

# Construction and Scheduling of Multi-resolution LOD Terrain in Super Low-Altitude Flight

Peng Zou<sup>1</sup>, Yanshan Bian<sup>2\*</sup>, Zhike Chen<sup>2</sup>, Ronghuan Yu<sup>1</sup>

<sup>1</sup> Science and Technology on Complex Electronic System Simulation Laboratory, Equipment Academy, Beijing 101416, China.

<sup>2</sup> Department of Graduate Management, Equipment Academy, Beijing 101416, China.

\* Corresponding author. Tel.: +8618606511598; email: bys02201@163.com

Manuscript submitted December 18, 2014; accepted April 27, 2015.

doi: 10.17706/ijcee.2015.7.3.159-167

---

**Abstract:** In visual navigation system, a method of construction and scheduling of multi-resolution LOD terrain is designed according to the influence of terrain precision on super low flight. First, the tiled pyramid is constituted to manage the terrain data, which is organized by quadtree. The terrain complexity-based node selection criterion is used to select the required blocks for rendering. Then, the structural similarity of quadtree nodes is made use of to symmetrically cluster two-level nodes, which is suitable for batch scheduling of terrain data. Experimental results show that this method improves the terrain precise from a certain range of the aircraft, reduces the frequency of data scheduling, and enhances the rendering frame rates of navigation scene.

**Key words:** Super low-altitude flight, visual navigation, level of detail (LOD), batch scheduling.

---

## 1. Introduction

In three-dimensional terrain scene of the visual navigation system, such as synthesis vision system (SVS) [1], combined vision system (CVS) [2] and so on, terrain data has important influence on superlow-altitude flight which has a flight altitude below 100 meters above the ground. The influence lies not only on the precise of terrain data, but also on the altitude of ground objects based on the terrain data. However, with the quick improvement of terrain precise, the data volume of terrain data by the form of digital elevation model (DEM) is getting bigger and bigger, which leads to the incapacity of dealing and rendering in real time because of the bottlenecks of graphic display hardware system and memory/video memory's transmission bandwidth. To generate the multi-resolution DEM with LOD is an effective method to boost the rendering efficiency. So, efficiently rendering huge planet-sized terrains of extremely high-resolution is important just like many other applications.

The key of constituting a multi-resolution model is to constitute an approximate model of the original DEM which makes the difference as small as possible. DEM data is managed and stored in external storage and transferred to memory when needed. Many methods are researched to manage DEM data efficiently, such as view-dependent LOD [3], incremental data scheduling [4], layer and block partition model [5], multiply technologies based mixture methods [6], and so on. Among those methods, the layer and block partition model is high-efficiency and can realize the displaying of multi-resolution terrains, which has been widely researched and employed.

By making use of quad tree structure, Bernhardt *et al.* [7] proposed a perfect solution to constitute

real-time and multi-resolution terrains dynamically, and to resolve the discontinuity of grids. Li *et al.* [8] proposed an algorithm of modeling and solving of multi-resolution mesh of visual sphere, which can generate a sphere mesh under an arbitrary viewpoint and view direction and can expand multi-resolution structure to spherical surface. Those methods had advantages in simplification efficiency and storage structure.

In the scheduling of terrain data, Pajarola [9] used the out-of-core method to divide the original terrain into small tiles which were then ranked by code rule and scheduled according to memory requests. Dai *et al.* [10] proposed a tiled pyramid structure and target tile search algorithm based on Pajarola's method, which can realize the dynamic updating of local terrain.

Two important batch-based terrain rendering algorithms have been researched. Batched dynamic adaptive meshes (BDAM) [11] and HyperBlock-QuadTIN [12] show that semi-continuous irregular triangular mesh optimization can still be done efficiently, avoiding bottlenecks encountered by CPU-based techniques such as ROAM [13]. For instance, BDAM uses a top-down refinement which is based on bintree. It clusters triangles into a hierarchy of regions. It performs very well on modern hardware. However, those methods need a lot of preprocessing on the terrain dataset and cannot be generated real-time.

