

Interactive 3D Terrain Exploration and Visualization

Stephanie Smullen, Ph.D.
Computer Science

The University of Tennessee at
Chattanooga
Chattanooga, TN 37403
(423) 425-4395

Stephanie-Smullen@utc.edu

Clinton W. Smullen, III, Ph.D.
Center of Excellence for Computer
Applications

The University of Tennessee at
Chattanooga
Chattanooga, TN 37403
(423) 425-4787

Clinton-Smullen@utc.edu

Carlos M. Santa
Center of Excellence for Computer
Applications

The University of Tennessee at
Chattanooga
Chattanooga, TN 37403
(423) 425-4787

Carlos-Santa@utc.edu

ABSTRACT

The CECA VisLab is currently investigating the educational uses of a GeoWall 3D Stereo Rendering system. Many commercial and non-commercial programs are currently available for geographic virtual reality. We have explored the uses of one of these programs, Walkabout (under development by the Electronic Visualization Laboratory at the University of Illinois at Chicago), a program that allows the user to move in relationship to the data in X-Y-Z directions to explore the terrain. We discuss several enhancements for this 3D stereo terrain visualization tool: the ability to directly load Digital Elevation Model (DEM) files and the implementation of a Level of Detail Algorithm (LOD) that optimizes the data in order to increase rendering performance. We discuss the performance of the enhanced program with a data set of particular interest to us, the Tennessee River Gorge including nearby mountains and the Chattanooga, Tennessee metropolitan area.

Categories and Subject Descriptors

I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism - *virtual reality, visible line/surface algorithms*

General Terms

Algorithms, Experimentation, Human Factors

Keywords

GeoWall, Terrain Visualization, Walkabout, LOD algorithms, DEM

1. INTRODUCTION

The GeoWall Consortium [4] is a joint effort of scientists at universities and research centers to make stereo-display technologies available to a broad audience. The technology consists of affordable equipment and open source code for the creation and display of three-dimensional applications with stereo-display capabilities. One goal is to enhance traditional teaching methods that use 2D representations of maps and other physical phenomena with a view that appears to be more three-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA, USA. Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

dimensional. In addition, due to their low cost, GeoWall systems are currently being used by emergency and disaster response teams and land management personnel.

A GeoWall basically consists of the following components: a (controller) computer, a dual output or quad-buffered enabled graphics card, two projectors, polarization filters for each projector, a polarization-preserving screen, polarized glasses, and a stereo-capable application.

To create a stereo display, two images, one that corresponds to the view in the right eye and the other that corresponds to the view in left eye are created. Both images are then displayed on the same space on a screen. The overlapping images are displayed by using two projectors, one for each eye, so that the viewer sees only one image with each eye. The light from each projector is polarized differently, and the projection screen has special properties that preserve polarization. Glasses with matching polarity are used by the viewers. The light from only one image enters the appropriate eye because the polarization of the viewing lens matches the polarization of the projector lens. The mind interprets the two images as one 3D image when they are displayed at an interval corresponding to the separation of human eyes.

2. RELATED WORKS

The CECA VisLab is currently investigating the educational uses of a GeoWall 3D Stereo Rendering system. Many commercial and non-commercial programs are currently available for geographic virtual reality (see for example [1, 9, 10, 13, 14]). Specifically, we want to exploit the 3D stereo capabilities of the terrain visualization tool Walkabout [15] by extending it with two major improvements: the ability to directly load Digital Elevation Models (DEM) files and the implementation of a Level of Detail Algorithm (LOD). The reason for including these two enhancements is that there is a considerable amount of terrain data available and easily obtainable in DEM format and an LOD algorithm was necessary to improve the rendering performance of high resolution terrain data.

Walkabout is an open source, multi-platform terrain visualization tool under development by the Electronic Visualization Laboratory group at the University of Illinois at Chicago. The user can load geo-referenced data representing height field maps and drape a texture to add features and characteristics to the terrain. Several features are provided for terrain exploration, such as multiple user interactions in the terrain, tracking of the users' explored paths, exaggeration of the altitude of the landscape, and walking as well as flying over the landscape. The application relies on the following libraries: CavernSoft G2, FLTK, Open

Inventor, Coin3D, and Libtiff [15]. Currently, Walkabout is limited to only two data formats, VRML files and IV files (the Open Inventor standard data file). Other data must be converted to one of these file types to be used by Walkabout.

