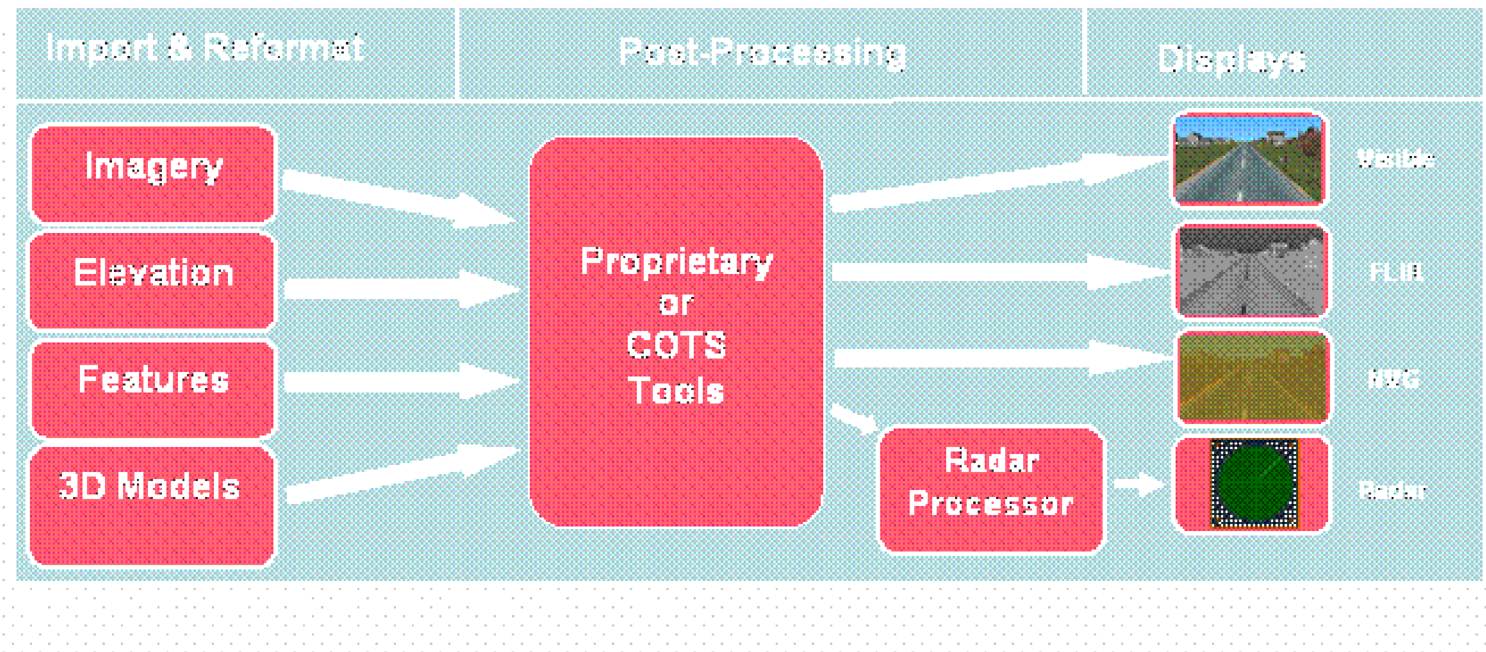


Figure 1. A broad conceptual design of the synthetic natural environment

Donovan (2000) expanded on the concept of the Environmental State by defining an open architecture for the development of virtual environment databases. This open architecture was designed to use traditional sources of raw geographic information to create visual and sensor run-time databases and generate SEDRIS transmittals. This conceptual design was heavily dependent on the use of U.S. National Geospatial Intelligence Agency (NGA) (formerly the National Imagery and Mapping Agency NIMA) data as a source of raw geographic information. The following discussion expands on Donovan’s open architecture by outlining a conceptual design for a database generation system expressly designed for the creation of next generation real-time simulation databases. This conceptual design has been developed with simplicity of processing and efficient use of software in mind.



Bitters [2004] has recently expanded this concept incorporating data warehousing and data mining techniques to allow the exploitation of disparate sources of spatial data in the data fusion process (Figure 3). This conceptual design includes functionality for advanced feature data conflation, point data de-confliction, and feature-to-image matching – all common issues encountered when performing data fusion of disparate sources of spatial data.

