Leveraging Graphics Processor Capability to Uncover Hidden Terrain Database Errors

Tim Palmer
TransLucid, Inc.
Huntsville, AL
tpalmer@translucidinc.com

Ryan Boyd
Science Applications International Corporation (SAIC)
Orlando, FL
ryan.boyd@saic.com

ABSTRACT

At the heart of most training and analysis systems used by the warfighter is a consistent, realistic, and valid visual and topologic representation of the terrain; visual terrain features such as rivers and roads; and visual models of vehicles, trees, and buildings. Although the industry has made significant advances in standardization, multiple simulation systems when interoperating still require multiple runtime terrain formats. Even without the interoperability use case, the simulation engineering community is often faced with datasets of the same terrain in many different formats. Achieving sufficient correlation between these multiple representations of the terrain is a necessary condition to support the warfighter, but just what is sufficient? Most existing methods of terrain correlation rely on imperfect assumptions, are manpower-intensive and time-consuming (and thus error-prone), and are even somewhat ad hoc. Using advances in processing power, especially those in graphics processing unit (GPU) technology, we examine in detail a range of possible terrain correlation problems in both elevation and line of sight (LOS), two key measures of correlation. This paper explains both the basic methodology behind this advanced terrain correlation testing and summarizes quantitative results. Using these techniques, our detailed examination of very large amounts of data in multiple datasets reveals potentially significant and hitherto uncovered problems in terrain correlation. A more detailed understanding of these problems is expected to provide insight into the quality of existing databases, the impact on training effectiveness of inadequately tested terrain databases, and even how very large datasets can be compared for other correlation problems.

ABOUT THE AUTHORS

Tim Palmer is president of TransLucid, Inc., an engineering and software development company in Huntsville, Ala., specializing in real-time graphics, GPU processing, and sensor simulation. Prior to founding TransLucid, Palmer was co-founder and vice president of CG2, Inc. and chief technical officer at Quantum3D, Inc., where he developed commercial software products for the visual simulation and training market. Mr. Palmer has a B.S.E.E. degree from the University of Alabama in Huntsville, where he also managed sponsored post-graduate research, including the development of terrain processing and visualization systems and advanced visualization technologies. In 2010, Palmer had a working relationship with SAIC to develop a terrain visualizer and correlation test tool.

Ryan Boyd is an SAIC simulation database engineer. He helps lead the design and creation of simulation database products for government and commercial customers. Prior to his work with SAIC, he worked as a systems and integration engineer on the Synthetic Environment (SE) Core Database Virtual Environment Development (DVED) program. His responsibilities included database integration, testing and delivery of SE Core databases on the confederate training systems, including the Close Combat Tactical Trainer (CCTT), Aviation Combined Arms Tactical Trainer (AVCATT) and the Common Driver Trainer (CDT).

Leveraging Graphics Processor Capability to Uncover Hidden Terrain Database Errors

Tim Palmer
TransLucid President
Huntsville, AL
tpalmer@translucidinc.com

Ryan Boyd
Science Applications International Corporation
Orlando, FL
ryan.boyd@saic.com

BACKGROUND

Today's warfighters require fully realistic, exact and accurate representations of the terrain and all of its features in their training and analysis systems to ensure a "fair fight." In order to ensure a "fair fight," correlated data and output formats are needed. The modeling and simulation industry continues to struggle with this problem. These errors can be a result of a wide range of issues. Source data, software design, and format specifications are just a few of the many potential areas where issues could be introduced. Many attempts have been made at solving the correlation problems in terrain data representations.

Type of Correlation Testing

The identification of errors is a result of correlation testing which is currently performed in two ways.

First, tools have been created to test one or more output formats against each other. Examples of these tools are Side-by-Side™ (CAE, 2011), SEE-IT (Synthetic Environment Evaluation – Inspection Tool) (SEDRIS, 2006), Zcap (UCF IST, 2011), and OneSAF® (Department of the Army) ATT/VTT (Automated Test Tool/Visual Test Tool). There are many pros and cons to each toolset, but both individually and collectively, they are limited in test scope. For example, Side-by-Side can consume multiple output formats, but there is no automation, so it is a quasi human-in-the-loop solution. Another example is SEE-IT. SEE-IT can conduct a consistency check on data, but it is done on an intermediate format that is internal to the tool. As a result, it is not able to verify the data from individual output formats. Those are just a sampling of the tools available to test correlation.

The second way to test correlation is human-in-the-loop. Many programs (PEO STRI, 2011) (Campbell, 2011) (RDECOM-STTC, 2011) (UCF, 2011) have written test procedures to test correlation on their systems. These test procedures lack the thoroughness and quality that is needed to verify that there are no "fair fight" issues in training databases. Correlation tests such as these include manual "fly-to" and "fly-through" to prior determined locations to verify positional correlation for each database format. This is generally done by visual inspection of the output scene. There are limitations to this approach since some of the output formats do not have a visual element that can be inspected, the overall location sampling is small compared to the size of the training databases, one format may break the rules and ignore the data from other formats (for example, image generator (IG) terrain clamping), intensive manpower requirements, and susceptibility to human error.

