White Paper

Geospecific Databases for Battlespace Visualization

By Graham Beasley and John Burwell

Geospecific Databases for Battlespace Visualization

Abstract

The simulation and training industry has developed a process to produce real-world, geospecific visual databases for pilot training. The process involves the collection and processing of raw image data to produce an earth-referenced product that can be visualized in 2D and 3D using high-performance graphics hardware and software. Pilots training in these simulators are presented with scenes that look exactly as they do in the real world. The value of this fidelity is rapidly perceived as pilots almost instantly move beyond basic part-task training to full mission training. The true value of geospecific visual databases goes far beyond their use in training and simulation. Because the visual database is earth-referenced, it can be fused with a variety of other earth-referenced data. It can be presented in a variety of different formats using different types of display products. It can be annotated and modified using a variety of collaborative tools. At the same time, we are seeing the commercial availability of high-resolution, panchromatic, and hyperspectral satellite imagery coming from a variety of sources. All these elements, when properly combined and integrated, create an opportunity to develop a series of battle-space visualization capabilities for decision support.

This paper outlines the process, findings, and solutions related to the gathering, formatting, and processing of geospecific data. It also outlines some of the display environments and decision-support concepts that can be developed to achieve information dominance.

1.0 Introduction

For more than 35 years, a variety of companies have worked hard to engineer digital computing systems and display environments capable of creating scenes that replicate the real world. Those who have architected image-generation systems have worked to increase scene realism and improve simulator fidelity. This effort has resulted in evolutionary improvements from one iteration of image-generation systems to the next. Major evolutionary enhancements have come in several areas including lighting and shading, anti-aliasing, and texture capability.

Lighting and shading capabilities have improved, and simple flat shading, which was formerly an industry standard, has been replaced with smooth-shaded polygons made possible by implementing Gouraud shading algorithms in hardware. Phong lighting followed to add specular reflections and highlights. The capability of image generators to create real-time, anti-aliased scenes composed of complex polygonal structures has further increased scene realism.

The single greatest improvement in image-generator scene quality in the 20th century was derived from the ability to utilize texture mapping. Textures can be generic noise patterns and can be derived from photographs of real objects. They help to add realism to the computer-generated image. By utilizing imagery, image generators can display visual databases that appear to look like a general (geotypical) or specific (geospecific) area of the world. This allows simulators to more closely resemble the real-world view with which students will be interacting in reality. The simulator is thus more realistic or believable, and the transfer of training from simulator to real-world device increases. An example of the image quality achievable is shown in figure 1.

Fig. 1

As we progress into the 21st century, the biggest development in image-generation technology relates to putting sufficient mathematical rigor behind the geospecific imagery to allow mensuration. When the synthetic environment created in the computer is mathematically accurate and represents the real world, the application of the visual database expands beyond the training device into a support tool for command and control or intelligence applications. These smart databases, displayed in new types of immersive environments more suited for collaborative visualization, realize the vision of battlespace visualization discussed a decade ago. Figure 2 shows a representative example of a battlefield visualization system.

Fig. 2

Some of the original concepts for image-based databases can be traced to the mid-1980s, when mission-rehearsal systems were proposed based on ultra-custom, very expensive image generators and often relied on classified imagery that was highly localized and specific for a training function. An excellent example of this technology is the multimillion-dollar Special Operations Forces Aircrew Training System, which used custom-built ESIG-4000s.[1] This system was prohibitively expensive and complex and did not receive widespread acceptance (fewer than a dozen were made). It also suffered from database generation woes and was often starved for source data. Source data becomes the content, and content is king.

Now, just over a decade later, high-quality 1 m imagery can be purchased on the Web from Space Imaging. The next few years promise to bring commercial hyperspectral imagery that can be used for material classification of images. There will also be more data sources, which should lead to lower pricing. Today, a huge growth in the geographic information systems (GIS) market fuels development of tools to process the imagery. Many of these tools are well suited to build and interact with the synthetic environment.

