Modeling of Earth’s Gravity Fields Visualization Based on Quad Tree

LUO Zhicai1,2, LI Zhenhai1, ZHONG Bo1

1. School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan 430079, China
2. Key Laboratory of Geospace Environment and Geodesy, Ministry of Education, 129 Luoyu Road, Wuhan 430079, China

© Wuhan University and Springer-Verlag Berlin Heidelberg 2010

Abstract The problems of the earth’s gravity fields’ visualization are both focus and puzzle currently. Aiming at multiresolution rendering, modeling of the Earth’s gravity fields’ data is discussed in the paper by using LOD algorithm based on Quad Tree. First, this paper employed the method of LOD based on Quad Tree to divide up the regional gravity anomaly data, introduced the combined node evaluation system that was composed of viewpoint related and roughness related systems, and then eliminated the T-cracks that appeared among the gravity anomaly data grids with different resolutions. The test results demonstrated that the gravity anomaly data grids’ rendering effects were living, and the computational power was low. Therefore, the proposed algorithm was a suitable method for modeling the gravity anomaly data and has potential applications in visualization of the earth’s gravity fields.

Keywords LOD; Quad Tree; earth’s gravity fields; multiresolution rendering

CLC number P223

Introduction

With the progress of gravitation measuring technique and the optimization of data processing methods, it is convenient and quick to get large scale and high accuracy gravitation data. How to express the earth’s gravity field exactly is one of the goals the geodetic scholars want to achieve. The presentation method and technology of the earth’s gravity field need to be broken through. The visualization technology can provide new research way for the earth’s gravity fields’ representation, research, and application. For modern PC, the rendering speed of large scale and high resolution’s gravity data is very slow, the rendering is unreality, so simplification methods are required. In this paper, we apply LOD algorithm in multiresolution rendering to build model of the gravity fields data. Finally, we want to achieve multiresolution rendering of the earth’s gravity fields.

LOD (level of detail) is one kind of valid methods for large area topography simplification. When a target is far from the observer, the whole area becomes a very small one after projection and transformation, so we need not to draw the whole area’s details, and we can merge some triangles properly so as to keep visualization effects.[1] There are two classical simplification methods, one is regular simplification for regular data, and the other is irregular simplification for discrete data. The regular simplification usually adopts a...
strategy of Up-To-Down, has two typical models: Binary Tree and Quad Tree. They begin with the lowest level of the scene and then increase details continuously under some conditions. Irregular simplification usually adopts a strategy of Down-to-Up, and this method is less used.

It has been widely studied for terrain LOD simplification. Hoppe (1995) put forward a model of progressive mesh to solve the dynamic terrain simplification and solved some problems that appeared in static terrain simplification. Lindstrom (1996) employed recursion Quad Tree to cut one terrain into many small pieces (tessellates) and then established an approximate height diagram. This algorithm can combine the characteristics of regular grids’ structure and the high efficiency of irregular grids’ rendering. Moreover, he proposed real-time terrain simplification method that took the visual space errors as evaluation criterion. A description of ROAM (real-time optimization self-adapting mesh) was given by Mark Duchaineau. Henri Hakl introduced “Rhombus Terrain Algorithm” method to avoid T-cracks.

For the most gravity data are grid, the Quad Tree structure is employed as a suitable simplification method for the gravity data’s modeling. The modeling of the Earth’s gravity field data is discussed in the paper by using regular simplification method and terrain partition LOD of Quad Tree. The constructing of Quad Tree, node evaluation system, the modeling, and rendering for the grids’ model are discussed in Section 1. The experiment results are given and analyzed in Section 2. Finally, some conclusion and suggestions are drawn from the test results.

1 Dynamic multiresolution simplification

The definition of dynamic multiresolution simplification is as follows: During the simplification, the suitable nodes’ integrations are selected dynamically under different resolutions, and they have to completely overlay the whole area, so as to attain the multiresolution rendering of data over the whole district. While roaming, the 3D data volume of different resolutions can be seen by browser according to the viewpoint and current data’s undulation. The node integrations of dissimilarity resolutions are given by the partition of evaluation system, so how to construct the evaluation system is of utmost importance.