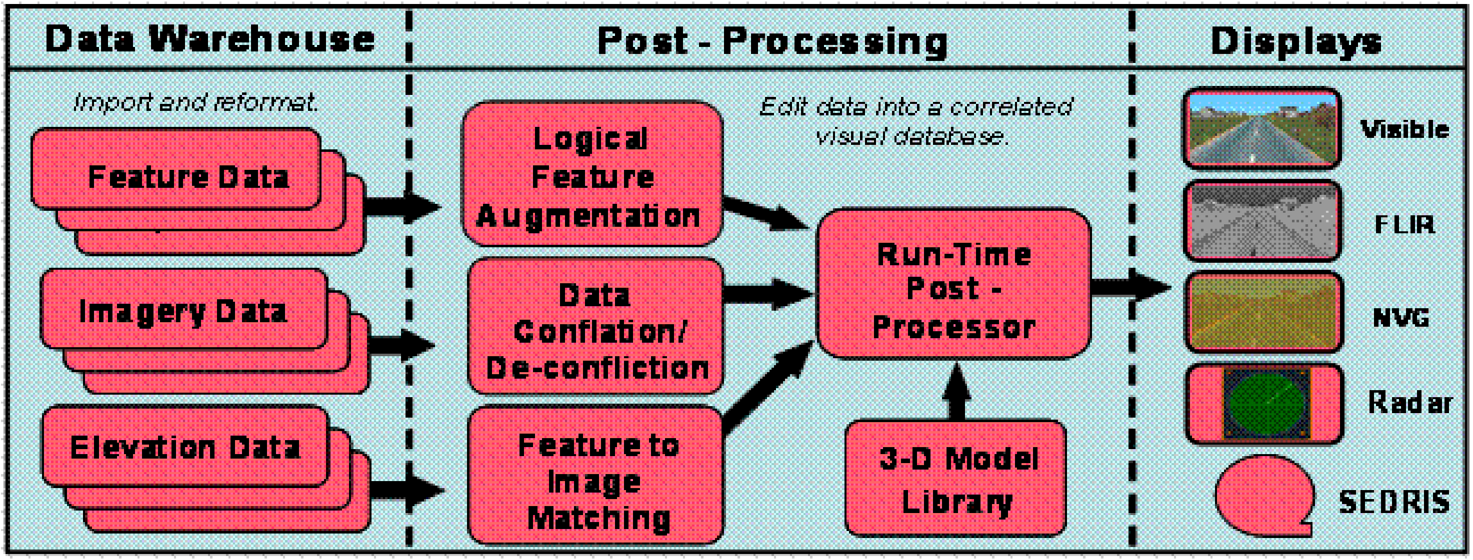


Figure 3. This conceptual design for a database generation process incorporates a formal data warehouse element Where disparate data are imported, reformatted, and stored and a post-processing element where data editing, data augmentation, data deconfliction and conflation, and feature-to-image matching are performed prior to the run-time post-processing of "cleansed" data into various runtime databases.

This conceptual design operates on the premise, as is the case with most traditional visual database post-processing software, that only the highest resolution of data will be available for display. In this way, the post processing software would perform intelligent tiling and automatically produce reduced resolution data sets. This is also the case in ArcGlobe. For any piece of ground within the ArcGlobe data structure, only the highest resolution data set is stored. This style of data storage (Figure 4), sometimes call chip-stack, is a very efficient means of storing spatial data, where by only the highest resolution of the original source data is maintained for any point on the ground. In this way, after data is loaded into ArcGlobe and lower resolution data has been removed, tiled reduced resolution data sets are generated or updated for the entire data structure using only the highest resolution imagery.

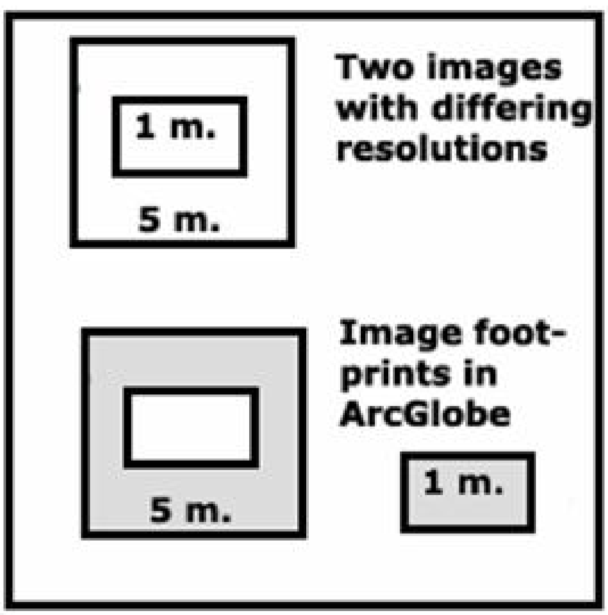


Figure 4. The "clip-stack" data storage used in ArcGlobe. Note that only the highest resolution data for any point on the ground is stored.

