05-LD

The 2nd International Conference on Cybernetics and Information Technologies, Systems and Applications

ISAS
CITSA
2005

11th International Conference on
Information Systems Analysis and Synthesis

July 14-17, 2005 Orlando, Florida ~ USA

PROCEEDINGS

Volume II

Edited by
Jose Aguilar
Hsing-Wei Chu
Elena D. Gugiu
Ilka Miloucheva
Naphtali Rishe

IIIS
Organized by
International Institute of Informatics and Systemics
Member of the International Federation of Systems Research (IFSR)

Level of Detail Control and Multi-Resolution Model for Online 3D GIS

Naphtali D. RISHE, Yanli SUN, Maxim CHEKMASOV, Andriy B. SELIVONENKO,
Malek ADJOUADI, Armando BARETTO, Scott GRAHAM
School of Computer Science, Florida International University
Miami, Florida 33199, USA

and

Ben WONGSAROJ, Royel HAYNES, Kiesha PIERRE, Asha BRITO
Division of Computer Sciences and Mathematics, Florida Memorial University
15800 NW 42nd Ave Miami Gardens, FL 33054, USA

ABSTRACT

For online 3D GIS, the data model complexity has to be limited to allow users with conventional PCs and network bandwidth to view animation at a reasonable frame rate. For this purpose, we have developed a view-dependent multi-resolution terrain model. This multi-resolution terrain model is constructed using a distance-based region subdivision technique. According to the distance from the viewpoint, the entire scene is divided into sub-regions of different levels of details. The resolutions of sub-regions in the rendering model are adaptively determined by a view-dependent level of detail control algorithm.

Keywords: Geographic information system, terrain modeling, three-dimensional visualization.

This material is based on work supported by the National Science Foundation under Grants No. HRD-0317692, EIA-0320956, and EIA-0220562.

1. INTRODUCTION

A 3D (three dimensional) terrain model contains two types of information: geometry related and surface related. Geometry information is represented by a mesh. Surface information is 2D remotely sensed imagery; we refer to [1] as an example. 3D GIS (geographic information system) visualization is inherently data- and computation-intensive. Large-scope terrain visualization results in significant volumes of data in a rendering model. Pipelined 3D graphics handling with multiple stages thus exhibits long processing times.

Current 3D GIS applications are mainly used in special environments such as scientific and military endeavors, where powerful servers for computation and storage, high speed networking resources, and high performance workstations with dedicated graphics accelerators are available. However, the hardware performance of graphic subsystems has greatly improved in recent years, so these applications will soon be available to casual users. Since a conventional PC (personal computer) still cannot handle the complexity of today's 3D GIS applications, algorithmic solutions for efficient data storage, transmission, modeling, and simplification are essential to optimize the performance of existing systems to improve the QoS (quality of service) of 3D GIS. Among these solutions, feature-preserving data reduction is the most effective approach in QoS management of 3D GIS. This technique aims to reduce the highly detailed data model to convincing level of realism, thus enhancing the rendering throughput. We refer to [2] for further discussion.

In 3D presentations, the data model complexity affects both display quality and rendering performance. Rendering large-scope terrain with an excessively detailed data model is not practical, especially for online 3D GIS. It is important for the QoS management framework to carefully choose the LOD (level of detail) in the rendering model to optimally balance the trade-off between display quality and rendering performance [3]. Multiple resolution rendering and view-dependent LOD control algorithms are used to effectively reduce the data complexity and to preserve the display quality. The LOD in the rendering model is dynamically adjusted by many factors. In principle, high-resolution data is used in rendering small areas to reflect the geographic details of the underlying terrain and low-resolution data is used to display more distant large terrain areas. Based on distance to the viewpoint the entire visible scene can be divided into multiple sub-regions of different detail levels where sub-regions closer to the viewpoint are rendered with higher resolution and distant areas are rendered with lower resolution [4, 5]. During terrain animation, navigation speed is one of the major factors that affects data transmission payload, so it should be considered by the LOD control algorithm. The LOD control model we propose is based on a distance-based region-subdivision mechanism. The number of detail levels in the rendering model and their resolutions are adaptively adjusted according to dynamic viewing and navigation conditions.

2. VIEWING FACTORS IN CONSIDERATION

The complexity of data model arises from the area of the visible scene and the underlying data resolutions. The applied mesh/texture resolutions in the rendering model affect both rendering performance and terrain visualization quality. For the sake of presentation quality, the data resolutions in the rendering model have to be carefully selected to optimally balance the performance and the display quality. In perspective viewing, the terrain area covered inside the view frustum depends on viewing height, view frustum size (fovy angle and x/y ratio) and viewing transformation angles (pitch, roll). In general, larger viewing height and fovy values (the field of view (fovy) specifies the angle of the view volume) indicate a larger visible area. For display quality, high-resolution data is preferred in rendering geographic details of small areas and lower resolution data is preferred for rendering vast areas. Within the terrain model, distant regions should be rendered with low-resolution data and regions close to the viewpoint should be rendered with high-resolution data.