1.1 Constructing of Quad Tree

The whole district is considered to be one original root node, and we check out whether this node satisfies some certain partition condition; if not, it will not be partition and be saved as a leaf node. Otherwise, the root node will be continuously divided up to four subnodes by recursion, until the level is up to the biggest. At last, all of the leaf nodes will be rendered as simplification result. This is the whole process of the Quad Tree database construction of multiresolution 3D data model. The more partition times, the bigger resolution value, namely, the sampling density will be improved 100% if the partition time is added 1. It is noticed that a node is not explained to a vertex but represents a rectangle of district piece, saving nine records, including one central point, four corner points, and four edge middle points. The Quad Tree structure must have a node evaluation system to judge when a node need to be partitioned continuously, or to be abandoned directly, and how to achieve multiresolution rendering through the distance between the viewpoint and a node’s center.

1.2 Node evaluation system

While roaming on the LOD 3D data volume, the details will become abundant gradually when the viewpoint is closer to the surface of 3D data but will become insufficient when the distance is long. Therefore, the node evaluation function needs to be established to partition the nodes that should match the vision characteristic during the 3D data simplification. It has two parts, the first one is viewpoint related, and the other one is the data roughness related. It is decided whether the node needs to be partitioned continuously through the node evaluation system. The apparent distance $l$ and the node’s size $d$ are shown in Fig.1.

![Fig.1 Schematic diagram of apparent distance related](image-url)
Viewpoint related. The more details of the data surface are expected to achieve when the apparent distance becomes shorter; otherwise, less details will be obtained. Another important factor that must be considered is the node’s size, a combination of the two main factors leads to the viewpoint-related formula:

\[
\frac{l}{d} < C
\]  

(1)

Where \( l \) is the apparent distance between viewpoint and the node’s center, \( d \) is the node’s size, and \( C \) is a regulation factor that can be changed. The larger \( C \) is, the more details are rendered. If the current condition satisfies the Eq. (1), the node needs to be divided repeatedly.

Roughness related. It means that the rougher region is, the more details should be rendered for approaching the real abrupt area, but it need not excessive grids on the flat area. The roughness-related errors are shown in Fig.2.

\[
\frac{l}{d} < C
\]  

(2)

Where \( C \) is a factor for roughness’ regulation. The bigger \( C \) is, the more details are rendered. Therefore, the combined node evaluation formula is

\[
f = \frac{l}{d \times r \times C_1} < 1
\]  

(3)

If \( f < 1 \), the node needs to be divided repeatedly.

### 1.3 Modeling and rendering the grids’ model

The Quad Tree should be updated to satisfy with the requirements of the standard before the grids’ rendering. The difference value between the two border nodes is no more than 1, or else, T-cracks will appear in the joint side. Fig.3(a) is a legal case of the Quad Tree example, and Fig.3(b) is an illegal case, for the difference value between the two neighbors nodes’ level is 2.

In order to achieve the legal Quad Tree structure, a set of rules must be set up. Generally, the method of twice roaming Quad Tree is adopted. For the first time, the earth’s gravity data grids’ drawing is achieved, for the second time, the grids are rendered, and the T-cracks between the neighbor nodes are eliminated. A more effective method to generate Quad Tree structure is employed in the paper, i.e., traversing Quad Tree structure by Breadth-First Traversal. That is, all nodes that belong to one level are generated in one time; therefore, the Quad Tree data structure is traversed only once. While traversing, two queues are adopted: one is used for recording all treating nodes in current level, and another queue is used for recording the generated partition nodes in next level. After all nodes are processed in current level, the nodes in next level begin to process by simply exchanging two queues. All nodes that do not need to be
divided or already arrived at the last resolution will be sent to API for rendering. The invisible nodes are directly thrown away.

The gravity data grids’ rendering is generated by the recursion method too. While traversing the whole Quad Tree structure, all nodes that cannot be divided will be rendered, that is, they have arrived at the last level and become leaf nodes. Triangle fan method was employed to render the nodes in the paper. It is a reasonable way for drawing; the reason is one node includes one central point and some points around the central one. This arrangement can just right form a triangle, and Fig.4(a) shows this structure.

![Fig.4 Two methods for processing T-cracks](image)

As shown in Fig.4(b), T-cracks appeared between two dissimilarity resolutions nodes must be considered for grids’ rendering. For eliminating these T-cracks, Fig.4(c) shows one processing method by adding one line to the jointed side, and this method is complex but general; for the difference value between two neighbors nodes’ level, it can be any value. Furthermore, Fig.4(d) shows another method for eliminating the T-cracks by removing one line. This method of eliminating T-cracks is easy, the difference value between two neighbors nodes’ level is no more than 1, and it is employed in this paper.