Elevation data provides a source of visual relief representation for the virtual environment. Triangulated networks are the most efficient form of data storage for elevation data that is destined for use in any real-time simulation environment. In the past, both triangular regular networks (TRN) and triangular irregular networks (TIN) have been used in real-time simulation systems. ArcGlobe uses a TIN as the basic data storage model for elevation data.

Efficient processing of virtual environments is dependent on overall polygon count of each scene. This overall polygon count per scene includes each triangular elevation entity within the scene. For this reason, it is essential that the resolution of the final production TIN is optimized to a resolution compatible with efficient display operation. Currently there are no tools that will allow an accurate preliminary database design based on raw data resolution and the resulting polygon counts, for use in the development of visual databases in the ArcGlobe environment. Therefore, it is a purely educated guess at what will be the optimum resolution for a TIN.

Database Development – Imagery (Ground Texture)

Imagery not only provides visual cues for detailed recognition of the natural and cultural landscape, but it also serves as the cartographic base for all feature data that will be displayed above the terrain. Therefore, extreme care must be taken in preprocessing imagery for use in the virtual environment. Precision rectified and ortho<-rectified imagery is an essential element of the virtual environment database. Precision rectification insures that each image is positionally correct relative to the ground. Ortho-rectification insures that each image is positionally correct based on a detailed elevation data set. Performing rigorous and precise rectification and ortho-rectification insures that imagery is both aesthetically appealing and positionally correct.

If you have several resolutions of imagery over a particular project area of interest, it is best to insure that each different resolution is a power of 2 reduced resolution of the most detailed image. For instance, if for a particular area, you have a 5 meter image and within that 5 meter image you have a small piece of 1 meter imagery for a high-interest area. To assist in the rapid ingestion, processing, and display of these two adjoining images, a base resolution of 1.0 meter is defined and all other imagery is preprocessed to a power of 2 lesser resolution. Therefore, the five meter imagery must be resampled to a 4.0 meter ground sample distance to meet the power of 2 requirement.

Database Development – Point Features

Point feature layers contain all those cultural and natural features that can be represented as discrete points. Storing cultural and natural features as point features has nothing to do with their actual size or shape in ground space. It has to do with the availability of a 3D model to use as a representation of the feature within virtual space. It is often advantageous to store a wide variety of objects as point features, such as trees, buildings, refinery complexes, even airfields. If a representative 3-D model is available for a feature, no matter what the actual size, the feature is probably best stored as a point feature.

Generally, at least two point layers are created: one for storing point vegetation data; and another for storing all other point features. The reason for segregating point vegetation features is because point vegetation data is often created by scattering random point feature within the boundaries of each individual areal vegetation feature. This process creates a point vegetation layer with literally hundreds of thousands of point vegetation features. With a data set this large it is essential to have a separate layer that can be controlled independently during the post-processing and display functions.

Database Development – Linear Features

Linear feature layers are used to store various forms of linear features – features with significant length and “minimal” width (relative to length). Object such as railroads, highways, streams, fences, and power lines all have a significant width. However, their width relative to length is “insignificant.”   Examples of texture maps for some of the more common highway surfaces are shown in Appendix B.

Generally, two linear layers are created to store the majority of all linear feature data: one for linear features that are flat on the terrain, e.g. railroads, highways and linear; and another for linear features that have a height above the terrain surface e.g. fences, powerlines, and walls. Segregating these broad categories of linear features is essential because two different processes are involved in the display of each. The display of linear features that are flat on the terrain involves the iterative application (tessellation) of a representative texture image along the full extent of the linear feature. The display of linear features that stand up above the terrain surface must first be represented by 3D geometric objects and then rendered wit tessellating texture maps.

Database Development – Area Features

Area feature layers are used to store various forms of natural and manmade areal feature data. A cultural or natural feature is considered to be areal if it has significant length and width. Examples are: open water features, forests, grass areas, and parking lots. Examples of some typical areal texture maps are shown in Appendix C. Generally, two layers are used to store the preponderance of areal feature data: one for features with no height above the terrain surface, e.g. parking lots, grass, and areal water features; and another layer is used to store area features that are raised above the terrain surface, e.g. forest canopies, clouds and other atmospheric effects. These broad categories of areal features must be segregated because different processes are required to build and display these features in real-time.