Chunked LOD [14] is a simple technique which is based on a quadtree structure of square terrain "chunks". Each chunk consists of a single batch. It is very easy to implement. Geometry Clipmaps [15] renders terrains using nested regular grids surrounding the viewpoint. Unfortunately, it does not tolerate a dynamic field of view, which is required by the superlow-altitude flight.

Under the condition of superlow altitude flight, the multi-resolution terrain should be built according to terrain complexity in order to enhance the precise of visual navigation system. Also, the rendering frequency of navigation scene is higher under that condition and the real-time scheduling of terrain data should be improved. According to the analysis above, the construction and scheduling of multi-resolution LOD terrain in super low-altitude flight is designed in this paper.

## 2. Influence Factors of Super Low-Altitude Flight Safety

It is dangerous for a superlow-altitude flight because of the complex terrains and low-altitude ground objects. For instance, the theoretical response time of human brain is only 0.6~0.7 s, but the real time can be expanded to 2 s according to the flight inertia. So for a helicopter flying with the speed of 80 m/s, the maximum flight distance within this period might exceed 160 m. Because of the quick variation of aircraft posture such as sudden low-altitude penetration and big turn, the low-altitude ground objects and rugged topography are potentially dangerous in this distance, as is shown in Fig. 1, which can be worse under the condition of low visibility.

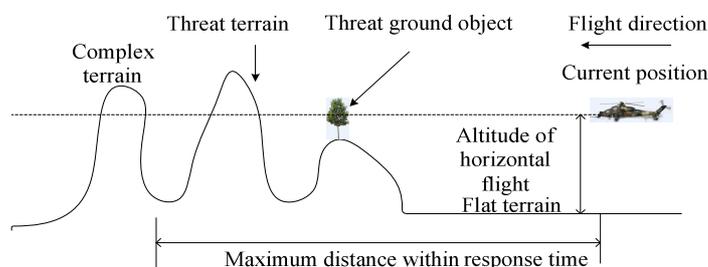


Fig. 1. Threat terrains and ground objects.

A pilot need sufficient time and space to shy away from threat terrains and ground objects, i.e., the terrain precise loss should be less than a threshold ( $\gamma$ ) within a distance ( $R_s$ ) from the aircraft when constituting the multi-resolution terrain models, if not so then pilot might cause mistakes on terrain and ground objects

which would lead to the inability of adjusting the routes and even the air crash at last.

So it is critical to set a maximum threat circle area to ensure the flight safety. The radius of the circle is  $R_s = v_h \cdot t$ , where  $v_h$  is the current horizontally flight speed and  $t$  represents for the response time. In this area, the simplification error of terrain should be less than  $\gamma$ . This area should be enlarged if the ground objects are taken into consideration on the other side. The topographic relief of a local terrain block can be measured by terrain complexity and terrain can be simplified by terrain complexity and loss threshold of terrain precise  $\gamma$ .

### 3. Constitution of Multi-resolution LOD Terrain

#### 3.1. Tiled Pyramid

Because of the large amount of terrain data, high precise of LOD models are only needed in those areas that are closed to an aircraft or are highly complex. Those areas are potentially dangerous to flight safety. To reduce the data amount and ensure the safety of superlow flight, different resolution models are need to represent different terrains according to terrain complexity and view distance, i.e., multi-resolution LOD model [16]. Then terrain data is managed by tiled pyramid, and organized by quadtree. The key of tiled pyramid is to constitute multi-resolution LOD terrain data using a layered and blocked strategy. The resolution of DEM is lowest at the top and highest at the bottom of pyramid. Then the proper quadtree nodes representing terrain blocks are selected to rendering according to node selection criteria.

#### 3.2. Terrain Complexity Based Quadtree Node Selection Criterion

Different resolution meshes of DEM are need to represent complex terrains in the frustum of rendering, which can be archived by selecting proper quadtree nodes. Small set of nodes means low resolution meshes and flat terrain while large set of nodes means high resolution and undulate terrain.

