REPORT DOCUMENTATION PAGE

AD-A255 481

Form Approved OMB No. 0704-0188

(2)

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 3-7 August 1992 — Scientific Paper
4. TITLE AND SUBTITLE Adaptive Mesh In Visualization Of Terrain Elevation Data	5. FUNDING NUMBERS
6. AUTHOR(S) J. John Kim	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Topographic Engineering Center Attn: CETEC-LO Fort Belvoir VA 22060-5546	8. PERFORMING ORGANIZATION REPORT NUMBER R-182
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; Distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The initial step in terrain visualization is to utilize terrain elevation data to portray the earths surface. After that, features are overlaid as terrain colors, textures, or models on the visualized elevation data. Terrain elevation data are often visualized in a wire-frame mesh where grid posts represent the elevations of corresponding points in the terrain. When terrain is displayed in a perspective view, the grid sizes in the wire-frame mesh must be varied adaptively according to the distance betwe n the viewpoint and the terrain so that the area nearest to the viewer is more densely sampled than the area further from the viewer. A wire-frame mesh with such varying grid sizes is called an adaptive mesh in this paper. A loosely defined adaptive mesh often creates anomalies between two areas with different grid sizes. This paper presents a few schemes for adaptive meshes that do not create anomalies.

14. SUBJECT TERMS 1. TERRAIN VISUALIZATION, 2. WIRE-FRAME MESH 3. ADAPTIVE MESH			15. NUMBER OF PAGES 6 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT

ADAPTIVE MESH IN VISUALIZATION OF TERRAIN ELEVATION DATA

J. John Kim
US ARMY Topographic Engineering Center
Fort Belvoir, VA 22060-5546

ABSTRACT

The initial step in terrain visualization is to utilize terrain elevation data to portray the earths surface. After that, features are overlaid as terrain colors, textures, or models on the visualized elevation data. Terrain elevation data are often visualized in a wire-frame mesh where grid posts represent the elevations of corresponding points in the terrain. When terrain is displayed in a perspective view, the grid sizes in the wire-frame mesh must be varied adaptively according to the distance between the viewpoint and the terrain so that the area nearest to the viewer is more densely sampled than the area further from the viewer. A wire-frame mesh with such varying grid sizes is called an adaptive mesh in this paper. A loosely defined adaptive mesh often creates anomalies between two areas with different grid sizes. This paper presents a few schemes for adaptive meshes that do not create anomalies.

INTRODUCTION

In a fly-through model of the terrain visualization(TV) system, the fly-through speed is affected by hardware performance rate, terrain data density, eye position and looking direction, viewing distance, and so on. When the number of items to process such as pixels, lines, polygons exceeds the hardware's real time performance rate, graphics rendering is delayed and the time gap between the movement of a positioning device (eg, mouse or spaceball) and the scene drawing is noticeably non-uniform. Especially, the real time fly-through over a terrain with a dense data is difficult to achieve even in a high end graphics machine. To achieve the real time graphics rendering, many researchers in the TV field have tried to find a good method of reducing the number of polygons with the minimal degradation of the viewing resolution. One such method is to define many levels of detail (LODs) and to apply a different LOD to each subdivided region of the terrain on the basis of the distance between the region and the current eye position. For example, a region far away from the viewing point appears too small to see its details in a perspective view and need not be presented in the same LOD as used to present a region closer to the viewing point. Figure 1 shows examples of LODs implemented as sampling grid resolutions where the sampling grid space for the region far from the viewing point is larger than that for the region near the viewing point.

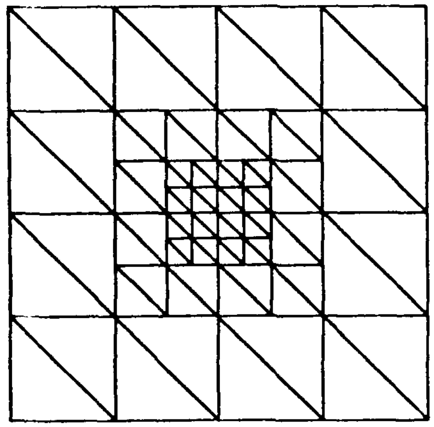

Figure 1. A sampling grid with 3 levels of detail at the moment when the viewing position is right above the center of terrain.

This approach, however, generates anomalies where two neighboring regions with different LODs meet. Figures 2-4 show how an anomaly occurs. Consider Point ***a*** in Figure 2. When Figure 2 is represented by a polygonal mesh of many LODs and Point ***a*** falls on the boundary between two regions with different LODs as shown in Figure 3, Point ***a*** will be represented as Vertex ***A*** of Polygon ***X*** and Point ***A'*** on the sides of Polygons ***Y***. Then, Vertex ***A*** of Polygon ***X*** is assigned the sampled value at Point ***a*** and Point ***A'*** of Polygon ***Y*** is assigned an interpolated value using the sampled values of two vertices of the side of Polygon ***Y*** which contains Point ***A'***. Consequently, when the interpolated value and the sampled value have a significant difference, the rendered screen image displays it as if a cliff or a hole exists in the terrain as shown in Figure 4.

