

# THE RESEARCH OF THREE-DIMENSIONAL SCENE CONSTRUCTION BASED ON MULTI-RESOLUTION DATA

LIN Zhiyong

School of Remote Sensing and Information Engineering  
Wuhan University  
129 Luoyu Road, Wuhan, China, 430079

LI Wenjing

College of Resources and Environmental Engineering  
Wuhan University of Science and Technology  
947 Hepin Road, Wuhan, China, 430081

## Abstract

The research on how to implement a global virtual terrain environment has become the hot point in spatial information study. Three-dimensional scene construction is the base to all further application in Virtual Reality. In order to find a solution to this problem, we concentrate on some key technologies, such as designing a multi-resolution hierarchy for global massive terrain data, building an optimized terrain model and texture model, building and saving the terrain index, requesting terrain data of interest as far as possible from database during real-time rendering and so on. In this paper, we put forward a method to use multiple resolution data to construct three-dimensional scene and build up a prototype system.

**Keywords:** Multi-Resolution Data, LOD, Scene Construction, DEM

## 0 Preface

With the development of Digital Earth and Global Information Grid, the research on how to implement a global virtual terrain environment has become the hot point in spatial information study, and till now there are still many challenging problems in the air, such as Virtual Realty, LOD and so on. Virtual Reality has been widely used in many scientific and application fields such as simulation flying, Cyber City, three-dimensional GIS, Digital Earth, etc. three-dimensional scene construction is the base of all Virtual Realty Environment. When we build Virtual Realty, the first thing is to construct a three-dimensional scene. The data sources of three-dimensional scene include DEM, texture and three-dimensional building model. Among these, DEM and texture is the base to construct three-dimensional scene<sup>[1,3,5]</sup>.

Three-dimensional scene construction is the base to all further application in Virtual Realty, but how to balance the overload between visual fidelity and real-time rendering, which is caused by massive terrain data. In order to find a solution to this problem, we concentrate on some key technologies, such as designing a multi-resolution hierarchy

for global massive terrain data, building an optimized terrain model and texture model, building and saving the terrain index, requesting terrain data of interest as far as possible from database during real-time rendering.

In recent years, the methods to obtain data have grown up and we can get the texture data whose resolution includes the range from 0.1 meter to several kilometers. The texture data has the best visualization at the origin resolution, but the majority methods we use to construct three-dimensional scene currently is to generate the data at different LOD model from one source texture data and DEM data. That cannot achieve the effect whether through high-resolution data sampling to be low-resolution data or through low-resolution data interpolation to obtain high-resolution. So when we construct three-dimensional LOD model, we had better integrate of multi-resolution data. The pyramid model adopted for each level, with the level of data resolution of the nearest data source is constructed with the best visual the effect of three-dimensional scenes of the precondition.

As a result of multi-source data across multiple resolution data, it is more complicated thus building a three-dimensional scene of the pyramid model, than using a single data source to build the pyramid complex of multi-model. When we use the single data to build LOD model, we only need to consider the data in one direction to operate; but when we use multi-time data sources, we not only need to consider the data to a the direction of the operation, but also we must consider the different relationship between the resolution data, including data accuracy, data scope, other relations between the overlay and so on.

## **1 The LOD Model based on Multi-Resolution Data**

### **1.1 The Quad-tree LOD Model Construction**

The LOD (Level of Detail, LOD) technology is the main method to accelerate 3D scene rendering effective<sup>[2,4]</sup>. The LOD model is divided into three different types including discrete LOD 、 sequence LOD and multi-resolution LOD. LOD, through pre-built with multiple levels of detail of the discrete model, or is displayed based on the view-dependent real-time dynamic parameters with different levels of detail to build the multi-resolution model, to improve the rendering efficiency. In fact all of the LOD technology is a dynamic management of 3D scene display technology. At present, digital terrain data products will be divided into two types, one for the triangulated irregular network (TIN); another grid square (Regular Square Grid, Grid). Taking into account the massive terrain data products for the GRID structure of the general performance of the digital elevation model, we put the main research on the method of building the digital elevation model for large-scale virtual terrain environment dynamic scene management based on GRID<sup>[2,7]</sup>.