There are two criteria to decide the selection of nodes which are: 1) terrain complexity criterion reflecting the complexity of terrain block; 2) viewpoint criterion describing the relationship of viewpoint and terrain block. The former can be measured by the maximum error of the nodes that representing an area, i.e., the maximum error of elevation between multi-resolution models and the original terrain. The latter can be decided by both the projection ratio of triangle on orthographic projection area of sight and view distance. The former's influence reflects in complex mountainous regions for flight safety while the latter's in the terrain and ground objects near the aircraft. Fine quadtree nodes are selected for rugged regions and coarse ones are for flat areas. Complex terrains are dangerous for the super low-altitude flight.

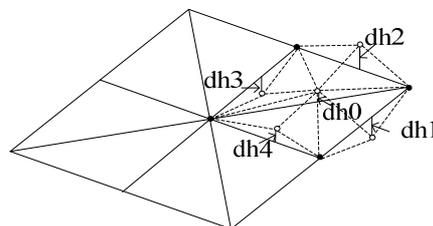


Fig. 2. Calculation of terrain complexity.

A terrain complexity criterion measures the influence of uneven terrain on node selection. The most common measurement is the vertical altitude difference between the simplification mesh and the original DEM. As is shown in Fig. 2, there are 9 vertexes in the mesh of the top-right corner, Errors are caused by refined or unrefined points of the central point and four boundary points of that mesh, which are  $dh_0 \sim dh_4$  and decide the complexity of that node. Besides, errors of the four child nodes of that node are taken into calculation, i.e.  $dh_5 \sim dh_8$ . So the preliminary criteria of terrain complexity is measured by  $r = \max(dh_0,$

dh1,..., dh8). Taking loss threshold of terrain precise  $\gamma$  (the terrain block's maximum vertex error) into consideration,  $r$  should meet the following equation:

$$\text{if } l \leq R_s, \text{ then } r \leq \gamma \tag{1}$$

where  $l$  represents the view distance from the central point to viewpoint and  $R_s$  represents the maximum radius of threat area. Out of the range of  $R_s$ , terrain errors can be augmented by adjustment factor  $\xi$ , i.e.  $r\xi$  ( $\xi \geq 1.0$ ).

The evaluation formula of terrain complexity of the current node is calculated by:

$$d/r < C \tag{2}$$

where  $d$  is the side length of this mesh, and  $C$  is the adjustment factor of complexity. If (2) is met for a given  $C$ , it means this mesh should be divided sequentially. The introduce of  $d$  is to describe the following situation: if an object is closer to viewpoint or it is bigger, the projection on the screen will be larger and more details are need to represent it.

A viewpoint criterion includes view distance and view sight. According to human visual system, the shorter the view distance is, the more details we can perceive. So, the node is evaluated by:

$$l/d < C' \tag{3}$$

where  $l$  is the view distance of this mesh, and  $C'$  is the adjustment factor of view distance ( $C' > 2$ ). The larger  $C'$  is, the more details this mesh represent.

Both view distance and terrain complexity are taken into consideration, and then a quadtree node selection criterion is decided by:

$$f = l / (d \times C' \times \max(C \times r, 1)) \tag{4}$$

If  $f < 1$ , it means the mesh needs a further subdivision.

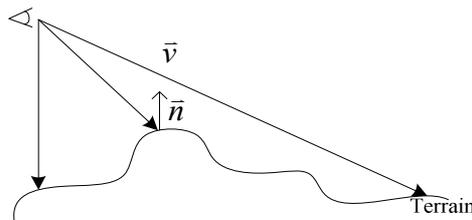


Fig. 3. The relationship between view and terrain.

The resolution of terrain mesh is related to view distance, but also the angle between view  $\bar{v}$  and terrain block, as is shown in Fig. 3. In the progress of superlow-altitude flight, lots of details are needed in those blocks that are set in the area of orthographic projection. This means high resolution nodes are used in those blocks. On the other side, low resolution nodes are needed in the other areas oppositely. The relationship  $\nu$  between view and terrain block is measured by the area-to-projection area ratio, i.e., dot product of view  $\bar{v}$  and normal vector  $\bar{n}$  of this block:  $\nu = |\bar{v} \cdot \bar{n}|$ . The influence of this factor on terrain resolution is given by:  $\tilde{C} = f(\nu)C$ , where  $C$  and  $\tilde{C}$  represent the adjustment factor of complexity before and after amendment respectively. And amendment function  $f(\nu)$  is:

$$f(\nu) = 1 / (1 + \lambda_\nu) = 1 / (1 + |\bar{v} \cdot \bar{n}|) \tag{5}$$

where  $\lambda_v$  is the adjustment factor of display effect and is set to be 1.0 acquiescently. And it controls the magnitude of changes of resolution. If the node is fully set in the area of orthographic projection, then  $v = 1$  and  $f(v) = 1/(1+|v|)$ , and then  $\tilde{C} = C/2$  where  $\lambda_v = 1$ . This means that resolution should be doubled and the mesh needs to be refined continually.

