Optimal Design of 2D/3D Hierarchical Content-Based Meshes for Multimedia

Işıl Celasun[1], Rupen Melkisetoğlu[1], and A. Murat Tekalp[2]

[1] Dept. of Electronics and Communication Eng.,
Istanbul Technical University, Maslak,
Istanbul, 34469, Turkey

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

Abstract. This paper proposes and compares methods for designing hierarchical 2D meshes for representation of object-based video and hierarchical 3D meshes for 3D objects used in telemedicine and multimedia applications. The same approach has been applied both in 2D and 3D but with different constraints. This representation consists of a hierarchy of Delaunay meshes, obtained by recursive simplification of the initial fine level-of-detail mesh geometry. There is no guarantee of an optimal mesh in 3D that uses a specific given set of node points whereas in 2D it is guaranteed that there is a unique 2D Delaunay mesh which uses all the node points for a specific set. To solve this problem an optimized alpha value is used in 3D Delaunay triangulation in the proposed algorithm. Mesh simplification entails removal of mesh nodes to reduce the level of detail. The selection of nodes to be removed is achieved by associating a cost with each mesh node. The Delaunay topology constraint on each mesh level not only helps to design meshes with desired geometric properties, but also enables efficient compression of the mesh data for multimedia applications.

1 Introduction

Dynamic mesh representation of video objects has recently been proposed for object-based video coding (by warped motion compensation), storage (e.g. video database query by motion) and manipulation (e.g. video animation). A 2D triangular mesh is initially designed on the first video object plane of a video object sequence, and subsequently tracked by motion estimation techniques. Hence, a 2D dynamic mesh compactly represents the shape and motion of a video object. We have recently proposed a hierarchical mesh-based representation of object-based video [1]. Hierarchical mesh representation has attracted attention recently, because it provides rendering at various level of detail (quality scalability), and allows progressive/scalable transmission of the mesh geometry and motion. Also, the hierarchical representation leads to improved motion track-

E. Andres et al. (Eds.): DGCI 2005, LNCS 3429, pp. 45–55, 2005.

ing performance [1]. This paper discusses optimal hierarchical construction of content-based 2D and 3D dynamic meshes.

A hierarchy of fine-to-coarse meshes can be obtained by repeated removal of some detail information from an initial mesh. A 2D mesh partitions the image domain into triangular patches, the vertices of which are referred to as *node points*. The straight-line segments between node points are referred to as edges. The *degree* of a node is the number of incident edges to the node. In the initial fine level-of-detail mesh design, node points are placed at salient edge and corner points of the boundary and interior of a video object plane. The number of nodes in the initial mesh can be found using the number of points detected by a corner detection algorithm. The initial mesh topology is constructed using constrained Delaunay triangulation of the node points, where the boundary edge segments serve as constraints. The Delaunay property of the initial mesh is preserved during mesh simplification by the constraint that only non-adjacent nodes can be removed, i.e., removed nodes must form an *independent* set. Each removed node leaves a hole in the mesh which is retriangulated using the Delaunay criterion.

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 [2]. 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.

Hierarchical representation of 3D 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 bit stream. Hierarchical representation of 3-D meshes has been addressed in computer graphics for adaptive level-of-detail (LOD) rendering of 3-D objects. A wavelet-based multiresolution mesh approximation was proposed in [3, 4, 5]. 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 new hierarchy of 3D Delaunay meshes and we only remove vertices in the fine to coarse design strategy. We do not reposition vertices nor edges which increase efficient use of bandwidth when compared to other methods mentioned so far.

In Section 2 and 3, algorithms for 2D/3D mesh simplification with their proper design parameters respectively are explained. In Section 4, experimental results for proposed decimation algorithms and conclusions are given.

2 Two Dimensional Hierarchical Mesh Representation and Design

A hierarchy of fine to coarse level meshes M_ℓ is defined, with $\ell = 0, 1, \ldots, L-1$, where each coarser level mesh is obtained by removal of some detail information from the finer level mesh. Thus, a hierarchical decomposition of the mesh data into L levels is obtained, consisting of a base level and $L-1$ enhancement levels, suitable for progressive transmission of a 2D dynamic mesh. The base level mesh contains a coarse version M_{L-1} of the original mesh, while each enhancement level contains detail information such that finer versions $M_{L-2}, \ldots, M_0$ of the mesh can be reconstructed, where M_0 is the original mesh. Hierarchical mesh representation and mesh simplification have been addressed in computer graphics for adaptive level-of-detail rendering of 3D objects. Hierarchical mesh modeling of video has been addressed previously in the context of uniform topology only.

