HIERARCHICAL DESIGN OF 3D MESHES IN REPRESENTATION/STORAGE OF VOLUME DATA FOR MULTIMEDIA APPLICATIONS

Işıl Celasun[1], Eren Manavoğlu[2], Barış Yeğen[1], A. Murat Tekalp[3]

[1]*Dept. of Electronics and Communication Eng., Istanbul Technical Univ., Maslak, 80626, Istanbul, Turkey*

[2]*Dept. of Computer En, Istanbul Technical Univ., Maslak, 80626,Istanbul, Turkey*

[3]*Dept. of Electrical Eng. and Center for Electronic Imaging Systems, Univ. of Rochester, P.O.Box 270126, Rochester, NY 14627, USA*

Abstract Hierarchical mesh representation and mesh simplification have been addressed in computer graphics for adaptive level-of-detail rendering of 3D objects used for animation. In this paper, we propose a new simplification method to design hierarchical 3D meshes such that each mesh level (representing a different level of spatial hierarchy) is a subset of the previous higher detail mesh and has Delaunay topology. This topology constraint on each mesh layer not only helps to design meshes with desired geometric properties, but also enables efficient compression of the mesh data for multimedia applications.

Keywords

Hierarchical mesh design, scalable geometry compression

I. INTRODUCTION

Three-dimensional (3-D) polygonal meshes have been popular in computer graphics to describe the geometry (structure) of world objects. They have been employed to view objects from different angles and/or to render photorealistic synthetic images by texture mapping [1]. For example, an animation can be created by repeatedly mapping a still image (texture map) onto a 3-D mesh subject to global transformations. Three-dimensional meshes are an elementary building block of the Virtual Reality Modeling Language (VRML), a standard for storing and interacting with graphics objects and virtual worlds over the World Wide Web [2].

Hierarchical representation of 3-D meshes have attracted attention because it: 1) provides rendering at various levels of detail (quality scalability); 2) allows progressive/scalable transmission or storage of the object geometry and motion information. Scalability means that terminals of different complexity can extract data of different quality levels from this single bitstream. Hierarchical representation of 3-D meshes has been addressed in computer graphics for adaptive level-of-detail (LOD) rendering of 3-D objects [3]. Geometric methods for fine-to-coarse 3-D mesh simplification include [4]. A wavelet-based multiresolution mesh approximation was proposed in [5]. In VRML, rendering at multiple resolutions is enabled by the LOD node which requires definition of L separate meshes that are stored or transmitted independently (simulcast). Meshes of tetrahedra have many applications, including interpolation, rendering, compression, and numerical methods such as the finite element method. Most such applications demand more than just a triangulation of the object or domain being rendered or simulated. To ensure accurate results, the tetrahedra must be "well-shaped", having small aspect ratios or bounds on their smallest and largest angles [6].

In this paper, we propose a hierarchy of 3D Delaunay meshes and we only remove vertices in the fine to coarse design strategy. We do not reposition vertices nor edges. In Section 2, algorithm for 3D mesh simplification with its proper design parameters is explained. In Section 3, experimental results and conclusions for "Torus" and "Cat" volume data are given.

II. ALGORITHM

II.1. Initial Fine Detail Mesh

The three dimensional Delaunay triangulation is defined as the triangulation that satisfies the Delaunay criterion for *n*-dimensional simplexes (in this case n=3 and the simplexes are tetrahedra). This criterion states that a circumsphere of each simplex in a triangulation contains only the n+1 defining points of the simplex. It has been proven that two dimensional Delaunay triangulation satisfies an "optimal" triangulation, but in three dimensional Delaunay triangulation the situation is not so, since a measurement for optimality in three dimension is not agreed on.

II.1.a Computation of the "Alpha" Parameter

A graph can be defined as $G = (V, E)$ Here V is the set of vertices, $V = \{ v_0, v_1, \ldots, v_1, \ldots, v_{V-1} \}$, and E is the set of edges, $E = \{ e_0, e_1, \ldots, e_i, \ldots, e_{E-1} \}$.
Optimal alpha value has to be computed in order to get a good approximation of the original image. The average tetrahedron edge length (l_e) of the convex hull can be used to determine this optimal value.

$$\alpha = \frac{1}{E} \sum_{i=1}^{E} l_i \tag{2.1}$$

Although the convex hull may contain edges of large length when connecting end nodes of the volume data, experiments show that the average edge length gives us a proper alpha value. In fact, to preserve all tetrahedra present in the 3-D mesh, maximum circumradius of them has to be used to determine alpha value, which in practice is expressed as average value when convex hull is employed. Again these "ill-conditioned" tetrahedra force us to choose average edge length instead of circumradius in alpha value determination.

