Defence Science Journal, Vol. 63, No. 1, January 2013, pp. 89-92, DOI: 10.14429/dsj.63.3768

Short Communication

Quad Tree-based Level-of-details Representation of Digital Globe

Sudhir Porwal

Defence Electronics Applications Laboratory, Dehradun-248 001, India
E-mail: sudhirporwal@yahoo.com

ABSTRACT

Three-dimensional visualization of the geographic data using a digital globe model has been an integral part of a modern GIS system. The visualization of the digital globe model presents many challenges not found in traditional terrain visualization system. The representation of the digital earth (globe) model is important to efficiently render the geographical data without any distortion either at equator or Polar Regions. This paper presents a uniform scheme for efficient quad tree based level-of-details (LOD) representation of the digital globe to minimize the distortion at Polar Regions and meets the requirement of fast frame rate rendering.

Keywords: Digital globe, terrain rendering, GIS, quad tree, level-of-details

1. INTRODUCTION

The real time rendering of the geographical data has been an interested topic for the researchers since decades. In early days, the geographical data was available through ground survey and very primitive methods were available for its visualization. The advancement of satellite based remote sensing and imaging has provided enormous amount of geographical data of sub-meter resolution. The digital terrain elevation data (DTED) of complete globe is available through SRTM[1] and ASTER datasets covering the earth at a resolution up to 30 arc sec. The high-resolution (sub meter) satellite imagery is available through remote sensing satellite i.e. CARTOSAT, TES, IKONOS, QUICKBIRD, etc. The real time visualization of the acquired ground data is used for flight simulation, mission planning & rehearsal, oil exploration, urban planning and many other areas of exploration. The conventional approach for terrain visualization assumes the terrain a flat surface and there exists many techniques for flat terrain visualization. It was suitable for visualization of few datasets only. The availability of ample amount of data has raised the need to visualize arbitrary number of datasets (each of very large size) in a continuous seamless manner over the 3D model of the globe. The traditional methods of modeling will not suffice to extend the planer visualization to globe based visualization. This requires a complete different scheme for geometrical modeling of the globe so that different datasets can easily fit over the globe for accurate visualization. The main challenge lies in the continuous visualization of multiple datasets from far distant point in space to a very close point on earth covering equatorial and polar regions. The suitable geometric model of the earth should be used to represent the datasets in such a way that the data can be accessed in a fast and efficient manner to meet the high frame rate rendering requirement of most of the applications.

The technique presented in this article discusses a method of quad tree based geometrical modeling of the globe including the Polar Regions for fast and efficient access of the datasets. This technique requires the input datasets should be tiled and mipmapped, a capability provided by geomatrix toolkit[2]. The research in the field of digital globe rendering is also useful to render the data of other planets like Mars, Moon etc. This work can easily be extended for visualization of other celestial bodies.

2. RELATED WORK

There exist many effective techniques for terrain representation in the literature. Before the advancement of 3D graphics hardware, most of the techniques for terrain rendering were CPU based i.e. Duchaineau[3], *et. al.* ROAM algorithm and Lindstrom[4-6], *et. al.* continuous level-of-detail method. The invention of powerful GPUs give birth to a new class of methods[7-10] for geometrical representation of the terrain based on graphics hardware. Kooima[11] has discussed a method for globe-based terrain rendering using the icosahedron as a base model of the globe. The triangles of icosahedron are recursively subdivided to generate more number of triangles to represent the finer details of terrain using some buffers and look up tables. This technique is very generic and can be applied to any planetary terrain rendering system. He has highlighted in his work that a single spherical projection cannot be used to represents the polar region and other parts of the globe. Yang[12] has also reported projection distortion at lunar poles in his lunar visualization method. He has used spherical projection to represent the complete moon surface. Zhang[13] has presented a quad tree based LOD algorithm for dynamic terrain rendering of planer surfaces. The work presented in this article will make use of quad tree data structure to effectively represent the globe geometry covering the equatorial and polar regions.

