A HYBRID LOD ALGORITHM BASED ON TIN MODELLING

Zhenhua. Feng [a, *], Hua. Qi [b]

[a]Surveying and Mapping Institute, Information Engineering University, ZhengZhou, HeNan, China, 450052, farmer398@sohu.com
[b]School of Civil Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China, 610031 - qi-3dgis@sohu.com

KEY WORDS: TIN, LOD, Muti-Resolution, Algorithm

ABSTRACT:

Triangulated Irregular Network (TIN) is a high-effect means expressing the terrain surface in the computer, compared with GRID model, TIN model can use less sampling to approach the terrain surface precisely, this is why we study Visualization of TIN model. The data structure of TIN model is complex, therefore to realize View-dependence LOD representation of TIN is difficult. General method is to partition the TIN, every tile adopt the different resolution according to the distance from viewpoint.Quadtree domain decomposition is adopted on TIN in this paper, and this is no longer simple rectangular partition to realize View-dependence LOD representation of TIN, but Through building a quadtree structure for tiling of TIN, and combining quadtree based LOD methods to realize controlling over the tiling of TIN effectively, this method can combines adjoining tile sometimes, the high limit cuts down the triangle quantity in the LOD mesh. This algorithm has combined TIN and quadtree structure, and combines quadtree based LOD methods and TIN based triangle mesh simplification methods to realize the view-dependence lever-of-detail controlling over the TIN model. Research indicates that this algorithm cuts down the triangle quantity in the LOD mesh and doesn't lose visual effect, a lever-of-detail mesh, similar with Grid and the TIN hybrid structure, has been acquired.

1. INTRODUCTION

Triangulated Irregular Network (TIN) is a high-effect means expressing the terrain surface in the computer, compared with GRID model, TIN model can use less sampling to approach the terrain surface precisely. The advantages of TIN model in the vertices count can achieve rendering small vertices in terrain visualization. This is why we study Visualization of TIN model. The data structure of TIN model is complex, therefore to realize View-dependence LOD representation of TIN is difficult. General method is to partition the TIN, every tile adopt the different resolution according to the distance from viewpoint.

This paper designed and implemented a algorithm to realize the view-dependence lever-of-detail controlling over the TIN model. The algorithm has two characteristics:

1. Quadtree domain decomposition is adopted on TIN, and this is no longer simple rectangular partition, but through building a quadtree spatial index structure to realize View-dependence LOD representation of TIN.
2. This algorithm combines quadtree based LOD methods and TIN based triangle mesh simplification methods to realize the view-dependence lever-of-detail controlling over the TIN model.

2. ALGORITHM IMPLEMENTATION

In this section we describe the algorithm details of the proposed method. At this point we assume we are given an irregular point data set, and the input point data set is distributed in rectangular area.

2.1 Overview

This approach uses quadtree domain decomposition to part the distribution area of the input point data set, and distributes the irregular points to the leaf nodes of quadtree. The input data is organized in an efficient spatial index structure that allows fast range query access for the quadtree construction process. We store the spatial index structure information and the irregular points in the leaf nodes into the file "QuadTree".

The rendering application reads data from the file "QuadTree", and then uses quadtree based LOD methods to triangulate, if the threshold of screen projection error cannot be met at leaf nodes, insert the irregular points contained in the leaf nodes in order of the value of the errors until the screen projection error is met or there are no terrain sampling points for insertion.

2.2 Quadtree Domain Decomposition

We split the domain of the point data set into quadtree tiling using max number of vertices in the quadtree tiles as a stop criterion (δ). The resulting quadtree hierarchy is not balanced, to avoid the T-shaped cracks, it is necessary to limit the differences of the adjacent leaf nodes at most one. Figure 1 shows the quadtree domain decomposition.

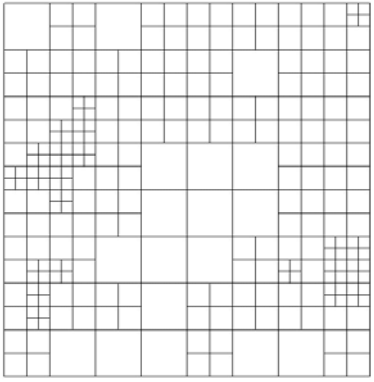


Figure 1. Quadtree domain decomposition (δ=50)

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

It is necessary to interpolate coordinates of corners and center of each leaf node, the x,y-coordinates are easy to get and the z-coordinates are averaged from these closes points. These interpolation points can be used in terrain expression.

We use a two-dimensional array to store quadtree information, in order to realize fast range query access. If the depth of the quadtree is N, the large of the array is $(2^N + 1) \times (2^N + 1)$. The quadtree hierarchy is not balanced, so some place is no data.

2.3 Restricted Quadtree Triangulation

In 2.2, we imposed an efficient quadtree hierarchy on irregular point set based input through the introduction of the interpolation points. These interpolation points form a grid. At real-time phase, firstly, triangulate on the grid using the method of restricted quadtree triangulation. If the threshold of screen projection error cannot be met at leaf nodes, insert the irregular points contained in the leaf nodes.