Database Development – 3D Models

An essential part of the ArcGlobe process is to use generic and specific 3D models (and associated texture images) to represent feature records stored within the visual database. Examples of generic 3D models are displayed in Appendix A. When 3D Analyst is delivered it comes with a small selection of 3D models and texture images. In the near future, the University of West Florida will be serving on the Internet a public-domain 3D model library of over 1000 3D models and over 4000 different texture images. In future revisions of, these may become an integral part of the standard deliverables for ArcGlobe.

To be able to display a visually recognizable representation of point feature objects, for each point record in a point feature layer, each different point feature record must be mapped to its counterpart 3D model object. This mapping can be explicitly stored in an attribute field of each feature record as a free text file name of the corresponding 3D model file name. In ArcGlobe, all 3D models that will be used must be represented in the geodatabase as stand alone feature classes. The **Import3DFile** coclass is intended for the import, into a geodatabase of 3D models that will be used within ArcGlobe. A Visual Basic tool that will import 3D Models into the geodatabase using the Import3DFile coclass is available on the ESRI ArcScripts website. It reads in a 3D model file and stores it as a stand alone feature class in a specified personal geodatabase. If there is already a stand alone feature class in the same database for that 3-D model, then the old 3-d model class will be deleted and replaced by the new model file. The 3D model file can be in either 3D Studio (*.3ds), OpenFlight (*.flt), or VRML (*.wrl) format. With the installation of all 3D models into the geodatabase as individual feature classes and the existence of explicit references to a 3D model filename within each feature record, ArcGlobe will take over and instance each appropriate 3D model in it proper ground space location and orientation within the display.

Optimizing Performance With ArcGlobe

Tuning a visual database is the process that adjusts the densities of data in a visual database to an optimal level of performance for a particular display system. If there are two many features in a particular portion of the database, fly-thru capabilities will be slowed, or, for all intent and purposes stopped. Prior to actually reducing the detail within a viual database, there are anumber of hardware and software adjustments that can be made to optimize display performance. What follows are a number of suggestions for the proper adjustment of many of the display and performance parameters within the ArcGIS environment, adjustments to software funtions for display graphics boards, and some quick options to reduce the density of data within a visual database.

> Use a computer with Dual Processors. ArcGlobe does not use both processors but a system with “dualees” will allow system overhead operations and any other processing to take place with minimal interruption to the ArcGlobe.

> Use a computer system that is equipped with a high performance OpenGL Graphics card. In addition to meeting the

minimal requirements for running ArcGIS, a PC running 3D Analyst, ArcScene and/or ArcGlobe will perform better with a minimum of 1 Gb (for best results 2 Gb) of RAM and an OpenGL graphics card containing at a minimum 64 Mb (for best results 128 Mb) of texture memory. Texture memory controls the total amount of imagery and feature texture that can be used. The more RAM and texture memory that is available the better ArcScene and ArcGlobe will perform.

Use the same projection for all data. Insure that all data within the project is transformed into the same projection. ArcGIS's capability to perform on-the-fly projection of data is a very convenient feature. However, when using ArcScene or ArcGlobe, the additional behind-the-scene-processing required to perform the on-the-fly projection transformations will seriously impair screen refresh rates. Use ArcCatalog to make sure all data in a scene is in the same projection prior to attempting an ArcScene or ArcGlobe session. Use the data Management Tools in the ArcToolBox to define, reproject, or transform the coordinate system for images, coverages, grids, shapefiles, or geodatabase layers.

Display data based on its size. Improve performance when displaying a scene containing complex data by changing the rendering properties for the layer. Right click on the layer and click on the *Rendering* tab. Under *Visibility*, change the default setting of the *Render At All Times* to *Render Layer Only While Navigation Has Stopped* and adjust the rate next to *Draw Simpler Level of Detail if Navigation Refresh Rate*.

Set the fixed extent for layers. When appropriate, limit the extent of a scene. The extent is the minimum bounding rectangle that contains the outer boundary of the layer or scene. By default, the extent of the scene is the combined extent of all the layers of the scene. To limit the amount of data that ArcScene or Arc Globe have to load in to memory, the extent can be changed to the extent of one layer or can be expressed as an x-y coordinate pair. Data that falls outside of this user specified extent will not be displayed. Perform the following steps to change the scene extents:

In the ArcScene table of contents, right click on Scene Layer and choose Scene Properties.

In the Scene Properties dialog box, click on the Extent tab.

