行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　※
※　　**虛擬實境中多層次精細度模型技術之研究**(I)　　※
※　　Level-of-Detail Modeling Technique for Virtual Reality（I）　　※
※　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：☑**個別型計畫**　　□**整合型計畫**
計畫編號：NSC 89-2218-E-009-012
執行期間：89 **年**　8 **月** 1 **日至**　90 **年**　7 **月** 31 **日**

計畫主持人：莊榮宏
共同主持人：

本成果報告包括以下應繳交之附件：
□**赴國外出差或研習心得報告一份**
□**赴大陸地區出差或研習心得報告一份**
□**出席國際學術會議心得報告及發表之論文各一份**
□**國際合作研究計畫國外研究報告書一份**

執行單位：國立交通大學資訊工程系

中　華　民　國　90 **年** 11 **月** 22 **日**

行政院國家科學委員會專題研究計畫成果報告

Level-of-Detail Modeling Technique for Virtual Reality（I）

計畫編號：NSC 89-2218-E-009-012
執行期限：89年8月至90年7月
主持人：莊榮宏　　國立交通大學資訊工程系

中文摘要

本計畫針對即時且平順地顯像大規模的幾何場景問題，利用多層次精細度模型技巧分別針對一般幾何物體及地形資料，發展出觀視點不相依與觀視點相依之多層次精細度模型技術。對於一般物體模型，提出與觀視點無關、具有幾何與拓樸化簡能力、並且能保持幾何特徵與顏色邊界之模型化簡技術。而對於大規模的地形資料，動態的產生觀視點相依多層次精細度模型，在不影響顯像品質情況下儘量的化簡地形網格之三角形數目。此外，我們設計了溝渠式的區間分割，有效地解決在區間邊界上，因多層次精細度選取的不一致而產生幾何不連續的現象。此外，也提出保留可延續使用的多層次精細度模型之方法，有效地利用時間與空間的共效性，獲得高度的顯像加速，使得地形顯像更為即時與逼真。

關鍵詞：虛擬實境，多層次精細度模型，地形顯像，選擇性化簡

Abstract

For problems of constant and high frame rate rendering of complex virtual scenes, this project proposes two level-of-detail (LOD) modeling techniques: view-independent LOD modeling for geometric models and view-dependent LOD modeling for terrain data. We present a mesh reduction scheme that combines geometry and topology simplification, and produces a progressive mesh which generally collapses more than three vertices between two adjacent levels. The method also preserves the color discontinuity, which is perceptually important; but has been less studied in the literature. For large terrain data, we propose a geometrically continuous view-dependent level-of-detail (LOD) modeling aiming to speed up the generation of terrain mesh. The terrain data is subdivided into blocks and each of which will process its own LOD mesh that is dynamically determined according to the viewing parameters. Between each pair of two adjacent blocks, a dike structure is proposed that aims to provide a smooth blending between two meshes of different levels of detail. We also propose a mechanism that caches the LOD of a block for the possible reuse in the following frames after it is generated.

Keywords: Virtual reality, Level-of-detail modeling, terrain rendering, selective refinement

1、Introduction

In virtual reality applications, maintaining a fast and constant frame rate is crucial for achieving a smooth and realistic visual perception. This is a challenge for recent hardware and algorithm of graphics.

Levels of detail (LOD) is a common heuristic technique to fast render complex environment [5, 12, 7, 14]. A detailed triangle mesh is used when the object is close to the viewer, and coarser approximations are substituted as the object recedes. The recently introduced progressive mesh (PM) representation captures a continuous sequence of meshes optimized for view-independent LOD control, allows fast traversal of the sequence at runtime and avoids popping effect.

Current progressive meshing algorithms, however, tend to collapse only edges or triangles, and hence possess a very long sequence of meshes. Another issue that is important but has been addressed less is the preserving of material property, especially the discontinuity of material attributes such as color.

In addition, a terrain rendering system in flight simulation, tank simulation, or other GIS applications usually cover a very large area, and hence require a large-sized terrain grid, which is often divided into blocks. The triangulation of terrain grid usually results in too many polygons to be efficiently rendered by the current hardware, LOD modeling has been proven to be a very effective technique for reducing the number of polygons sent to the graphics pipeline.