II.1.b Boundary Extraction Algorithm

The boundary extraction algorithm for 3-D images represented by the Delaunay tetrahedralization, uses the sum of the solid angles at every vertex to determine whether the vertex is on the boundary or not. The solid angle at the vertex v_i of the tetrahedron T (v_0, v_1, v_2, v_3) is defined to be the surface area formed by projecting each point of the face not containing v_i to the unit sphere centered at v_i .

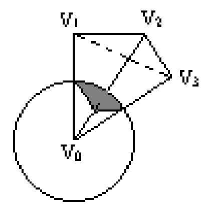

Figure 1 - The shaded area is the solid angle of the tetrahedron T (v_0 , v_1 , v_2 ,v_3) at vertex v_0

For each tetrahedron, the solid angle at each corner is calculated. A vertex is said to be on the boundary, if the sum of the solid angles of every tetrahedron the vertex belongs to at this node, is less than the surface area of a unit sphere, 4π. If the sum of the solid angles equals 4π, the vertex has to be in the interior of the image. It is obvious from the definition of the solid angle of a tetrahedron, that there is no possibility for the sum of the solid angles to be greater than the surface area of the unit sphere, provided that the tetrahedrons do not intersect.

To compute the solid angle (Ψ) of a tetrahedron:

$$\begin{aligned} \Psi_A &= \text{(dihedral angle at edge AB)} \\ &+ \text{(dihedral angle at edge AC)} \\ &+ \text{(dihedral angle at edge AD)} - \pi \end{aligned} \tag{2.2}$$

where the dihedral angle (Θ) at the edge e_{AB} is the angle between the intersection of the two faces containing the edge e_{AB} and a plane perpendicular to this edge:

$$\Theta_{AB} = \pi - | \arccos (face_{ABC} \bullet face_{ABD}) | \tag{2.3}$$

II.2 Mesh Boundary Simplification

A sequential simplification algorithm is used to remove boundary vertices going from one hierarchy level to the next. The boundary simplification algorithm uses a distance parameter D_{max} to control the shape error. D_{max} specifies the pixel error tolerance and the choice of this parameter is discussed in detail in [7]. For each boundary vertex, a candidate edge segment is drawn between an initial boundary vertex and the vertex under consideration, the chord vertex. For every vertex between the initial and the chord vertices, the distance d from the vertex under consideration to the candidate edge segment is computed. Every vertex having d less than D_{max} is a candidate for removal. The first vertex for which d is greater than D_{max}

becomes the new initial vertex. Note that only an independent subset of the removable nodes is actually removed, so as to conform to the hierarchy. This procedure is repeated until all the vertices have been processed. This is an exact generalization of 2D algorithm to 3D case.

A constant D_{max} value is applied to form all levels and either the user is asked to input a certain possible amount of error for each level and the algorithm stop to remove nodes when it can not pass this amount error or vertices up to an average cost value are removed in a level and the algorithm stops when there is no more removal on the boundary.

II.3 Mesh Interior Simplification

The simplest method for 3D mesh simplification is to select a subset of nodes of the finer level mesh according to an importance criterion associated with each vertex, and employ Delaunay triangulation of the selected subset at a given level to obtain the coarser mesh. This method, however does not provide a hierarchy in the topology in the sense that tetrahedra of one level are not necessarily a subset of those of the previous finer detail level. If a hierarchy in both node positions and topology is desired, then a coarse-to-fine design strategy where tetrahedra of the coarser mesh are subdivided. However, this approach usually results in poor aspect ratio tetrahedra. To this effect, we choose an approach where some part of the mesh topology is maintained between successive levels of resolution. This is achieved by starting with the finest resolution mesh, and removing an independent set of vertices at each level.

An “independent” set is a set of vertices among which no two vertices are adjacent to each other. The removal of an independent set of nodes implies that the bounding polygons of the removed nodes can be preserved from one level to the next. Retriangulation of the interior of each bounding polygon using the Delaunay criterion completes the mesh definition at each level.

The aim of the simplification of the interior of a mesh is to remove the maximum number of independent vertices. The general idea is to retain the vertices which are expected to be important, going from one hierarchy level to the next.