For online terrain animation, the performance of rendering a single frame depends on the data access time and terrain rendering time. The per-frame data retrieval overhead is roughly proportional to the area updated for each frame. This in turn is related to the horizontal navigation speed. The higher the navigation speed, the more terrain data needs to be retrieved to update the scene for each frame.

Based on these observations, our model simplification algorithm adopts viewing distance based on a region-subdivision strategy. The entire scene is decomposed into sub-regions of different LOD according to distance to the viewpoint. The texture and mesh resolutions for each sub-region are determined by dynamic viewing status to guarantee a desired level of realism in the rendering result.

3. MULTIPLE LOD RENDERING MODEL

The algorithm for multi-resolution model construction includes three steps:

- View culling estimation - estimating the visible terrain region for the current frame.
- Distance-based scene subdivision - sub-dividing the entire scene into sub-regions based on distance to the viewpoint.
- Surface crack handling - calculating texture/mesh resolutions for every sub-region in the terrain model so that the entire rendering surface appears to be spatially smooth and continuous.

3.1. View Culling Estimation

The view culling processing is an estimation process. The visible terrain area is calculated according to the viewpoint coordinate, view volume and view transformation angles in the following steps:

- The depth of the viewing volume depth is set large enough to accommodate the potential geographic scene. The distance from the viewpoint to the 'near' side of the frustum is set to be 1/100 of the viewing height and the depth of 'far' side is set to be 9,999 times the viewing height.
- View rotation is performed along the viewpoint according to Pitch, Roll and Heading angles.
- The estimated visible area is obtained by intersecting the view frustum with plane of zero height (z=0). The result is given as a polygon with four vertices in anti-clockwise order: (x1, y1, 0), (x2, y2, 0), (x3, y3, 0) and (x4, y4, 0).

The above view culling process is a 'quick' estimation. It does not consider actual height of the terrain. The result based on sea level assumption may not be accurate to reflect the actual visible terrain in 3D viewing. We also propose to enlarge the polygon by 20% in size to ensure that the actual visible terrain inside the view frustum is completely enclosed in the estimation result.

3.2. Distance Based Scene Subdivision

To determine the region sub-division before mesh data is loaded and to simplify distance calculation in 3D space, the region subdivision is based on projected distance calculated in 2D space without elevation. Given vertex P (x, y, z), its projected distance to viewpoint VP (x0, y0, z0) is defined as:

$$D = \max(|x - x0|, |y - y0|)$$

Based on the current viewing status, the region sub-division factor R is calculated as:

$$R = Height \times \tan\left(\frac{\max(|pitch|, |roll|, Fovy)}{2}\right) \times \max(1, x\_y\_ratio)$$

Where

•	Height	Viewing height
•	Fovy	The field of view angle
•	X_y_ratio	The height/width ratio of view frustum
•	Pitch	Viewing transformation angle pitch
•	Roll	Viewing transformation angle roll

Based on region sub-division factor R and projected distance D, the entire scene is decomposed into up to four sub-regions, see Table 1.

Sub-region	D Value	Description
1	D≤2R	Sub-region closest to the viewpoint, highest data resolutions should be applied to this sub-region.
2	2R<D≤ 8R	Sub-region with medium distance to the viewpoint and medium data resolutions should be applied.
3	8R<D≤ 20R	Sub-region considered far enough to the viewpoint and lowest data resolutions should be applied.
4	D>20R	Areas that are too far to the viewpoint are ignored in terrain rendering.

Table 1. Region Subdivision Criteria.

Distant areas near the horizon line (in the case D > 20R) on the scene are ignored from rendering. Since these areas will be ultimately minified to a narrow strip near the horizon line in terrain rendering, cutting them off does not disrupt the realism of display but can significantly reduce the data model complexity.