For solving these problems, this project proposes two LOD approaches; view-independent LOD modeling for simplifying general geometric models and view-dependent LOD modeling for large terrain data. For geometric models, we are primarily concerned with how to achieve several desired features that come from several mesh simplification methods all together in a single reduction scheme. The method proposed employs both geometry and topology simplification, and in the meantime, preserves the color discontinuity. For the terrain data, techniques are proposed for removing cracks that occur between two adjacent blocks of different LOD, and for caching LOD of a block for possible reuse in the following frames.

2、Related Works

Part A: View-independent LOD modeling for geometric objects.

Many mesh simplification algorithms have been proposed. Most algorithms work by applying local geometry based criteria for simplifying small regions on the meshes [4, 6, 14]. The criteria are iteratively applied until the criteria are no longer satisfied or a user-specified reduction rate is achieved. Some algorithms optimize the geometric simplification globally over the whole mesh [1, 5, 15]. Most of these methods have satisfactory capability on preserving shape; but might fail to achieve very low reduction rate mainly due to their constraint on topology preserving. Some local geometry based algorithms have been extended to modify topology while doing geometry simplification [11, 13]; the topology simplification is, however, limited. The method that simplifies both geometry and topology is the vertex clustering algorithm [12]. The method is extremely general, as it works on any type of meshes, it can achieve very low reduction rate, and it can eliminate small geometric features and change the topology of a model. It, unfortunately, does poorly on shape preservation.

Part B: View-dependent LOD modeling for terrain.

LOD modeling techniques for terrain grid can be classified into two major mesh structures: regular square grid (RSG) [2, 8, 9] and triangulated irregular network (TIN) [3, 7, 10].

We choose RSG approach because RSG approach has several advantages. For examples, Delaunay triangulation can be easily maintained for view-dependent selective refinement, switching between levels can be efficient and simple, and fast triangle strip can be easily constructed. It, however, produces for each block a mesh that is usually not optimal, and has cracks between two adjacent blocks of different LOD resolutions.

4、Our Approaches

Part A: View-independent LOD modeling for geometric objects.

Most traditional methods for generating progressive mesh based on geometric simplifications, such as edge or triangle collapsing and vertex decimation, and some local topology modification as well. Our algorithm aims to produce an effective progressive mesh by (a) allowing more than three vertices to be collapsed or clustered at each level, (b) employing geometric simplification as well as topology simplification that involves local and global topology modification, and (c) using effective criteria to preserve geometric shape, especially sharp feature, and color discontinuity.

Algorithm begins with a preprocessing, in which each vertex is classified into five categories and evaluated to yield a weight and a priority value, then the bounding box of the given mesh is uniformly subdivided into cells of size τ. The algorithm then enters a simplification loop, in which each cycle yields a simplified mesh. In each cycle of mesh simplification loop, doing the following:

1. Select a vertex with the highest priority value to be the representative for the next clustering operation.
2. Create a floating cell of size τ that is centered on the representative to confirm the spatial range of vertex clustering.
3. Start at the representative and generate the spanning tree for all vertices that are inside the floating cell and can be clustered to the representative.
4. Cluster all vertices in the spanning tree to the representative. Delete the triangles that contain two or three clustered vertices, and replace the clustered vertex by the representative for triangles that contain one clustered vertex.
5. Record the clustered vertices, vanishing triangles, and the vertex replacements.
6. Update the weights and priority values for the representative and its neighboring vertices.

The cycle is repeated until a user specified reduction rate is reached. The loop yields a sequence of meshes M^n, M^{n-1},…, M^0 , for some n, in which M^n is the original mesh and M^0 is the most simplified mesh called *base mesh*. The resulting progressive mesh consists of the base mesh M^0 and the sequence of recorded information necessary for the refinement.

Part B: Effective criteria are proposed to preserve geometric shape, especially shape edges, shape corners, and color discontinuity.

In the preprocessing phase, we divide the terrain grid into blocks with a dike between each pair of adjacent blocks. In the navigation phase, blocks are first tested for view-volume culling for each new viewpoint, and for each of those blocks intersecting the view volume, we check to see if its cached LOD can be reused in the new frame. That is, the cached LOD can be reused if its projected error with respect to the new viewpoint is within a pre-specified error, or a new LOD should be re-generated if the test fails. The test is block-based; rather than vertex-based, and thus can be performed very efficiently. After the LOD of all blocks within the view volume are available, we triangulate all the dikes in such a way that the LOD models of different resolutions can be smoothly blended.