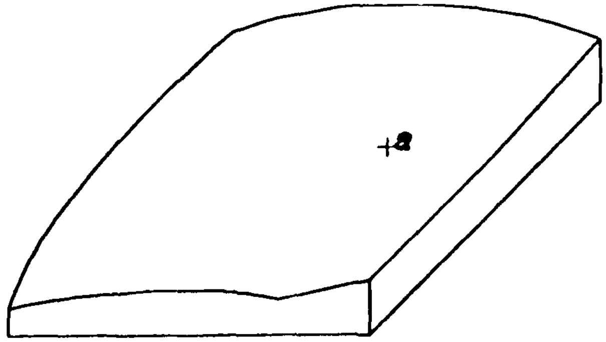

Figure 2. An example of terrain

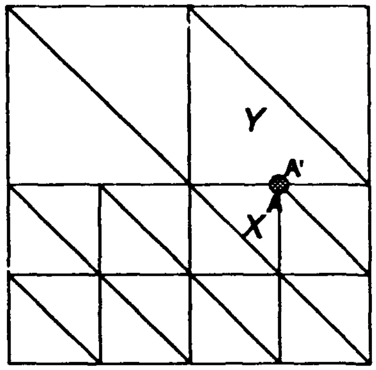

Figure 3. An example of sampling grid with two levels of detail

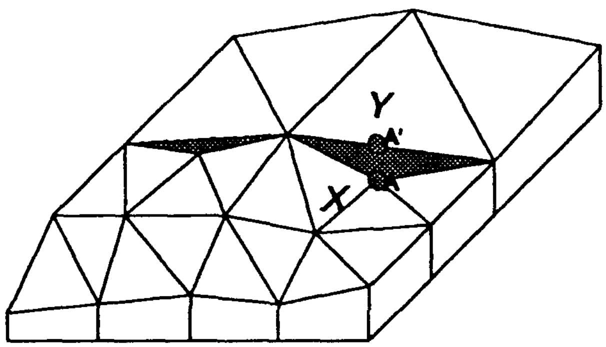

Figure 4. Resulting anomalies on the boundary of two region with different levels of detail

Three methods to eliminate this kind of anomaly are presented in this paper. The first one is to use reconciling regions between two discretely changing LODs and the second one is to use continuously changing LODs instead of using the discrete LODs which tends to cause the anomalies. The third one is to combine the first and the second ones. These methods are described in the next section.

PROPOSED METHODS

In this section, three methods are described which reduce the number of polygons adaptively without creating the anomalies described in the previous section.

Method One
Reconciling Region Between Discretely Changing LODs

The anomalies in the LOD approach can be prevented by reconciling the area where two regions with different LODs meet so that a vertex of a polygon never falls on a side of another polygon. The patterns of reconciled polygonal meshes are illustrated in Figures 5-6.

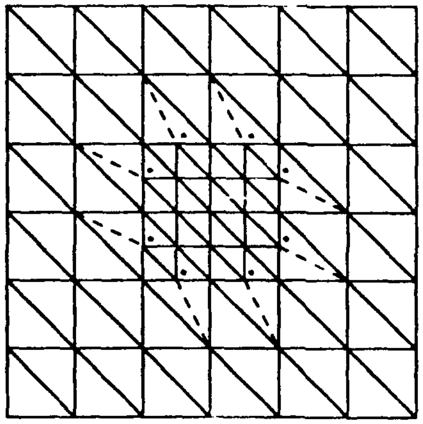

Figure 5. An example of adaptive mesh with polygons splitted

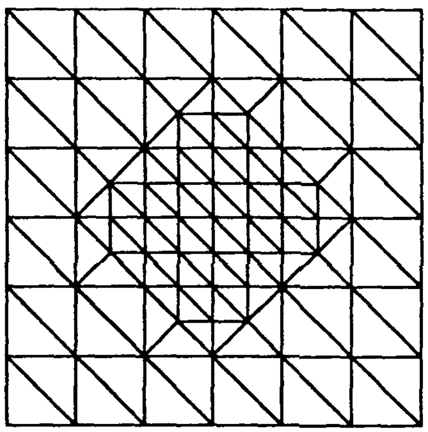

Figure 6. An example of adaptive mesh with irregular patterns

In Figure 5, each of the points marked '*' is a vertex of a triangle and falls on the side of an adjacent bigger triangle. Falling on a side is avoided by splitting the bigger triangle into two smaller triangles, where the point marked '*' becomes a shared vertex. Figure 6 shows another way to reconcile the problem area, where the vertices falling on the sides of other triangles are avoided at the cost of the irregularity of the triangular mesh. There can be many different polygonal meshes which prevent a vertex from falling on a side. Their computational efficiencies depend on a host graphics system.