To set the extent using a layer, click the radio button next to Layer and click on the adjacent drop-down box. All the layers in the scene will be listed. Select the layer that will be used to define the extent of the scene.

To specify the extent with coordinates, click on the radio button next to Custom and define the extent by typing in the maximum and minimum x and y values

Click OK.

Consider down-sampling elevation surface resolution. When draping layers over the terrain surface, reduce drawing time by down sampling the underlying surface resolution. If the rendering resolution of the layer is not vital, lower the default values for the surface resolution. Right-click on the elevation layer and click on the Base Heights tab in the Properties dialog box. Click on the Raster Resolution button and set the x and y cell sizes to a lower value to increase navigation speeds.

Consider down-sampling imagery resolution. When draping layers over the surface, reduce drawing time by down sampling the imagery resolution. For initial testing, a significant increase in display refresh rates will result with reduced resolution ground texture. Right-click on the image layer and click on the Base Heights tab in the Properties dialog box. Click on the Raster Resolution button and set the x and y cell sizes to a lower value to increase navigation speeds.

Don't automatically make new layers visible. By default, ArcMap makes all layers visible. A significant increase in refresh rates can be attained by disabling this feature. In the ArcMap menu, choose Tools - then Options. In the *Options* dialog box, click on the *Applications* tab. Uncheck the box *Make Newly Added Layers Visible*.

Modify OpenGL graphics card settings. When working with complex data sets, turn off anti-aliasing and vertical synchronization. Changing these features will sacrifice visual quality for improved navigational and display refresh performance.

Conclusions

This report was prepared using an early release of the ArcGlobe software. Much of the functionality described was only partially functional and much of the functionality that is customary to a typical real-time simulation software system was not available. Never the less, this technology has the potential to allow the viewing of current or future changes in our environment without being on site. It can assist in decisions to develop a new building downtown, and allow the visualization of what the building will actually look like; or from what floors will be able to view the surrounding environment. These types of questions can not be answered without an accurate simulation tool. Additionally, with a high-end desk-top computer it will be possible to perform interactive fly, drive, or walk-thru of realistic cultural and manmade environments.

In the future, with significant improvements in bandwidth and data compression, hosting this type of virtual environment on the Internet, with the ability to navigate freely, could provide information to businesses, travelers, and tourists. The potential applications of this type of simulation capability are limited only by the imagination of visual database engineer and the end users.

References

Bitters, Barry. (2004) Real-Time Simulation Database Generation: A Conceptual Model for the Future. *Proceedings of the IMAGE 2004 Conference*. Scottsdale, Arizona

Birkel, Paul A. (1999) Synthetic Natural Environment (SNE) Conceptual Reference Model. *1998 Fall SIW Conference*, September 1998. www.sisostds.org/swi/98fall/view-papers.htm

Donovan, Kenneth B., et al. (2000) An open architecture for synthetic environmental database generation. *1999 Fall SIW Conference*, September 1999. www.sisostds.org/swi/99fall/view-papers.htm

SEDRIS. (2002) Environmental Data Coding Specification (EDCS), ISO/IEC Final Draft. Available at http://www.sedris.org/.

DIGEST. (2002) *Digital Geographic Information Exchange Standard (DIGEST), Part 4, Annex B, Attribute and Value Codes*, Edition 2.1, September 2000.

Appendix A. Examples of Generic 3D Models Used to Represent Point Feature Data.

Appendix B. Examples of Highway Texture Images used for Tessellation on Linear Highway Features.