Quad-tree model is a model of multi-resolution detail levels. When the system renders

the 3D scene, we need to use data of DEM and texture image data with different resolution[4,8]. We build a LOD pyramid model of quad-tree by pre-treatment of origin data. In the scene rendering, we can access the data directly in real time without having to resample. Using this method, we reduce the 3D scene rendering time by increasing the storage space when consumption and improve the implementation of input and output efficiency, so as to enhance the system overall performance.

When we build a pyramid, it is usually in the original data with the highest-resolution as the bottom of the pyramid and its sub-blocks to form a layer 0 into the grid. And then the layer 0 in the first layer on the basis of each  $2 * 2$  pixels for a pixel to generate the first layer, and its sub-block to form a layer 1 block pyramid. Because the LOD based on multi- resolution data, when we build a higher layer, we need to compare the resolution between this layer and the original data, if the original data has the resolution closer to this layer, we should use the original data as the base. Under such circumstances, the entire pyramid is constituted.

We use texture data in grid as an example to explain this process. Suppose the pixel matrix size layer  $i$  is  $R_i * C_i$ , the resolution is  $RES_i$ , the grid size is  $N * N$ , then the current size of the grid matrix is  $GR_i * GC_i$ ;

$$GR_i = \lceil R_i / N \rceil; \quad (1)$$

$$CR_i = \lceil C_i / N \rceil; \quad (2)$$

The symbol  $\lceil \rceil$  means for the next round. By  $2 * 2$  pixels for the synthetic generation of a pixel, the matrix size of the layer  $(i + 1)$  is  $R_{i+1} * C_{i+1}$ ,  $R_{i+1} = R_i / 2, C_{i+1} = C_i / 2$ ; its resolution  $RES_{i+1} = RES_i * 2$ . The Operation of synthesis of pixels is carried out from the pixel matrix of the lower-left corner, from left to right from bottom to top in order; the data block operation is started from the lower left-hand corner, from left to right followed by bottom-up.

## 1.2 LOD Building based on Multi-source data

The process of Multi-resolution terrain model is based on quad-tree as follows. The whole terrain, as root, starts from the root node to check whether the root of a partition to meet the conditions, if it does not meet it should be preserved as a leaf node without spited; otherwise, the root node should be split continuous recursive into four equal sub-node region until it cannot be split up. Finally, all the leaf nodes construct the simplified result for rendering. The data with greater depth of division, and it has the higher the resolution. When the depth of each layer of partition increases, the resolution of sample density is double. A leaf node here means a rectangular block in which nine terrain data is recorded.

In order to facilitate the view of the terrain levels of detail dependent on scheduling and the establishment of the scene, we need split scenes in memory into each square unit.

The vertex number of every square grid unit is  $n$ , and  $n = 2^k + 1$ , in which  $k$  is the level of quad tree. From bottom to top, the vertex number of each unit in quad tree is followed by multiple of decreasing the number of relations, the data grid resolution and the size of the relationship is followed multiple of growing. Because of this relationship among the grid unit in quad tree, the bottom data of quad tree has the character of the smallest and highest resolution, the largest number of vertex. The vertex number of the bottom level is  $2^{(m-1)} + 1$ , in which  $m$  is the level counts of the whole quad tree. The resolution of upper layer is twice of the current layer and the number of the current vertex is  $n = 2^m - 2 + 1$ , And so on, the number of top-level module for the vertex  $n = 2^m - 2 + 1$ .

In the Actual operation, the background database according to the pyramid has a database of the highest resolution (the resolution of the underlying data), that is the definition of the bottom scene vertex grid resolution. Because the scale and resolution is not a one-to-one relationship and the resolutions in different digital elevation model data are generally used in different coordinate reference systems, this situation must be considered during the process of the virtual scenes generation. Because of the resolution limit of quadtree, its various layers quadtree pyramid corresponding resolution and the data in the database layer are very difficult to be the same. During the real-time data scheduling, we need automatically select the resolution closest to the database layer of the pyramid by resampling the data, and provide the required data resolution according to the required resolution. In this way, for the above-mentioned resolution, the maximum span can create a resolution of 5 m, the minimum resolution of 320m, from the bottom-up resolution of 5 m, 10 m, 20 m, 40m, 80 m, 160 m, 320 m, etc. 7 quadtree level unit.