Received 24 September 2012, revised 15 December 2012, online published 24 January 2013

3. QUAD TREE REPRESENTATION OF DIGITAL GLOBE

Since the globe is spherical in shape, it is not a good idea to represent the complete globe with a single quad tree or some other representations. The globe is divided into six faces (T0 to T5) in our method. The four faces (T0 to T3) are along the equator and two faces (T4, T5) cover the north and south poles respectively as shown in Fig. 1. The size of each face is 90° x 90°.

Figure 1. Six faces of the globe.

The four faces along equator cover the earth area ranging from 45° latitude to –45° latitude. The two polar faces are circular section of the globe at 45° latitude. These faces of the globe are used as a root of the quad trees. These faces are recursively divided into sub faces to reach the desired resolution of the dataset. The power of two subdivision of the 90° face requires less than 32 levels of a quad tree to represent a sub centimeter resolution dataset as shown in Table 1, sufficient for present and future high resolution datasets.

Table 1. Pixel resolution in longitudinal direction at various levels of quad tree

Level #	Degree/pixel	Meters/pixel
9	0.17578125	19,567.949
10	0.08789063	9,783.975
11	0.04394531	4,891.987
…	…	…
24	0.00000536	0.597
25	0.00000268	0.299
26	0.00000134	0.149

It is assumed that the geographical data is in WGS84 coordinate system and the coordinate of a grid point is represented by B, L, H where B, L and H is denote longitude, latitude, and elevation respectively. The following transformation functions are used to convert the WGS84 coordinates to open GL world space coordinate system.

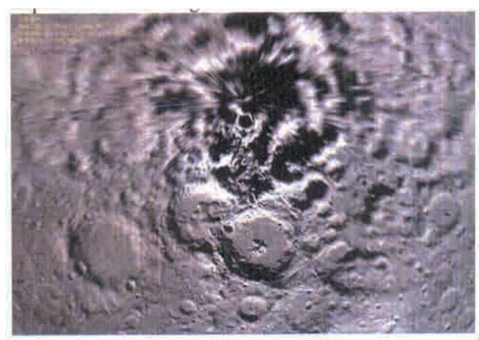
$$X = (N + H) \cos B \cos L$$

$$Y = [N(1\text{-}e^2) + H] \sin B$$

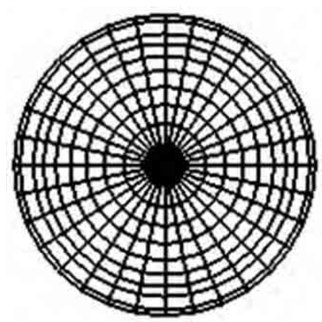
$$Z = (N + H) \cos B \sin L$$

where $N = a/\sqrt{(1\text{-}e^2 \sin^2 B)}$ $e^2 = (a^2 - b^2)/a^2$ a = major radius of the globe, b = minor radius of the globe.

This coordinate conversion can effectively be implemented in GPU. This quad tree decomposition of the globe describes different LOD of the globe by different levels of the quad tree. The quad tree decomposition of the region along equator (between –45° to +45° latitude) perfectly aligns with the natural latitude-longitude boundaries on the globe. It helps in searching and loading of a suitable LOD tile during the rendering process. If the same quad tree decomposition is extended for the polar region, the projection distortion increases due to convergence of all the longitudes at the poles[12] as shown in Fig. 2.



(a)



(b)

Figure 2. (a) Projection distortion in lunar pole[12], (b) longitude convergence at pole.

3.1 Equator Region

The globe is divided in to six faces of 90° x 90° in our method. There are four faces along the equator. These four faces are the root tiles of the quad trees. These root tiles are sub divided by two along latitude and longitude to generate four sub tiles at next resolution level. These tiles are recursively subdivided up to a desired depth of partition based on the LOD requirement of the rendering process. The quad tree decomposition of the globe face is shown in Fig. 3 (a).

The quad tree decomposition makes use of the frustum parameter to decide the depth of partition of a tile. The tiles closer to the viewer eye are partitioned with a greater depth than those tiles that are farther from the viewer eye. A LOD evaluation system decides the depth of portioning a tile in the quad tree. Each square tile regions in the quad tree are rendered by the four triangles. The final rendering of tiles requires refinement of the triangular mesh. There exists different level of depth in the quad tree representation of each globe face. This difference creates crack during the rendering process as shown in Fig. 3 (b).