2.0 Database Generation

Traditional database generation was a very labor-intensive task that was project specific and difficult to repurpose or reuse. Most databases were optimized for a specific image generator and specific training requirements. The only real similarities between a database for a tank and a fighter were the tools used to construct the database and some of the sources of data (for example, digital feature-analysis data or digital terrain-elevation data). Although the resolutions and feature densities were very different, the database construction techniques were very similar.[2]

Modern tools such as MultiGen software have made database construction techniques easier, but it is still a manual process that requires individual tuning and construction techniques. Terrain skins can be generated automatically, but visual databases are still a custom process. The focus on using real imagery in simulators has helped fuel the need for clean imagery. Clean imagery is a product that has been processed from its raw state into a visible source that is useful to the image generator and that will ultimately provide more complete and accurate battlespace visualization value.

The data processing required to transform raw imagery into something visible in real time and mensurable is not trivial. Once the appropriate source imagery is collected, the challenge lies in creating a mosaic of the images. In an ideal situation, a perfect mosaic can be generated using a series of rigid transformations on the different images. Because the world is round, the terrain being modeled is typically not flat; thus, geometric distortions and radiometric differences must be taken into consideration. Recent research in creating automated tools to orthorectify the imagery has resulted in toolkits that reduce the time required to make raw imagery useful. Most tools utilize stereo imagery pairs, and automatic stereo-matching algorithms are used to correlate data pairs. The data pairs are used to compute the seam lines for the mosaic and to compute the geometric-correction transformations. The rest of this paper concentrates on the details of this process.

2.1 Source Imagery

A revolution has occurred with the abundance of commercial high-resolution imagery from a variety of vendors. Data is now widely available from sources such as SPOT, Landsat, ORBIMAGE, Space Imaging, Earth-Watch, and even on the Web from Microsoft! It is literally possible to buy "spy photo"–quality imagery at prices that are less than the cost of aerial photo surveys.[3, 4, 5]

Aerial photography is by no means dead (one of the reasons the U.S. keeps U2s active and recently reactivated the SR-71s). For commercial applications on a small scale, aerial photography is still hard to beat. High-quality images can be taken quickly anywhere that can be flown over. It is possible to get stereo pairs, and different filters and film types may be used depending on the type of data sought. Typically, panchromatic black and white with a yellow (minus blue) filter is used to help eliminate the UV and the blue wavelengths scattered by the atmosphere. Color IR and natural color filters are also very popular. The problems with these methods include cloud coverage, the need to fly over territory that may be inhospitable, and limited range and loiter capability. The price/performance is hard to beat, however, and devices such as digital cameras and global positioning systems are making this technology easier to use and more efficient then ever.

Another breakthrough that has helped the data revolution is that the modeling, simulation, and imaging communities are all using similar commercial off-the-shelf computer hardware that is affordable and available for widespread use. Now, the challenge is getting high-quality imagery converted quickly and efficiently into visual databases. This new data paradigm is opening market opportunities beyond traditional training and simulation areas while shifting the visual database problem to storing and utilizing the imagery in an efficient manner.

2.2 Orthorectification

Orthorectification is a process of transforming the image into a desired map projection. Any satellite or aerial image represents the surface of the earth from exactly one perspective, which is where the sensor or camera was located at the time of collection. As a natural part of this process, the data is georeferenced. The convergence of technologies from GIS and modeling and simulation has produced commercial toolkits that greatly ease the pain of the orthorectification task, including toolkits from companies such as ERDAS, PCI, ER Mapper, etc. The orthorectification process is as follows:

- Convert imagery to digital format if necessary
- Collect ground control points
- Find fiducial marks to determine orientation of image
- Generate the camera model to calculate the exterior orientation
- Correlate the image with a digital elevation model
- Compute a transformation matrix
- Resample the imagery

Many data providers can orthorectify imagery as part of their service. Many can also give the data to users in any map projection required.