4. Experimental Results

Part A: View-independent LOD modeling for geometric objects.

The proposed method for view-independent geometric simplification has been implemented using C language and several models have been tested on a PC with Pentium II 233MHz CPU, 64M RAM, and Window NT 4.0. We first demonstrate the computation efficiency of our method. As shown in Table 1, each of ten models has been simplified to about 500 triangles in 1 to 40 seconds. In each clustering step, in average 4:10 to 20:56 triangles are clustered.

The number of levels is in general dependent on model’s geometry and the number of triangles in its original mesh. For the test models, 412 to 3484 levels are obtained. Levels of all models are also reduced to 100 levels. On each model, we have performed 20000 runs each of which refines the mesh from M^0 to M^n followed by simplifying from M^n to M^0. Table 2 shows the timing data of this experiment for the progressive meshes with original levels and reduced levels. For progressive meshes with higher number of levels, the efficiency gain from level reduction is more significant. For different level of model, the rendering results were shown as Figure 1.

Part B: View-dependent LOD modeling for terrain.

About this part, our experiments have been performed using a terrain data of Dan-Shoei River. Results are obtained on a PC with Pentium III 660Mhz CPU, 128MB Ram, and GeForce 256 3D graphics card. The terrain data includes an area of 26,400m×26,400m, and is divided into 20×20 blocks, each of which has 33×33 grid vertices. A complete triangulation of this terrain data has 868,488 triangles. We set up a navigation path with height about 1,000m, 40 degrees for field of view, and a display window of 800×800 pixels. The rendering result was shown in Figure 2.

Using LOD caching, we have 92% to 98% speed-up in LOD construction time and 25% to 46% speed-up in frame time. Note that the LOD models obtained using LOD caching have less number in triangles, ranging from 2.2% to 8.7% in our experiment.

5. Conclusion

We have proposed a mesh reduction method that combines the geometry and topology simplification, and their advantages as well. The method produces a progressive mesh by using a more general collapsing scheme, in which more than three vertices can be clustered in each reduction step. The method also preserves the discontinuity of color attribute (color borders), which is perceptually important but has been less addressed in previous methods.

Excitingly, we have found that the proposed penalty function can effectively preserve geometry features and is able to eliminate small features of high curvature. The involved clustering operation has also shown its strength on topology simplification. The color discontinuity is also effectively preserved in such a way that the color border is simplified as well when the mesh is reduced. As the future research directions, we are currently extending the proposed method to support a view-dependent refinement of progressive mesh.

Regarding our another proposed view-dependent LOD techniques for large terrain data, we have presented a terrain rendering system in which a dike structure and a LOD caching mechanism have been proposed to, respectively, remove cracks usually occurring in the boundary of adjacent blocks and speed up the LOD selection by reusing previously constructed LOD models. The experiments we have done revealed that dike structure successfully blends two LOD models of different resolution, and the LOD caching mechanism is able to speed up the LOD construction by 92-98%, and the frame time by 25-46%. Among future study plans, we will focus on frame-time control and the integration of hybrid rendering techniques into terrain rendering systems.

6. References

[1] J. D. Cohen, A. Varshney, D. Manocha, G. Turk, H.Weber, P. Agarwal, F. P. Brooks, Jr., and W. Wright. Simplification envelopes. In *Proceedings of ACM Siggraph ’96*, pages 119–128, 1996.

[2] M. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. Mineev. Roaming terrain (real-time optimally adapting meshes). In Proceeding of IEEE Visualization '97, 1997.

[3] R. J. Fowler and J. J. Little. Automatic extraction of irregular network digital terrain models. Computer Graphics (Proceedings of ACM Siggraph '79), 13(2):199-207, 1979.

[4] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of ACM Siggraph ’97*, pages 209–224, 1997.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. *In Proceedings of ACM Siggraph’93*, pages 19–26, August 1993.

[6] H. Hoppe. Progressive meshes. In *Proceedings of ACM Siggraph ’96*, 1996. *Computer Graphics*, pages 455–465. Springer-Verlag, 1996.