This problem is resolved by partitioning the triangles at the tile boundary up to a depth level of its neighboring tile as shown in Fig. 3 (c). This way different level of depth of quad tree is used to represent the face of the globe that is triangulated and rendered during the rendering process.

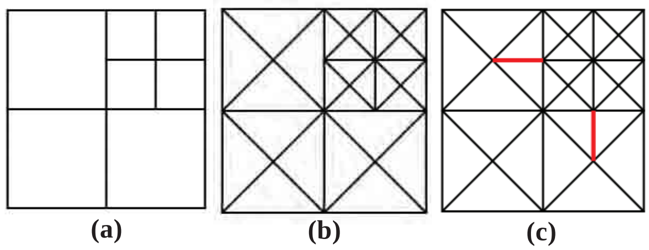


Figure 3. (a) Quad tree decomposition (b) Tiles rendered at different depths, and (c) Tiles refinement.

3.2 Polar Region

As discussed earlier that the similar quad tree decomposition will not work for the polar region where all the latitudes converge. The region along the longitude gets compressed but remains unaltered along the latitudes. This results in a radial blur centered at the poles. A different partitioning scheme is required for polar region so that the quad tree decomposition can be extended for the polar region without facing any distortion at the poles. As discussed earlier, the polar face is circular in nature. This circular face is partitioned from center in to four equal regions resulting in four-quarter circular regions. Each quarter circular region is partitioned using quad tree approach as shown in Fig. 4. The quarter circular polar region has three corner points, the fourth point is assumed as the mid point of the circular edge. Now the quarter circle has the four edges. The mid-point of these edges are used to partition this in to four regions as shown in Fig. 4.

This partitioning results in two circular regions and two quadrilateral regions. This decomposition can recursively be done for any depth of the quad tree to represent the polar region at different levels as shown in Fig. 4. The tiles of the quad tree are triangulated and it is refined also for the cracks at the adjacent levels in a similar manner it was done for equator region.

The quad tree decomposition of the polar region improves the quality of rendering as the quad regions at all levels remain in good quadrilateral shapes. This scheme does not require any additional computational step as the number of regions generated remain same as in simple spherical representation case.

3.3 Selection and Rendering of LOD model

The datasets representing the globe are huge in size. All of then cannot be accommodated in the main memory or the video RAM of the computer. These datasets are tiled and mipmapped before using them for the visualization. The six faces of the globe are represented by the quad tree structures. These quad trees remain in the main memory. The selection of the appropriate tiles is done based on the frustum parameters and location of the viewer. The distance of the tile from the eye is used to compute the resolution level of the tile as shown in Fig. 5.

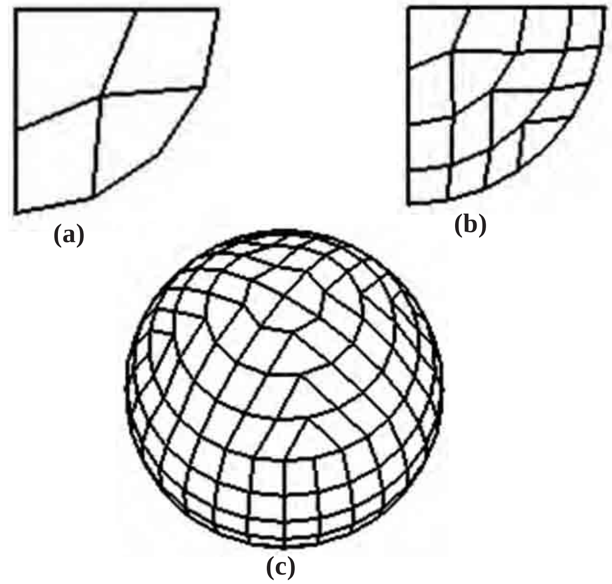


Figure 4. Quad tree decomposition of the polar face (a) level 2, (b) level 3, and (c) complete globe.