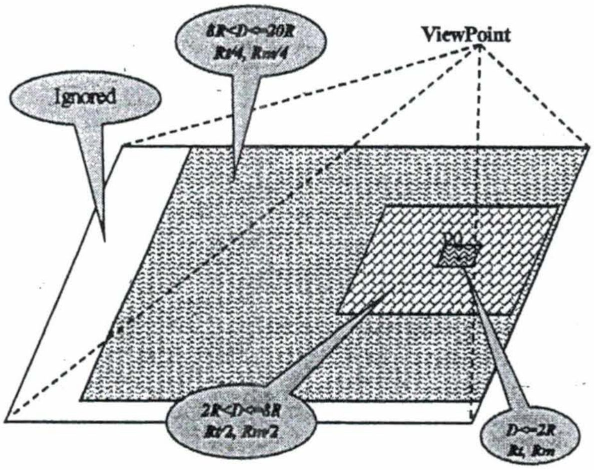

Figure 1. Illustration of Region Based Multiple LOD Terrain Model.

The rendering model may have at most three LODs. The number of LODs applied in the rendering model depends on the viewing status. When looking straight downward (under zero pitch and roll angles) the entire scene is completely within sub-region 1 and there is only one LOD. Under certain conditions, there may be two or three detail levels in the rendering model. When viewing under large Pitch and Roll angles, the three detail levels exist and region cut-off is applied. The distance based region subdivision is illustrated on Figure 1.

Sub-region 1 is the closest sub-region to viewpoint and it is assigned the highest data resolutions in the scene. The texture resolution R_t and mesh resolution R_m for sub-region 1 are calculated as follows:

$$\mathrm{Speed\_Factor} = 1 - \min\left(\frac{\mathrm{HSpeed}}{\mathrm{FrameRate} \times \mathrm{C}}, 0.5\right)$$

$$\mathrm{Rt} = \frac{\mathrm{Height}}{\mathrm{HR\_RATIO} \times \tan\left(\frac{\mathrm{Fovy}}{2}\right)} \times \mathrm{Speed\_Factor}$$

$$\mathrm{Rm} = \frac{\mathrm{Rt} \times \mathrm{Tile\_Size}}{\mathrm{Mesh\_Grid\_Size}} \times \mathrm{Speed\_Factor}$$

Where

•	HR_RATIO	An empirical constant number 400, determined by experiment results
•	Tile_Size	The texture image tile size in pixel
•	Mesh_Grid_Size	The grid size in a mesh tile
•	Hspeed	Horizontal navigation speed
•	FrameRate	Animation frame rate
•	C	A constant number 100. This is selected based on experiments.

For sub-regions 2 and 3, as the viewing distances to the viewpoint increase, the applied data resolutions decrease accordingly. In sub-region 2, the texture and mesh resolutions are lowered to half of that for sub-region 1. Similarly in sub-region 3, the texture and mesh resolutions are lowered to half of that for sub-region 2. The data resolutions applied to the three sub-regions are listed in Table 2.

Sub-region	**Resolutions**	
	Texture	**Mesh**
1	R_t	R_m
2	$R_t/2$	$R_m/2$
3	$R_t/4$	$R_m/4$

Table 2. Resolutions of Sub-Regions in the Terrain Model.

3.3. Surface Crack Handling

A challenging issue in multi-resolution rendering is to make the rendering surface appear to be spatially smooth and continuous. For large-scope terrain, independent adjustment of local mesh resolution may cause cracks in the rendering surface along the borders of sub-regions of different LODs. At the borders between sub-regions in our multi-resolution terrain model, mesh tiles of different resolutions may not spatially match with each other (shown as a shaded area on Figure 2). To avoid the terrain surface cracks in the rendering result, tiles along the sub-region borders are specially processed.

At the border of two sub-regions, the mesh resolution in the sub-region of higher LOD is twice as much as the mesh resolution in the sub-region of lower LOD, therefore a mesh tile in the lower LOD sub-region always connects exactly two mesh tiles of higher resolution (As mesh tiles A, B and C shown on Figure 2). For the mesh tiles of higher resolution, the border cells connecting the lower resolution mesh tile are adjusted to exactly match the border of the lower resolution mesh tile, see Figure 3). This mesh adjustment ensures that all mesh tiles in the rendering model are seamlessly connected without cracks to form a smooth terrain surface.

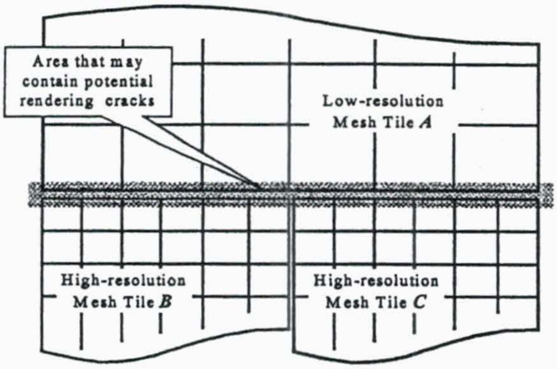

Figure 2. Mesh Tiles of Different Resolutions Do Not Match at the Borders.

