# REAL-TIME RENDERING OF THE EARTH SURFACE BASED ON THE ELLIPSOIDAL TRIANGULAR MESHES

BAI Jianjun[a] , ZHAO Xuesheng[a] , CHEN Jun[b]

[a] China University of Mining and Technology (Beijing), D11 Xueyuan Road, Haidian District, Beijing, China, 100083-jianjunb@sina.com
[b] National Geomatics Center of China, 1 Baishengcun, Zizhuyuan, Beijing, China, 100044—chenjun@nsdi.gov.cn

**Commission VI, WG VI/4**

**KEW WORDS:** Real-time rendering; Ellipsoidal triangular meshes; Digital elevation model; Data tiling; Multi-resolution model; Geomorphing;

## ABSTRACT:

Real-time rendering and multiresolution modeling of the earth surface has attracted growing interest over the last decade. In order to visualize and model the earth surface, a digital elevation model based on the ellipsoidal triangular meshes is made to approximate to the earth surface in this paper, and then two key components of the visualization of the surface are described: management of the massive terrain data; and real-time rendering of the earth surface. First, we organize the terrain data through different resolution levels and data tiling in the same level, and give an approach of viewpoints-based data extraction from terrain data. Second, similar to several previously proposed methods for tile-based refinement, a multi-resolution model is built through recursive subdivision of each Diamond. As a component of the rendering, the crack prevention between the different blocks and geomorphing are done together with the refinement algorithms to enhance the fidelity of scenes. At last, a group of experiment results are given, which illustrate that the methods is an effective one to realize fast-speed rendering of earth surface.

## 1. INTRODUCTION

The modeling and visualization of the earth surfaces has been one of the research hotspots in geography, especially with the origin of the concept of Digital Earth put forward with the published text of a speech made by former U.S. Vice President Gore in 1998(Gore 1998), it has been attached to more and more important in many country. In GIS, the regular grid DEM based on the idea of map projections, is effectively a form of modeling the terrain on the small spherical surfaces as a flat or planar surfaces, and there emerge many desirable algorithms (Lindstrom *et al.* 1996,2002;Duchaineau *et al.* 1997; Hoppe 1996;Pan *et al.* 1998) about the visualization of the terrain over the last decade. Although the traditional grid DEM suits small-scale terrain modeling and visualization adequately, they do have some significant drawbacks for modeling the large-scale spherical surfaces. These are unacceptable distortions, broken topology, space inefficiency, and difficulty with sharing data between projects caused by a lack of good global representation schemes (Nicholas *et al.* 1995). They are all still consider the earth as a quilt made up of piecewise flat regions.

To address these shortcomings, this paper put forward a multi-resolution digital modeling of the earth surfaces based on the Quaternary Triangular Mesh (QTM) on the ellipsoidal surface. Ellipsoidal surface is the space mathematical base of the digital model, which was built through QTM hierarchical tessellation of the ellipsoidal surface and the function defined over a triangle region of QTM. The diamond partition and level of detail (LOD) are proposed in horizontal and vertical direction respectively to manage and render the global terrain data. The diamonds at different LOD are bring in and out of RAM based on the viewpoint and screen resolution, and the multi-resolution model be built through recursive subdivision of each diamond in the RAM.

The remainder of this paper is organized as follows. In section 2, the current methods of modeling and visualization of the earth surface are critically reviewed. Following this we create a digital model to earth surface based on hierarchical triangular subdivision in section 3. In section 4, the data organize is introduced with respect to the diamond partition. Based on recursive subdivision of the Diamond we realize the visualization of the global DEM in section 5 and section 6. Finally this paper concludes in section 7 with some remarks about the global DEM and our work in the future.

