ACCV2002: The 5th Asian Conference on Computer Vision, 23--25 January 2002, Melbourne, Australia.

Triangular Mesh Segmentation Based On Surface Normal

Dong Hwan Kim
School of Electrical Eng.
Seoul Nat'l Univ.
Seoul, 151-742, Korea
donggory@diehard.snu.ac.kr

Il Dong Yun
School of Electronics and Inform. Eng.
Hankuk Univ. of F.S.
Yongin, 449-791, Korea
yun@hufs.ac.kr

Sang Uk Lee
School of Electrical Eng.
Seoul Nat'l Univ.
Seoul, 151-742, Korea
sanguk@sting.snu.ac.kr

Abstract

This paper presents an algorithm for segmentation of 3D triangular mesh data. The proposed algorithm uses iterative merging of adjacent triangle pairs based on the orientation of triangles. The surface is segmented into patches, where each patch has a similar normal property. Thus, each region can be approximated to planar patch and its boundaries have perceptually important geometric information of the entire mesh model. The experimental results show that the proposed algorithm is performed efficiently.

1. Introduction

In computer vision and computer graphics, polygonal models are often used for efficient representation of the individual objects that are acquired by laser scanners and so on. Simplex polygons, *i.e.*, triangles, are used primarily because they are easy and efficient to render. However, in many cases, meshes that consist of such polygons have no explicit higher-level structure. One way to impose a higher level structure on such a mesh is to segment it into a set of connected pieces, which themselves have relationships to other pieces. This process is called *mesh segmentation*. In mesh segmentation, each boundary of the segmented pieces contains important topological information of entire mesh and becomes a kind of 3D-edge. These 3D-edges can be used in 3D feature detection and recognition. Also, the result of mesh segmentation can be applied to multi-resolution surface visualization technique, which is also known as LOD (Levels-Of-Detail) problem and is used to render the complex mesh efficiently. As performing edge detection and segmentation is important for analytic and systematic understanding of an object in 2D image, mesh segmentation is very important for mesh understanding.

Until now, the research for the segmentation of 3D data has been generally focused on range data, *i.e.*, 3D points [1] – [5]. However, 3D range data is only image that has information of a viewing part of object from a specific direction. Thus, 2D segmentation technique can be easily applied because the range image is the same as 2D image of which pixel values are depth data. In contrast with 3D range image, 3D mesh data that has geometric information of whole object cannot be represented to 2D rectangular structure like 2D image. Thus, for segmentation of 3D mesh data it is needed that a new approach different from existing range image segmentation technique.

Mesh data can be segmented with various criteria according to the purpose of segmentation. Also, to judge whether the segmentation result is good or not depends on the purpose of segmentation. In this paper, mesh segmentation algorithm using surface normal is proposed for analytic and systematic understanding of mesh. The mesh model that consists of many triangles as shown in Fig. 1a is segmented as shown in Fig. 1b. Since each region consists of the triangles that have similar geometric property, now the input mesh can be considered as set of regions not triangles. Thus, we can consider it as 3 regions, *i.e.*, upper face, lower face and side face, and understand it analytically and systematically through connection information among the regions. Also, the mesh model can be segmented into planar patches as shown in Fig. 1c. The result can be easily applied to the application such as multi-resolution modeling technique. Since the proposed mesh segmentation algorithm supports these two different types of segmentation, it supplies the framework for

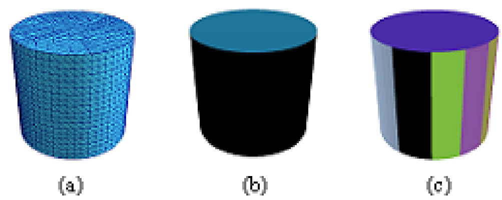

Figure 1. Mesh segmentation. (a) input mesh. (b) segmented mesh (3 regions). (c) segmented mesh (14 regions).

understanding mesh analytically and systematically.

The remainder of the paper proceeds as follows: The next section reviews some aspects of the literature that are related to the proposed algorithm. Section 3 explains how mesh model is segmented. In Section 4 experimental results are represented and final concluding remarks are given in Section 5.