2.1 Delaunay Hierarchy

Here, the decomposition of the initial mesh is restricted such that each mesh M_ℓ reconstructed at level ℓ has Delaunay topology. Thus, no topology data has to be encoded on any level of the decomposition. Only mesh node locations need to be encoded.

A simplification step starts by removing an *independent set* of nodes from the mesh at a certain level. An independent set of nodes is a set of nodes among which no two nodes are connected to each other by an edge. Edges that are incident to a removed node are removed as well. Each removed node of the mesh leaves a hole in the mesh which is to be retriangulated using the Delaunay criterion. It has been noted by de Berg et al. [7] that, after locally retriangulating the areas around removed independent nodes, the global triangulation remains Delaunay. The nodes that were removed in this step form the detail information to be encoded in an enhancement layer. This removal step is iterated a number of times, thus defining L versions of the mesh from fine to coarse. This strategy leads to a well-defined hierarchy of triangles and Delaunay meshes for all levels $\ell = 0, \ldots, L-1$, in which part of the topology is preserved going from one hierarchical level to the next. The approach represents the middle ground between enforcing a strict triangular hierarchy as one extreme, and ignoring topology preservation completely by picking nodes to be removed freely as the other extreme. It is naturally desirable to retain important nodes while going from one level to the next, such that essential mesh features are not removed. In this paper, we determine the importance of a node adaptively using image-based and shape-based criteria. The nodes on a 2D mesh boundary, convey essential object *shape* information.

2.2 Removal of Mesh Boundary Nodes

A sequential simplification algorithm is used to remove boundary nodes going from one hierarchy level to the next. The algorithm uses a distance parameter

D_{max} to control the error in the polygonal boundary shape, which is increased from level to level. A candidate approximating edge segment, or chord, is drawn between an initial boundary node point and another node point, the chord node. Initially, the chord node is the second node on the boundary with respect to the initial node. For each node between the initial boundary node and the chord node, the distance d from the candidate node to the chord is computed and compared to D_{max}. If d is smaller than D_{max} for all candidate node points, the next node on the boundary becomes the new chord node and a new chord is drawn. If d is greater than or equal to D_{max} for one or more candidate nodes, the candidate node with maximum distance d is retained and all nodes between this node and the initial node become removable nodes. The node that is retained becomes the initial node in the next step. Note that only an independent subset of the removable nodes is actually removed, so as to conform to the hierarchy discussed in the previous subsection. These steps are repeated until all nodes on the boundary have been processed.

2.3 Removal of Mesh Interior Nodes

The set of interior nodes to be removed while going from one hierarchy level to the next can be determined using both spatial and temporal image-based criteria. That is, one desires to retain nodes with locally important motion activity, as well as nodes that are salient in terms of "edgeness" and "cornerness" measures. The following measure of the saliency of node n is used:

$$C_n = \left|I_x\left(\vec{p}_n\right)\right|^2 + \left|I_y\left(\vec{p}_n\right)\right|^2 + \mathit{\Gamma}(\vec{p}_n) \tag{1}$$

where $\vec{p}_n(x_n, y_n)$ is the location of node n, $I_x\left(\vec{p}_n\right)$ and $I_y\left(\vec{p}_n\right)$ are partial derivatives of the intensity image at $\vec{p}_n$, and $\mathit{\Gamma}(\vec{p}_n)$ is the response of a corner detector at $\vec{p}_n$, based on a non-linear measure of similarity between the image intensities in a neighborhood around $\vec{p}_n$. Here, the intensity image refers to the initial VOP of the video object from which the mesh was derived. Using this importance measure, interior nodes are removed using an iterative greedy algorithm. Initially, all nodes that were connected directly to a removed boundary node are marked. Then, during each iteration, the unmarked node with the smallest value of the importance measure is removed and all nodes connected directly to this node are marked. Repeat until there are no more unmarked nodes. Note that the marking of nodes ensures that the nodes that are removed form an independent set. We allow a node to be removed only if its degree (number of incident edges) is at most 6. One can still ensure that a certain percentage of nodes will be removed.