3. TERRAIN DATA REPRESENTATION

Terrain files represent the topography of an area with elevation data called height maps. These height maps represent the terrain $w(v)$ with a set of vertices (v) , where $w(v)$ is of the form $w(v) = (vx, vy, z(v))$ and (vx, vy) are the coordinates of the vertex v , and $z(v)$ is the height at v . A common file format, DEM, stores the terrain elevations in a ragged rectangular array. The points are taken at regularly spaced horizontal intervals in a Universal Transverse Mercator (UTM) projection or a geographic coordinate system. The grid cells are ordered from south to north and from west to east (see Figure 1, taken from [12]). The United States Geological Survey (USGS) has made available DEM files for all 50 states and territories. Currently, the USGS produces five

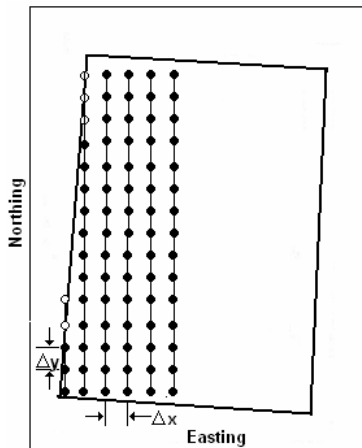


Figure 1. DEM Quad Format

different types of such files: 7.5-minute DEM, 30-minute DEM, 1-degree DEM, 7.5-minute Alaska DEM, and 15-minute Alaska DEM [12]. Thirty meter spacing is used for the 7.5-minute DEM files; for selected locations, 7.5-minute DEM files with 10-meter spacing are available. For this project we used 30-meter and 10-meter DEM data files.

3.1 Reading DEM files

Since a considerable amount of terrain data is available in DEM format, we simplified using Walkabout by allowing it to directly load data in this format. To that end, we implemented a class that reads in the height field data points that define the terrain and stores the points in dynamic arrays for later use by the renderer. Since our LOD rendering algorithm requires a rectangular grid, we developed a function to fill in the ragged edges from the DEM file using a nearest neighbor averaging algorithm. The result is a rectangular array of height field values.

3.2 Real-Time Terrain Rendering

The simplest method when rendering a terrain scene is to traverse the rectangular array and build a triangle mesh by connecting neighboring height field values. A mesh can be defined as a set of

vertices and a set of faces. Each vertex specifies the (x, y, z) coordinates of a point in space, and each face defines a polygon by connecting together an ordered subset of the vertices. However, this approach proved unacceptable due to the poor frames per second rate caused by the large number of triangles. For example, the 10-meter Chattanooga DEM produced over 500,000 triangles.

To reduce the total number of triangles, yet still render sufficient detail where needed, we implemented a Level of Detail (LOD) algorithm. By using an LOD algorithm we can filter the original triangular mesh and render only the triangles necessary to supply sufficient detail in the terrain while improving performance. In Figure 2 note the number of triangles used to render the flat portion of the terrain. These triangles are not needed for accurate rendition of the terrain.

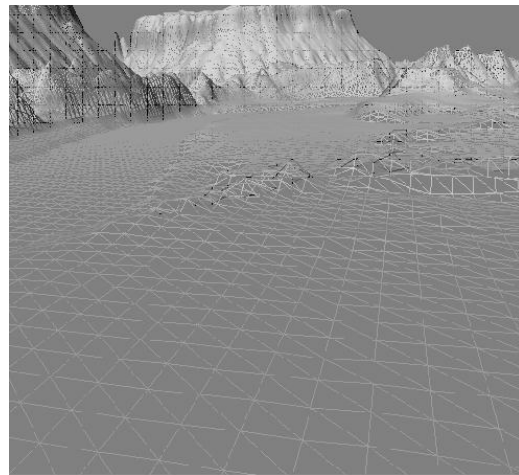


Figure 2. Chattanooga mesh without LOD

4. LOD ALGORITHMS

Significant progress has recently been made in developing algorithmic techniques to obtain different level of detail in the terrain [5, 8].

4.1 Progressive Meshing (PM)

Hugues Hoppe, from Microsoft Corporation, is a pioneer in PM algorithms [6]. His idea is to build a data structure that allows rapid extraction of different level of detail meshes that represent a good approximation of the target triangle mesh. Progressive meshing allows smooth transitions among levels of detail, supporting progressive transmission and effective compression. His algorithms are directed toward dynamic LOD.