## 2. RELATED WORK

Many scholar and academic organization (Falby *et al.* 1993;Hitchner *et al.*1992; Reddy *et al.*1999; Sun *et al.* 2000; Koller *et al.* 1995;Faust *et al.* 2000; Lukatela 2000;Chen 2001) have studied the modeling and visualization of earth surface. The DEM data we often used are regular grid scheme gridded by latitude and longitude in large scale DEM database. The GTOPO30 and ETOPO5 data supplied by US Geological Survey, and the JGP95E5′ data compiled by The Defense Mapping Agency and NASA/Goddard Space Flight Center are all in this format. Also 1:250000, 1:50000 DEM database in china offer the same format data. The three dimensional earth figure is expressed using JGP95E5′ data in paper (Sun *et al.* 2000). In Gerstner (1999), it is shown that compressed the regular grid DEM data gridded by latitude and longitude can be used to interactive visualization of the whole globe, but it is concern little about the global topographical model. The latitude/longitude grids DEM that deal with global coverage create a lack of uniformity among some geometric properties of the grid cells. A grid cell covers a larger area near the equator than it does near the poles, thus the latitude/longitude grid DEM data have a lot of redundancy.

Virtual Geographic Information System（VGIS）(Koller *et al.* 1995;Faust *et al.* 2000) developed by Georgia Institute of Technology create a hierarchical terrain model based on the partition and map projection of the spherical surface. The whole earth surface is divided into many zones along the latitude and longitude. Each zone has its own quadtree and local coordinate system; all are linked so that the terrain crossing boundaries can be rendered efficiently. This system has some shortcomings: it is still based on the map project in local areas, thus causes some projection distortions. And there are difficulties with sharing data between different zones due to different local coordinate systems. It is still consider the Earth as a quilt made up of piecewise flat regions. Paper (Falby *et al.* 1993;Hitchner *et al.*1992; Reddy *et al.*1999;) all have the similar problems,

Hipparchus System (Lukatela 2000) construct seamless global triangular networks similar to the planar TIN commonly used to facilitate terrain modeling and volumetrics. It is based on global coordinates and a planetary surface tessellation using spheroidal Voronoi polygons. Duo to the TIN is difficult to construct and the storage requires is very large, it is not desirable to be used to model the earth surface as a whole.

## 3. A HIERARCHICAL MODEL TO EARTH SURFACE BASED ON HIERARCHICAL TRIANGULAR SUBDIVISION

In this study, we choose six ground sampling points including the North Pole, the South Pole and four points which are the crossed points of the equator and the 0th, 90th, 180th and 270th meridians respectively. Given each point a height value, we connect them together using straight-line as shown in figure 1, so that they form ellipsoidal triangular mesh (or octahedron). It is an inscribed polyhedron that can be seen one of the coarsest approximation to the earth surface. We also call the facet of the octahedron initial ellipsoidal triangle since these points that construct it are all based on the ellipsoidal surface.

As for each facet of the octahedron, it can be subdivided recursively. When it is subdivided, the latitudes and longitudes of pairs of its vertices are averaged to yield three new edge midpoint locations. These points are connected to form four new smaller ellipsoidal triangles, as shown in figure 1(b), which is a more refined approximation to the earth surface. This process can be conducted recursively until a desirable approximation reaches.

These ellipsoidal triangles created by recursive subdivision are called Quaternary Triangular Mesh (QTM), which take ellipsoidal surface as the space mathematical base. The vertices that make up QTM are called *QTM vertices* whose position can be identified by latitude/longitude coordinate. Thank for the regularity of the subdivision, the topology of the QTM is simply and the coordinates of the points can be implicitly figured out according to its storage location and the levels of the subdivision. Thus, DEM can be represented by the array of elevation values of the vertices.


(a) （b）

Fig.1 (a) Triangular mesh connected by six sample points on the ellipsoidal surface. (b) Hierarchical subdivision of triangular mesh on the ellipsoidal surface.

## 4. DATA ORGANIZING

This section addresses the problem of laying out the DEM data on disk to achieve efficient out-of –core performance. In order to convenience data management, we conceive of pairs of adjacent triangles as Diamonds (Figure 2), and organize the DEM data based on Diamond of given size. In the database system it is a basic storage unit corresponding to a record, while in the file system it corresponds to a data file.