2. Related Works

The method of mesh segmentation is dependent on its purpose. Yan *et al.* [6] proposed a mesh partitioning algorithm for coding of 3D graphic models. They proposed the algorithm that repeatedly merges adjacent triangle not indexed yet into the present region. However, the purpose of this algorithm is to divide the whole mesh into a set of small regions so that each region can be encoded and decoded independently. Also, this segmentation algorithm cannot extract higher-level geometric information of entire mesh, and the size of each partitioned part is determined by the channel error rate that has no concern with the shape of the mesh. Thus, this segmentation is not fit for applications where geometric information of the model is important, like the LOD. Cutzu [7] proposed an algorithm for shape segmentation. In this algorithm, The number of parts and images projected from several views of the object are needed. Thus, this segmentation is not fit for mesh segmentation using only input mesh itself.

However, Mangan and Whitaker [8] so performed mesh segmentation that each region contains the shape information of model. They generalized watershed algorithm used in image segmentation and applied it to 3D mesh data. In this case, the adjacent vertices of a surface vertex become the neighbors connected to the vertex with one edge. In application of watershed algorithm they chose top-down approach and use the curvature as the height function. However, the watershed algorithm is sensitive to even the smallest fluctuations in surface shape, and every local minimum of curvature establishes its own region. Thus, it is necessary to merge the regions together in order to avoid over-segmentation. They proposed the algorithm that merges the regions of which watershed depth is below a given threshold. Since this algorithm uses the curvature of the mesh surface, it can segment the model that has uniform curvature only as shown in Fig. 2. However, this algorithm cannot segment it into planar patches that are fit for multi-resolution modeling. Also, this algorithm is sensitive to noise because it uses the curvature.

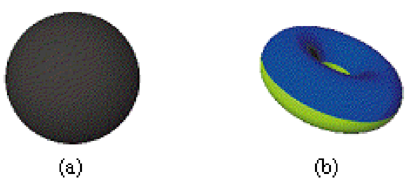

Figure 2. Mesh partitioning using watershed algorithm. (a) sphere (1 region). (b) torus (2 regions).

3. Proposed Algorithm

The proposed mesh segmentation technique partitions input mesh model into several regions that have similar geometric property for analytic and systematic understanding of it. The criterion by which adjacent triangles are merged into same region uses surface normal of the mesh model.

The proposed algorithm has the advantage that it supports the option for various segmentations according to the aim of application. While the segmentation algorithm using surface curvature proposed by Mangan and Whitaker [8] can segment a object which has uniform curvature like sphere into only one region, the proposed algorithm can segment it into one region or several approximated planar patches by only adjusting control parameters. Also, error of the approximated planar patches can be controlled, therefore the proposed mesh segmentation algorithm has another advantage that it can be easily applicable to multi-resolution modeling technique that requires various mesh representation according to the resolution. The block diagram of the proposed algorithm is shown in Fig. 3.

3.1. Iterative Triangle-Merging

The proposed mesh segmentation technique using

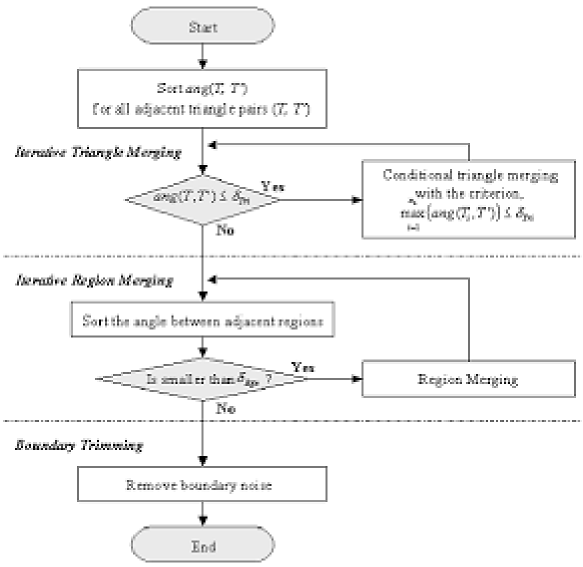

Figure 3. Block diagram of the proposed algorithm.

surface normal of the mesh merges adjacent triangles that have similar geometric property by iteration. For this, in this paper the triangle-merging operator for adjacent triangle pair is proposed. The proposed algorithm merges adjacent triangles using triangle-merging operator at each stage of iteration. Thus, under given constraints the proposed segmentation technique always yields the unique segmentation result in contrast with the segmentation technique that extends each region from a seed triangle.