The restricted quadtree triangulation (RQT) method is an adoptive, hierarchical triangulation model to tessellate surfaces. RQT subdivision is constraint such that the levers of adjacent quadtree nodes differ by at most one.

In the top-down quadtree subdivision and triangulation process, there are some differences from the RQT in the leaf nodes triangulation, the triangulation of the leaf node needs that all the corners, center and the midpoint which had data are used in the triangulation. Therefore, the initial triangulation of leaf nodes can be foreseen in the pre-processing stage (Figure 2). In the next section we will describe the details of the triangulation of the leaf node.

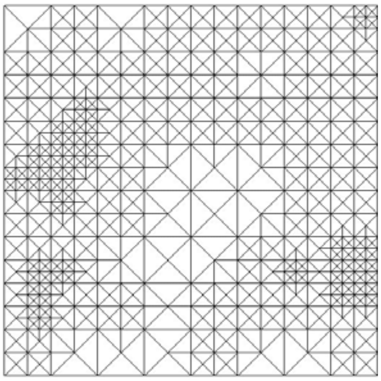


Figure 2. The initial triangulation of leaf nodes

2.4 Triangulation Of Leaf Nodes

In 2.3, we achieve the initial triangulation of leaf nodes, and then, it is necessary to constantly insert the largest local errors points in initial triangulation, while a modification when the point is inserted specifies a local change in a triangle mesh, which replaced a subset T_1 of its triangles with another set of triangles T_2 in such a way that T_2 fits the hole left in the triangle mesh by the removal of T_1 (Figure 3), the modification is described as a pair $M = (T_1, T_2)$, and we record the modification of by the data structure had defined. These are completed at the pre-processing phase, and it does not take up time of real-time rendering phase.

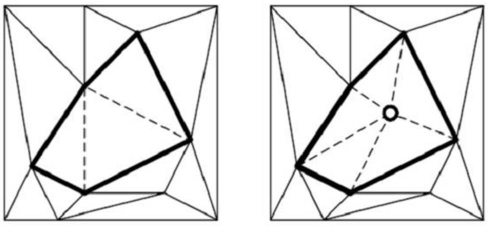


Figure 3. The modification when the point is inserted (Prof. P. Widmayer , 2001)

At real-time rendering phase, we use the modification to achieve the point insertion operations (M) or deletion operations (M_{-1}), Figure 3 shows the direction of movement in the point list when the point is inserted and delete.

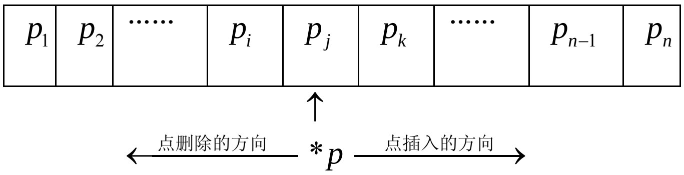


Figure 4. The direction of movement in the point list

2.5 Real-time rendering

Before a scene can be rendered, a criterion is necessary to judge when quadtree nodes need subdivision, and when leaf nodes insert the irregular sampling point. Criterion should be established to consider two aspects: first View-dependent; second the geometric error of terrain. Screen projection error covers these two aspects. The formula of screen projection error is:

$$\rho = \lambda \frac{\delta}{d} \qquad (1)$$

δ is the geometric error, d is the distance to the view point, $\lambda = w / \phi$ 。

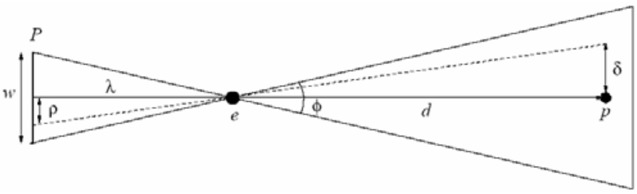


Figure 5. The screen projection error

C	Number of points (N_P)	Number of interpolation points (N_C)	N_C/ N_P	Number of triangles in original TIN(N_T)	Number of triangles in LOD mesh (N_LT)	N_LT/ N_T
10	35418	340	0.96%	195029	12816	6.6%

Table 1.The important data in an experiment

The geometric error of point to be inserted is:

$$\delta(p) = \left| H(p) - Interpolate(p) \right|$$

The geometric error of quadtree node is the greatest error of the irregular points contained in nodes:

$$\delta(node) = Max(\delta(p_i)) \quad (i = 0, 1, 2 ......)$$

The criterion of subdivision is:

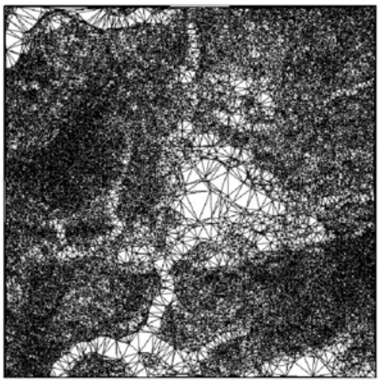
$$C < \lambda \frac{\delta}{d} \tag{2}$$

At real-time rendering phase, if the screen projection error of quadtree node meet the formula (2), the nodes need further subdivision; if the screen projection error of quadtree node can not meet the formula (2) at leaf nodes, insert the irregular points contained in the leaf nodes in order of the value of the errors until the screen projection error is met the formula (2) or there are no terrain sampling points for insertion.