The selected resolution level tile is loaded to video memory for rendering using the least recently used (LRU) policy. This application is implemented using the Geomatrix software development tool kit. It uses cache size of 128 MB for texture data and 64 MB for geometric data.

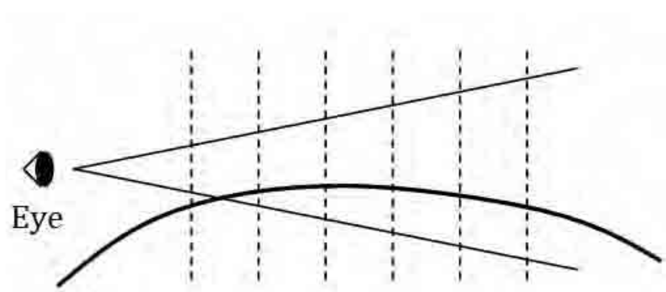


Figure 5. View Frustum on Globe surface & LOD ranges.

4. RESULTS AND CONCLUSION

The test hardware is a dual AMD Opteron 252 at 2.59 GHz with 4 GB of RAM and an NVIDIA Quadro FX 3400 with 256 MB VRAM. The cache size of 128 MB is used for texture and 64 MB for geometric data. The SRTM digital elevation data (25 GB) of complete globe is used at a resolution of 90 arc s The Blue Marble texture image at 1 km per pixel resolution is used as a base map of the globe. A predefined path is used to record the performance of this application. It begins with a very wide-angle view of the globe and moves to a closer view in J&K region in India. The rendering of the globe at different altitude shown in Fig. 6.

The execution starts with an empty data cache and data are loaded as needed during the rendering of globe at different resolution levels. The recorded results are shown in Fig. 7. It shows the frame construction time in milliseconds and the number of triangles rendered in each frame during the execution.

The initial frame construction time is high due to loading of data in to empty cache. The sharp peaks in the graph show the page faults. The consistent frame construction time shows that this approach is useful for real time rendering applications. The less time required to construct each frame also guarantees the high frame rate rendering.

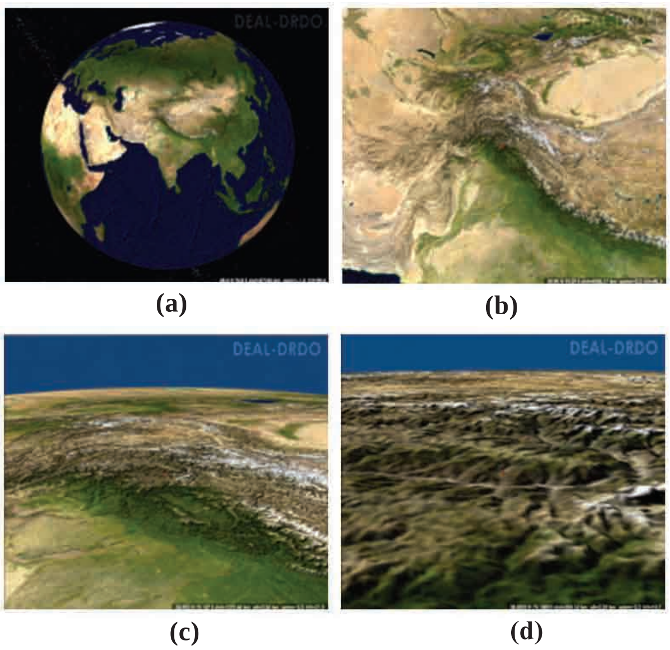


Figure 6. Globe views from (a) 27180 km, (b) 4828 km, (c) 1278 km, and (d) 204 km.