The triangle-merging operator, first, compares the angle between the surface normals of the two adjacent triangles with given error bound. Where the error means the angle difference of the surface normal of the triangles indexed to the same region, and the bound is given by angle threshold. The angle threshold is determined experimentally according to the resolution of the input mesh model. In high resolution the small value of angle threshold will be sufficient, however, in lower resolution larger value is needed. When the angle between the normals of the two triangles is beyond the error bound, the two triangles are separated into other regions, respectively and the edge between the two triangles becomes the boundary of the two regions. Otherwise, the segmentation is performed in following cases :

- case 1) The two triangles are not indexed yet :
  The two triangles are merged into new same region.

- case 2) One triangle is indexed and the other is not indexed yet :
  Let T and S denote a triangular face and a segment, respectively. Let T' denote a triangle that is not indexed yet and kth segment S_k denote the segment that includes the other indexed triangle. Assume that $S_k = \{T_1, T_2, \ldots, T_{n_k}\}$. T' is merged into S_k when the following decision rule is satisfied.

$$\max_{T_i \in S_k}^{T_{n_k}} \{ang(T_i, T')\} \le \delta_{Tri} \quad (1)$$

  where δ_{Tri} is the angle threshold and the function $ang(T_I, T')$ means the angle between the normals of triangle T_i and T'. If triangle T' does not satisfy the decision rule, Eq. (1), T' becomes a new region and the edge between the two triangles becomes the boundary of the two regions.

- case 3) Each triangle is indexed and each indexed region is different each other :
  Eq. (1) is applied to each triangle and all triangles included in each region. When the two triangles satisfy the decision rule, Eq. (1), the two regions are merged into same region. Otherwise, they are not merged.

In iterative triangle-merging algorithm, the triangle pair where the angle between the surface normals is the smallest of all adjacent triangle pairs', is selected and where triangle-merging operator is applied at each stage of iteration. Thus, the unique segmentation result is always yielded under given error bound.

If we aim to segment the object of which surface normal is continuously changed like sphere into one region, following equation substitutes for Eq. (1).

$$ang(T_i, T') \le \delta_{Tri} \ . \quad (2)$$

We call the segmentation in this case the segmentation without local shape constraint (LSC) and denote it as LSC OFF. LSC ON means the segmentation with local shape constraint.

3.2 Iterative Region-Merging

The iterative region-merging algorithm performs the segmentation by merging the adjacent triangles that have similar surface normal efficiently. However, in the segmentation under LSC ON mesh surface is partitioned into so many regions, because the proposed algorithm under low error bound is sensitive to small fluctuations in surface shape. Thus, it is needed that the process that merges the small over-segmented regions into adjacent regions that have similar normal.

In the proposed mesh segmentation algorithm, to obtain the optimum segmentation result under given error bound by iterative region merging, the region-merging operator is adopted. The region-merging operator merges two adjacent regions into one same region when the angle between the normal of the two regions is in given error bound. We denote the angle threshold that determines error bound in region-merging operator as δ_{Rgn} and determine the value of it experimentally according to the resolution of input mesh model. In iterative region merging, first, principle normal of each region is calculated, and region-merging operator is supplied to the region pair where the angle between the mean normal of the adjacent regions is the smallest of all adjacent region pairs, at each stage of iteration.

3.3 Boundary Trimming

In the proposed mesh segmentation algorithm, iterative triangle-merging and iterative region-merging processes are performed in succession, and they yield the optimum segmentation result under the given error bound. However, in the segmentation under LSC ON the boundary of a region is sometimes degraded by noise when the region includes a triangle with relatively high normal error as compared with other triangles of the region. For a triangle that composes the boundary, if we consider the angle

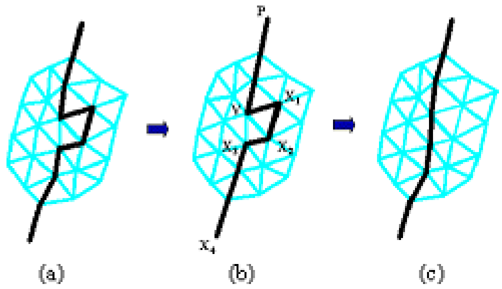

Figure 4. Bopundary trimming. (a) noisy boundary. (b) extraction of segment-edge. (c) noise-removed boundary.

between the normals of the triangle and each adjacent triangle, sometimes, it can be right that the triangle is included in the adjacent region. Because of a triangle with high surface normal error, the triangle did not satisfy Eq. (1), therefore it separated into other region. Also, in case that a triangle is separated into other region but it is reasonable to merge it into the same region, the triangle is thought to be a noise. The purpose of boundary trimming in the mesh segmentation is, to remove the noise in the degraded boundary and describe the shape of the entire mesh efficiently using the boundary information of each region. The noisy boundary is such a part as the protrusion of the boundary represented by black line in Fig. 4a. The end of boundary trimming is to correct boundary as shown in Fig. 4c.