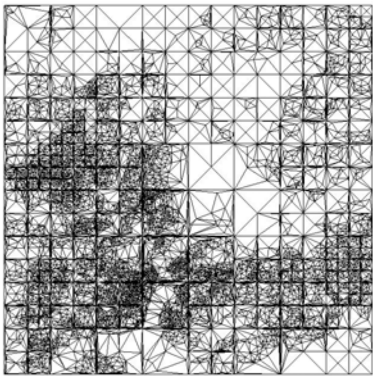
3. EXPERIMENTS

Author has realized this algorithm on PC platform using C++ and OpenGL API. Figure 6(1) shows the original TIN, and 6(2) shows the LOD triangle mesh of TIN at 600 moment (where C=10, each leaf node contains 50 points at lest), 6(3) shows the terrain expression. The following table is the important data in an experiment.

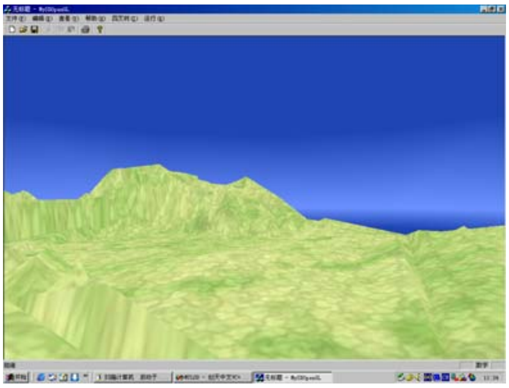
From the table we can know: The number of triangles in LOD triangle mesh of the number of triangles in the original TIN in the ratio is 6.6% under C=10, which shows that this LOD algorithm has been greatly reduced the number of triangles; the introduction of the interpolation points of the original sampling point set in the ratio is 0.96%, the interpolation points would not introduce significant error.



(1) The original TIN



(2) The LOD triangle mesh of TIN



(3) The terrain expression

Figure 6. Experiments

4. CONCLUSION

In this project we construct a quadtree spatial index structure from irregular set of elevation points at the pre-processing phase, and combine the advantage of a TIN based mesh simplification preprocess with high-performance quadtree based LOD triangulation. we achieve a LOD triangle mesh similar to hybrid structure of TIN and GRID, and realize the view-dependence lever-of-detail controlling over the TIN model.

REFERENCES

P. Lindstrom, D., 1996. Koller, W. Ribarsky, L. Hodges, N. Faust, and G. Turner. Real-time, continuous level of detail rendering of height fields. In ACM Computer Graphics (SIGGRAPH '96 Proceedings), ACM Press, pp. 109-118.

Stefan Röttger Wolfgang Heidrich, Philipp Slasallek and Hans-Peter Seidel, 1998. Real-Time Generation of Continuous Levels of Detail for Height Fields. Winter School in Computer Graphics (WSCG Proceedings) '98.

Renato Pajarola, 1998. Large Scale Terrain Visualization Using The Restricted Quadtree Triangulation . Proceedings IEEE Visualization'98, IEEE Computer Society, pp. 19-26.

Mark Duchaineau, Murray Wolinsky, David E. Sigeti, Mark C. Miller, Charles Aldrich, and Mark B. Mineev-Weinstein, 1997. Roaming terrain: Real-time optimally adapting meshes. In Proceedings Visualization 97, pp. 81–88. IEEE, Computer Society Press, Los Alamitos, California.

Renato Pajarola, 2002. Overview of Quadtree-based Terrain Triangulation and Visualization.

H. Hoppe, 1996. Progressive meshes. In Computer Graphics (SIGGRAPH '96 Proceedings), ACM Press, pp. 99-108..

H. Hoppe, 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. In Proceedings IEEE Visualization'98, IEEE Computer Society, pp. 35-42.

Renato Pajarola, Marc Antonijuan and Roberto Lario, 2002. QuadTIN: Quadtree based Triangulated Irregular Networks . In Proceedings IEEE Visualization, pp. 395–402.

Roberto Lario, Renato Pajarola and Francisco Tirado, 2003. HyperBlock-QuadTIN: Hyper-Block Quadtree based

Triangulated Irregular Networks. In Proceedings IASTED VIIP Conference.

Michael Garland and Paul S. Heckbert, 1995. Fast Polygonal Approximation of Terrains and Height Fields.

Prof. P. Widmayer, 2001. Multiresolution Terrain Triangulation using a Delaunay Hierarchy.

Yoon-Sig Kang , Taixup Lee , Sung-Bong Yang and Woo-Chan Park, 2000. A Fast Digital Terrain Simplification Algorithm with a Partitioning Method . IEEE 2000 .

Leila De Floriani and Paola Magillo, 2002. Regular and Irregular Multi-Resolution Terrain Models: a Comparison. GIS'02, November 8–9, 2002 McLean, Virginia, USA.