Considering all the factors above, the final quadtree node selection criterion is calculated by:

$$f = 1 / (d \times C' \times \max(C \times r / (1 + \lambda_v |\bar{v} \times \bar{n}|), 1)) \quad (6)$$

If  $f < 1$ , it means the mesh needs a further, otherwise it stops subdivision.

So terrain nodes are selected according to (6). Then seamless treatment is processed for multi-resolution terrains. And then images are mapped onto terrain meshes to render the multi-resolution LOD terrain.

#### 4. Batch Scheduling Based on Symmetrically Cluster

According to the node selection criterion, proper quadtree nodes are selected to represent the visible terrain blocks. For example, if a block contains 185 quadtree nodes, 185 scheduling requests are needed to transfer those nodes using the traditional algorithms, which occupy much communicating time. With the rocket development of computer hardware, time spend on 185 scheduling has exceeded time of rendering. So, the traditional ways based on single node are not consistent with the hardware developments. Therefore, a scheduling method based on symmetrically cluster of quadtree nodes is needed to improve the scheduling efficiency of terrain data.

John Judnich *et al.* [17] proposed a LOD algorithm based on symmetrically cluster (SCSLOD) to achieve the batch scheduling of terrain nodes. Each parent node is subdivided into four quadrants for a quadtree structure, and each of the four quadrants ends up being either a leaf node or not. So there are 16 possible clusters of terrain blocks resulting from one subdivision of one node. And this is the sigle-level node clusters. Those 16 clusters can be fit together to produce any nontrivial terrain quadtree, and the computation is very simple and trivial. However, not all these 16 states need represented by their own cluster batch if we observe rotational symmetry. In all, six seeds are required to produce all sixteen clusters by applying a 90 degree rotation to those seeds. Rendering is performed the same as before, but with the appropriate rotation applied to each cluster instance which is computationally negligible for the GPU. Single-level node clusters can be applied to two-level node clusters using the same way. 23 cluster batches are resulted from this. It reduces the number of classification of two-level subdivisions from 16384 types to only 23 types. When rendering clusters of different resolution, seams will result from the differing contour of each block's edges. Then vertical wall "Skirt" is used to fill the seams.

#### 5. Experimental Results and Analysis

The following performance analysis measurements were performed using an NVIDIA GeForce GT 630 and Intel Core (TM) 2 Duo CPU 2.93GHz processor. Visual C++ 2005 and OpenGL are used for software environment. The bus bandwidth of CUP/GPU was 4 GB/s. All test were rendered at 1024×768 resolution. Three-dimension digital earth was used to test the scene of a superlow-altitude flight. The precision of terrain data was 10 m, and the resolution of images was 1 m as a reference. In order to verify the efficiency of batch scheduling of multi-resolution LOD terrains in superlow-altitude flight, a terrain dataset of 2405×1852 points was displayed without texture mapping. Our method was employed to render this terrain block through batch scheduling.

The initially regular meshes generated by tiled pyramid and organized by quadtree are shown in Fig. 4(a), and the multi-resolution LOD terrain meshes generated by our method are shown in Fig. 4(b) for

comparison. As can be seen from Fig. 4, the complex terrains are described with more meshes, which mean the precise of local terrain being improved.