Method Two
Adaptive Polygonal Mesh With Continuously Changing LODs

It is notable that the anomalies are the products of discrete LODs. This notion led us to consider "continuous" LODs as an alternative. It is meant by the "continuous" LODs that a terrain is tessellated into polygons whose size are varying gradually or the data sampling spaces between points in a terrain are varying continuously. This can be accomplished by putting an imaginary viewing grid at a fixed distance from the eye position and then sampling the data points at the grid points on the terrain projected by the viewing grid. Figure 7 shows the concept of "continuous" LODs, where the imaginary viewing grid is equally spaced but the projected grid on the terrain base is gradually varying in its post distances (***a*** > ***b*** > ***c*** > ***d***).

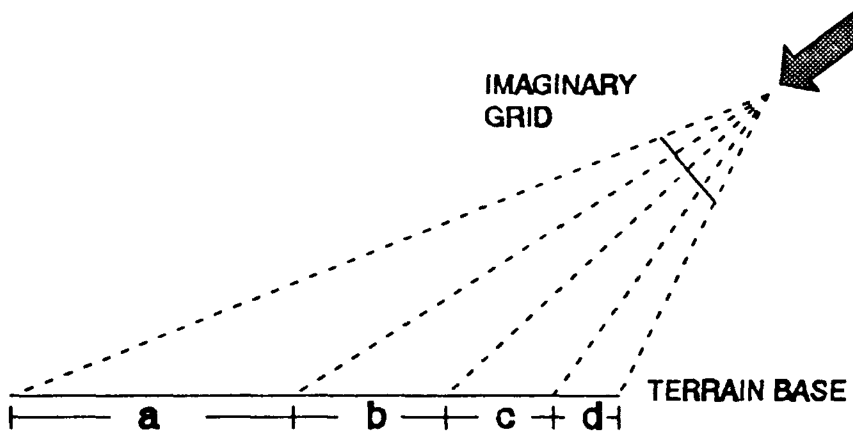

Figure 7. When the imaginary grid of an equal post space is projected on the terrain base a grid with continuously varying post spaces is obtained.

This method is useful when an object to visualize is not very voluminous, since the highest portion of such an object would be nearer to the viewing point (meaning denser sampling required) even if that portion is far back in the terrain base, and when the viewing direction is not perpendicular to the terrain base. However, the domain of terrain visualization is mostly the earth surface which is relatively flat and the viewing direction in a fly-trough model is seldom perpendicular to the terrain base, which makes this method applicable.

A major advantage of Method Two over Method One is that the fly-through speed over a terrain is dependent only on the resolution of the imaginary viewing grid and hence, the desired uniform response time can be achieved when the viewing mesh resolution is properly set according to the performance rate of a host computer. Also, it is possible to achieve the real time fly-through at the sacrifice of the rendering resolution by controlling the viewing mesh interactively.

Method Three
Combination of Methods One and Two

As many readers may have already noticed, Method One or Method Two has its own limitation: Method One does not have the mechanism that responds to the change of viewing direction, and Method Two is less sensitive to the change of viewing distance. It is notable that they complement each other. This fact gives rise to the idea that these two methods should be combined to maximize the effect of the LOD approach. Here is the way it can be done: The imaginary viewing grid in Method Two is replaced by the reconciled adaptive mesh of Method One. The imaginary viewing grid has discretely changing levels of detail with their boundaries between different LODs reconciled. The number of LODs and the number of grid posts in the imaginary viewing grid are controlled by an end user interactively, depending on the performance rate of a host and the desired speed of fly-through.

CONCLUSIONS AND FURTHER WORK

In this paper three methods of adaptive mesh generation that do not create anomalies are proposed. These methods are now being implemented in C programming language for upgrading an existing fly-through terrain visualization system. Compared to an adaptive mesh without LOD boundaries reconciled, the extra overhead of Method One is negligible. However, Methods Two and Three have some overhead since the location of the projected point on a terrain base has to be calculated in more complicated ways for each post in the imaginary viewing grid. A special hardware device for the computation would alleviate the problem of overhead. One of the biggest advantage of Methods Two and Three is the uniform response time because the number of items to render is constant after the imaginary viewing grid is defined.

The current techniques need to be further studied. For instance, this paper introduces the concept of "continuous" LODs using an imaginary viewing grid. The optimal definition of the imaginary viewing grid should be sought since the orientation, the distance from the viewing point, the grid definition, and the LOD definition of the imaginary viewing grid may have a great impact on the performance of a terrain visualization system.

ACKNOWLEDGEMENTS

The author would like to thank Jim Gladden and C. Daniel Carleton for their helpful discussion and the initial implementation of the proposed methods. Thanks are due also to Thomas Jorgensen for his encouragement.