2.3 Map Projections

Map projections are attempts to draw the 3D earth on a 2D surface. The latest breed of image generators (such as the SGI® Onyx® 3000 series systems) is capable of rendering round earth models, but it is still necessary to project the imagery (agree or standardize on a projection). There are several ways to model the round world, including simple spheres and oblique spheroids. Because most applications utilize 2D maps, it is possible to easily create the visual database with a specific map projection in mind. Distortions of conformality, distance, direction, scale, and area always result from this process. Maps are made with different applications in mind (i.e., there will be an error, but a judgment must be made about which error is the most tolerable). Some projections minimize distortions in certain properties at the expense of others. For a good reference on map projections, see the University of Texas Web site.[6] Major terminology for map projections is as follows:

- Conformality: A map projection is conformal when at any point the scale is the same in every direction. Meridians and parallels intersect at right angles, and the shape of very small areas and angles with very short sides is preserved. The size of most areas, however, is distorted.
- Equal area: A map projection is equal area if every part, as well as the whole, has the same area as the corresponding part on the earth at the same reduced scale. No flat map can be both equal and conformal.
- Equidistant: Equidistant maps show true distances only from the center of the projection or along a special set of lines. No flat map can be both equidistant and equal area.
- Linear scale: Linear scale is the relation between a distance on a map and the corresponding distance on the earth. Scale varies from place to place on every map. The degree of variation depends on the projection used in making the map.

Other painful considerations include geodetic datums. There are hundreds of different datums used throughout the world. Different datums exist for many reasons, mainly due to the ever-increasing accuracy with which the earth can be measured. They have evolved from a spherical earth to a very precise ellipsoid. The reason these issues are important relates to data fusion. Fusing different types of data sources is the ultimate goal of battlespace visualization. When a projection is picked, it must be consistent with each type of data used. For example, if you project your imagery in WGS-84 and later want to overlay signal intelligence data collected by a different source, the projections have to match in order to correlate. Without all the parameters of the collection (i.e., camera models, geographic positions, etc.), you may not be able to reproject the data and obtain correlation. It is important to understand the datums before designing the system.

2.4 Image-Processing Techniques

Once the data has been collected and mapped onto a reference system, it is necessary to perform image-processing functions to make the best possible "clean data" from the source data. For a detailed explanation of various image-processing functions, refer to Schowengerdt (1997) and Gonzalez and Woods (1993).[7,8]

The following operations are most commonly used in processing the data:

- Histogram: The goal of histogram equalization is to help distribute image intensities more consistently. Histogram matching is useful when two or more images must be mosaicked together, but have different color combinations because they were taken on different days or have different sun positions. It is not useful when images come from different times of year and the content of the images is different (for example, one image has snow on the ground and the other does not). Some graphics hardware, such as the SGI Onyx 3000 series systems, has the ability to do texture lookup tables in hardware, which greatly accelerate the image-mapping function.
- Intensity transformation: A simple intensity transformation is creating an image negative. Other common transformations are gamma corrections. This is a nonlinear function.
- Convolutions: A convolution is a weighted sum of pixels in the neighborhood of the source pixel. The weights are determined by a matrix called the convolution mask, or kernel. Each element of the kernel is called a coefficient. The kernel is centered on a pixel, and each coefficient is multiplied by the pixel underneath it. All of these products are summed, and the total is divided by the sum of the kernel values. There are many different types of convolutions, and different ones are used depending on the filtering function to be performed on the source imagery. Common filtering functions and their uses are:
  - –High-pass filter, typically used for edge sharpening
  - –Low-pass filter, used to blur or average the image
  - –Median filter, used if there is noise or the image is corrupted and erroneous pixels must be replaced
  - –Min/max filtering, used for removing spikes in the image

Processing of large databases can be very computationally intensive and take hours to run, especially when done using standard computing architectures. It is possible to perform many of these advanced filters (convolutions) in graphics hardware, greatly speeding up the whole process. Taking advantage of graphics hardware in this way requires that image-processing applications are developed using OpenGL® application programming interface.

Many commercially available image-processing packages simplify the image clean-up process. Examples of this software include ERDAS IMAGINE, ER Mapper®, ESRI® ARC/INFO®, and PCI®.

2.5 Assembly and Tuning

Database generation begins once the image data is ready. This is another offline process where the needs of the particular mission profile are assessed and the appropriate 2D and 3D data are assembled. Typical simulation requirements are for high-resolution imagery around an airport and high-3D-feature content in low-level training corridors and target areas. Key way-points or other visual references may have to be merged. Common database-generation goals are to provide detailed imagery of sufficient quality to support the training activity, but only in the areas where it is needed.

