RESEARCH ON
MULTI-SCALE TERRAIN MODEL LOD

Xia Qing, Zhang Shitao, Song Guoming
Department of Cartography, Institute of Surveying and Mapping
66 Longhai Road, Zhengzhou 450052, P.R.China
E-mail:windmill@public2.zz.ha.cn

ABSTRACT

In recent years, large scope terrain visualization has attracted general attention. In order to describe large scope terrain, a large quantity of digital terrain models are required, while in order to enhance the truthfulness of the scene, careful digital terrain models are adopted. As a result, terrain database becomes too large. Enlarging digital terrain model database gives rise to many knotty problems, esp. in real-time interface system. Now, efficient management and hierarchical terrain modeling have been developed, but compared with the present problems, there is still a long way to go. For some special system, such as Digital Earth that requires multi-scale terrain visualization, no effective measures are available. What is going to talk about in this article may be of some help in large scope and multi-scale terrain visualization.

1 INTRODUCTION

The result of research into terrain visualization indicates that hierarchical terrain modeling is the key to the real-time render of terrain. LOD(Level of Details) is generally classified into Real-time LOD and Pre-computed LOD. Real-time LOD redefines triangles depending on where the view is. Therefore, Real-time LOD is also known as view-dependent LOD. Pre-computed LOD is where a certain number of models, which have different precisions and different amount of triangles, are created in advance, and the important basis of modeling is the errors given by users, irrelevant of observer's position. Therefore, pre-compute LOD is also called view-independent LOD.

The aforementioned two LOD have their advantages and disadvantages respectively. Pre-computed LOD is comparatively simple, easy to be achieved, and because the errors are able to be limited, it is especially applicable to the occasion where the modeling errors are highly limited. But in the Pre-computed LOD, a series of models with different precisions need to be created and stored ahead of time, and the data are large, which bring pressure on the storage and transmission of the data. Moreover, instantaneous switching between LOD meshes may lead to perceptible “popping”. Real-time LOD is able to solve the “popping” and reduce data on hierarchical models. It is applicable to the occasion where data transmission and storage are limited, such as Internet. In fact, most Real-time LODs need data pre-processing to make the order of collapse. When the original model changed, this order will be re-made, so Real-time LOD doesn’t fit dynamic modeling.

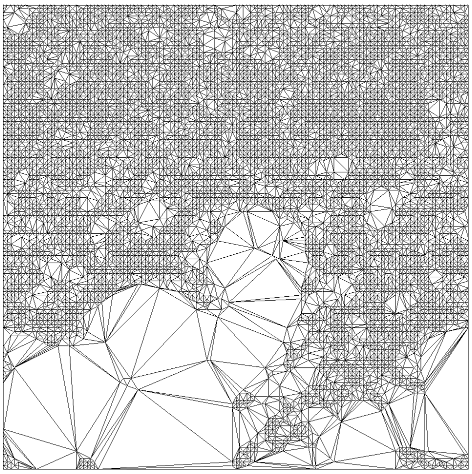
View-dependent LOD can confine the amount of triangles to be rendered within necessary limit. But because triangles should be redefined in real-time, the data sent to graphics-generator are not continuous, and then the graphics-generator isn’t performed effectively. View-independent LOD will cause the existence of superfluous data, but without real-time data processing, it can send triangles data continuously to graphics-generator and exert all system's powers. Especially in some complicated simulation applications, large numbers of data irrelevant to rendering need to be processed, so time saved by view-independent LOD plays an important role in increasing performance of system. In a word, there is no difference in performance between the view-dependent LOD and view-independent LOD.

LOD has been put into a wider user in VR, and there are many successful examples. These approaches are widely used for static isolated objects, such as house, car, aircraft and so on, but are not generally well suited for terrain heightfields. When LOD is applied to the rendering of objects like terrain whose meshes are continuous and space is limitless, two processing work will to be done. First, a large terrain model is divided into many small blocks and hierarchical models are created for each block. Then, identical vertexes must be required on common edge between two blocks in order to keep surface is continuous. If above two conditions are met, LOD can be used on terrain model visualization. It is shown in Figure 1.

