Course Notes for Volume Graphics, Course 41, Proceedings of the 26th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1999), Los Angeles, CA, USA, August 8-13, 1999.

A Brief Terrain Rendering Literature Review

J. Edward Swan II

1 Introduction

Rendering terrain data is an important application area in computer graphics. Much early work in computer graphics was motivated by building flight simulators. This is a very taxing application, which requires real-time rendering of potentially very large databases. In addition, terrain rendering is important for geographic information systems and architectural landscape previewing systems.

The most common source of terrain data is earth terrain data obtained from aerial or satellite imagery [2]. The terrain data typically comes in the form of two datasets: a color or texture image (typically from an aerial or satellite photograph) and topology or elevation samples. The color dataset is usually at a much higher resolution than the elevation dataset [1]. These two data sources are registered and merged [2] to form two arrays: a 2D array of height values, and a 2D array of color values. Typically the color values have three channels corresponding to red, green, and blue; but other color channels (such as infrared, ultraviolet, etc.) are possible. These two arrays are the typical input to a terrain renderer.

Another source of terrain data are fractal techniques (e.g. Musgrave et al. [15]), which generate realistic-looking yet artificial terrains. These techniques motivated much of the terrain rendering research in the 1980s.

2 Previous Work

Terrain rendering algorithms can be classified according to how the terrain is rendered. This classification groups the algorithms into ***ray casting***, ***shear-warp***, and ***object order*** methods.

2.1 Ray Casting

The ray-casting methods typically cast a ray from each pixel and sample the terrain at the resulting intersection point. Unlike volume ray casting, which samples each ray many times, terrain ray casting algorithms usually only sample one point per ray. These algorithms typically store the height and color information in grid form; the grids are interpolated at the intersection point. They can be further characterized by how they represent rays and step through the data structure:

Incremental Techniques: Dungan [7], Coqillart and Gangnet [6], and Musgrave [14, 15] all trace the 2D projection of the ray across the baseplane of the terrain grid. At each step, the height of the ray is compared to the height of the terrain data at that step. When the ray height drops below the terrain, the exact intersection point is found.

Space-Leaping Techniques: Cohen-Or et al. [3] and Lee and Shin [12] use a similar incremental technique, but accelerate the process by starting the traversal of each ray above the intersection point of the previous ray.

Hierarchical Techniques: Cohen and Shaked [5] store the height grid at multiple resolutions in a quadtree data structure. The ray steps first across the largest quadtree node, which contains the highest point of the terrain. If the ray height is below this, then the ray is recursively compared to the heights of the proper quadrants. This continues until the ray intersects the terrain data at a quadtree leaf node.

Distance Transform Techniques: Paglieroni and Petersen [16, 17] compute a distance transform of the height grid. This gives an area of empty space around every voxel, where it is guaranteed that the ray will not encounter other voxels. The ray is stepped from a given voxel to the edge of the voxel's distance transform.

Because ray casting point-samples the terrain grid, all of these methods are particularly subject to aliasing artifacts. Many of the above methods perform some anti-aliasing by storing the terrain grid at multiple levels of resolution; the lower resolution grids are used for pixels which show areas of the terrain that are far from the view point. This still results in aliasing artifacts, however. Cohen-Or [4] describes a ray-casting terrain rendering technique that area-samples the terrain grid, resulting in high-quality terrain images.

2.2 Shear-Warp

The shear-warp methods create a perspective projection of the height and color grids through a series of 1D shearing and warping operations. The data is resampled into a regular grid which is designed so that all voxels which might occlude a particular voxel are located in the same row or column. Then a back-to-front traversal along these rows or columns results in the correct visibility. This is followed by an inverse resampling that restores the data to a perspective projection. The basic implementation of this technique is described by Robertson [18]. Another implementation using a spherical projection is given by Miller [13]. The technique's advantages are that each 1D pass can be made quickly, and that the technique is easy to parallelize. Vezina and Robertson [21], Kaba et al. [9], and Kaba and Peters [10] all describe parallel implementations, some of which can render 30 frames per second [9, 10]. Robertson [19] also describes how the technique can be extended to rapidly generate shadows on a terrain surface.

2.3 Object Order

Most of the object-order techniques fit polygons or patches to the terrain height field, and then render these using standard polygon or patch rendering techniques. The terrain color grid is texture mapped onto the polygons or patches. An example is Coqillart and Gangnet [6], which fits the surface with bilinear patches. Another is Kaneda et al. [11], which fits polygons that vary in size in such a way that they sample approximately equal areas of the terrain when rendered from a perspective projection. This is similar to Geymayer et al. [8], except that the polygons are pre-fit into a pyramid data structure, and thus the terrain does not have to be resampled with each frame. Kaneda et al. [11] also demonstrates an advantage of polygon/patch fitting methods: it is easy to include additional polygons representing buildings or other ground structures.

Agranov and Gotsman [1] describe a hybrid order algorithm. They use ray-casting to determine the pixels along the screen border; this projects the screen as a polygon onto the terrain dataset. They then project all the triangles contained in this polygon onto the screen using a z-buffer.

Wright and Hsieh [23] model the terrain as a set of voxel grids at different resolutions. The voxels are visited and projected onto the image plane in order of increasing distance from the image plane; as the distance to the image plane increases voxels from lower resolutions are used. This ensures that the projected voxels span approximately the same area of the image plane.