The collection, processing, and maintenance of data that is of higher resolution than required by the training profile greatly increases the up-front costs of database construction. Run-time costs (amount of disk online, memory, and even CPU allocation) are also negatively impacted. It is advantageous to have a hardware platform that is scalable in all areas (disk, processors, memory, and graphics capability) to get the optimum price/performance. The database modeler must become a system architect and balance the needs of the visualization with what is possible from the database (including construction cost, source imagery availability, turnaround time, etc.) and what is possible from the run-time hardware producing the imagery.

3.0 Real-Time Processing

Section 2 concerned issues in creating the visual database. This function is performed offline and is usually a combination of interactive and batch processes. The hardware requirements for these tasks may be compute-intensive for functions such as convolutions without being time-critical. Low-cost Windows NT® OS-based, PCs or UNIX® OS-based workstations are well suited for these efforts.

Visualizing the resulting databases in real time is another matter. Typically, imagery-based databases start at 100GB and go up from there. Fusing the data with other information results in cluttered displays with thousands of overlaid symbols. Input and output can vary greatly. Well-known schemes to manage polygon loads have been developed, about which countless papers have been written. The goal is to be sure that every rendered polygon contributes significantly to the scene; otherwise, it is thrown out. On the other hand, little has been written about pixel and texel management, where one makes sure that every pixel and texel contributes significantly to the scene or else it is not rendered. Because pixel and texel processing become much more relevant with image-based databases, the following subsections discuss load management mechanisms for these scarce resources.

3.1 Pixel Fill Load Management

The color and visibility of every polygon on the screen must be calculated to a discrete picture element or pixel level. When polygons overlap, each visible pixel on the screen must be computed over and over before the actual rendered pixel is obtained. As thousands of unique pixels can exist in each polygon, this process is much more computationally intensive than transforming polygon vertices. Therefore, pixel fill rates are a much better indicator of system performance than polygon rates when filled and, especially, textured filled polygons are rendered. Methods to manage pixel rendering can greatly increase actual performance over pure benchmark numbers.

A unique feature of the SGI Onyx 3000 series with InfiniteReality3™ graphics that fits into this category is dynamic video resizing.[9] When enabled, this technique modifies the computed resolution of the scene and scans out an image that fits in the frame time. For example, if a 1280x1024 pixel display device is used and insufficient pixel writing capacity is available at high frame rates, the systems can reduce the computed resolution slightly to 1200x980 and still drive the display at the required 1280x1024. Note that the video clock stays the same, and 1280x1024 pixels still compose the image, but the effective resolution is 1200x980.[10] This feature can be varied on a frame-by-frame basis, so be sure that frame rate is maintained. Features like this are critical in fixed-frame-rate applications. Also, great flexibility can be obtained in a command post, where variable resolution displays can be mapped to any display device in the center.

3.2 Texel Load Management

For approximately 15 years, visual systems have been capable of using photo-realistic texture to increase apparent scene quantity and limit database modeling times. However, when not properly filtered, textures cause distracting visual anomalies due to the effects of digital sampling when polygons are rendered at an oblique angle. The use of photo texture became really useful when algorithms like mip-mapping became widely used. Creating mip-maps or producing down-sampled versions of the primary pattern and having the hardware automatically select the correct version in real time eliminates the scintillation evident without it. This hardware feature enabled database modelers to quickly use texture (commonly a photograph) overlaid on a polygon for greatly enhanced realism. For example, in the past modelers would avoid the impractical use of multiple polygons by "painting" a wall red rather than modeling the individual bricks in a brick wall. This method potentially removed so much detail that training effectiveness could be negatively impacted. The benefits are clear. The drawbacks are that every level of "mip" requires one-fourth the amount of storage of the level above it, or 1.3 times the size of the primary pattern. Usually, one wants these textures stored in dedicated texture memory so that the rendered polygons can be filled quickly enough for real-time performance. This greatly impacts the price of the rasterization hardware, as expensive SDRAM (high-speed memory) must be used. This means that if one is texturing a 1024x1024 pixel polygon and uses 4 m imagery, then a 256x256 mip-map would suffice. If one were to store three-component (RGB) imagery at 8-bit color resolution, then the highest level of mip would take 256 * 256 * 3, or 196,608 bytes of storage. However, if one wanted to utilize 1 m imagery, the storage requirements for the top level would be 1K * 1K * 3, or 3MB just to store one level! Even with a small 20 km by 20 km gaming area at this resolution, the texture memory capacities (and therefore the prices) would be prohibitive. Imagine a battlespace visualization application where one wants to visualize an entire theater with thousands of square miles of imagery.