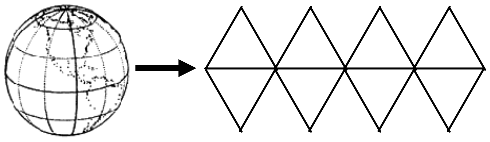

Figure 2. Four base diamond of the octahedron

Similar to White (2000), we conceive of the surfaces of the octahedron as composed of pairs adjacent triangles, or diamonds, that tessellates, or cover the surface. On the polyhedra, the diamonds are bent, so to speak, across the narrow diagonal, but for data structure purposes they can be considered entire. The octahedron has four diamonds as shown in Figure 2.

In QTM each ellipsoidal triangle in the octahedron can be divided into four smaller triangles by breaking each edge into 2 pieces and connecting the midpoints with lines. Recursively subdividing the triangles thus obtained in the same manner yields QTM (Figure 3). Like the quadtree subdivision of the square, each diamond in the octahedron can be divided into four smaller diamonds (Figure 3). These two kinds of subdivision are essentially the same, so the QTM can be regarded as diamond meshes.

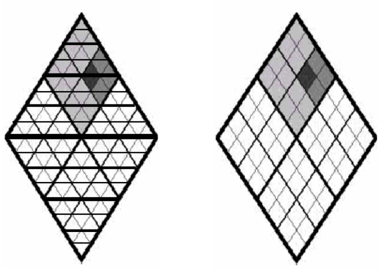

Figure 3. QTM and the diamond tessellation at the third level

The surface of the earth can be represented as a quadtree that the root (corresponding to the surface of the earth) has four children node (four Diamonds), and the internal node has four children node (four child Diamonds) as shown in Figure 4. Each Diamond is assigned a quadcode, and the leaves can be labeled according to the Z spacefilling curve. We organize the DEM data based on levels of Diamond. The Diamonds of different level, which correspond to different resolution traditional map, are basic storage unit. In the file system they are the binary terrain data files in which the height values of the *QTM vertices* within the Diamond region are stored in Binary Large Object （BLOB）.The final form of the dataset make up of many binary files, each containing a description of terrain for given size of Diamond.

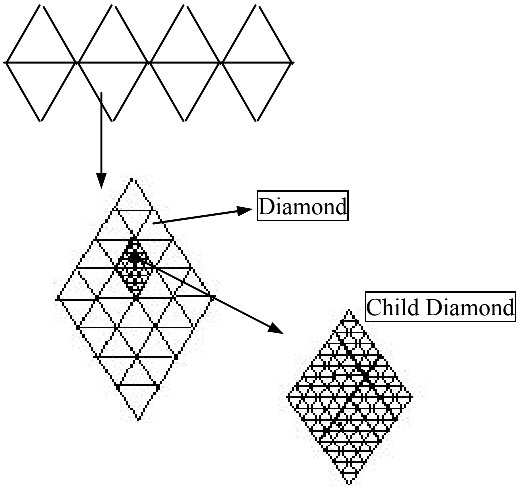

Figure 4. the terrain data organizing based on the levels of diamond

## 5. THE VISUALIZATION OF THE GLOBAL DEM

### 5.1 Tile-based LOD

Prior to the widespread adoption of GPU (Graphic Processing Units), LOD algorithms for the terrain visualization focused on minimizing the number of triangles drawn, at the expense of the limited amount of extra CPU processing. Various published algorithms for this, such as CLOD (Lindstrom 1996),ROAM(Duchaineau 1997) and PM(Hoppe 1996) ware adopted and adapted to become a usefull component in the game programmer's toolbox. With the advent of GPU, developers focus on approaches with very low CPU overhead and high triangle throughout, like Chunked LOD (Ulrith 2002), VIPM (Bloom 2000) and Dynamic fixed-representation LOD (David 2002).unfortunately, Chunked LOD and VIPM are all require substantial pre-computation and storage space for the different representations of terrain, And Dynamic fixed-representation LOD does not consider hypsography of terrain in simplification.

In this paper, our approach is similar to (David 2002); additionally we take into account terrain fluctuation in the simplification of the tile of terrain. In our system, the Diamond is organized into a tile-based quadtree in which each node represents a terrain tile. If the LOD algorithm determines that a tile is too coarse for the given tolerance value, the tile is split into four children (four smaller tiles). Each child is considered in turn, recursively subdividing it as needed. Each node in the quadtree is rendering using 9×9 sample points. The sample points for parent node are derived form the sample points for child node through interval sampling, thus the ratio of the grid interval between parent node and child node is 2.