When roaming the virtual scene, we use the point of view with the quadtree as the choice of the distance unit quadtree level metrics. With the view from the nearest unit is the use of the highest resolution data, with the perspective of the unit distance is the use of low-resolution data. Quadtree element data shows that the resolution as the unit of distance and perspective to reduce the increase. This method only needs the unit near the point of view higher resolution read into the data, while others more distant from the point of view of the unit (the majority) only need to read the data to a lower resolution. This reduces the flow of network data, but a corresponding increase in levels of detail of the memory management complexity. In order to avoid adjacent quadtree as the resolution of the difference between units is too large and the "cracks", we limit the adjacent unit difference between the level of resolution cannot exceed the maximum level.

## **2 Scheduling and multi-resolution data visualization**

The purpose of multi-source data pyramid model is to fast three-dimensional visualization of the scene to provide support. During the process of 3D data visualization, firstly we need to determine the scope of visibility of the region and resolution of data based on the current position of the point of view and the direction

and angle of sight; and then, in accordance with the established requirements of the regional scope and resolution to obtain relevant data; finally, to stitch and render obtained data to achieve the purpose of visualization.

## 2.1 Establishment of visual resolution and regional

As we all know that 3D data is the massive of data, during user interaction and rendering the process of visualization, all data cannot be transferred to the system memory one-time. Therefore, it is necessary to schedule and render data dynamically in accordance with the process of human-computer interaction. As the three-dimensional scene is a stereoscopic scene, the distances of every part in the scene to the current point of view are inconsistent, there is a need in accordance with the direction and angle of the pilot to determine the current view in the distance and the relationship between scenes, and then based on the size of the distance to determine the need for by scheduling the data blocks in which the level of the pyramid.

Based on the current coordinate , the direction and angle of the point of view ,we can calculate the average terrain level body with the plane intersecting the regional level and regional scale terrain visualization. In Figure.1 ,we can see that XOY is the average level for the terrain , E is the point of view , EM is the line of sight is the intersection between the EM and XOY, , point of view in the plane XOY projection is  $M_0$  , and the points of A'、 B'、 C' and D' is the four intersection between visual body E-ABCD and XOY。 So the visibility of the terrain is region-wide quadrilateral A'B'C'D'。 After determining the visible region, we need determine the pixel resolution based on perspective in light of the current level of the average distance.

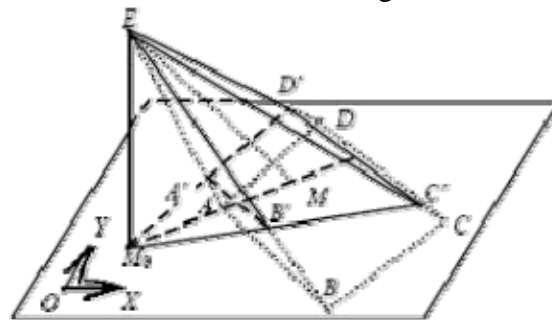


Figure.1 The region of visualization in 3D Scene

$X_w$  is the width of screen window for 3D Visualization for the region, and  $Y_w$  is the height of it. The unit of  $X_w$  and  $Y_w$  is the pixel. The horizontal and vertical fields of view in visual body are FovX and FovY. To illustrate the problem, we assume that the current visual region A'B'C'D', and the window is just filled with A'B'C'D'. the window horizontal and vertical pixels corresponding to the average distance from the ground are  $D_x$  and  $D_y$ .

$$D_x = EM * \tan ( \text{FovX}/2 ) * 2.0 / X_w \quad (3)$$

$$D_y = EM * \tan ( \text{FovY}/2 ) * 2.0 / Y_w \quad (4)$$

When the resolution in x direction is less than  $D_x$  or the resolution in y direction is less than  $D_y$ , the terrain will reduce the accuracy of the visualization. On the contrary the data will generate unnecessary redundant and at the same time the resolution in terrain rendering of  $D_x$  and  $D_y$  is the best in theory. Because the terrain rendering is usually using the same sample resolution in X direction and Y direction, we need to obtain the minimum value of  $D_x$  and  $D_y$ , and use this as the best resolution of the terrain for practical application, then determine the level of the pyramid based on the best resolution. The usually method is the use of the window center data block corresponding to the resolution as the best resolution, and the window to a different location by the corresponding lower resolution make the appropriate adjustments. Through such operations, the visualization of the region in line is fit for the eye rule of observation, and it can reduce the number of terrain triangles to improve the display speed.