Before describing the proposed boundary trimming algorithm, we describe the several terminology first. *Segment-vertex* is defined as the vertex that determines the overall shape of the region of all vertices that compose the boundary of the region. *Segment-edge* means the edge that connects the two adjacent segment-vertices. The segment-vertex is determined by *ABV(Angle of Boundary Vertex)* of each boundary vertex. The ABV is defined at each boundary vertex and it means the angle between the two connected edges, as shown in Fig. 5a. In the process of extraction of the segment-vertex, it is applied through modification that the mesh simplification technique using the edge-based vertex-remove operator originally proposed by Choo *et al*. [9]. The segment-vertex extraction algorithm is as follows.

(i) Let the all boundary vertices of each region be included in candidate set.

(ii) Of the boundary vertices between the only two regions, the vertex of which ABV is the closest to 180° is selected. When it satisfies following Eq. (3), it is removed from candidate set and ABV of the adjacent vertices is updated as shown in Fig. 5b.

$$\left|180^{\circ} - ABV\right| \le \delta_B \tag{3}$$

where δ_B is the angle threshold and is determined in the range of 10 – 20° according to the resolution of input mesh model.

(iii) The step (ii) is repeated until no vertex in candidate set satisfies Eq. (3). Finally, the remained vertices in candidate set are the segment-vertices.

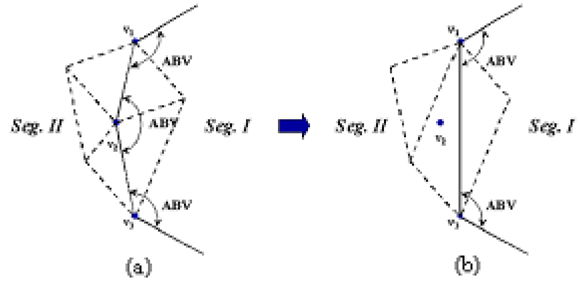

Figure 5. Extraction of segment-vertex and segment-edge.

In the proposed boundary trimming algorithm, first, segment-vertices and segment-edges are extracted in each region. Then the noisy boundary is detected by examination of the direction of each segment-edges along the segment-edges. For example, assume that the segment-vertex V in Fig. 4b is investigated. Each X_n is a neighbor of vertex V in n-th order neighborhood structure. Where the order of the neighborhood structure means the minimum number of the edges that connect the two vertices V and X. In this case, n, the order of the neighborhood structure is increased, whether the angle between the segment-edges VP and VX_n is close to 180° or not, is investigated. The inner product form of this criterion is as following equation.

$$\left\langle \overrightarrow{VP}, \overrightarrow{VX_n} \right\rangle \le \cos\left(180^{\circ} - \delta_B\right) \tag{4}$$

where δ_B is the same value in Eq. (3). The boundary noise is removed as shown in Fig. 4c by merging the noisy part, $VX_1 \ldots X_n$ into adjacent region when the segment-vertex X_n that satisfies Eq. (4) is neighbor within n-th neighborhood structure. However, if the segment-vertices, $X_1, \ldots, X_{n-1}$ are not the boundary vertices between only two regions, the noisy part can not removed.

In this paper, we limit the maximum order of neighborhood structure to 3. In case of more than 3, we cannot consider that the boundary is not degraded by noise, even though Eg. (4) is satisfied. Thus, the boundary is considered as important segment-edge that determines the shape of the region, and is not removed.

4. Experimental Results

To evaluate the performance of he proposed mesh segmentation algorithm, experiments carried out on the icosahedron model added Gaussian noise of which standard deviation is determined by following equation.

$$\sigma = D_m \times \frac{X}{100} \quad (5)$$

where D_m is mean edge-distance of the input mesh model, X is percentage.