4.2 Real Time Optimally Adapting Meshes

ROAM developed by Mark Duchaineau et al. [3] operates on a binary tree constructed from the triangular mesh. This structure is used, along with two priority queues, for incremental refinement of the scene.

4.3 Stateless One-pass Adaptive Refinement

This algorithm, called SOAR, developed by Peter Lindstrom et al. [7] combines the techniques mentioned above but extends them by recursively dividing the triangular mesh using longest-edge

bisection. Its main features are its adaptive refinement algorithm along with optional frustum culling and on-the-fly triangle stripping based on projected error.

4.4 Adaptive Quadtree Continuous LOD

In order to increase rendering performance, we implemented a terrain rendering algorithm to give us different level of details at different points in the terrain using the algorithm by Ulrich [11]. The algorithm employs an adaptive quadtree to represent the terrain height information. A quadtree is a data structure where the root represents the whole surface, and the four children subdivide the parent into four smaller partitions (Figure 3, taken from [11]). The algorithm can be divided into two main functions, Update() and Render(). During Update() the algorithm

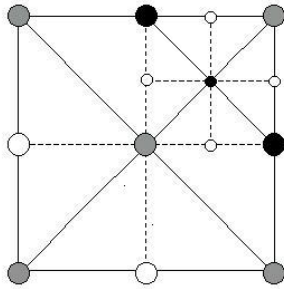


Figure 3. Quadtree refinement

decides based on the gradient in the area which vertices to include in the output mesh. The function recursively subdivides a quadrant, and marks the center and corner vertices as enabled. By recursively subdividing a square, new vertices can be added and treated the same way the vertices of the original square were treated. Then, during Render() a triangle mesh is generated that includes the vertices that were enabled during Update().

Because of the quadtree’s adaptive nature, the terrain can be adapted or updated on the fly at different resolutions in different regions in the terrain. One advantage of this is to generate finer detail only around the viewer’s area and less detail in areas of the terrain where the level of detail is not necessary, thus freeing up computer resources as the viewer moves.

5. IMPLEMENTATION

As discussed above, we implemented a class that could directly load DEM files and store a triangular mesh used to render the terrain. This method enabled us to readily explore 30-meter DEMs. However, when we rendered a scene containing three 10-meter DEM files the performance was unacceptably slow (5 frames per second). Rather than seek a faster hardware solution we implemented an LOD technique that helped us gain rendering performance while maintaining the important terrain detail. We selected Ulrich’s approach because we could obtain acceptable performance without the overhead of the other LOD algorithms.

Figure 4 shows the Chattanooga DEM section using the quadtree LOD algorithm. Compare the triangular mesh with that of Figure 2. Figure 2 represents a display with over 500,000 triangles whereas Figure 4 contains just over 100,000 triangles, a reduction of 1:5. Sufficient detail is maintained in the image by adaptively refining the rapidly varying parts of the terrain.

6.RESULTS AND DISCUSSION

To evaluate the rendering improvement with the LOD technique, we implemented the portion of the code that loads the DEM data files, renders the image, and performs the LOD as a separate application. A menu and selected performance statistics (total triangles, frames per second and time to create the initial quadtree and rendering of the terrain) are available while the user explores

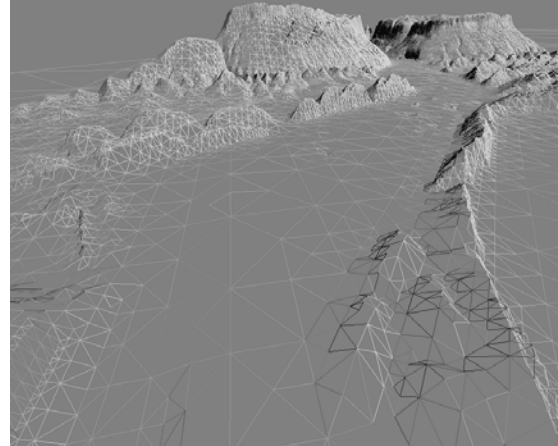


Figure 4. Chattanooga with LOD refinement

the terrain. A fly-through is available, as well as an option to render the terrain in wire-frame mode to easily review the terrain triangulation. We experimented with 10-meter DEMs of Chattanooga, Sequatchie and Wauhatchie, as well as, a combined DEM of the three quads. We used a PC with a 2.80 GHz Pentium 4 CPU, 1GB system memory, and NVIDIA Quadro FX 1100 with 128MB video memory.