Because of the difficulty of comprehensive testing, some have asserted (Presagis, 2011) (SE Core, 2011) that using the same source data ensures correlation among the various and disparate formats. There are weaknesses in this logic, and even if accepted, it only solves one piece of the correlation problem. Software can read common source data, but software tool processing necessarily modifies the source data, introducing the possibility of correlation degradation. Some output formats require the re-projection of data or the remapping of feature types. These types of processes can also have a significant negative impact on correlation.

CORRELATION CHALLENGES

Exhaustive and automated terrain correlation testing presents several intrinsic challenges.

Multiple formats

One goal is to test published output terrain databases in their native formats. There are many terrain formats used in the simulation and training industry, each of which requires unique software to read and transform the terrain data into a format usable for comparison testing. A sample listing of frequently encountered formats includes OneSAF Object Terrain Format (OTF) (OneSAF, 2010), paged and non-paged OpenFlight (OF), TerraPage (Presagis, 2009), and Virtual Battlespace 2 (VBS2) (Bohemia, 2011).

Coordinate systems

Each terrain database represents spatial locations in a particular coordinate system. Some formats support only a single coordinate system (such as OTF and VBS2) while others (such as OF) allow a choice of coordinate system. Since the primary goal of correlation testing is fundamentally to compare terrain in various formats, each terrain must be converted to a common coordinate system. This conversion process, itself, can present certain challenges. Projection precision, for example, both in the original terrain publishing and in the test tool can introduce correlation differences due to translations in the position of terrain skin vertices or three-dimensional (3D) features. It is also possible for large polygons to introduce elevation errors when projected, since they are intrinsically "flat" (shown exaggerated in Figure 1).

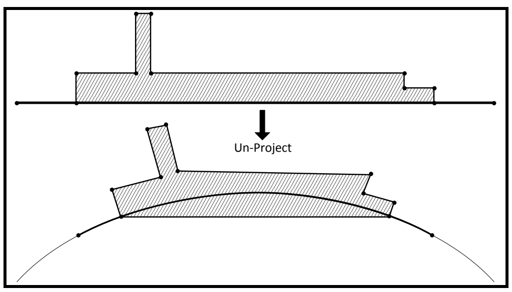

Figure 1. Distortion caused by projection change

DEM Thinning and Triangulation

Even though a single digital elevation model (DEM) source is used for the terrain skin polygon model, the publishing process may "thin" the elevation data in different ways for each database format. For example, some formats (e.g., VBS2) use a regular grid triangulation (Bohemia, 2011), while others (e.g., OTF and OF) use an integrated triangulated irregular network (ITIN) (see Figure 2) (Presagis, 2009) (OneSAF, 2010). The dissimilarity of these approaches can result in elevation correlation differences.

Level of Detail Handling

Terrain database formats may support the concept of level of detail (LOD) for reducing memory footprint and rendering load for distant objects and terrain. Differences in representation due to LOD handling can affect, for example, line-of-sight inter-visibility during training.

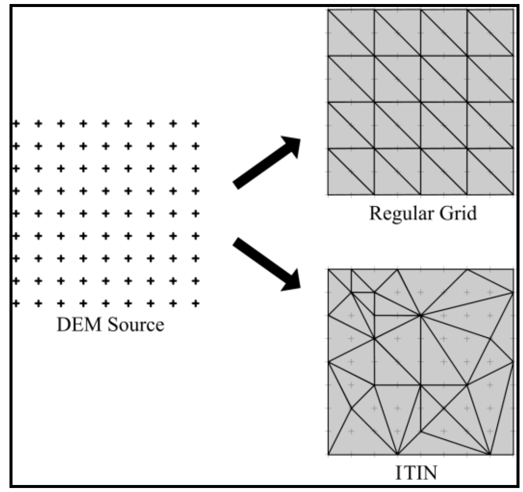

Figure 2. DEM triangulation methods

Feature Coding

Some formats, such as OTF and OF, allow the embedding of feature codes into the terrain database (OneSAF, 2010). Since this is not always needed for out-the-window visual simulation terrains, feature codes are often omitted. Even when feature codes are present, the environment data model (EDM) used may vary between terrains (e.g., Feature Attribute Coding Catalogue (FACC) versus Environmental Data Coding Specification (EDCS)).

Feature Representation

Terrain databases usually include 3D features such as trees, buildings, and bridges. The description and representation of features varies between database formats.

As an example, consider individual trees. OTF stores a parametric definition of individual trees using EDCS feature codes and attributes. This provides a rough definition of the tree geometry (e.g., trunk height and diameter, crown height and diameter) and metadata including the species and occlusion parameters (see left side of Figure 3). By contrast, OF terrains usually reference external or embedded polygonal models that may be as simple as a billboard with a texture with transparency cut-outs or as complex as individual limbs and leaves (see right side of Figure 3). OF, by itself, does not include all of the metadata present in the OTF.

This difference in feature representation complicates correlation testing, especially line-of-sight inter-visibility testing.