For the entire earth, four unique quadtree, each corresponding to a Diamond, are used to form the surfaces of the earth. Each tile in the quadtree is subdivided as necessary, thus it forms a tile-based LOD for the planet.

In this paper, the object-space error metric is used to determine if tiles need to be split or not. An object-space error metric is defined by calculating bounding priorities at each node and a bottom up traversal of the tile-based quadtree, similar to the approach presented in (Duchaineau 1997). This method ensures the monotonic of the error metric in the quadtree hierarchy.

In order to facility Crack filling, it is necessary to enforce the property that no tile differs by more than a single level from any of its neighbors. To implement this, each tile is given a pointer to each of its four neighbors. Thus ensuring that adjacent tiles only differ by one level of detail.

### 5.2 Crack Filling

Cracks occur between neighboring tiles if they are at different levels of detail. In section 5.1, we ensure that adjacent tiles differ at most by one level of detail through tile tessellation algorithm. These crack can be fixed without modifying the vertex data of either tile (Figure 5). Instead, an alternate indexing scheme is used that creates coarser triangle along the edge of the finer-detailed tile, while the coarser-detailed tile remains untouched (David,2002;Sun,2002).

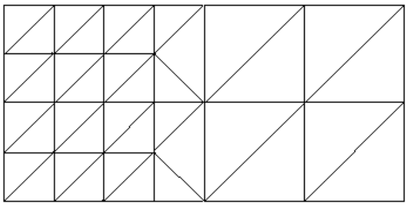

Figure 5.Cracks between the adjacent Diamonds eliminated

### 5.3 Avoiding Pops

We can hide the ugly visual transition between the different frames. The solution involves applying morphing to the vertical coordinate *h* of each vertex in a triangular mesh, using a single uniform morph parameter *f* over the whole tile. Morphing is based on entirely on viewpoint distance, which has the nice property that the mesh only morphs when the viewpoint moves.At rendering time, we will compute a tile's morph parameter, ensuring that the morph parameter is always 0 when the tile is about to split, and 1 when the tile is about to merge. Otherwise the morph parameter *f* can be linearly interpolated in between those distances, as the viewpoint gets close to the tile.

Let the vertical coordinate of a vertex in a tile before splitting and after splitting is $h_0$ and $h_1$ respectively, the vertical coordinate *h* of this vertex at one given distance can be calculated from the formula(1):

$$h=h_0+f(h_1-h_0) \qquad (1)$$

### 5.4 Data paging

In the database, different level of data has different resolution, and also has different size of the cell (QTM). As the amount of dataset is too much to store in main memory at one time, in the visualization of terrain data, only one level of data are paging in according to proportion between the screen space distance and its corresponding object space distance. When the projection area of given objects in the screen is larger, the object space distance per pixel is smaller, thus the data resolution needed in visualization is higher; otherwise, and it is reverse. We select the needed QTM cell interval through calculating the object space distance per pixel according to projection relation in the visualization, thus decide which level of data to page in.

Usually the screen resolution is a constant. With the changing of the viewpoint and angle of view, the object space distance corresponding to the adjacent pixel in the screen is changed. The projection relation is shown in Figure 6:

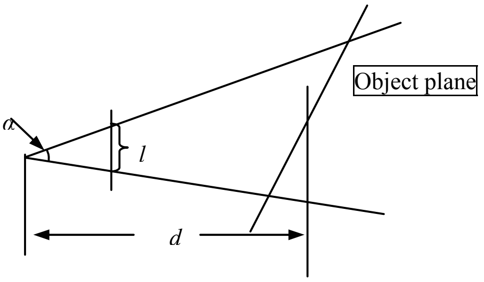

Figure 6. The projection relation in visualization

Let $\lambda$ is number of the pixels per distance unit in the projection plane; $d$ is the distance between the view and the object plane; $\alpha$ is angle of view; and $l$ is the length of the projection plane; we have the object space distance corresponding to the adjacent pixel:

$$D=\frac{2\times d\times tg\frac{\alpha}{2}}{l\times\lambda} \qquad (2)$$

With the changing of the viewpoint and angle of view, we calculate the object space distance per pixel according to equation (1) in the visualization, thus decide which level of data to be paged in.