3 Three Dimensional Hierarchical Mesh Representation and Design

3.1 Initial Fine Detail Mesh

The 3D 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 3D Delaunay triangulation the situation is not so, since a measurement for optimality in 3D is not agreed on in the literature.

Computation of the "α" Parameter. A graph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V}$ is the set of vertices, $\mathcal{V}$=$\{$v$_0$, v$_1$, ... , v$_1$, ... , v$_{v-1}\}$, and $\mathcal{E}$ is the set of edges, $\mathcal{E}$=$\{$e$_0$, e$_1$, ..., e$_i$, ..., e$_{E-1}\}$. The α parameter used in the 3-D Delaunay tetrahedralization tool specifies the radius of the circumsphere of each tetrahedron. Only tetrahedrons lying within this circumsphere are allowed. If the α value is zero, the output of the tetrahedralizator is a convex hull. Usually the output is a tetrahedral mesh, but if α distance value is not to be set as zero, then output data consists of tetrahedra, triangles, edges, and vertices lying within the α radius. Optimal α 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 is used in the proposed decimation method to determine this optimal value. 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 α value. In fact, to preserve all tetrahedra present in the 3-D mesh, maximum circumradius of them has to be used to determine α 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 α value determination.

When used a *global* α, some regions in the mesh, as pointed in Figure 3 (a), (c), (e) cannot be filled. The reason is that, an optimal Delaunay triangulation in 3D using all given node points cannot be guaranteed 3D which is different from 2D case [6, 7]. These vertices cannot be located in the spheres by using global α value in the Delaunay triangulation topology. There may be non connected regions since α value does not guarantee a connected mesh. The α variable in the *local* Delaunay topology, is obtained by vertex density value calculated by scanning the number of neighbours of a vertex in a region. Optimization is realized consecutively by bounding the alpha variable with a minimum and maximum value. Region with a density of vertices with constrained Delaunay triangulation, and the boundaries of these regions are kept and after the application of the same process for other density value regions, these regions boundaries are connected. So, generally a Delaunay mesh is formed by connected submeshes which are constrained Delaunay triangulated.

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 $\mathcal{T}$(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$_i$. For each tetrahedron, the solid angle at each vertex 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. The solid angle (Ψ) of a tetrahedron is as:

$$(\Theta_{\mathrm{e}_1\mathrm{e}_2} : \text{dihedral angle at edge } \mathrm{e}_1\mathrm{e}_2) \quad \Psi_A = (\Theta_{AB}) + (\Theta_{AC}) + (\Theta_{AD}) - \pi \quad (2)$$

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:

$$\Theta_{AB} = \pi - |\arccos(\mathrm{face}_{ABC} \bullet \mathrm{face}_{ABD})| \quad (3)$$

A sequential simplification algorithm is again used as in 2D case for 3D too to remove boundary vertices going from one hierarchy level to the next.

Mesh Interior Simplification. Importance function assigns an importance value for each vertex in the mesh. If the vertex is an important vertex that should not be removed, his importance value is also large.The importance (IP_1) value for an interior node is defined as the ratio of the sum of its neighbors' tetrahedral volume to its tetrahedron volume. This IP_1 stress on the connectivity of the mesh and try to retain 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. 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. 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.

The importance value for an interior vertex, is defined as the multiplication of his degree with the ratio of the total volume of the tetrahedrons formed by its neighbors to its own volume. If the volume of a tetrahedron related to a vertex, is smaller then the volumes of the tetrahedrons related to its neighbors, this means that this vertex contains a detail

$$IP(n) = Degree(n) * \frac{\sum_{i=1}^{neighbors-1} volume(i)}{volume(n)} \quad (4)$$

information and removing will be deletion of this detail information.