Even more significant than texture memory are database paging requirements. If the viewpoint is placed high over the terrain and directed to look off toward the horizon, the number of textured polygons in the scene increases dramatically. Each of these polygons must have the entire mip-map structure paged from disk into the graphics pipeline, even though only the lowest level of detail will be visible throughout the scene and especially on the horizon. This is a huge waste of resource. The tradeoffs in this scenario are reduced resolution and update rate, much fewer and smaller high-detail areas, or not using geospecific imagery. Fortunately, there is a better way.

3.2.1 Clip-Mapping

Clip-mapping is a texture-processing algorithm implemented in the Onyx 3000 series with InfiniteReality3™ graphics. The clip-mapping algorithm is derived from the mip-mapping algorithm, but is unique in that unique levels of the clip stack can be rendered without requiring the entire clip stack to be loaded in texture memory. This greatly reduces online memory and paging requirements. High-resolution imagery is used only in areas where it is needed. A stack of clip-mapping is tightly coupled with the real-time software (and the hardware) to page the appropriate (rather than the entire mip-map) texture from disk into main memory and, ultimately, into the high-speed (but expensive and limited) texture memory. Clip-mapping is fundamentally virtual texture that is allocated based on the oblique angle of the polygon relative to the field-of-view frustum. It is a caching paging technique that swaps high-resolution imagery in and out at different mip levels. The maximum size of this virtual texture is 231 by 231 or 1 billion by 1 billion. Thus, if one clip-map was composed of 10 cm imagery (aerial photo based!), then the maximum size would be 10,000 km by 10,000 km. (Now that's a gaming area!)

It is also possible to have multiple clip textures active at the same time. One other major difference between clip-mapping and mip-mapping is that mip-mapping is all or nothing at each level. It is not possible to have mip levels that are only partially filled; thus, users who want to have 1 m high-resolution imagery around an airport, for example, need to fill the entire level with 1 m imagery. With clip-mapping it is possible to have detailed imagery where it is needed or where it is available. Data of varying resolutions may be used in the same clip-map. This makes database modeling easier and enables an elegant scheme for updating data when changes or higher-resolution data becomes available.

The optimum texture level to be used is one that supports a one-to-one correspondence between pixels and texels. For example, if a polygon in a simulator's visual system represents a real-world size of 100 m on a side and the polygon never gets bigger then 50 pixels wide when rendered on the screen, then 2 m imagery could be used. The optimum texture filter would apply a texture that is 50 texels wide for a polygon that is 50 pixels wide. Half-meter imagery would be a waste.

Clip-mapping supports multiple resolutions but requires custom hardware to deal with the complex filtering in real time. Clip-mapping works by implementing the notion of a clip center that is updated in real time. The appropriate texels are paged in from main memory to the high-speed texture memory. The high-resolution texture maps are divided into tiles, and the tiles are loaded as appropriate. The key is to page in the appropriate tiles before they are needed. This sounds difficult, but it is actually quite simple and not dissimilar to database paging, which has been widely used for more than a decade. Most training platforms simulate real-world devices that move in a deterministic fashion and are relatively predictable. For the clip-mapping scheme to work, it is essential that the system has the appropriate bandwidth between hard disk, main memory, and texture memory. Clip-mapping provides a placeholder for high-resolution data.

With clip-mapping a typical database-generation scenario would be as follows:

- Three-dimensional models are constructed in the traditional polygonal fashion with database modeling software (such as MultiGen).
- Polygonal terrain skins are also generated in the traditional method (as above).
- Imagery is "cleaned" (as described above) and the clip-map is constructed. For example, a flight simulation scenario may have an overall base imagery of 50 m. High-resolution corridors may require 10 m imagery; within 10 km of airports (or target areas) a 1 m imagery may be required.