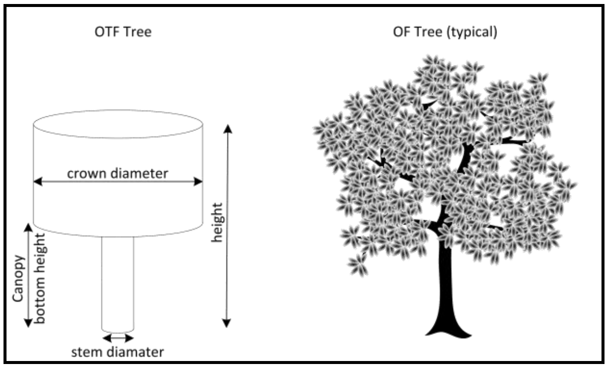

Figure 3. OTF and OF Trees

Data Quantity

Training databases are often very large, both in terms of geographical coverage and density. The large size makes it difficult to manually inspect by fly-through of the terrain, even using a side-by-side technique.

The size and density can also present a memory resource problem, both for the central processing unit (CPU) and the graphics processing unit (GPU), if the entire database has to be loaded for comparison. Paging techniques can usually be used to overcome most or all of this difficulty, however.

TESTING METHODOLOGY

The primary goal of correlation testing is to compare two or more terrain databases and provide a quantitative output report that can be used to locate and correct miscorrelations, if deemed necessary. To achieve this goal several major processes and requirements were identified.

Multiple Formats

Comparison testing requires multiple terrain databases to be loaded simultaneously. A flexible testing tool should have extensible support for new terrain formats. Software plug-in modules, one for each format, provide the required flexibility and allow new formats to be easily integrated over the test application life cycle.

Exhaustive Terrain Coverage

Unlike ad hoc fly-through testing, this project is intended to exhaustively test the entire terrain database coverage area. Modern CPUs, GPUs, memory architectures, and storage technologies make iterative, brute-force testing possible and a compelling choice.

2D/3D Preview

While the testing should be automated and exhaustive, there are times when ad hoc fly-through and inspection can be very helpful, especially when reviewing the uncovered correlation errors. A 2D (top-down) and 3D viewing capability, with user camera control, is necessary for such inspection.

GPU Acceleration

For automated correlation testing to be relevant in a build-test-fix cycle, the test phase must provide timely results. The parallelism achieved by the modern GPUs can greatly accelerate the large number of computations required for correlation testing (Feldman, 2010)). Visual terrain formats, including OF and VBS2, are intended to be rendered using modern GPUs. They already include data structures and formats that are readily consumed by GPUs, further suggesting the GPU as a natural and convenient choice for accelerating correlation testing.

Feedback to Build Process

The purpose of correlation testing is to locate, quantify and, if required, fix correlation errors. The testing process requires an efficient means of communicating correlation errors back to the terrain database building process, where they can be further evaluated and fixed if deemed necessary.

Accuracy

The accuracy correlation test results are affected and, in part, governed by the accuracy of the underlying mathematics. Since terrain database formats may require re-projection to facilitate automated comparison, the accuracy and robustness of the coordinate system conversion is, thus, of key importance.

TECHNOLOGIES EVALUATED AND USED

Several technologies were evaluated for the automated testing project.

Windows, WPF, and C#

Microsoft Corporation's Windows® Presentation Foundation (WPF) was evaluated and selected for high-level coding, primarily for required graphical user interfaces (GUIs) (Microsoft, 2010). GUI prototyping and development has been greatly accelerated by the simplicity of WPF. The associated C#® (Microsoft Corporation) language, and its associated common language runtime (CLR), allows interoperability with native C++ code modules, which is important since some of the other selected technologies (e.g., OpenSceneGraph, SEDRIS application programming interfaces (APIs)) include native C++ components.

SEDRIS SRM

The SEDRIS spatial reference model (SRM) was selected for coordinate system conversions, including terrain data re-projection. The SEDRIS SRM uses proven and robust algorithms for coordinate conversion, and is a de-facto standard in the simulation and training community (SEDRIS, 2009).

SEDRIS EDCS

The SEDRIS Environmental Data Coding Specification (EDCS) was selected as the standard for EDM (SEDRIS, 2005). The SEDRIS EDCS Mapping APIs include convenient mappings between EDCS and FACC, and OneSAF defines a mapping between EDCS and the OneSAF EDM.

OpenSceneGraph

The OpenSceneGraph (OSG) open-source rendering library was evaluated and selected for 2D and 3D rendering. OSG supports Windows, Linux® (Linus Torvalds), and other operating systems, and interoperates with other key technologies such as OpenFlight and C for Graphics (NVIDIA Corporation's Cg®) (OpenSceneGraph, 2010).

Cg

Leveraging the GPU for accelerating correlating testing requires programming for the GPU in a high-level language (NVIDIA, 2003). NVIDIA Corporation's Cg language was evaluated and selected. Cg is well supported by multiple graphics card vendors, provides all of the required language features, and readily interfaces with OSG.