4 Experimental Results and Conclusions

4.1 Results for Design of 2D Meshes

Initial fine level-of-detail content-based meshes were designed on initial Bream and Akiyo video object planes, as illustrated in Figure 1 (a) and Figure 2 (a) respectively. These fine level-of-detail meshes contain 165 nodes and 210 nodes, respectively. Successively coarser level-of-detail meshes, obtained using the simplification methods are illustrated in Figure 1 (b), (c), (d) and (e) for Bream,

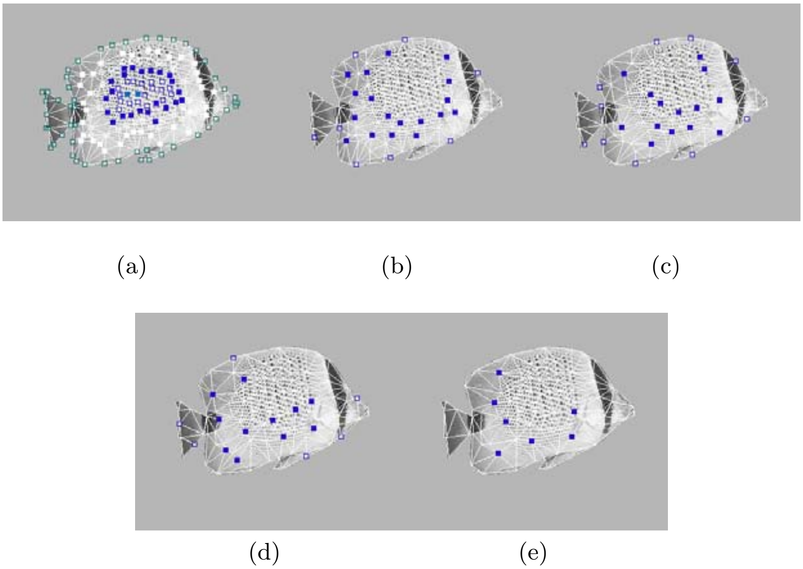

Fig. 1. (a) Illustrates six node layers of the original Bream mesh used in mesh simplification, by successive empty and filled squares. In (b), (c), (d) and (e), the hierarchical mesh for four consecutive levels of Bream is shown overlaid onto the original video object plane, where empty square symbols indicate the boundary nodes that will be removed at that hierarchy level and filled squares indicate the interior nodes to be removed at that level

Table 1. Number of removed boundary nodes and number of removed interior nodes at each level of the mesh hierarchy for the initial Bream video object plane

Hier. level	D_{max}	***# rem. boundary***	***# rem. interior***
1	0.8997	8	18
2	1.6246	10	14
3	1.8342	6	11
4	2.0613	0	7

Table 2. Number of removed boundary nodes and number of removed interior nodes at each level of the mesh hierarchy for the initial Akiyo video object plane

Hier. Level	D_{max}	***# rem. boundary***	***# rem. interior***
1	0.4662	4	32
2	1.3083	6	23
3	3.3609	6	19

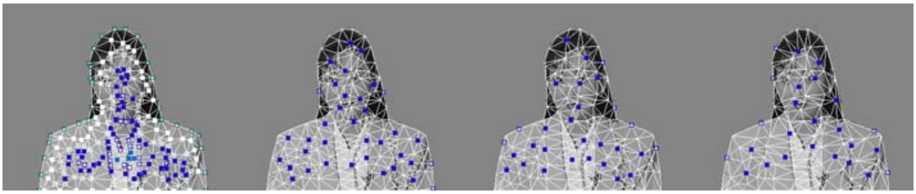

Fig. 2. (a) Illustrates six node layers of the original Akiyo mesh used in mesh simplification, by successive empty and filled squares. In (b), (c) and (d), the hierarchical mesh for three consecutive levels of Akiyo is shown overlaid onto the original video object plane, where empty square symbols indicate the boundary nodes that will be removed at that hierarchy level and filled squares indicate the interior nodes to be removed at that level

and in Figure 2 (b), (c) and (d) for Akiyo. The number of nodes removed from the boundary and from the interior of each 2D mesh at all hierarchy levels are summarized in Tables 1 and 2, for the Bream and Akiyo sequences, respectively.