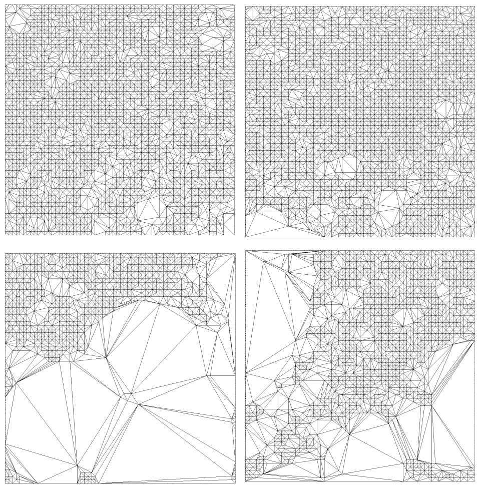
Although LOD can be applied on terrain visualization when some conditions are satisfied, bad effects will be brought. On the one hand, because there are fixed vertexes on edges of hierarchical models, narrow triangles may be redefined, especially in coarse models. On the other hand, these conditions limit LOD to be applied on multi-scale and dynamic terrain visualizations. In this article, a LOD algorithm on multi-scale terrain visualization will be introduced.

2 SCALE vs. LOD

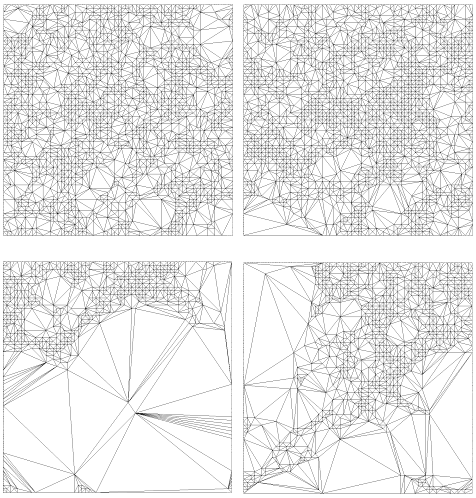
It is thought by someone that scale of map and terrain data will become past history if LOD is used widely. But it is not the case, however, for there is some difference between the scale and LOD. The precision of map data on any scale is confined with necessary limits, so is the precision of the terrain model on the basis of the map data. Compare with a DEM on the basis of 1:100,000 map data and the other on the basis of 1:10,000 map data, density of vertexes of both models are perhaps the same, but the precision of the former is lower. If a DEM is replaced by other DEM on basis of larger scale map data, the users have to be faced with existence of superfluous data and processing more basic data and lack of larger scale map. Besides, the map data on any scale is acquired through cartographic generalization, and spatial information is accepted and rejected scientifically, not increase or reduce simply. At this point, cartographic generalization can’t be replaced by LOD fully and the scale will exist for a long time.



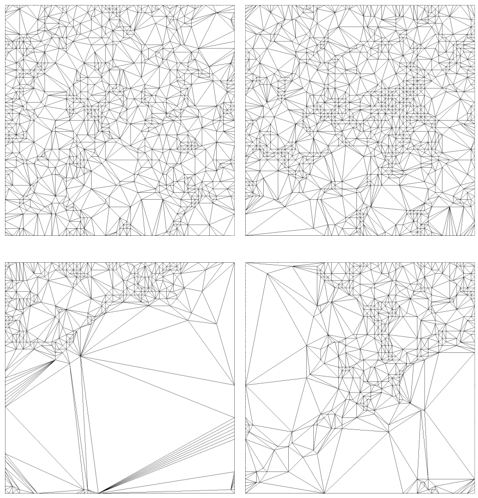
a. Original Mesh
(16648 faces)



b. Divided into four blocks but not be simplified
(16978 faces)



c. Simplified meshes of four smaller blocks
(10720 faces)



d. Simplified further meshes
(5774 faces)

Figure1. Terrain model dividing and LOD

In some applications, such as Digital Earth, multi-scale terrain visualization is required. At the same time, perceptible “popping” resulting from switching between LOD meshes and the pause resulting from data reloading must be avoided. In general, it is difficult to keep identical vertexes on the common edges between different scale DEMs, so it is impossible to keep surface continuous and not allowed that there are different scale terrain models in a scene. When the different scale terrain models are required, all models in the scene are replaced. Although it seems simple, the data reloaded are large, the time of reloading is long, and it is oppressive for user to keep waiting. Therefore, it is necessary to develop multi-scale LOD to meet the requirement of application.

3 LOD ON MULTI-SCALE TERRAIN MODEL

It is well known that precisions of different scale terrain models are different, so is area of terrain described by them are different. The large the scale of model is, the small area it covers. In this algorithm, the scope of models are not necessarily equal, but it must satisfy some conditions. The scope of a smaller scale terrain model is the total of scope of some larger scale models, and the scope of a terrain model must be included in the scope of one smaller scale terrain model only. To manage effectively, it is prescribed that the area of the terrain described the larger scale model is the total of that four smaller scale terrain models. For example, if a 100km x 100km terrain is described by a 1:100,000 terrain model, then it is described by four 1:50,000 terrain models whose area is 0 x 50 km^2, and sixteen terrain models on the scale of 1:25,000 whose area is 25 x 25 km^2. It is shown in Figure 2.