Clip-mapping enables users to fuse data from different sources (sensors). For example, sources today might include satellites for the overall base imagery and aerial photographs for high-detail insets. In the military world, UAVs, RPVs, drones, etc., are gaining widespread acceptance as a low-cost tool for augmenting data collection. With clip-mapping, this imagery can be fused to make one seamless, contiguous data set that can later be viewed in real time.

It is still possible to have an overload condition on an image generator equipped with clip-mapping, but there are more graceful degradation functions that help the image generation system do the right thing. For example, if a system was architected to support a training scenario where an aircraft was flying at 500 knots and 100 m, it may be optimal to have 2 m imagery. If a new aircraft system was added such that a mission profile required a 500-knot speed but at a 50 m altitude, 1m imagery may be necessary for maximum realism (and, consequently, maximum training value). One major benefit of clip-mapping is that it is possible to revisit the old database and burn in new 1 m data to support this new training need. With clip-mapping one does not have to re-architect the whole database and start over; in fact, it is possible to literally try it and see what happens. The clip-mapping algorithm always loads the coarser texture maps needed first and radiates from the clip center, loading as much fine-level texture as possible in the available time. This means that if the user architected and bought a system that was tuned for a maximum of 2 m imagery, then there might not be enough time to load all of the 1 m imagery. The visual artifact of being out of time rather than being a stalled image generator or a missed frame may simply be more blurry texture in the periphery. If one is doing 500 knots at 50 m, one probably will not be looking at the periphery anyway.

In order to take advantage of features like clip-mapping, custom high-speed dedicated hardware (InfiniteReality® graphics) was built and is now available. InfiniteReality® supports handling algorithms that are engineered into the Application-Specific Integrated Circuits to perform a variety of functions such as tile edges, filtering, and determining what to do if the data is not there when the fetch occurs. Other specialized areas of sophistication are maximum bandwidth between hard disk and main memory and main memory to texture memory. Feeding the pipeline is another major issue. Without sophisticated multi-processing and high bandwidth among memory, CPU, and the graphics pipe, features like clip-mapping cannot be implemented on large data sets. Although commodity graphics technology is advancing rapidly, the underlying PC platform technology will limit applications at this fidelity for years to come.

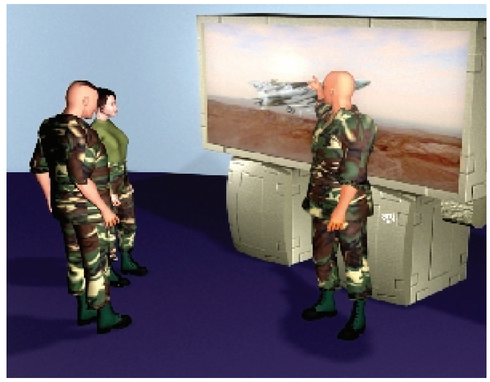

Fig. 3

Fig. 4

Fig. 5

4.0 Repurposing Visual System Databases for Battlespace Visualization

Developing a high-fidelity, image-based database can be an expensive proposition. The potential to reuse the same database for intelligence and operational uses makes the investment decision much easier. As mentioned earlier, if properly constructed, this same database can be fused with different data sources to produce a variety of battlespace displays. For example, using products from ESRI, displays of the underlying GIS layers can be superimposed over the photospecific 3D representation. High-resolution video can be projected over the underlying terrain. Military symbology can be used to indicate the positions and status of friends and foes. Two-dimensional map displays can be overlaid on top of 3D terrain or switched with the underlying terrain. Elaborate modeling of chemical and biological dispersions can be programmed and displayed over the terrain. Almost any combination can be derived based on the requirements.

Once the requirements are defined for the actual appearance and content of the displays, many choices are available for a display environment. In some cases, a simple monitor is sufficient. However, many new large-format options are available, primarily targeted to the concept of collaborative visualization. Examples of these are shown in figures 3, 4, and 5.