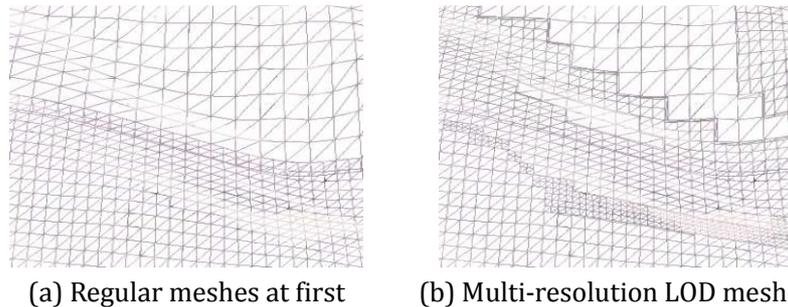


Fig. 4. Regular meshes and multi-resolution LOD meshes generated by our method.

Because the data cache of CPU was 32 KB, in order to test the influence of different scales of blocks on data batch scheduling, we set the block grid sizes to be 8×8, 16×16, 32×32, 64×64 and 128×128, respectively. And the other two methods were introduced into our experiments for comparison. The first method was a single quadtree node based data scheduling method (CDLOD) [18], where each  $n \times n$  grid block was rendered with exactly one draw call. The second implemented a batch scheduling method (Chunked LOD) [14], where the whole multi-resolution block preprocessed was rendered with one draw call. Each chunk was exactly a static cluster of triangles with different resolution. It rendered the scene using a quadtree of fixed-resolution mesh blocks.

Table 1 showed the rendering frame rates with different block grid sizes of those three methods. The maximum horizontal flight speed was set  $v_h = 80$  m/s, and screen-space pixel error tolerance was set to be 1.0 pixel.

Table 1. The Rendering Frame Rates of Different Tile Scales (Frames/Second, fps)

Methods	Block grid size				
	8×8	16×16	32×32	64×64	128×128
CDLOD	11	15	32	37	31
Chunked LOD	18	21	26	33	35
Our method	23	28	48	31	20

Our method uses two-level cluster nodes for symmetrically cluster, so the number of cluster nodes is  $22 \times 22 = 16$  times as large as that of CDLOD. When the grid size is 64, the amount of node clusters has exceeded data cache of CPU. The maximum amount of CPU cache is not sufficiently used, which leads to the decrease of rendering frame rates caused by an unsynchronized data transfer between CPU and GPU. Chunked LOD has more and more redundancies as the grid size gets smaller and smaller. As is shown in Table 1, the rendering frame rates of our method is improved by approximately 30% compared with CDLOD, and approximately 37% compared with Chunked LOD at the peak. Because our method, CDLOD and Chunked LOD obtain the maximum frame rates at the grid size of 32×32, 64×64, and 128×128, respectively, these three sizes are then used for comparison in the following tests. Accordingly, those three methods are marked as our method-32, CDLOD-64, and Chunked LOD-128, respectively.

The value of screen-space pixel error tolerance has an effect on the loss threshold of terrain precise  $\gamma$ , and then the quadtree node selection criterion  $f$ , which can be calculated by:  $\rho = w / (2l \tan(\theta/2))$ , where  $w$  is the width of screen visible area (in pixel),  $\theta$  is the horizontal field of view, and  $l$  is the view distance from the viewpoint to the central point of the current terrain block. The averaged rendering frame rates of different methods are measured through setting different values of  $\rho$  ( $1.0 \leq \rho \leq 10.0$ ). At the same

time, in order to ensure the safety of superlow-altitude flight, the related settings about potential threat area  $R_s$  are shown in Table 2.

**Table 2. Settings of Potential Threat Area**

Parameters	Properties	Values
$\theta$	the horizontal field of view	90°
$w$	the width of screen visible area	800 pixels
$\kappa$	the response time	2 s
$\gamma$	the loss threshold of terrain precise	10 m
$v_h$	the maximum horizontal flight speed	100 m/s
$R_s$	the radius of potential threat area	200 m

So the relationship between potential threat area, the screen-space error tolerance value, and the maximum tolerance view distance is given by:  $\rho = 2000 / l$ . To ensure the flight safety, i.e.,  $l \geq R_s$ , the screen-space error tolerance values should range from 1.0 to 10.0 pixels. When  $\rho = 1.0$ , the maximum tolerance view distance  $l = 2000 \geq R_s$ , therefore, the accuracies of those terrain blocks outside of  $R_s$  should be reduced according to our method. So the rendering data can be reduced.