Other conditions should be satisfied during terrain modeling. First, there is an edge of a triangle overlapping the edge of model at best. Otherwise, the triangle must be redefined. It is shown in Figure 3. Second, if a Grid DEM is created, the step between two vertexes is several times as long as that of the lager scale DEM. If the step of the 1:250,000 Grid DEM is 50m, then the step of the 1:100,000Grid DEM must be 25m or 12.5m.

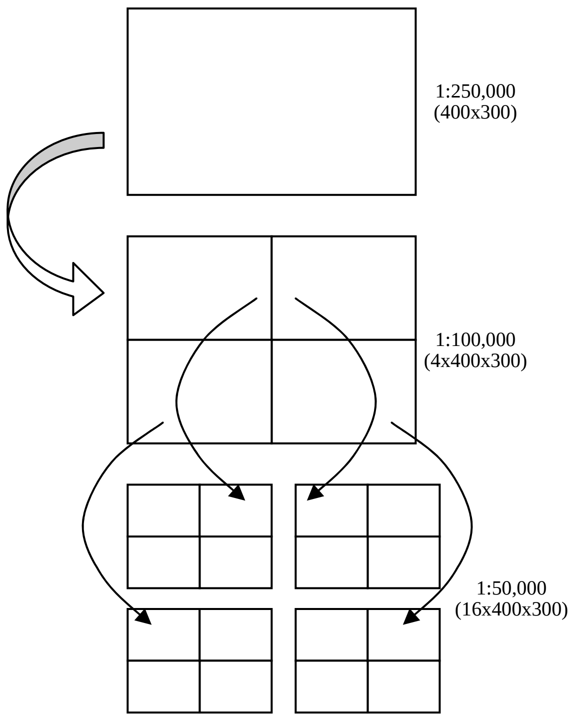


Figure2. Hierarchical relation of multi-scale terrain models

The different scale models can be used as view-independent hierarchical models, but it is different from general hierarchical models, scopes of models on different details level are not uniform. So, when a model on lower details level whose scope is larger switches to other on higher details level, four larger scale terrain models will be loaded in fact. And when a model on higher details level meets the conditions to switch to lower details level models, the switching will be not executed until all of other models whose scope is included in the scope of the lower details level model meet the conditions of switching.

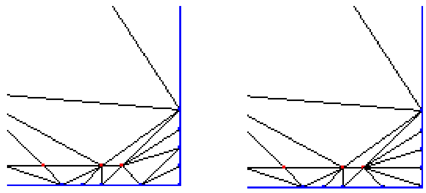
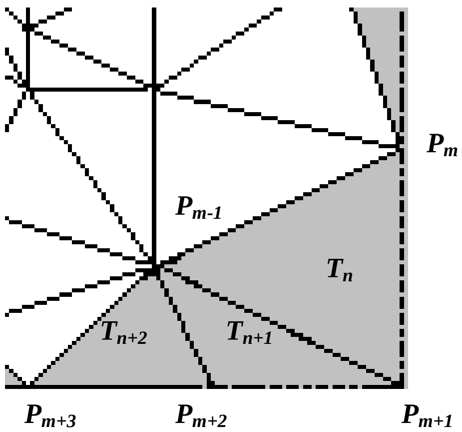


Figure 3. There is an edge of a triangle overlapping the edge of model at best

To speed the switching, two indexes will be created. One is about hierarchical relation of models, which used to get model will be load rapidly. The other is about the border of model, which is used to find out the vertexes or triangles on the edges.

In general, the former can be saved alone, while the latter is saved with other data and information of the model. In order to record the second index, data structures will be redefined. Records of vertexes and triangles on the four edges of model are defined as illustrated in Figure 4. Indexes of triangle on edges are recorded and triangles are redefined to make sure that the edge defined by first two vertexes of triangle record overlaps the border of model.