II.1.1 Importance Value Function

The importance (*IP*) value for an interior node is defined as the ratio of the sum of its neighbors' volume to its volume.

This IP stress on the connectivity of the mesh and try to retain

$$IP(n) = Degree(n) * \frac{\sum_{i=1}^{neigbors-1} volume(i)}{volume(n)} \tag{2.4}$$

detailed regions' vertices of the volume data.. Degree(n) is criterion of the connectivity of relevant vertex v_n. It represents how many edges are connected to the vertex. From equation (2.4) it is obvious that more smaller the volume(n) and more larger the sum of volume(i) is more higher the importance value of relevant vertex because small volume represents a more detailed region and vertex in this region is more important.

II.3.1.a The Computation of Importance Value

Before the calculation of the importance value function, the volumes of the tetrahedrons and the neighbor vertices and tetrahedrons of each vertex has to be determined.

The volume of a tetrahedron is computed using the following formula:

$$\text{volume} = (1/6)\, \mathbf{a} \bullet (\, \mathbf{b} \times \mathbf{c}\,) \tag{2.5}$$

where **a** is the vector from v_0 to v_1, **b** is the vector from v_0 to v_2, **c** is a vector from v_0 to v_3, in the tetrahedron T (v_0, v_1, v_2, v_3). Using this formula, the volumes of the tetrahedrons can be calculated by an algorithm with a complexity of O(number of tetrahedrons). After the volumes of the tetrahedrons are calculated and the neighbors of each vertex are known, the importance value can be computed by the algorithm given below.

II.4. Interior Vertex Remove Algorithm

The interior vertex remove algorithm is a simple Greedy-type algorithm that removes the vertex having the smallest importance value among the unprocessed ones, and keeps its neighbor vertices. A fast implementation of quicksort algorithm is used to sort the importance values of the vertices.

III. EXPERIMENTAL RESULTS AND CONCLUSIONS

Experimental results related to boundary and interior vertex simplification are given for "Torus" and "Cat" volume data. Boundary node simplification is accomplished for "Torus" with D_{max} = 3.0 and D_{max} = 2.0 for "Cat". "Cat" volume data has a boundary consisted of concave and convex regions so its boundary has to be processed carefully for its global shape information. That is why a low D_{max} value is chosen for "Cat". "Torus" volume data has instead a more uniform shape and a higher value of D_{max} is suitable for it. In fact, this reasoning coincides with the uniform distribution of the vertices on the boundary on in the interior of the volume data modeled by a 3D Delaunay mesh. We end up with four levels of hierarchy for "Torus " volume data where the stopping criterion that there is no more removed vertices on its boundary is applied (Figure 2-16). For "Cat" volume data, we have a huge amount of vertices at the finest level and only three levels of hierarchy is shown in Figure 2-5 Our criteria for the formation of hierarchy of 3D meshes is to remove the maximum number of independent vertices. This will help us in handling huge amount of volume data and so higher compression ratios can be obtained. The tradeoff between maximum removal of vertices and quality of the so formed hierarchy meshes is accomplished by allowing removal of maximum number of vertices so that the formed coarser mesh has an acceptable value for the minimum of the solid angles. In 3D, Delaunay meshes do not satisfy the maximization of the minimum angle criterion. 3D Delaunay tetrahedralizators often produce tetrahedra having faces with fine minimum angle values, but are still badly shaped, just like the sliver. Instead of the minimum angles of the faces, minimum solid angles of the tetrahedra are used to guarantee the existence of the Delaunay property.

Table 1. "Cat" data hierarchy levels vs solid angles

Hierarchy Level	Average of the minimum solid angles
Original Finest	0.016755
First coarser	0.023105
Second	0.026592
Coarsest	0.029761

The average of the minimum angles of the tetrahedra is calculated at each hierarchy level to give an idea of the quality of the mesh produced. The solid angles of a tetrahedron are equal to each other and have a value of 0.551286, which is the maximum value that the minimum of the solid angles may have.

Table 2. "Torus" data hierarchy levels vs solid angles

Hierarchy Level	Average of the minimum solid angles
Original Finest	0.06752
First coarser	0.06940
Second	0.07267
Third	0.079
Coarsest	0.080

The result for "Cat" and "Torus" given show that the quality of the 3D Delaunay mesh is not satisfactory, but the simplification algorithm discussed in this paper improves the quality.

IV ACKNOWLEDGMENTS

This work was supported by TÜBİTAK under contract EEEAG-198E011

V. REFERENCES

[1] D. Hearn and M. P. Baker, *Computer Graphic* , 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1997.