## 2.2 Quick Search Algorithm of data grid

The data quick search of grid data block in pyramid model is the basic conditions for rapid rendering. If you only use just one resolution, it can only involve one suitable search result to LOD level and the level corresponding to the region. In the use of view-dependent LOD terrain visualization technology systems, we need to satisfy the requirement that in different regions of the terrain data is different resolutions, so in the process of grid data searching we need to test resolution. As follows, in Figure.2, E is the point of view, white areas means intersect with the vision and the body that need to be involved in mapping the terrain grid data. The more intensive grid is, and the higher the resolution grid is corresponding to.

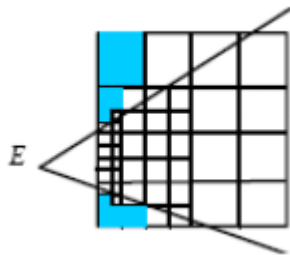


Figure.2 The graph of terrain grid resolution

In order to access data fast, we use the way of coarse-to-fine. First of all, to determine whether the visual region indicated that the terrain data, to determine the overlap with the visual region of the top-level data grid to meet the requirements of resolution, if the requirements are not satisfied, then continue to traverse the next branch of LOD, search until all the requirements to meet the grid resolution. During the intersection test to determine, as long as the grid of the four corners at any point is in the visual region and grid is intersected with the visible region. For the resolution of testing, we need to

calculate the resolutions of four corner points of grid firstly. If the resolutions of four corner points values are greater than or equal to the grid where the resolution pyramid level value, the grid meets the requirements of resolution. Or we need to continue to traverse the grid nodes corresponding to the four branches.

Grid search algorithm is a typical recursive process. During the process of determining in the resolution, we only need to judge for the grid coordinates of four corner, so is efficient. In addition, the array index from the grid to extract grid the sub-branch is needed to determine the spatial location, and only need to calculate net offset bounds, this can significantly improve the algorithm performance. In addition, the array index from the grid to extract grid the sub-branch need to determine the spatial location, and only need to calculate net offset bounds, this can significantly improve the algorithm performance.

### **2.3 pre-read technology of data grid**

Grid data pre-read technology can enhance the massive terrain data in real-time visualization of system performance through the establishment of appropriate mechanisms for data buffer. When the system is running, if the grid through data pre-processing technology to the system can be used to obtain data, and early data from the disk can be obtained and stored in the buffer zone, when the real needs of the application of these data happens, this can be read directly from the buffer, thereby eliminate the need for the data from the disk to read the time. When we carry out massive terrain data, the data predict the need for scheduling and mapping the scene at the same time. Therefore, we use multi-threading technology in which scheduling data and mapping scenarios use two separate threads to complete. Data buffering and scheduling complete through the operation of the scene using two separate threads. Forecast data buffer through the grid to achieve the request. Data request is divided into three levels, namely high, middle and low of the three priorities. High-priority request grid data is immediately processed and participate in mapping the terrain, the middle-priority request in the absence of data on high-priority data grid request to read to the buffer zone, but not to participate in mapping the terrain, data on the composition of the buffer; low-priority request grid data only to read to the buffer zone when the system is free, and not to participate in mapping the terrain.

Grid data-priority request is as follows. Assumed that the window for the three continue scenes are made up of the window 1, window 2 and the window 3. In order to achieve grid data buffer, the system sends window of a data request grid, with sending window 2 and window 3 data request, only the priority of the grid data request is different. Grid window 1 data is the high-priority request, the region in window grid 2 and out of window 1 data request is the middle-priority, window 3 grid data requests with the exception of window 1 and window 2 outside of the public part of is set to low-priority.