XML

Data interchange with the upstream terrain build process is required to provide feedback of correlation test results. The ESRI® (Environmental Systems Research Institute, Inc.) shapefile file format and Extensible Markup Language (XML) were evaluated as possible output formats. XML was ultimately selected, primarily because it is widely supported by a variety of external software tools (W3C, 2008).

PROTOTYPE DEVELOPMENT

A prototype application, SAIC's LightBox Terrain Correlation Testing Prototype, was designed and developed for conceptual testing. It is a Windows 64-bit application written in C# and C++. C# WPF is used for the GUI (see Figure 4), and C++ is used for rendering and test management. C# and C++ communicate using CLR interoperation techniques, including a managed "wrapper" library.

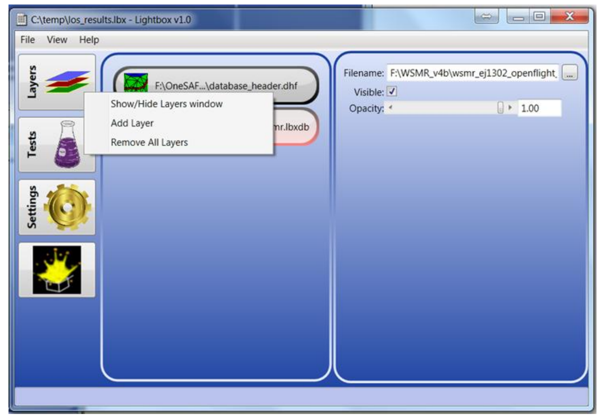

Figure 4. Prototype User Interface

A "layer" concept was implemented, where a layer is either a terrain or a test output overlay. Test overlays are 3D scene-graph files that are generated during testing and overlay the terrain rendering to highlight areas where correlation issues were detected. Each layer can be inspected in 2D (see Figure 5) or 3D modes (see Figure 6).

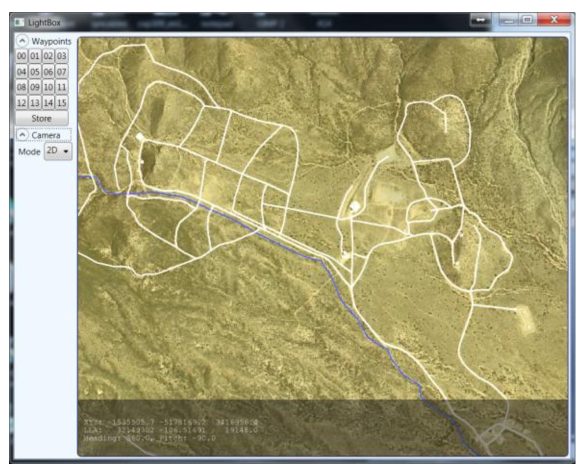

Figure 5. 2D Preview

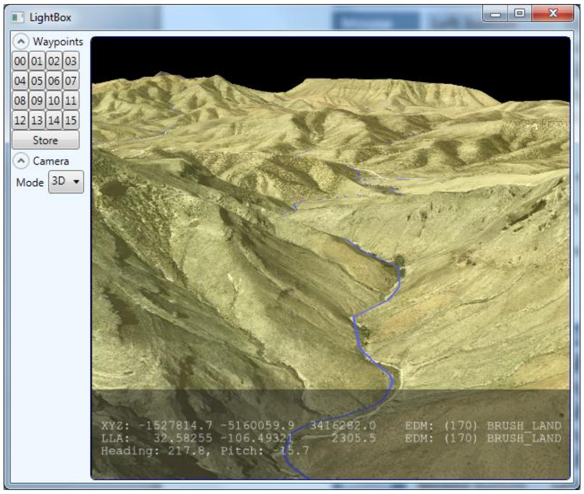

Figure 6. 3D Preview

To assist in investigating correlation issues, terrain and overlay layers can be shown or hidden, and the layer opacity can be adjusted. Also, terrain layers can be rendered in wireframe and "point" modes (see Figure 7).

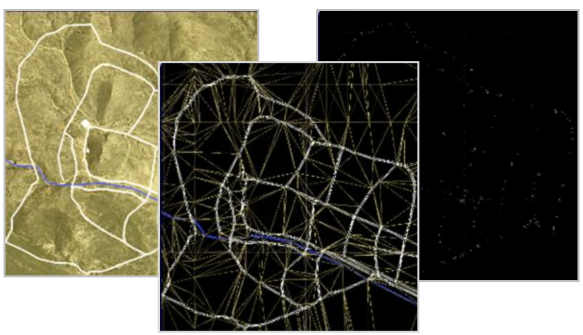

Figure 7. Solid, Wireframe, and Point Rendering

Testing

SAIC's LightBox Terrain Correlation Testing Prototype allows any number of tests to be created and configured using the GUI (see Figure 8). When testing is activated, all enabled tests are executed serially.

The target test region is divided into a grid of cells based on the defined test parameters and available GPU resources. Testing proceeds on a cell-by-cell basis. The terrain paging subsystem pages in the minimum amount of each terrain required to fill the cell. A Hilbert space filling algorithm is used to optimize cell processing sequence in order to minimize paging resources such as disk access, memory footprint and fragmentation, and execution time.