[2] J. Hartman and J. Wernecke, *The VRML 2.0 Handbook*. Reading, MA: Addison Wesley, 1996.

[3] P. S. Heckbert and M. Garland, "Multiresolution modeling for fast rendering," in *Proc. Graphics Interface '94*, Banff, Alta., Canada. May 1994.

[4] H. Hoppe, "Progressive meshes," in *Computer Graphics-Proc.* SIGGRAPH ' 96, 1996, pp. 99-108.

[5] M. Eck, T. DeRose, T. Duchamp, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in *Computer Graphics-Proc.* SIGGRAPH ' 95, 1995, pp. 173-182.

[6] J. R. Shewchuck, "Tetrahedral mesh generation by Delaunay refinement," in *Proceedings of the fourteenth annual symposium on computational geometry*, Association for Computing Machinery, June 1998, pp. 86-95.

[7] I. Celasun, E. Ilgaz, A. Murat Tekalp, P. J. L. van Beek and N. Zhuang, “Optimal Hierarchical Design of 2D Dynamic Meshes for Video”, ICIP98, Chicago.

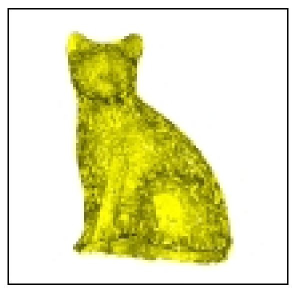

Figure 2 - Original volume graphics model-3D Delaunay Mesh l_0 , alpha = 0.0223, total vertex number = 10000 boundary vertex number = 1973

Figure 3 - 3D Delaunay Mesh $l_1$alpha = 0.028938, removed interior vertex number = 3785, removed boundary vertex number = 1221

Figure 4 - 3D Delaunay Mesh l_2 alpha = 0.061, removed interior vertex number = 1304, removed boundary vertex number = 491

Figure 5 - 3D Delaunay Mesh $l_3$alpha = 0.098, removed interior vertex number = 810, removed boundary vertex number = 305

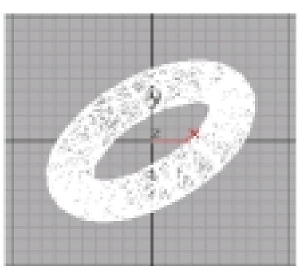

Figure 6 - Original volume graphics model - 3D Delaunay Mesh l_{0} , alpha = 19.6261 ,total vertex number = 288 , boundary vertex number = 180

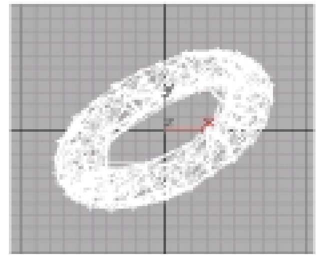

Figure 7 - 3D Delaunay Mesh l_1 , alpha = 37.649 , removed interior vertex number = 25 ,removed boundary vertex number = 25

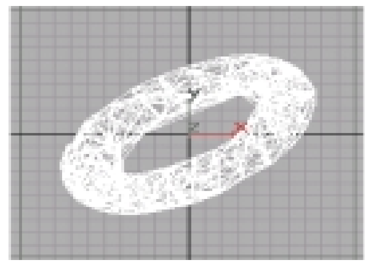

Figure 8 - 3D Delaunay Mesh l_2, alpha=37.5246, removed interior vertex number=18, removed boundary vertex number= 5

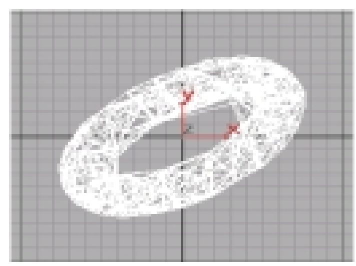

Figure 9 - 3D Delaunay Mesh l_3 , alpha=41.802 , removed interior vertex number=14 , removed boundary vertex number=1

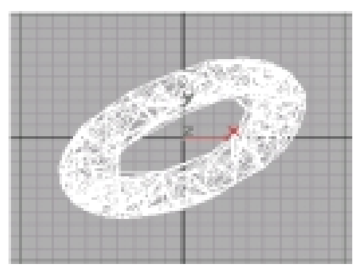

Figure 10 - 3D Delaunay Mesh l_4 , alpha=41.832 , removed interior vertex number=11 ,removed boundary vertex number= 0