For all Diamonds located in the same level, we only select the Diamonds inside the view frustum to page in, and not consider the Diamonds located outside the view frustum, for the efficient rendering. 4 Diamonds are chosen as the active areas based on considerations for memory size of available PC, frame rates, and desired range of view. This amount of terrain data is in the main memory at any given time and available for rendering. The Diamonds needed to page in can be decided according to the same projection principle. When the rendering algorithm is initialized, the viewpoint is centered on the 4 diamonds. A bounding box is established around the screen center as shown in Figure 7. When the user reaches the bounding box in any direction, memory space is freed in the direction opposite of travel, terrain is paged in the direction of travel, and the bounding box moves. The dynamic algorithm of paging Diamonds data in the database is as follows:

1. Calculate the needed QTM cell resolution according to projection relation, and then determine which level of Diamond is needed in the visualization.
2. For the level of Diamond, compare the bounding box and the scope of the diamonds, and determine which diamonds should be paged in.
3. With the changing of viewpoint and angle of view, some diamonds far from the bounding box are paged out and other diamonds near the bounding box are paged in.

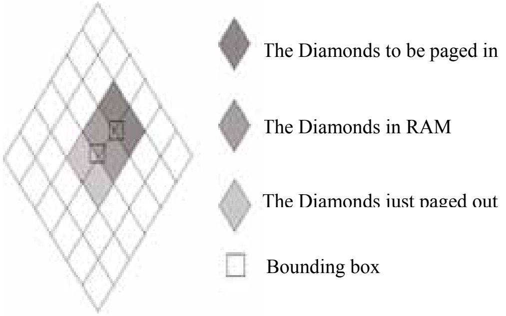

Figure 7. The dynamic paging of the diamonds

## 6. EXPERIMENTS

Before visualization, the data must be prepared and structured in a form that closely matches the internal representation so that it can be organized into Diamond based file. The global terrain data (the gtopo30 dataset) are from Courtesy of US Geological Survey. The gtopo30 dataset was interpolated onto 4 Diamonds, 16 Diamonds, 64 Diamonds and 256 Diamonds dataset respectively; each Diamond includes 2049*2049 vertices. The four Diamonds dataset make up a multi-resolution pyramid. There are 4,16,64 and 256 Diamonds file at one level of the pyramid respectively. Each elevation value in the Diamond file represents the height value in meters above sea level and is stored using 2 bytes as a signed integer. Ocean areas are marked by *nodata* values. Using our data structures, this dataset occupies roughly 2.7GB on disk.

We used a processor 933MHz Pentium III PC, with 256MB of RAM and 32M GeoForce graphics. Figure 8 shows a subset of the whole data set that is visualized in an interactive application. The triangles are color shaded using a simple geographical colormap. Ocean areas are not drawn. All height values are exaggerated by a factor of 100.we achieved a drawing rate of 500000 per second independent on the current position and zoom factor.

## 7. CONCLUSION AND FUTURE WORK

The modeling and visualization of the earth surfaces is one of key part of Digital Earth. The Global DEM based on Ellipsoidal Triangular meshes described in this paper provide a efficient way to modeling the earth surface. The ellipsoidal triangular meshes are based on the ellipsoidal surface, thus it avoids the gaps caused by map projection. It is near regular that makes it convenient for data organization and compressed storage. Our multi-resolution model based on Diamond is suited for visualization.

The main future work we anticipate in this area is better integration of our global DEM and the visualization algorithms, and based on this, to develop some spatial analysis algorithms on the ellipsoidal surface.

Fig 8 the delineation of global terrain based on hierarchical subdivision of the triangular meshes on ellipsoidal surface

## ACKNOWLEDGEMENTS

The work described in this paper was supported by the National Outstanding Youth Science Foundation of China (under grant No.40025101).