Since mesh models have been generally obtained from CAD, they have been usually considered as the model without noise. However, in rescent years mesh models are often acquired directly by laser scanner. Since the laser scanner can obtain the data of the only visible part of the object from the specific view direction, it is necessary to register and integrate the partial data obtained from the different view directions one another. The mesh model is generated through these processes. However, the cumulative error in each process becomes a kind of noise in generating mesh. As a result, the mesh model also can be degraded by noise. Thus, experiments for noisy mesh models are needed.

The icosahedron model shown in Fig. 6a consists of 2,562 vertices and 5,120 triangles with 10% Gaussian noise. Fig. 6b shows the result of iterative triangle-merging under LSC ON and δ_{Tri}=20°. Then, the icosahedron was over-segmented into total 1256 regions. When iterative region-merging under δ_{Rgn}=30° is performed on the result, small regions are merged into total 54 regions as shown in Fig. 6c. Fig. 6d is the result of boundary trimming on the output of iterative region merging, and the number of regions is 22. The results show that the icosahedron model is segmented as our perceptual understanding even though the model is degraded by noise.

Fig. 7, 8 show the segmentation results on the fan model. Fig. 7a shows the fan model used in experiments. It consists 1,375 vertices and 2,750 faces. Fig. 7b shows the results of iterative triangle merging under LSC ON and δ_{Tri}=20°. Then, the fan model was over-segmented into total 313 regions. The results of iterative region-merging and boundary trimming are shown in Fig. 7c, 7d, respectively. Then, the fan model is segmented into 89, 68 regions, respectively. Fig. 8a shows the results of iterative triangle-merging under LSC OFF and δ_{Tri}=20°. Then, the fan model was segmented into 31 regions. Since the local shape constraint is off, the result, in contrast with the results in Fig. 7., shows that the surfaces of the wings and the central cylinder are merged into only one region, respectively. Fig. 8b is the result of the iterative region-merging under δ_{Rgn}=35° and boundary trimming, performed on the result in Fig. 7b. Then the number of regions is 38. Since the error bound is increase, the wings of the fan model are merged into only one region, respectively, in contrast with the result in Fig. 7d. Although the result is under LSC ON, it is similar to the result of the segmentation under LSC OFF shown in Fig. 8a.

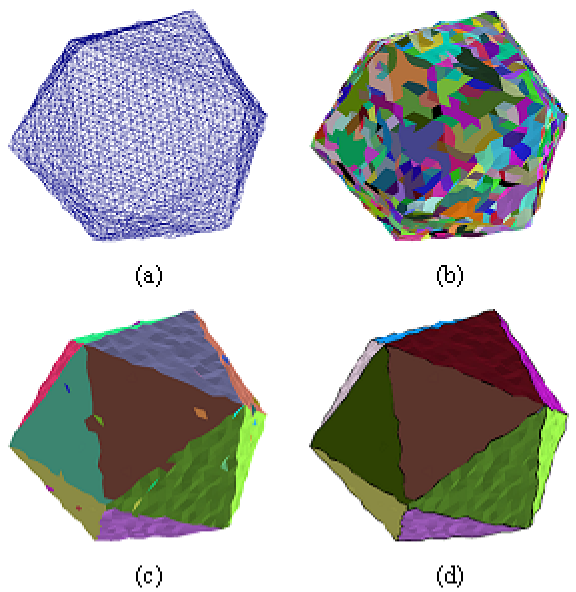

Fig. 6. Segmentation of icosahedron with 10% Gaussian noise. (a) input mesh model (2,652 vertices, 5,120 faces). (b) iterative triangle-merging under LSC ON and δ_{Tri}= 20° (1,256 regions). (c) iterative region-merging under δ_{Rgn}= 30° (54 regions). (d) boundary trimming (22 regions).