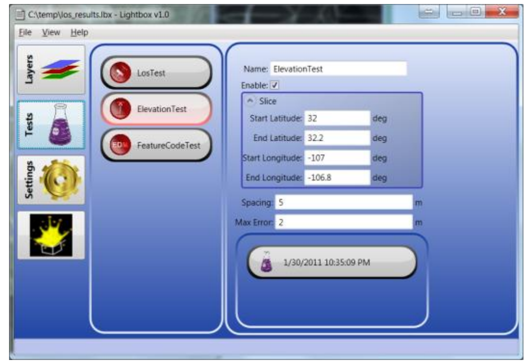

Figure 8. Sample Test Configuration

During testing, a GUI is displayed showing real-time progress of the test (see Figure 9).

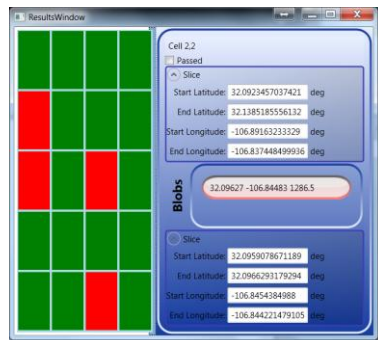

Figure 9. Sample Real-Time Progress Display

A green cell indicates no correlation errors for the corresponding cell. A red cell indicates one or more correlation errors found. Untested cells (not shown above) are colored gray.

The real-time progress display also shows the location of uncovered correlation errors (labeled "Blobs" in the figure) and allows the user to quickly navigate to the blob center in the 2D or 3D preview display.

When the test completes, an XML file is created containing the location and description of correlation errors. A test overlay scene-graph file is also created. This file can be loaded into SAIC's LightBox Terrain Correlation Testing Prototype as a layer and used to visually locate correlation errors (see Figure 10).

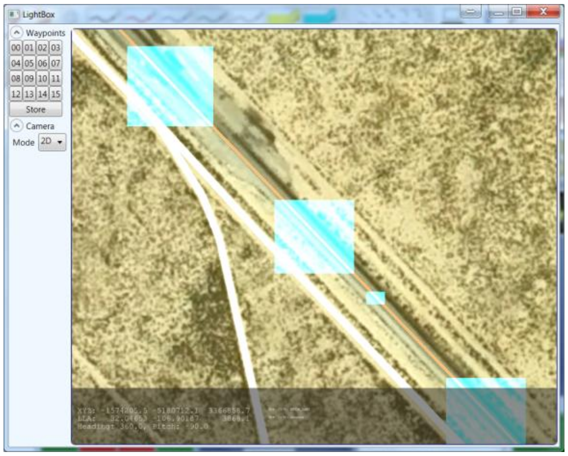

Figure 10. Test Result Overlay (cyan color)

Terrain Loaders

The prototype implementation includes support for paged and non-paged OF and OneSAF Objective Terrain Format version 8 (OTF 8) terrains. The C++ loader framework uses plug-ins to add new formats, allowing more formats to be easily developed when needed.

GPU and CPU Utilization

In addition to using the GPU for 2D and 3D preview rendering, the GPU is also used during test execution. Various tests are likely to require similar steps in their execution. To avoid redundancy, a number of generic GPU and CPU operation "passes" were developed. Examples include depth rendering, data compositing, histogram generation, error point consolidation (i.e., blobbing), and XML result output.

Passes can be created, configured, and connected as needed to define the operation of a given test. The C++ pass framework is modular, allowing new passes to be developed as needed to support new test capability.

PROTOTYPE TESTS

The prototype application includes support for three test types: elevation, feature code, and line-of-sight (LOS). These were chosen to prototype because of their utility and the difficulty of performing them by other, conventional, means such as terrain fly-through.

Elevation Test

The elevation test compares the elevation of all terrains using a fixed grid sampling. The minimum (min) and maximum (max) elevations are found and the difference is computed. If the elevation difference exceeds a user-specified threshold, a correlation error is declared for the grid point. This is illustrated for two terrains in Figure 11, where the (A) and (B) lines represent the terrain elevations of terrain A and B, respectively.

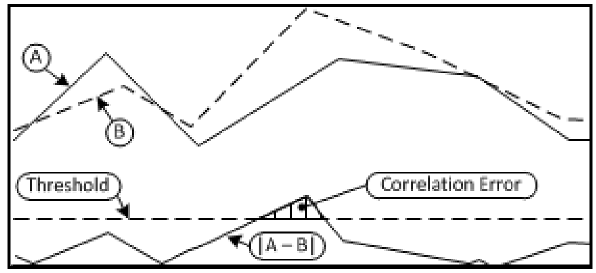

Figure 11. Correlation Error (for Terrains A and B)

The inputs to the terrain test are:

- Test region (minimum/maximum geodetic location)
- Grid size (meters)
- Elevation error threshold (meters)

Feature Code Test

The feature code test is used to find differences in feature coding. As with the elevation test, this test is performed on a regular grid. However, the grid spacing for this test is typically smaller to account for smaller linear features such as roads and streams. To accommodate different EDMs used by each terrain, a feature code mapping is performed in the GPU to convert input terrain feature codes to output EDCS codes.