5.0 Conclusions

The commercial availability of 1 m image data is beginning to change the paradigm in image generation systems. Concurrently, this same technology can be used to create powerful battlespace visualization systems. As the level of functionality available from well-designed, image-based databases and flexible, general-purpose hardware increases, the two technologies begin to merge. Simulators turn into mission-rehearsal devices. Operational command-and-control assets can now be used for simulation. With convergent technologies from the image-processing world, prices will decrease for hardware, and software capabilities will increase. There will be increased automation of database generation and more commonality and benefit derived from proximity markets such as GIS and image processing.

Programs such as Distributed Mission Training (DMT) in the U.S. are pushing mission-rehearsal technology into the mainstream and challenging industry to find a better way. DMT also looks to use the same devices for command-and-control. This merging of technologies will be a watershed event for the command and control community, which will benefit from years of development in synthetic environments on the simulation side. The result will be true information dominance.

6.0 Acknowledgments

The authors would like to acknowledge the special contributions made by Michael Jones in pushing to develop the technology and by Mark Aubin in clarifying the terminology. Also, many thanks to Luis Barcena for developing technical reports. Thanks go out as well to all of the ASD engineering staff at SGI for implementing clip-mapping, building the SGI Onyx 3000 series hardware on which it runs, and continuing to push the envelope.

7.0 Author Biographies

Graham Beasley has been in the simulation industry for approximately 15 years. Currently, he is a member of the management team at the SGI Mountain View, California, headquarters. He has worked at SGI for the past nine years and has specialized in the modeling and simulation areas. Currently, he manages two engineering teams (High-End Graphics and Modeling, Simulation, and Imagery). He has published numerous articles and presented at I/ITSEC, ITEC, IMAGE, SIGgraph, and many other simulation-related conferences.

Before coming to SGI, Mr. Beasley worked at CAE Electronics in Montreal in its visual systems group. His specialties include image generation, real-time simulation, visual database, networked simulation, and helmet-mounted display technologies. He received his bachelor's degree in computer science from Concordia University.

John Burwell has been in the defense and intelligence industry for 17 years. Currently, he is the director of the Government Industry team at the SGI Mountain View, California, headquarters. He is based in Washington, D.C. He has worked at SGI for the past eight years and has specialized in defense solutions. He has published numerous articles and presented at a variety of defense-related conferences.

Prior to working for SGI, Mr. Burwell was with Evans & Sutherland in Salt Lake City, Utah, as a program manager and with Boeing and McDonnell Douglas in a variety of engineering and management roles. He received his bachelor's degree in electrical engineering and computer science from the University of Colorado, Boulder, and his master's degree in international management from the American Graduate School of International Management (Thunderbird).

8.0 References

1. Cosman, Michael A. 1994. Global Terrain Texture: Lowering the Cost. In Proceedings of Image Society 94, 52–64.
2. Liddiard, Todd. 1996. Manipulation and Refinement of Digital Source Data for Ground Warfare Databases. In Proceedings of SimTecT 96, 137–42.
3. Landsat: (*http://edcwww.cr.usgs.gov/Webglis/glisbin /search.pl?LANDSAT_TM*)
4. SPOT: (*http://spot4.cnew.fr*)
5. Space Imaging: (*www.spaceimaging.com*)
6. Map projections: (*www.utexas.edu/depts/grg/gcraft/notes/mapproj/mapproj.html*)
7. Schowengerdt, Robert A. 1997. Remote Sensing. 2nd ed. San Diego, Calif.: Academic Press.
8. Gonzalez, Rafael C., and Richard E. Woods. 1993. Digital Image Processing. Reading, Mass.: Addison-Wesley.
9. Silicon Graphics Onyx2 Technical Report. 1998. Silicon Graphics.
10. Burwell, John. 1996. Redefining High-Performance Computer Image Generation. In Proceedings of Image Society 96.

Corporate Office
1600 Amphitheatre Pkwy.
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

U.S. 1(800) 800-7441
Europe (44) 118-925.75.00
Asia Pacific (81) 3-54.88.18.11
Latin America 1(650) 933.46.73

Canada 1(905) 625-4747
Australia/New Zealand (61) 2.9879.95.00
SAARC/India (91) 11.621.13.55
Sub-Saharan Africa (27) 11.884.41.47

3279 (05/02) J13672