Triangulated Network Data Struct {
………
South: T_{n+1} , T_{n+2} , …...
East: T_n , …...
North : ……
West: ……
………
}
T_n {P_{m+1}, P_m, P_{m-1}}
T_{n+1} {P_{m+1}, P_{m+2}, P_{m-1}}
T_{n+2} {P_{m+21}, P_{m+3}, P_{m-1}}

Figure 4. Records of vertexes and triangles on the four edges of model

Usually, it is difficult to ensure that there are identical vertexes on the common edge between the two models on the basis of different scale, and it is difficult also to keep the surface continuous. So a technology to adjusting edge self-adaptability is developed. It is key to multi-scale terrain model render.

In general, there are more vertexes on the border of the model on large scale. To keep vertexes identical on the common border between two models on different scale, there are two options. One is to create some vertexes on the common border of the smaller scale model, and the other is to delete some vertexes on the border of the larger scale model. In this algorithm, the former is selected to increase the precisions of meshes. As a result, the triangle whose one edge overlap border of the model will be divide into more small narrow triangles. Concerning performance of system, these narrow triangles are not optimized.

Another problem arises. The vertexes of terrain model of smaller scale and on the common border may not subset of those vertexes of the higher scale terrain model. In this case, the algorithm is illegitimate. So the pre-processing must be done to solve the problem.

Two cases should be taken into consideration. One is about Grid DEM. If models meet the above requirements and elevation is not regarded, the set of vertexes of model on the smaller scale on the common border is subset of that of the model on larger scale. But different original data of models will lead to different elevation of vertexes whose plane position is the same. If so, elevation of vertexes of smaller scale model is changed to that of higher scale model, because the precision of larger scale model is superior to that of smaller scale model. The other case is about TIN DEM. If vertexes of TIN are identical with that of digital contour line, then plane position of vertexes, which are on common border and whose elevation is the same, are different. So the plane position of vertexes of smaller scale model is adjusted on the basis of that of larger scale model. Actually, only one coordinate is adjusted.

There are two approaches to adjust the vertexes and triangles on the common border between two different scale models: Pre-computed adjustment and Real-time adjustment. Pre-computed adjustment is where two pairs of fixed triangulated networks are defined. Here, the triangulated network of the model is divided into parts. One is composed of triangles on the border (one vertex of the triangle is on the border of the model at least.), and called edge-part. The other is composed of the inter-triangles (no vertex of the triangle is on the border of model), and called inter-part. In fact, edge-part triangulated network includes four smaller parts to four borders of the model. Each smaller edge-part triangulated network has two groups of triangulated networks: one has the created vertexes on the border and more triangles defined, which is called redefined-group and is loaded when the model border upon larger scale models, and the other has original vertexes and triangles, which is call original group. So the edge-part triangulated network has four original-groups and four redefined groups. Because the model borders larger scale model on one side at a time, so a redefined-group will be loaded at most. Real-time adjustment is where the side-part triangulated net will be redefined according to the models bordered during rendering.

The two adjustments have their advantages and disadvantages respectively, just like Pre-computed LOD and Real-time LOD.

In the above, different scale models are used as hierarchical terrain models. For a model on any scale, the Real-time LOD, such as progressive mesh, can be used. About Real-time LOD, many articles have released, so it is no need to talk about in this article.

Rendering the terrain models on different scale in a scene is the key to multi-scale and large scope terrain visualization. The algorithm described in this article is helpful to keep the terrain surface continuous when two different scale models border upon each other. Of course, there are many unsolved problems in multi-scale terrain visualization, for example, how to build the texture of multi-scale terrain model. These problems will be discussed in other articles.

REFERENCES

[1] H. Hoppe, Progressive meshes, *Computer Graphics (SIGGRAPH 1996 Proceedings)*, 99-108.

[2] H. Hoppe, Smooth view-dependent level-of-detail control and its application to terrain rendering, *IEEE Visualization 1998*, October 1998, 35-42.

[3] H. Hoppe, View-dependent refinement of progressive meshes. *Computer Graphics (SIGGRAPH 1997 Proceedings)*, 189-198

[4] L. De Floriani, E. Puppo, P. Magillo, Selective Refinement of Triangle Meshes:

Data Structures and Algorithms, *Technical Report* DISI-TR-98-02, Department of Computer and Information Sciences, University of Genova, 1997
[5] P. Lindstrom, D. Koller, W. Ribarsky, L.F. Hodges, N. Faust, G.A. Turner, 1996, Real-Time, Continuous Level of Detail Rendering of Height Fields, *SIGGRAPH 96 Conference Pro-ceeding*s, 109-118, 1996.
[6] Michael Garland and Paul S.Heckert, “Fast Polygonal Approximation of Terrain and Height Fields”, *CMU-CS-95-181*, 1995