[7] H. Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. *In IEEE Visualization’98*, pages 35–42, October 1998

[8] P. Lindstrom, D. Koller, W. Ribarsky, W. Hodges, L. Faust, and G. Turner. Real-time, continuous level of detail rendering of height fields. In Proceedings of ACM Siggraph '96,

[9] R. Pajarola. Large scale terrain visualization

using the restricted quadtree triangulation. In Proceeding of IEEE Visualization '98, 1998.

[10] M. F. Polis and D. M. Mckeown. Iterative tin generation from digital elevation models. In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pages 787-790, 1992.

[11] J. Popovi´c and H. Hoppe. Progressive simplicial complexes. In *Proceedings of ACM Siggraph ’97*, pages 217–224, 1997.

[12] J. Rossignac and P. Borrel. Multi-resolution 3d approximations for rendering complex scenes. *Modeling in Computer Graphics: Methods and Applications*, pages 455–465, June 1993.

[13] W. J. Schroeder. A topology modifying progressive decimation algorithm. In *Proceedings of IEEE Visualization’97*, 1997.

[14] W. Schroeder, J. Zarge, andW. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, July 1992.

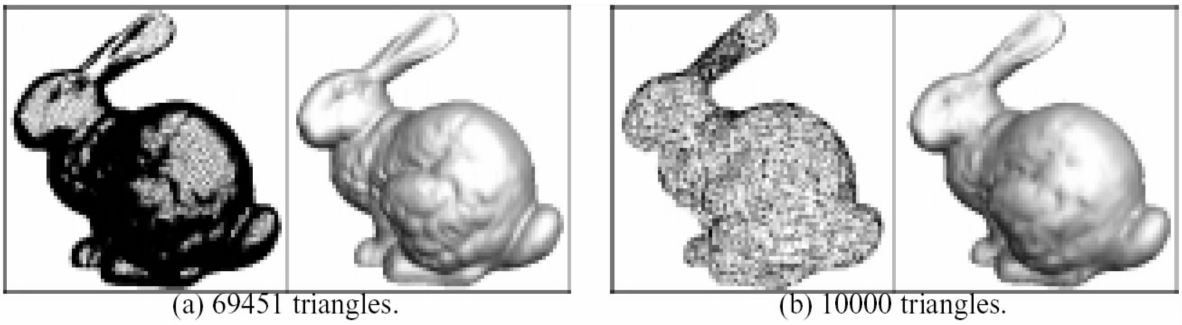
[15] G. Turk. Re-tiling polygonal meshes. *Computer Graphics (Proceedings of Siggraph)*, 26(2):55–64, 1992.

Table 1: *Computation efficiency and clustering performance.*

Models	Size of M^n	Size of M^0	Reduction time in second	# of levels	Average # of triangles clustered in each level
Canstick	4150	498	1	734	4.98
Footbones	4204	498	1	417	8.89
Easter	4976	500	1	450	9.95
Beethoven	5030	500	2	1105	4.10
Cow	5804	496	2	501	10.60
Streetlamp	8828	498	8	1225	6.80
Teaport	9216	498	9	831	10.49
Spider	9286	498	3	867	10.14
Dog	33885	499	40	1624	20.56
Bunny	69451	500	32	3484	19.79

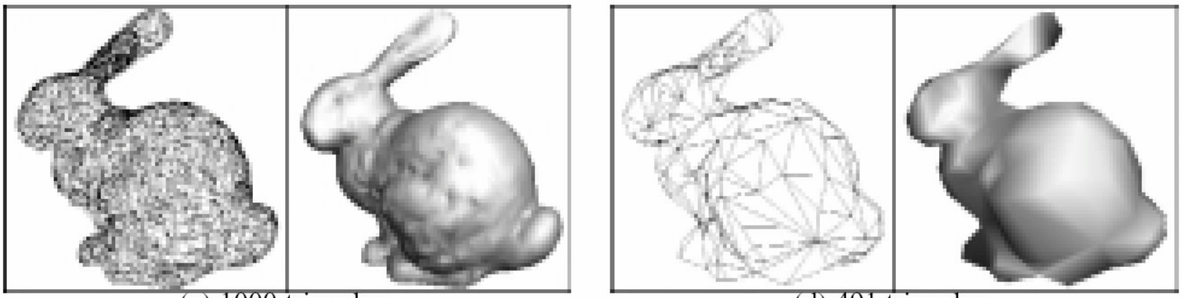
Table 2: *Average time in second for refining from M^0 to M^n.*