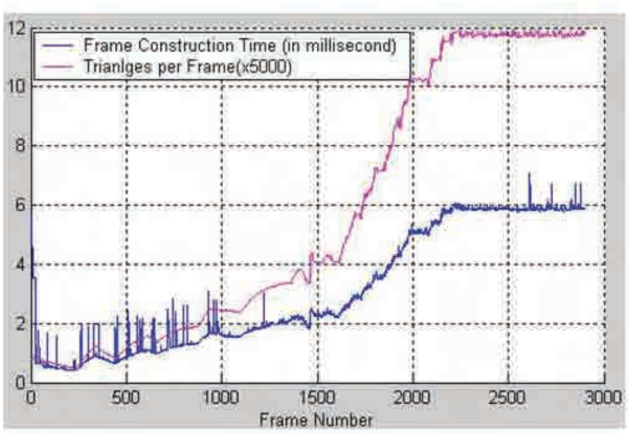


Figure 7. Frame rendering performance at 1280 x 1024 screen resolution.

ACKNOWLEDGMENTS

This work was done at Image Analysis Center (IAC), Defence Electronics Applications Laboratory (DEAL), Dehradun, India. The author is thankful to Dr SC Jain, Group Director, IAC for providing all the support and resources to complete this work. The author is also grateful to Mr RC Agarwal, Director, DEAL for giving the opportunity to work on this project.

REFERENCES

1. Farr, T.G. *et al.* The shuttle radar topographic mission. *Rev. Geophysics*, 2007, **45**(2), RG2004.
2. Geomatrix software development toolkit. http://www.geofusion.com (Accessed on 14 Dec 2012)
3. Duchaineau, M.; Wolinsky, M.; Sigeti, D.E.; Miller, M.C.; Aldrich, C. & Mineev-Weinstein, M.B. ROAMing Terrain: Real-time optimally adapting meshes. Lawrence Livermore Nat'l Laboratory, US, Oct. 1997. Technical Report No. UCRL-JC-127870.
4. Lindstrom, P.D.; Ribarsky, K.W.; Hodges, L.; Faust, N. & Turner, G. Real time, continuous level of detail rendering of height fields. *In* the Proceedings of ACM SIGGRAPH, 1996, pp. 109-18.
5. Lindstrom, P.; Koller, D.; Ribarsky, W.; Hodges, L.; Bosch, A. den & Faust, N. An Integrated global GIS and visual simulation system. Graphics visualization, and usability center, Georgia, Inst. of Technology, 1997.
6. Lindstrom, P. & Pascucci, V. Visualization of large terrains made easy. *In* the Proceedings of IEEE Conference on Visualization, 2001, pp. 363-74.
7. Boer, W. H. de. Fast terrain rendering using geometrical mipmapping, http://www.flipcode.com/, Oct. 2000. (Accessed on 14 Dec 2012)
8. Levenberg, J. Fast view-dependent level-of-detail rendering using cached geometry. *In* the Proceedings of IEEE Conference on Visualization, 2002, pp. 259-65.
9. Pajarola, R. & Gobbetti, E. Survey of semi-regular multi-resolution models for interactive terrain rendering.The Visual Computer, 2007, **23**(8), 583-605.
10. Losasso, F. & Hoppe, H. Geometry clipmaps: Terrain rendering using nested regular grids. *In* the Proceedings of ACM SIGGRAPH, 2004, pp. 769-76.
11. Kooima, R.; Leigh, J.; Johnson, A.; Roberts, D.; SubbaRao, M. & DeFanti, T.A. Planetary-scale terrain composition. *IEEE Trans. Visualization Comp. Graphics*, 2009, **15**(5), 719-33.
12. Yang, Z.; Qing, X.; BaoMing, Z.; JianSheng, L. & ChaoZhen, L. Lunar geomorphy 3D visualization method, ISPRS TC VII Symposium, Austria, 2010, **38**(7B), pp. 674-79.
13. Zhang, R.; Lu, C. & Qu, W. A study on the quad tree-based LOD algorithm.. *In* the Proceedings of IEEE Conference on Communication, Software and Networks, May 27-29, 2011, pp. 546-49.

Contributor

Mr Sudhir Porwal received MTech (Comp. & Info. Technol.) from Indian Institute of Technology, Kharagpur in 2005. He is currently working as senior scientist in Image Analysis Center, Defence Electronics Applications Laboratory, Dehradun, India. His research interests are in the field of GIS, three-dimensional modeling and visualization, computer vision, video processing, software design & web applications.