2 Experiment results and analysis

This experiment applies the $2' \times 2'$ gravity anomaly data calculated by EGM96 model to check the rationality by using the method of simplification and algorithm of eliminating T-cracks. The data model’s size is $120 \times 120$, the original number of grid points is 14400, and the number of triangles that are generated by no simplification is about 57600. Treating so many triangles, if the simplification method is not applied, a great deal of system resources will be taken up, and the rendering time will be longer. Therefore, the result achieved through the method of LOD based on Quad Tree is more ideal. The test platform is listed as follows: Intel Core Duo CPU 2.53G, 2GB RAM, the DDR VRAM of the NVIDIA GeForce 9500 GT 512M. The operating system is Windows XP sp3.

![Fig.5 Grids modeling results](image)
(c) or comparing (b) and (d), the details vary with the level. Depending on all above studies and experiments, we can conclude three summaries.

The consumption of the memory. The consumption of the memory by using algorithm of LOD in this paper is related to the district’s size, and this kind of relation is linearity. For each point on the data’s district, the information needs to be known as follows: the value of the gravity anomaly (1 byte), roughness (float type, 4 byte), and Quad Tree information (1 byte). That is, each point is saved for 6 byte.

Speed of rendering and quality of image. After using the simplification method, the grid data rendering speed is faster than the traditional methods, such as static simplification and the simplification method based on single nodes’ evaluation system. The quality achieved in this paper is not as good as the TIN data, for TIN reflects the local data surface’s feature well.

The number of triangles. It is related not only with $C$ and the level but also with the undulation of the 3D data, namely, $C_r$. More triangles will be rendered if the apparent distance is getting shorter or roughness is bigger.

3 Conclusion

This paper mainly introduced the principle of Quad Tree simplification, the realization of LOD based on Quad Tree algorithm, researched on the data’s organization method during the modeling of the earth’s gravity field data, applied this method to the gravity anomaly data’s modeling, then organized and processed the data, controlled the earth’s gravity field data’s LOD by roughness and apparent distance, and solved the geometric deformation appeared during ranging. Experiment results showed the proposed algorithm could satisfy the data’s real-time rendering and eliminated the T-cracks. Therefore, the LOD based on Quad Tree is an effective, accelerative, real-time algorithm. As for how many levels should be given, it should be established on the basis of the request of application and the capability of PC. Moreover, original data may not satisfy the request, and resampling is needed. The effects of results’ precision caused by resampling still should be researched and resolved in the future. Based on the all above studies and algorithms, our future works are the earth gravity field’s visualization in some ways: the multidimensional representation, multisource data rendering, large-scale curve area visualization, etc. The further studies on LOD include the option treating and characteristic keeping on the data model’s surface, establishing the uniform criterion of the errors’ evaluation, the smooth transition of different levels, etc.

References

[1] Yao Huimin, Cui Tiejun, Shao Shixin, et al. (2007) LOD terrain model and data organization method based on Quad Tree structure [J]. Geomatics World, 6: 56-59 (in Chinese)

[2] Lindstrom P, Koller D, Ribursky W, et al. (1996) Real-Time, continuous level of the detail of rendering of height fields[C]. SIGGRAPH’96, Proceedings of the 23rd Annual Conference on Computer Graphics, New Orleans, LA

[3] Hoppe H (1998) Smooth view-dependent level-of-detail control and its application to terrain rendering[C]. Proceedings of the Conference on Visualization ’98, Research Triangle Park, NC

[4] Duchaineau M, Wolins M, Sigeti D E, et al. (1997) Roaming terrain: real-time, optimally adapting meshes[C]. IEEE Visualization ’97, Phoenix, AZ

[5] Wang Zhen, Hu Min, Li Xiang (2007) Quick simplification for LOD terrain with dynamic multi-resolution based on Quad Tree [J]. Computer Applications, 27(7): 1641-1643 (in Chinese)

[6] Liu Haifei, Zang Deyan, Che Jianren (2006) The LOD terrain simplification based on Quad Tree and geomorphing correction [J]. Geomatics & Spatial Information Technology, 29(2): 48-51 (in Chinese)

[7] Pan Liliang (2003) Research on the large scale terrain rendering based on LOD in outdoor scene [OL]. http://www.hellocpp.net/Articles/Article/145.aspx

[8] Jing Tao (2005) An efficient algorithm of LOD for 3D terrain simulation based on Quad Trees [J]. Computer Emulation, 22(11):123-126 (in Chinese)

[9] Zhou Faliang (2007) Application of LOD based quad-tree in terrain generation [J]. Computer Emulation, 24(1): 188-191 (in Chinese)