4.2 Results for Design of 3D Meshes

Experimental results related to boundary and interior vertex simplification are given for "Cat" and "Engine" data. The proposed algorithm works in general for volume data but it can also provide successful results for surface data in terms of minimized number of vertices for a good mesh quality. Boundary node simplification is accomplished for $D_{max} = 2.0$ for "Cat". "Cat" 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". The stopping criterion is such that no more removed vertices on its boundary is applied. Global α value and *IP* are used for "Engine" and "Cat" in Figures 3-5. As seen from Figure 3, there are some gaps (nonconnected regions) at the output the rendered object. To get rid of this, local α values are used as in Figure 3-5 with *IP* calculation. Although number of vertices are very small in Figure 4 and 5, good results in terms of visual information and node simplification are obtained as given in Table 3 and 4 where two test results are also added.

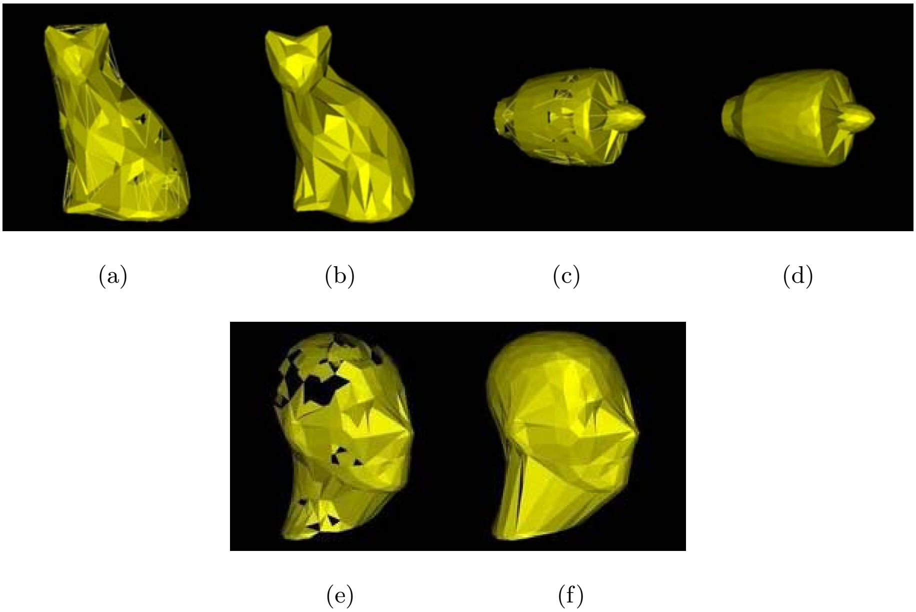

Fig. 3. “Results for 3D Delaunay meshes using global α are given in (a), (c), (e) and for 3D Delaunay meshes using local α s are given in (b), (d), (f) respectively for “Cat”, “Engine”, “Mannequin” data

Mesh quality criteria can be provided by a “Volume Test” and an “Angle Test”: First, ratio of inner radius (r) to outer (R) one for a tetrahedron cannot pass 1/3. In a sliver $r \to 0$ and $R \to \infty$, thus $\frac{3r}{R} \cong 0$. Using this ratio as a mesh quality measure, we can say that this mesh performance criterion is between 0 and 1. By calculating this ratio for each tetrahedron and taking the average value, we can determine the overall quality of the mesh at hand. This is called a “Volume Test”. Second, absolute value of the deviation of each angle of each triangle in the 3D mesh from 60 is calculated. Average of the deviation angles of all triangles in the 3D mesh is taken and “Angle Test” is thus calculated.

For both 2D and 3D hierarchical meshed only vertices have to be known, triangles are determined by Delaunay triangulation. The main difficulty with Delaunay triangulation in 3D is that its optimality is not proven. An optimal choice of alpha value controls this problem. In the proposed method, one only needs the vertices so a high transmission rate is possible for both 2D and 3D applications. Removal of maximum number of independent vertices will then provide high compression ratios by preserving mesh and thus rendered object quality.