Models	# of levels	Average refining time in second	Reduced # of levels	Average refining time in second
Canstick	734	0.000675	100	0.000500
Footbones	417	0.000550	100	0.000450
Easter	450	0.000650	100	0.000525
Beethoven	1105	0.000900	100	0.000675
Cow	501	0.000725	100	0.000550
Streetlamp	1225	0.001375	100	0.001325
Teaport	831	0.001175	100	0.000875
Spider	867	0.001175	100	0.000700
Dog	1624	0.003775	100	0.002875
Bunny	3848	0.004300	100	0.004550



(a) 69451 triangles.

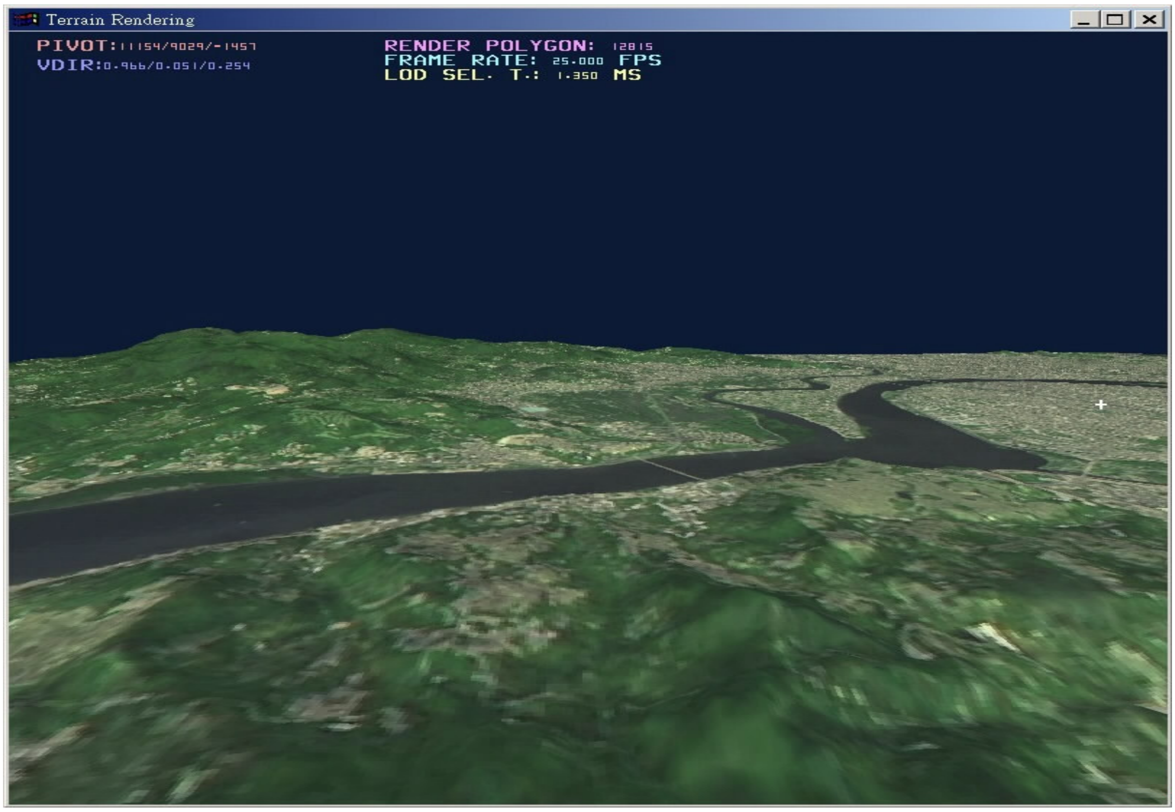
(b) 10000 triangles.



(c) 1000 triangles.

(d) 491 triangles.

Figure 1: *Bunny model.*



```
PIVOT:11154/9029/-1457                   RENDER POLYGON: 12815
VDIR:0.966/0.051/0.254                   FRAME RATE: 25.000 FPS
                                         LOD SEL. T.: 1.350 MS
```

Figure2: The run-time rendering result of terrain data of Dan-Shoei River