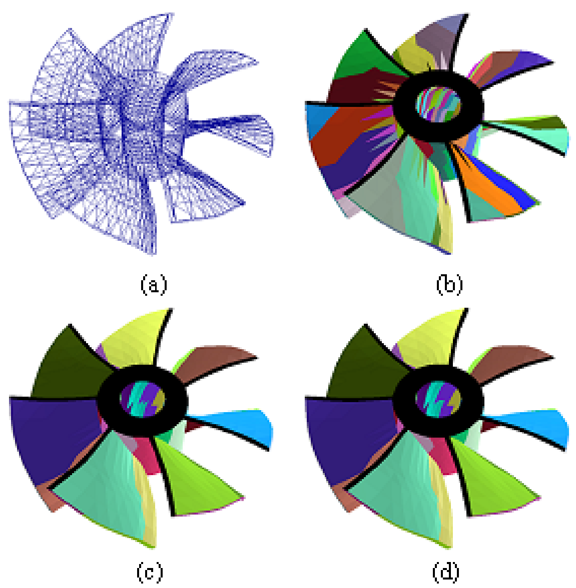

Fig. 7. Fan segmentations. (a) input mesh model (1,375 vertices, 2,750 faces). (b) iterative triangle-merging under LSC ON and δ_{Tri} = 20° (313 regions). (c) iterative region-merging under δ_{Rgn} = 25° (68 regions). (d) boundary trimming (68 regions).

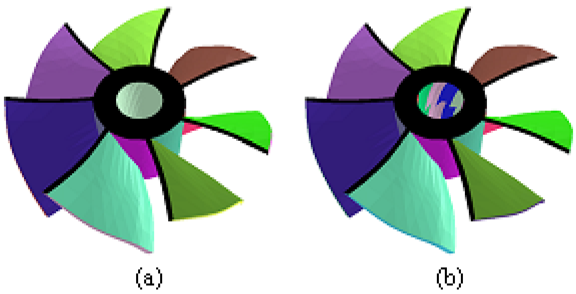

Fig. 8. Fan segmentation with various segmentation option. (a) iterative triangle-merging under LSC OFF and δ_{Tri} = 20 ° (31 regions). (b) segmentation under LSC ON, δ_{Tri} = 20 ° and δ_{Rgn} = 35 ° (38 regions).

In the proposed mesh segmentation algorithm, even the same mesh model can be segmented variously by adjusting control parameter of segmentation as shown in Fig. 7, 8. Thus, this advantage will make it possible to apply the proposed segmentation technique to various applications. Also, it will be helpful to understand the object in various points of view.

5. Conclusion

We have described the mesh segmentation technique using surface normal. In the proposed algorithm, the part where the geometric property of mesh surface changes steeply becomes a boundary of each region. Also, the segmentation results show that the output is similar to our perceptual segmentation.

The experimental results show that the mesh segmentation under LSC ON can be easily applied to multi-resolution modeling technique where the maintenance of the topological shape is important than any other thing in the process that yields simplified model in low resolution. In the proposed algorithm, since the boundaries of each region contain the shape information of entire mesh, the simplified model can be efficiently yielded by maintenance of the boundaries. Also, since the connection information among the regions segmented by mesh segmentation supplies the framework of analytic and systematic understanding, it is necessary that the problems such as mesh detection, mesh recognition and so on are investigated as future works.

6. References

[1] M. A. Wani and B. G. Batchelor, "Edge-Region-Based Segmentation of Range Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 314-319, Mar. 1994.

[2] R. Krishnapuram and S. Gupta, "Morphological Methods for Detection and Classification of Edges in Range Images," *Mathematical Imaging and Vision*, vol. 2, pp. 351-375, 1992.

[3] D. J. Langridge, "Detection of Discontinuities in the First Derivatives of Surfaces," *Computer Vision, Graphics and Image Processing*, vol. 27, pp. 291-308, 1984.

[4] P. Checchin, L. Trassoudaine, and J. Alizon, "Segmentation of Range Images into Planar Regions," *Proceedings of International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, May, 1997.

[5] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 673-689, Jul. 1996.

[6] Z. Yan, S. Kumar, J. Li, and C. -C. Jay Kuo, "Robust Coding of 3D Graphic Models Using Mesh Segmentation and Data Partitioning," *Proc. IEEE International Conference on Image Processing*, vol. 4. pp 352-356, Kobe, Japan, Oct. 1999.

[7] F. Cutzu, "Computing 3D Object Parts from Similarities among Object Views," *Proc. IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 95-100, South Carolina, Jun. 2000.

[8] A. P. Mangan and R. T. Whitaker, "Partitioning 3D Surface Meshes Using Watershed Segmentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 308-321, Oct.-Dec. 1999.

[9] K.W. Choo, I. D. Yun, and S. U. Lee, Edge-based Approach to Mesh Simplification," *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling*, pp. 368-377, Oct. 1999.