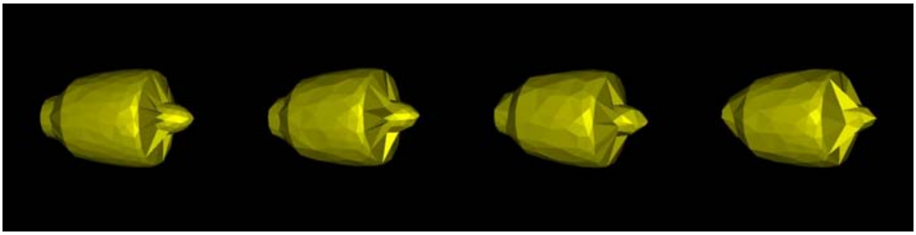

Fig. 4. (a) Original “Engine”graphics model with 426 vertices -3D Delaunay Mesh ℓ_0 (b) 3D Delaunay Mesh ℓ_1 hierarchy level 1 with 366 vertices (c) 3D Delaunay Mesh ℓ_2 hierarchy level 2, with 291 vertices (d) 3D Delaunay Mesh ℓ_3 hierarchy level 3 with 235 vertices

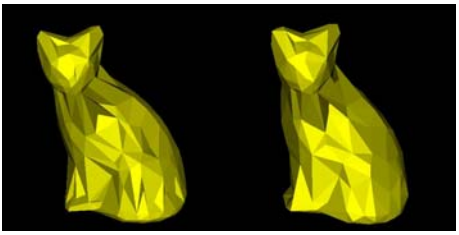

Fig. 5. (a) Original graphics model ”Cat” with 328 vertices,-3D Delaunay Mesh ℓ_0 (b) 3D Delaunay Mesh ℓ_3 hierarchy level 3 with 238 vertices

Table 3. Performance of the hierarchical design algorithm for “Engine” data as given in Figure 4 with IP calculation and local α values (V: vertices, B.V.: Boundary vertices)

	V	**B.V.**	**Volume Test**	**Angle Test**
Original	426	189	0.419	24.04
Hierarchy Level 1	366	161	0.419	23.85
Hierarchy Level 3	291	153	0.437	22.92
Hierarchy Level 5	235	144	0.462	22.09

Table 4. Performance of the hierarchical design algorithm for “Cat” data as given in Figure 5 with IP calculation and local α values (V: vertices, B.V.: Boundary vertices)

	V	**B.V.**	**Volume Test**	**Angle Test**
Original	328	149	0.284	28.02
Hierarchy Level 1	268	120	0.284	27.91
Hierarchy Level 2	238	120	0.294	27.45

Table 5. Local α values for "Engine" and "Cat" data for different hierarchy levels (E:Engine, C:Cat, ℓ_i:level i)

E ℓ_0	E ℓ_1	E ℓ_3	E ℓ_5	C ℓ_0	C ℓ_1
0.78	0.81	0.87	1.08	0.11	0.12
1.34	1.37	1.48	0.94	0.18	0.13
1.33	1.36	0.98	1.60	0.16	0.19
1.23	1.25	1.11	1.44	0.12	0.19
0.95	0.96	1.24	1.25	0.15	0.17
1.03	0.88	1.39		0.14	0.12
1.13	1.15			0.18	0.16
0.86	1.05			0.15	0.17
				0.17	0.14
				0.12	0.15
				0.13	

References

1. van Beek, P., Tekalp, A. M., Zhuang, N., Celasun, I., Xia, M.: Hierarchical 2-D mesh representation, tracking, and compression for object-based video. IEEE Trans. on CSVT 9, (1999)
2. Hearn D., and Baker M. P.: Computer Graphics, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, (1997)
3. P. S., Heckbert, M., Garland: Multiresolution Modeling for Fast Rendering, in Proc. Graphics Interface '94, Banff, Alta. (May 1994)
4. Hoppe H.: Progressive Meshes, in Computer Graphics-Proc. SIGGRAPH '96, (1996), 99–108
5. Eck M., DeRose T., Duchamp T., Stuetzle W.: Multiresolution Analysis of Arbitrary Meshes, in Computer Graphics-Proc. SIGGRAPH '95, (1995), 173–182
6. Shewchuck, J. R.: Tetrahedral mesh generation by Delaunay refinement, in Proceedings of the fourteenth annual symposium on computational geometry, Association for Computing Machinery, (June 1998), 86–95
7. de Berg, M., van Kreveld, M. , Overmars, M., Schwarzkopf, O.: Computational Geometry-Algorithms and Applications, Springer, 1997