The inputs to the feature code test are:

- Test region (min/max geodetic location)
- Grid size (meters)

Any difference in code is considered a correlation error, so no difference threshold is required.

Line-of-Sight Test

The purpose of the LOS test is to determine whether there are differences in visibility range for each terrain. A brute force test strategy is not practical in this test because of the vast amount of processing required. To thoroughly test LOS, a large number of "rays" in a 360-degree sphere would require testing for every point on or above the terrain. Instead, SAIC's LightBox Terrain Correlation

Testing Prototype tries to optimize testing by computing test points that are likely to uncover LOS correlation errors.

Test Point Selection

The first step in choosing testing points is to locate all vertical obstructions, such as buildings, trees, and bridges. A "depth complexity" GPU rendering pass is used to accomplish this. The efficacy of this algorithm depends on the fact that vertical obstructions sit atop the terrain skin and have (somewhat) horizontal polygonal faces. A vertical line segment would then pass through (black dot in Figure 12) only one polygon face where there are no obstructions, but more than one when there is a vertical obstruction present.

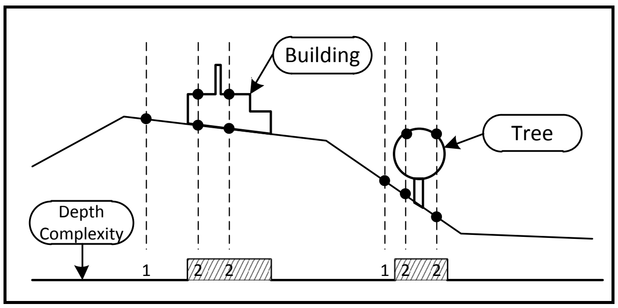

Figure 12. Depth Complexity Determination

An actual output of this process is shown in Figure 13. The white areas of the figure indicate where the depth complexity is one and, thus, there are no vertical obstructions present. The black areas indicate where the depth complexity is greater than one, indicating the presence of a vertical obstruction. In this figure, the obstructions are primarily buildings and trees.

Figure 13. Depth Complexity (white=1, black= >1)

The next step is to group nearby obstructions into rectangular "blobs", as shown in Figure 14. This step is performed in the CPU, after reading back the image shown in Figure 13 from the GPU. The algorithm groups dark images into rectangular regions based on their proximity to each other and pre-defined thresholds. Grouping the points into blobs simplifies the process of test point planning by reducing the number of considered regions.

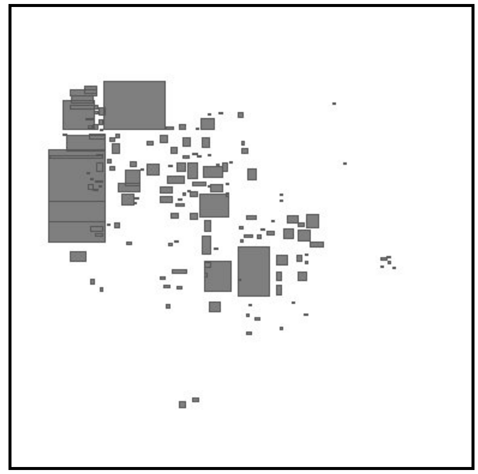

Figure 14. Blobbed Vertical Obstructions

The next step is to choose points around the perimeter of the obstruction blobs where testing will occur. The number of points around the obstruction is proportional to how near the obstruction is to its nearest neighbor, since the likelihood of LOS differences is increased. Finally, the total test point count is reduced by eliminating those that are too near to a neighboring test point to provide useful information. Figure 15 shows test points as red dots.

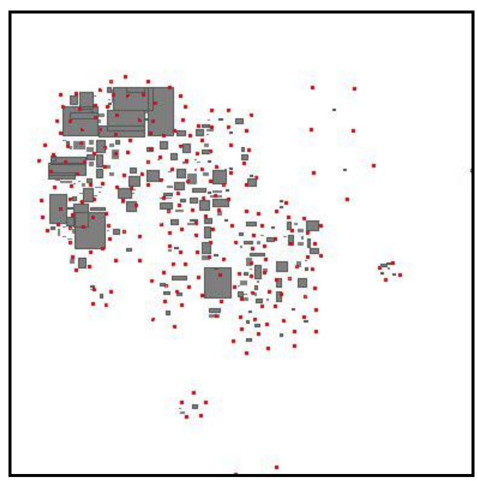

Figure 15. Final Test Point Locations

As can be seen in Figure 15, a majority of the terrain area does not need to be tested, since vertical obstructions are statistically rare in most regions.

Test Process

Testing consists of determining the nearest object along a segment with an origin that is at the test point and extending to a configured maximum range. A full 360-degree azimuth by 120 degrees elevation region is covered, with configurable spacing of points. The elevation slices extend from the zenith to 30 degrees below the horizon. The reason for extending below the horizon is to accommodate test points that are located on hillsides (see Figure 16).