Table 1. Statistics with and without LOD refinement

Quad	Original Mesh		
	# triangles	Initial Render time (seconds)	Frames per second
Chattanooga	545,916	0.032	14
Sequatchie	542,880	0.036	15
Wauhatchie	544,590	0.051	15
Combined	1,653,307	0.094	5
LOD Mesh			
	# triangles	Initial Render time (seconds)	Frames per second
Chattanooga	114,882	2.161	44
Sequatchie	100,456	2.513	44
Wauhatchie	155,352	3.1170	44
Combined	244,966	9.655	44

For our application we need performance in the range of 24 to 30 frames per second to provide realistic scene viewing. Table 1 shows unacceptable performance without the use of the LOD algorithm. A considerable increase in the frames per second rate was observed for all the data sets when the LOD technique was applied to the scene. The initial construction of the quadtree and

initial rendering time was increased but did not impact the scene viewing experience.

7. SUMMARY AND FUTURE WORK

A terrain visualization tool with stereo-display capabilities has been enhanced to directly render terrain data from a common terrain data format, DEM. To maintain acceptable display rates, we incorporated LOD techniques. The data shows that the LOD algorithm resulted in significant performance gains. We continue to investigate various filter algorithms that select vertices for expansion in the LOD process and measure their performance characteristics.

While commercial packages such as the recently released ESRI ArcView 3D Analyst [1, 2] provide stereo viewing of terrain data files, our extended version of Walkabout provides this capability in a cost effective, open source program.

8. ACKNOWLEDGEMENTS

The UTC Center of Excellence for Computer Applications supported this project.

9. REFERENCES

- [1] ArcView 3D Analyst, ESRI, <http://www.esri.com/software/arcview/extensions/3danalyst/>
- [2] Davis, Brian. Affordable System for Viewing Spatial Data in Stereo. In *ArcUser*, July-September 2004, pp. 48-49.
- [3] Duchaineauy, M., Wolinsky, M., et al. ROAMing Terrain: Real-Time Optimally Adapting Meshes. In *Proceedings of the IEEE 1997 Conference on Visualization (IEEE Visualization '97)* (Phoenix, AZ, October 19-24, 1997).
- [4] GeoWall, <http://www.geowall.org>
- [5] Hesse, Michael and Gavrilova Marina. A Quantitative Analysis of Culling Techniques for Real-time Rendering of Digital Elevation Models. In *SHORT Papers Proceedings of the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '2003)* (Bory, Czech Republic, February 3-7, 2003).
- [6] Hoppe, Hugues, Progressive Meshes. In *Proceedings of ACM SIGGRAPH 1996* (SIGGRAPH 1996) (New Orleans, LA, August 4-9, 1996), pp. 99-108.
- [7] Lindstrom, Peter and Valerio Pascucci, Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization, *IEEE Transactions on Visualization and Computer Graphics*, 8(3), July-September 2002, pp. 239-254.
- [8] Losasso, Frank and Hugues Hoppe, Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids. In *Proceeding of ACM SIGGRAPH 2004* (SIGGRAPH 2004) (Los Angeles, CA, August 8-12, 2004), pp. 766-773.
- [9] Terra Explorer, Skyline Software, http://www.skylinesoft.com/corporate/corporate_home.asp
- [10] 3DEM Software for Terrain Visualization and Flyby Animation, <http://www.visualizationsoftware.com/3dem.html>
- [11] Ulrich, Thatcher. Continuous LOD Terrain Meshing Using Adaptive Quadtrees. In *Gamasutra* (www.gamasutra.com), February, 28, 2000.
- [12] USGS National Mapping Program Standards, <http://rockyweb.cr.usgs.gov/nmpstds/acrodocus/dem/1DEM0897.PDF>
- [13] Virtual Field Course, <http://www.geog.le.ac.uk/vfc/>
- [14] Virtual Terrain Project (VTP), <http://www.vterrain.org/>
- [15] Walkabout, <http://www.ev1.uic.edu/aej/macagave/walkabout.html>