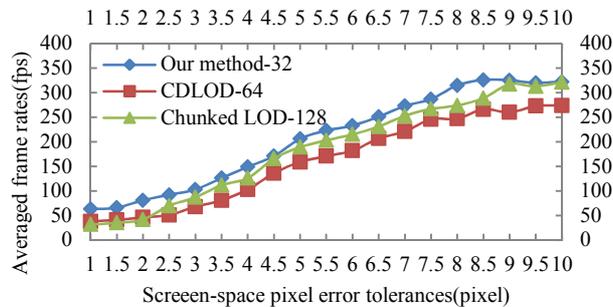


Fig. 5. Contrast of averaged rendering frame rates of terrains.

The rendering frame rates are shown in Fig. 5 with three methods. We can see that our method-32 has a 28 percent higher of the averaged rendering frame rate than CDLOD-64, and 11 percent higher than Chunked LOD-128. The growth trend of frame rates is decreasing as  $\rho$  is increasing. This is because that with the increasing of  $\rho$ ,  $l$  is decreasing, and then  $l$  is coming closer and closer to  $R_s$ , which leads to the gradually narrowing of the range of terrain precise reduced by our method. With the increasing of  $\rho$ , the frame rates of those methods are all improving. The growth rates of our method and Chunked LOD are quicker than that of CDLOD, which reflects the advantages of batch scheduling of terrain data. However, Chunked LOD has a significant amount of redundant data which leads to the low frame rates with small  $\rho$ . Even the frame rates of Chunked LOD are lower than that of our method and CDLOD for the aforementioned reason.

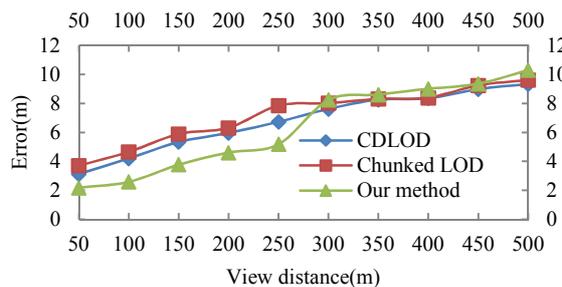


Fig. 6. Errors of multi-resolution terrains.

For comparison of the loss of terrain precise of multi-resolution terrain of different methods, the precise loss statistics are respectively performed outside and inside of the potential threat area, starting with the safety of superlow-altitude flight, as is shown in Fig. 6. In *Rs*, the errors of our method are smallest. With the increasing of view distance, terrain precise is getting lower and the loss of terrain precise is growing, which makes the error of our method is larger than both Chunked LOD and CDLOD. However, this is in line with the request of terrain precise in superlow-altitude flight. Our method retained the flexibility of Chunked LOD while improving the geometric batch efficiency, enabling performance more competitive to specialized techniques such as BDAM and Geometry Clipmaps.

## 6. Conclusion

A construction and scheduling of multi-resolution LOD terrain in superlow-altitude flight is designed. Based on the multi-resolution LOD thought, tiled pyramid is used to manage the terrain data, quadtree structure is taken use to organize the tiles, and then batch scheduling based on symmetric cluster of quadtree nodes is employed to transfer terrain blocks with different resolutions instead of the single quadtree node based scheduling. By doing this, the requests of data transfers are reduced greatly, and the rendering efficiency is improved, which provides SVS, CVS and other visual navigation platform with real-time scene.

## Acknowledgment

This work was supported in part by the National HGJ Major Project of China under Grant 2013ZX01045-004-002.