Swan et al. [20] utilize the ***splatting*** [22] technique to render terrain grids. They model the terrain as a thin voxel sheet, and visit the voxels in order of decreasing distance to the image plane. They describe an anti-aliasing technique which successfully handles the severe compression which occurs at the horizon of terrain datasets.

References

[1] Agranov, G., and Gotsman, C., "Algorithms for Rendering Realistic Terrain Image Sequences and their Parallel Implementation", *The Visual Computer*, 11, November 1995, pp. 455–464.

[2] Cohen, D., and Gotsman, C., "Photorealistic Terrain Imaging and Flight Simulation", *IEEE Computer Graphics and Applications*, 14(2), March 1994, pp. 10–12.

[3] Cohen-Or, D., Rich, E., Lerner, U., and Shenkar, V., "A Real-Time Photo-Realistic Visual Flythrough", *IEEE Transactions on Visualization and Computer Graphics*, 2(3), September 1996, pp. 255–265.

[4] Cohen-Or, D., "Exact Antialiasing of Textured Terrain Models", To be published in *The Visual Computer*, 1997.

[5] Cohen, D. and Shaked, A., "Photo-Realistic Imaging of Digital Terrains", In proceedings of *Eurographics '93*, 1993, pp. 363–373.

[6] Coquillart, S. and Gangnet, M., "Shaded Display of Digital Maps", *IEEE Computer Graphics and Applications*, 4(7), July 1984, pp. 35–42.

[7] Dungan, W., "A Terrain and Cloud Computer Image Generation Model", *Computer Graphics* (proceedings of SIGGRAPH), 13(3), August 1979, pp. 143–150.

[8] Geymayer, B., Prantl, M., Muller-Seelich, H., and Tabatabai, B., "Animation of Landscapes Using Satellite Imagery", In proceedings of *Eurographics '91*, 1991, pp. 437–446.

[9] Kaba, J., Matey, J., Stoll, G., Taylor, H., and Hanrahan, P., "Interactive Terrain Rendering and Volume Visualization on the Princeton Engine", In proceedings of *Visualization '92* (October 19–23, Boston, MA), IEEE Computer Society Press, 1992, pp. 349–355.

[10] Kaba, J., and Peters, J., "A Pyramid-based Approach to Interactive Terrain Visualization", In proceedings of *1993 Parallel Rendering Symposium* (San Jose, California, October 25–26), IEEE Computer Society Press, 1993, pp. 67–70.

[11] Kaneda, K., Kato, F., Nakamae, E., Nishita, T., Tanaka, H., and Noguchi, T., "Three Dimensional Terrain Modeling and Display for Environmental Assessment", *Computer Graphics* (proceedings of SIGGRAPH), 23(3), July 1989, pp. 207–214.

[12] Lee, C., and Shin, Y. G., "An Efficient Ray Tracing Method for Terrain Rendering", In proceedings of *Pacific Graphics '95*, 1995, pp. 180–193.

[13] Miller, G. S. P., "The Definition and Rendering of Terrain Maps", *Computer Graphics* (proceedings of SIGGRAPH), 20(4), August 1986, pp. 39–48.

[14] Musgrave, F. K., *Grid Tracing: Fast Ray Tracing for Height Fields*, Technical Report YALEU/DCS/RR-639, Yale University Department of Computer Science Research, 1988.

[15] Musgrave, F. K., Kolb, C. E., and Mace, R. S., "The Synthesis and Rendering of Eroded Fractal Terrains", *Computer Graphics* (proceedings of SIGGRAPH), 23(3), July 1989, pp. 41–50.

[16] Paglieroni, D. W. and Petersen, S. M., "Parametric Height Field Ray Tracing", In proceedings of *Graphics Interface '92*, May 1992, pp. 192–200.

[17] Paglieroni, D. W. and Petersen, S. M., "Height Distributional Distance Transform Methods for Height Field Ray Tracing", *ACM Transactions on Graphics*, 13(4), October 1994, pp. 376–399.

[18] Robertson, P. K., "Fast Perspective Views of Images Using One-Dimensional Operations", *IEEE Computer Graphics and Applications*, 7(2), February 1987, pp. 47–56.

[19] Robertson, P. K., "Spatial Transformations for Rapid Scan-Line Surface Shadowing", *IEEE Computer Graphics and Applications*, 9(2), March 1989, pp. 30–38.

[20] Swan II, J. E., Mueller, K., Möller, T., Shareef, N., Crawfis, R., Yagel, R., "An Anti-Aliasing Technique for Splatting", In proceedings *IEEE Visualization '97* (October 19–24, Phoenix, Arizona), IEEE Computer Society Press, 1997, pp. 197–204.

[21] Vezina, G., and Robertson, P. K., "Terrain Perspectives on a Massively Parallel SIMD Computer", *Scientific Visualization of Physical Phenomena* (proceedings of *CG International '91*), 1991, pp. 163–188.

[22] Westover, L. A., "Footprint Evaluation for Volume Rendering", *Computer Graphics* (proceedings of SIGGRAPH), 24(4), August 1990.

[23] Wright, J. R., and Hsieh, J. C. L., "A Voxel-Based, Forward Projection Algorithm for Rendering Surface and Volumetric Data", In proceedings of *Visualization '92* (October 19–23, Boston, MA), IEEE Computer Society Press, 1992, pp. 340–348.