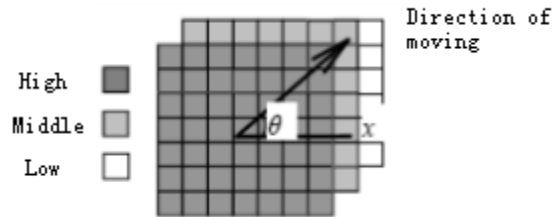


Figure.3 Priority of Data

In Figure.3, when the scene moves to window 2, the system sends window 2 grid data request, data prediction thread scheduling can be extracted directly from the buffer zone to participate in the terrain mapping data without real-time data from the disk to read data.

### 3 The experimental results and Conclusion

The prototype system uses an area of three-dimensional visualization of the terrain as an example. The data of region is only Grid DEM; DEM data has 45747 \* 46894 elevation points, the resolution of which is 30m. The coverage texture data in the region is two kinds of resolutions. One is 30m color resolution, and the other is 10m color resolution, so we build the six-layer pyramid and the associated index file based on the DEM and texture, the grid size of unit is 64 \* 64. The configuration of hardware and software in prototype is that the computer CPU is Intel P4 2.8 GHZ, RAM is 512 M, Hard Drive is 120 SATA, graphics is NVIDIA GeForce 7300 GS, the operating system is Windows XP professional, and three-dimensional rendering driver for OpenGL. The prototype system has the function of the real three-dimensional texture mapping and visualization, and achieving real-time roaming the terrain, zooming options, and interoperation. The display window size for the current is 1024 \* 768, and real-time rendering frame rate is 27.2 frames / sec. The rendering result is show in figure.4 as fellows.



Figure.4 The result of Prototype

The experiment result shows that multi-source data to build the pyramid model can integrate multi-source data to meet the massive terrain data in real-time rendering visualization and scheduling needs, can support the massive terrain in real-time rendering of geometric data and real-time data mapping of massive texture. Based on



the clipping terrain data outside the region and the resolution of the grid search algorithm, it can improve the system for data scheduling and mapping efficiency through the reduction of the terrain mapping data. On the way of the establishment of the data pyramid grid scheduling priority, data pre-processing, it can further enhance the speed of data acquisition. In this paper, results of research on massive data can be used in the expansion of the three-dimensional interactive visualization system roaming. This study grid data and rule is based multi-resolution texture data, not related to irregular grids of data, we will continue to do further study in this field.

### **Acknowledgment**

This work is supported by the National Science Foundation of China under Grant No.40701158 .

### **Reference**

1. Foster I , Kesselman C , Tuecke S. The Anatomy of t he Grid :Enabling Scalable Virtual Organizations [ J ] . International Journal Supercomputer Applications , 2001 ,15 (3) :200 ~ 222
2. LIU Hua, WANG Yandong, GONGJ ianya. KEY TECHNOLOGIES OF SPATIAL DATA ENGINER BASED ON MULTI2DATA SOURCE[J]. Journal of Geomatics ,Apr. 2005 ; 30 (2)
3. Ribelles J, Lopez A, Belmonte O, et al. Multiresolution modeling of arbitrary polygonal surfaces: a characterization. Compute Graph,2002, 26: 449—462
4. De Floriani L. A pyramidal data structure for triangle-based surface description. IEEE Compute Graph Appl, 1989, 8(2): 67—78
5. Yang B S, Shi W Z, Li Q Q. A dynamic method for generating multi-resolution TIN models. Photogramm Eng Remote Sens,2005, 71(8): 917—927
6. Xia J, El-Sana J, Varshney A. Adaptive real-time level-of-detailbased rendering for polygonal models. IEEE Trans Vis Comput Graph, 1997, 3(2): 171—183
7. WANG Yong2jun ,GONGJ ianya. An Approach to the Management of Dynamic Scene in Large Scope Virtual Terrain Environment. Journal of Institute of Surveying and Mapping, 20(3) Sept . 2003
8. Renato Pajarola. Large scale terrain visualization using the restricted quadtree triangulation [ A ] . Proceedings Visualization 98[C] . IEEE Computer Society Press , 1998