## REFERENCES

Bloom C.. View independent progressive meshes. 2000. (VIPM).http://www.cbloom.com/3d/techdocs/vipm.txt

Chen G,.Wan G,. and You Y,.Design & Realization of Earth-Based Terrain Visualization. Journal of System Simulation,2001,Vol 13 Suppl:, pp.282-285 (in chinese)

David Hill. An Efficient, Hardware-accelerated, Level-of Detail Rendering Technique for Large Terrains. Master Thesis University of Toronto,2002, pp. 1-99

Duchaineau M., Wolinsky M., Sigeti D.E., Miller M.C., Aldrich C.,Mineev and Weinstein M.B. ROAMing Terrain: Real—time Optimally Adapting Meshes, in Proc. Visualization '97, Oct. 1997. 1997. pp..81-88.

Falby J.S., Zyda M.J., Pratt D.R. and Mackey R.L. “NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation”. Computer and Graphics .1993,17(1),pp.65-69. Pergamon Press, UK.

Faust N., Ribarsky W., and Jian g T.Y., Real-Time Global Data Model for the Digital Earth. http://www.ncgia.ucsb.edu/globalgrids/papers/faust.pdf. 2000

Gerstner T. Multiresolution Visualization and Compression of Global Topographic Data. Http://www.issrech.iam.uni-bonn.de/research/Gerster/global.pdf

Gore Ai. The Digital Earth: Understanding Our Planet in the 21st Century. http://digitalearth.gsfc.nasa.gov/VP19980131.html

Hitchner L. E. A Virtual Planetary Exploration: Very Large Virtual Environment. Proceeding of SIGGRAPH 1992, pp.6.1-6.15,

Hoppe H. Progressive Meshes . SIGRAPH 96 Conference Proceedings , 1996, pp. 99-108

Koller D., Lindstrom p., Ribarsky W., Larry F., Hodges, Faust N., and Turner G. Virtual GIS: A Real-Time 3D Geographic Information System. In proceedings Visualization 95, pp.94–100. IEEE, Computer Society Press, Los Alamitos, California, 1995.

Lindstrom P., Koller D, Faust and Gregory. A Real-Time Continuous Level of Detail Rendering of Height Fields. SIGGRAPH’96 Conference Proceedings, 1996, pp.109-118.

Lindstrom P. and Pascucci V. 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.

Lukatela H. A Seamless Global Terrain Model in the Hipparchus System. Http://www.geodyssey.com/global/papers, 2000

Nicholas M. Short, Jr., Robert F. Cromp, William J. Campbell, James C. Tilton, Jacqueline J. LeMoigne, Gyorgy Fekete, Nathan S. Netanyahu, Keith Wichmann, Walter B. Ligon III . Mission to the planet earth： ai views the world http://csdl.computer.org/comp/mags/ex/1995/06/x6024abs.htm

Pan Z,. Ma X,.and Shi J,.Overview of Multiple Level of Detail Creation.Journal of image and Graphics, 1998(9):,pp.754-759 (in chinese)

Reddy M., Leclerc Y., Iverson L., and Bletter N. Terravision II: Visualizing Massive Terrain Databases in VRML, IEEE Computer Graphics& Applications, vol. 19, no. 2, pp. 30–38, Mar. - Apr. 1999.

Sun H,. Du D,. Li Z,.and Zhou Y,. Study on the 3D Visualizatin of Earth Figure.Journal of Wuhan Technical of Surveying and Mapping,2000,25(2), pp. 158-162 (in chinese)

Sun M.,Xue Y,.and Ma A,. 3D Visualization of Large DEM Data Set Based on Grid Division.Journal of Computer-aided Design& Computer Graphics, 2002,14(6), pp.188-193 (in chinese)

Ulrith T.. Rendering Massive Terrains Using Chunked Level of Detail Control. 2002,http://www.labri.fr/Perso/~preuter/imageSynthesis/03-04/presentations/fannyguillaume.ppt

U.S.Geogolical Survey: Global 30 arc second Elevation Dataset(GTOPO30). http://edcwww.ct.usgs.gov/landdaac/gtopo30/gtopo30.html

White. D. Global Grids From Recursive Diamond Subdivisions of The Surface of an Octahedron or Icosahedron. Environmental Monitoring and Asse ssment, 2000,4(1), pp.93-103.