ASPHALT TRAVELED-WAYS

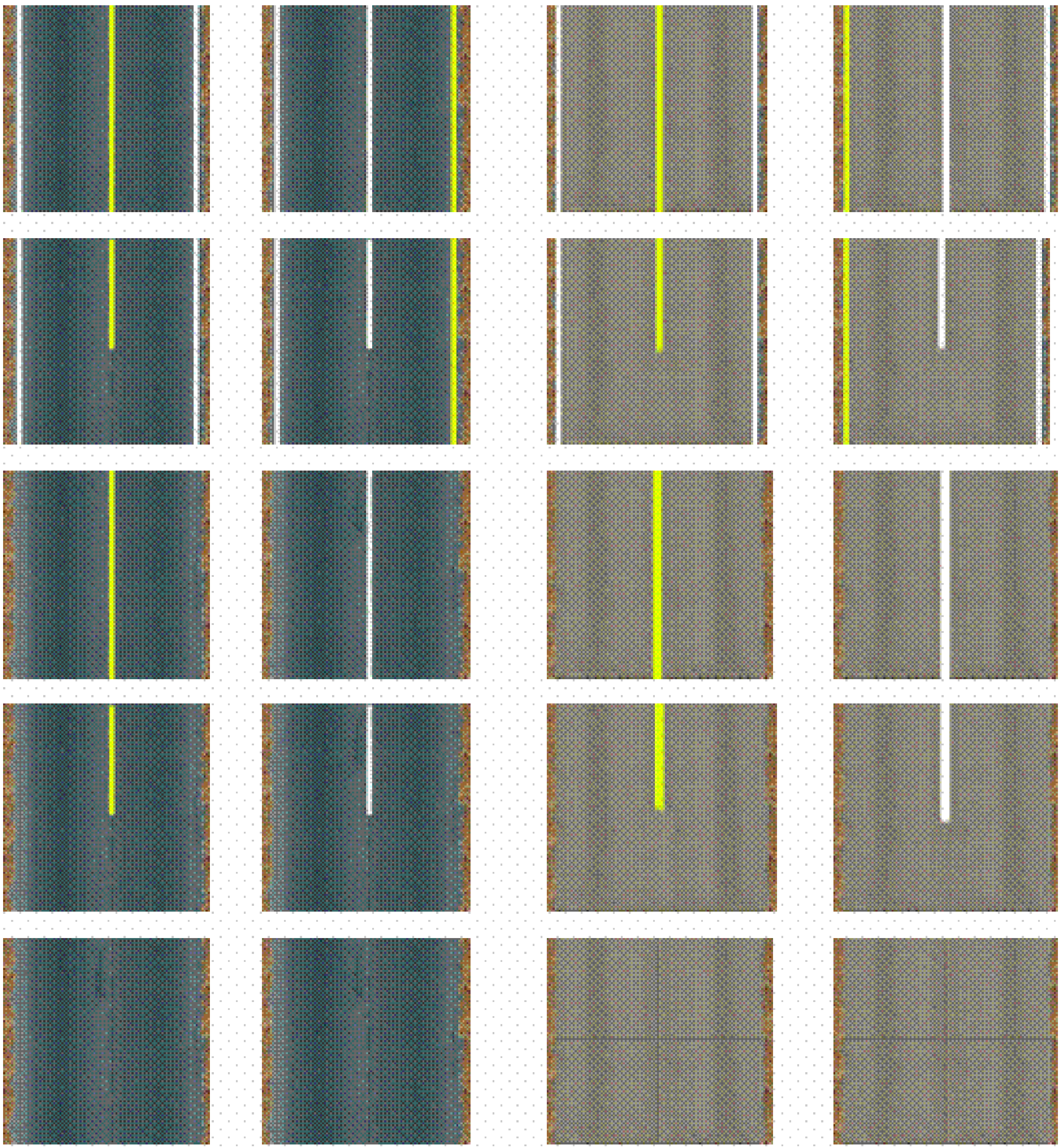
Two-Lane

Two-Way | One-Way

CONCRETE TRAVELED-WAYS

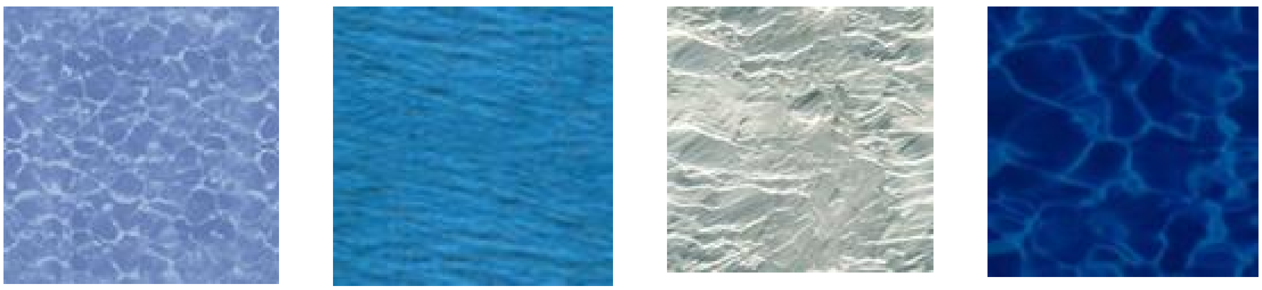
Two-Lane

Two-Way | One-Way



Appendix C. Examples of Various Areal Texture Images Used to Depict Area Features.

Water



Forest

Land