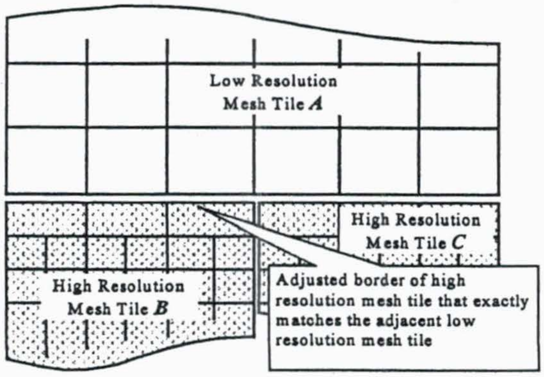

Figure 3. Adjustment of Mesh Tile Borders along Sub-region Borders.

Those mesh tiles with adjusted borders are specially marked to distinguish them from the other mesh tiles in the data model. The specially marked tiles are simplified in a mesh simplification process.

4. ANALYSIS OF LOD CONTROL ALGORITHM AND CONCLUSIONS

The simple region-based LOD control algorithm adaptively controls the LODs in the rendering model to maintain the virtual quality of the presentation at the desired level. It is very efficient and is suitable for real-time terrain animation.

When viewing under a narrow view frustum and small Pitch and Roll angles, a relatively small amount of terrain data is needed to render the visible terrain. In this case, the data model contains only high-resolution data to reflect the fine details of a small terrain area. On the other hand when viewing under wide view frustum and large viewing angles (Pitch and/or Roll), a large volume of data is needed to render the large visible terrain area. In this case, a reduction of the data complexity is needed. The data model of up to three LODs, combined with the artificial horizon cut-off scheme for distant areas, can significantly reduce the terrain data complexity. For mesh/texture tiles of fixed sizes, lowering the resolution by 50 percent means increasing the underlying area four times. In sub-regions 2 and 3, the data complexity can be reduced to about 75 and 94 percent compared to a single resolution model. A snapshot of a multi-resolution rendering result of the North California coast area is shown in Figure 4. The terrain model for this example contains three detail levels and the distant area near the horizon is artificially cut off.

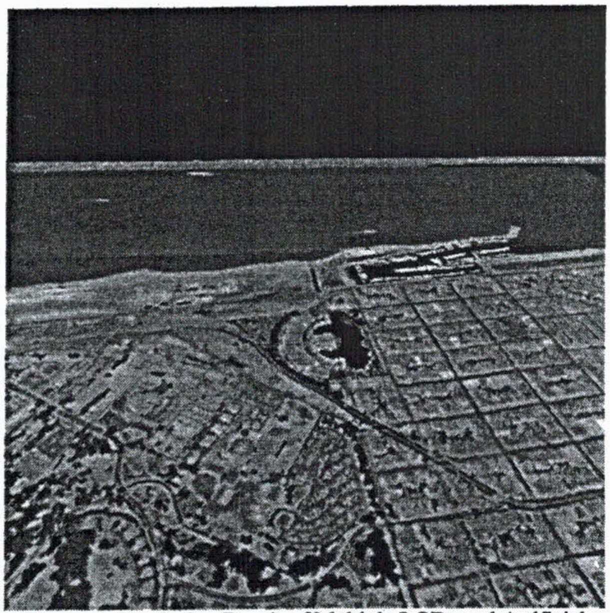

Figure 4. Rendering Result of Multiple LODs and Artificial Horizon Cut-Off

5. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grants No. HRD-0317692, EIA-0320956, EIA-0220562, CNS-0426125, IIS-0326284, CCF-0330342, IIS-0086144, and IIS-0209190.

6. RFERENCES

[1] N. Rishe. "TerraFly: A Web-Enabled Application for Visualization and Manipulation of Remotely Sensed Data." available at http://terrafly.fiu.edu/tf-whitepaper.pdf

[2] Yanli Sun. "3D TerraFly – Quality of Service Management of Online Interactive 3D GIS Presentation." PhD dissertation, Florida International University, 74 pp., 2004.

[3] J. H. Clark. "Hierarchical geometric models for visible surface algorithms." Communications of the ACM, 19(10):547-554, 1976.

[4] B.J. Grosz, C.L. Sidner. "Attention, Intensions and the structure of Discourse." Computational Linguistics (12), 1986, pp 175-204.

[5] A. Krger. "Automatic Graphical Abstraction in Intent-Based 3D Illustrations." In Catarci et al. (Eds) Advanced Visual Interfaces Workshop, AVI 98, Italy, 1998, ACM Press, pp 47-55.