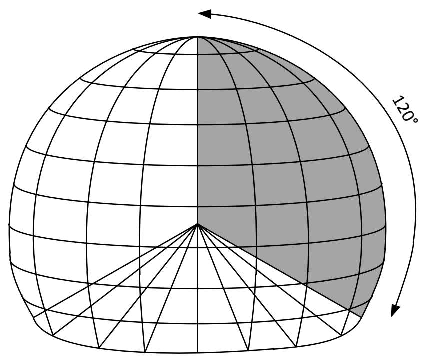

Figure 16. LOS Test Volume

Inputs to the LOS test include:

- Test region (min/max geodetic location)
- Test planning grid size (meters)
- Min/max perimeter points (per blob)
- Min test point spacing (meters)
- Min/max distance from perimeter (meters)
- Max LOS range (meters)

The test overlay output for LOS tests is different from the rectangle output for the elevation and feature code tests, owing to the nature of the test. Figure 17 shows an LOS overlay. The blue sectors indicate the directions, relative to the test point location, in which LOS correlation errors were detected. The GUI can be used to place the observer at the test point, looking through the error sector (see Figure 18).

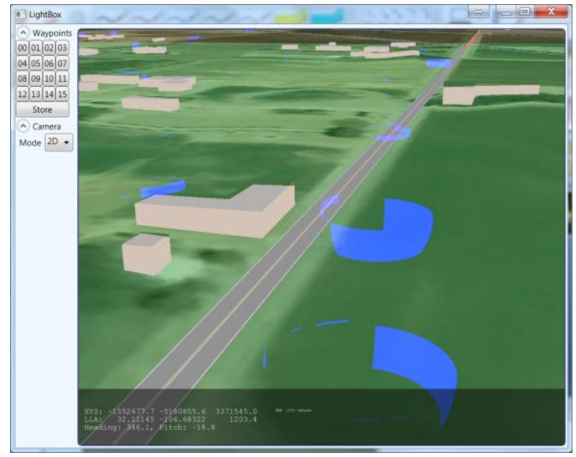

Figure 17. LOS Overlays (viewed from above)

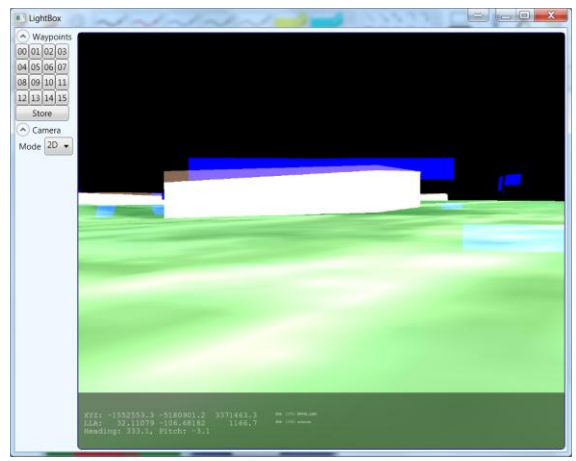

Figure 18. Looking From LOS Test Point Toward Error (Note difference in building elevation)

RESULTS

Prototype test results were very promising, proving the validity of the concept, at least for the prototype test types. Tests were executed on a typical gaming computer. Correlation tests between two or more terrains in OF and OTF formats were performed.

Typical results when comparing two terrains are shown below (Table 1).

Table 1: Typical Results for Two Terrains

Test Type	Region	Resolution	Time
Elevation	1 Geo-cell	1 m	10 minutes
Elevation	1 Geo-cell	10 m	4 minutes
Feature Code	1 Geo-cell	1 m	11 minutes
Feature Code	4 Geo-cell	1 m	38 minutes
Line of Sight	1 Geo-cell	1 m	29 minutes

The elapsed time required for performing the prototype correlation tests is encouraging, especially since the tests are exhaustive in nature. The one-meter resolution tests produce over 10 billion test points, and 10-meter resolution tests produce over 100 million test points per geo-cell.

Although no timing results were available, the authors estimate the time required to accomplish the same level of testing using legacy techniques and applications to be on the order of 100 times longer than those achieved by the techniques and application presented in the paper.

The times are especially small when compared to the time required to build and publish the input databases. Given that even these prototype tests provide exhaustive and quantitative results in a relatively short period of time, it seems natural to integrate such testing into the standard database building process.

FUTURE ENHANCEMENTS

SAIC's LightBox Terrain Correlation Testing Prototype was very successful in uncovering correlation issues between various terrain databases. Several enhancements have been identified that would further improve the automated correlation testing process.

Additional Terrain Formats

The prototype is capable of loading any terrain format supported by OpenSceneGraph, and OneSAF OTF 8 terrains. More formats could be added, including those identified earlier such as VBS2. In addition to published terrain formats, adding the ability to load Geographic Information System data files such as digital elevation models (DEM) and geospatial tagged image file format (GeoTIFF) would allow comparison of published terrains against source data, and even between various source data (i.e., "raw" and "cleaned").