## References

- [1] McKenna, E. (2012). Synthetic vision systems. from [http://www.aviationtoday.com/av/commercial/Synthetic -Vision- Systems\\_76212.html](http://www.aviationtoday.com/av/commercial/Synthetic-Vision-Systems_76212.html)
- [2] Rahman, A. (2012). Combined vision system (CVS) helps pilots avoid crash landings. from <http://thetechjournal.com/tech-news/combined-vision-system-cvs-helps-pilots-avoid-crash-landings.xhtml>
- [3] Wang, Y., Liu, J. Y., Jiang, N., *et al.* (2003). A dynamic triangulation algorithm for the view-dependent and real-time LoD model of terrain. *Acta Geodaetica et Cartographica Sinica*, 32(1), 47-52.
- [4] Li, S., Ji, J. F., Liu, X. H., *et al.* (2006). High performance navigation of very large-scale terrain environment. *Journal of Software*, 17(3), 535-545.
- [5] Shi, S. X., Ye, X. Z., Zhang, S. Y., *et al.* (2007). Partition based real-time rendering method for large-area terrain data. *Journal of Zhejiang University (Engineering Science)*, 41(12), 2002-2006.
- [6] Lauritsen, T., & Nielsen, S. L. (2011). Rendering very large, very detailed terrains. Form <http://www.terrain.dk>
- [7] Bernhart, A., Maximo, A., *et al.* (2011). Real-time terrain modeling using CPU-GPU coupled computation. *Proceedings of the 2011 24th SIBGRAPI Conference on Graphics, Patterins and Images* (pp. 64-71).
- [8] Li, S. J., Wu, L. D., Weng, S. S., *et al.* (2008). Modeling and solving of multi-Resolution mesh of visual sphere. *Mathematics in Practice and Theory*, 38(4), 120-125.
- [9] Pajarola, R. (1998). Large scale terrain visualization using the restricted quadtree triangulation, *Proceedings of IEEE Visualization 98* (pp. 19-26). North Carolina: IEEE Computer Society Press.
- [10] Dai, C. G., Zhang, Y. S., & Deng, X. Q. (2005). An organization and management approach of data for real-time visualization of massive terrain dataset. *Journal of System Simulation*, 17(2), 406-409.
- [11] Cignoni, P., Ganovelli, F., Gobbetti, E., *et al.* (2003). Bdam batched dynamic adaptive meshes for high

- performance terrain visualization. *Computer Graphics Forum*, 22(3), 505-514.
- [12] Lario, R., & De, D. A. (2003). Hyperblock-quadtin: Hyper-block quadtree based triangulated irregular networks. *Proceedings of IASTED Visualization, Imaging and Image Processing* (pp. 733-738).
- [13] Duchaineau, M., Wolinsky, M., Siget, D. E., et al. (1997). Roaming terrain: real-time optimally adapting meshes. *Proceedings of the 8th Conference on Visualization'97* (pp. 81-88). Los Alamitos, CA, USA.
- [14] Ulrich, T. (2002). Rendering massive terrains using chunked level of detail control. *Proceedings of SIGGRAPH 2002* (pp. 1-14). San Antonio: ACM Press.
- [15] Losasso, F., & Hoppe, H. (2004). Geometry clipmaps: Terrain rendering using nested regular grids. *ACM Trans. Graph.*, 23, 769-776.
- [16] Zhang, S. Y., & Deng, D. Y. (2009). 3D reconstruction based on seamless contiguity of TIN. *Geometrics and Information Science of Wuhan Univers.*, 34(1), 15-18.
- [17] Judnich, J., & Ling, N. (2012). Symmetric cluster set level of detail for real-time terrain rendering. *Proceedings of IEEE International Conference on Multimedia and Expo* (pp. 320-324). Melbourne.
- [18] Strugar, F. (2009). Continuous distance-dependent level of detail for rendering heightmaps. *Journal of Graphics, GPU, and Game Tools*, 14(4), 54-74.



**Peng Zou** was born in 1957. He is a professor at Key Lab. of the Academy of Equipment. His research interests are mainly in information systems theory and design.



**Yanshan Bian** was born in 1986. He is a student at Key Lab. of the Academy of Equipment, Beijing, China. He is now working towards the Ph.D. degree in virtual reality technology and virtual simulation in the Academy of Equipment, Beijing.



**Zhike Chen** was born in 1986. He is a student at Key Lab. of the Academy of Equipment, Beijing, China. He is now working towards the Ph.D. degree in information systems theory and design in the Academy of Equipment, Beijing.



**Ronghuan Yu** was born in 1983. He is the assistant researcher at Key Lab. of the Academy of Equipment. His research interests are mainly in virtual reality technology and virtual simulation.