Feature Code Translation

The prototype application translates all feature codes from their native EDM into EDCS. When investigating uncovered correlation errors, it may be convenient to report uncorrelated feature codes in the test terrain's native EDM (e.g., FACC or OneSAF EDM).

ESRI Shapefile Output

In the prototype, correlation errors were output to an XML format file. While an external tool can be used to convert the XML to ESRI shapefile format, it may be more efficient and convenient for the test application to produce ESRI shapefiles directly.

LOS Test and Point Features

The difference in tree vegetation representation between OneSAF and OpenFlight databases can cause undesirable "false positive" correlation errors in the LOS test. A means of modifying the test to accommodate differences in tree representation could reduce these spurious results. OneSAF inter-visibility calculations must be thoroughly investigated for this to be possible.

REFERENCES

Bohemia Interactive. *VBS2 VTK Developer's Reference* (section 6.16.3 – *Terrain Size and Resolution Requirements)*. Retrieved June 2011 from: http://developer.vbs2.com/onlinehelp/Content/Visitor4/Terrain_Size_Resolution.htm

CAE. *Side-by-Side*. Retrieved September 2010 from http://www.cae.com/en/military/side.by.side.asp

Chuck Campbell, Kevin Wertman, Julio de la Cruz . *A Framework for Generating High-Fidelity, Interoperable Urban Terrain Databases.* Retrieved May 2011 from: http://ntsa.metapress.com/link.asp?id=4mb469jegc8trmay

Michael Feldman, HPCwire Editor. *GPGPU Finds Its Groove in HPC*. Retrieved October 2010 from: http://www.hpcwire.com/hpcwire/2010-09-21/gpgpu_finds_its_groove_in_hpc.html

Microsoft Corporation (2010). Visual C# Developer Center (website). http:// http://msdn.microsoft.com/en-us/vcsharp

NVIDIA Corporation (2003). *Cg Toolkit, Release 3.0*. Retrieved September 2010 from: http://developer.nvidia.com/cg-toolkit

OneSAF Architecture and Integration (A&I) Task Order (2010). *Object Terrain Format Specification*. Retrieved November 2010, from OneSAF Web site: http://www.onesaf.net/community/systemdocuments/v.5.0/MaintenanceManual/erc/OTF_format_specification_Book.pdf

OpenSceneGraph (2010). *OpenSceneGraph-2.9.9.zip*. Retrieved September 2010 from: http://www.openscenegraph.org/projects/osg/wiki/Downloads/DeveloperReleases

PEO STRI. *SE Core Program Briefing UC09 – Womack.ppt* Retrieved May 2011 from: http://www.onesaf.net/community/systemdocuments/UsersConference2009/3.Wednesday%20Briefings/Grand%20Ballroom/1520%20-%20SE%20Core%20Program%20Briefing%20UC09%20-%20Womack.ppt

Presagis. *Creating Entire Virtual Worlds: Correlated Terrain & Beyond.* Retrieved May 2011 from:

http://www.onesaf.net/community/systemdocuments/UsersConference2009/2.Tuesday%20Briefings/Grand%20Ballroom/1605%20-%20Creating%20Entire%20Virtual%20Worlds%20UC09-%20Hebert.pdf

Presagis (2009). *OpenFlight Scene Description Database Specification, version 16.4*. Retrieved November 2010 from: http://www.presagis.com/files/standards/OpenFlight16.4.pdf

RDECOM – STTC. *Broad Agency Announcement (BAA) W91CRB-08-R-0073.* Retrieved May 2011 from: http://www.arl.army.mil/www/pages/8/research/W91CRB-08-R_-073_Amendment_0001_STTC_BAA.PDF

SE Core Database Virtual Environment Development (DVED). *I/ITSEC 2008 Status Update*. Retrieved May 2011 from: http://www.peostri.army.mil/PMConSim/Documents/SECoreDVEDIITSECBrief_Rev_2.0.pdf

SEDRIS Program (2005). ISO/IEC 18025 : *Information Technology – Environmental Data Coding Specification (EDCS).* Retrieved November 2010 from: http://standards.sedris.org/#18025

SEDRIS Program (2006). *Fact Sheet: Synthetic Environment Evaluation Tool – Inspection Tool.* (website) http://tools.sedris.org/seeit_4.1.htm

SEDRIS Program (2009). ISO/IEC 18026: *Information Technology – Spatial Reference Model (SRM)*. Retrieved November 2010 from: http://standards.sedris.org/#18025

University of Central Florida – Institute for Simulation & Training. *ZCAP Research and Development* Retrieved May 2011 from: http://www.ist.ucf.edu/pdfs/zcap.pdf

University of Central Florida. *The Future Aviation Simulation Strategy for the United States Army.* Retrieved May 2011 from: ftp://ftp.rta.nato.int/PubFullText/RTO/MP/RTO-MP-MSG-060/Supporting%20Documents/ MP-MSG-060-04.pps

*The World Wide Web Consortium (W3C) (2008). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Retrieved September 2011 from: http:// http://www.w3.org/TR/2008/REC-xml-20081126