A Hybrid Approach to Feature Segmentation of 3-Dimensional Meshes

Anshuman Razdan [*] MyungSoo Bae[†]

November 26, 2001

Keywords: 3D Segmentation, Shape Recognition, Feature Extraction, Watershed Segmentation, Curvature Estimation, Triangle Mesh.

Abstract

Segmentation of a 3-dimensional (3D) polygonal mesh is a method of breaking the mesh down into "meaningful" connected subsets of meshes called regions or features. Several methods have been proposed in the past and they are either vertex based or edge based. The vertex method used here is based on the *watershed segmentation scheme* which appears prominently in the image segmentation literature and was later applied to the 3D segmentation problem[16], [33]. Its main drawback is that it's a vertex based method and no hard boundaries (edges) are created for the features or regions. Edge based methods rely on the dihedral angle between polygon faces to determine if the common edge should be classified as a *Feature Edge*. However, this method results in many disconnected edges and thereby incomplete feature loops.

We propose a hybrid method which takes advantage of both methods mentioned above and creates regions with complete feature loops. Satisfactory results have been achieved for both CAD parts as well as other laser scanned objects such as bones and ceramic vessels.

1 Introduction

The domain of the problem, called a *3D mesh* (denoted by M), consists of a set of n points (vertices $\mathbf{v}_i \in \mathbb{E}^3$; $0 \leq i < n$) and a set of planar convex polygons (faces) made up of these vertices. 3D meshes are currently a popular surface modeling primitive used extensively to represent real world and synthetic surfaces in computer graphics. In order to generate meshes, real world objects are often either digitized, i.e. point samples are collected from the surface of the object and their connectivity generated or sampled as a volume from which an

---

[*] PRISM, Arizona State University, Tempe AZ 85287-5906. razdan@asu.edu

[†] PRISM and Computer Sc. & Engineering, Arizona State University, Tempe AZ 85287-5906. msbae@asu.edu

isosurface is extracted. Alternately, the points (and their connectivity) are generated using computer programs, either interactively (e.g. using a surface design tool) or automatically (e.g. from a mathematical function). A mesh is a digital or *discretized* structure representing some *continuous* surface.

Segmentation means breaking down an existing structure into meaningful, connected sub-components. In the context of 3D meshes, the sub-components of the structure being broken down are sets of vertices called *regions* which share some "commonality". 3D mesh segmentation has had numerous applications in the past. Moreover, several existing applications benefit from segmentation. Good examples are feature extraction, decimation, adaptive subdivision[1], and surface fitting for reverse engineering. The motivation for this research was provided by two application domains: optimized triangulated meshes in CAD (Reverse Engineering, Manufacturing, etc.) and over sampled meshes from laser scanned data such as ceramic vessels and bones. Since curvature is not scale invariant, we scale all input files to a unit-bounding box.

2 Related Work

Segmentation of 3D objects (solids, surfaces, etc.) has received considerable attention for some time. A significant amount of work has been done for feature recognition of Solid Models. Henderson[35] developed rules to extract design features such as keys, slots, and protrusions from solid model data. Kim[34] and Woo[36] took the volumetric approach which can be computationally expensive and, in some cases, does not converge. Razdan ([41], [42]) used a feature based approach to decompose a solid model volume for finite element analysis. Gadh ([39], [37], [38]) uses a viewpoint based approach to create differential depth filters and deduce features from silhouette edges. Rule based approaches work well on the Solid Model data where the topological relationships are well defined. For example, in a cylinder there is one topological surface or face that defines the cylindrical region, although geometrically it may be represented as more than one surface entity. Another approach used is graph matching. This is computationally expensive. In a more generalized sense, where the underlying topology (other than mesh connectivity) is not known, the above methods would be either computationally very expensive or simply ineffective.

Sonthi et al. [40] proposed a curvature region approach where the surface(s) comprising the 3D object are labeled as protrusions, depressions or flat regions based on Gaussian curvature. Neighboring regions define the interconnects or edges further divide them into more specific regions. There are two reasons why this approach is not effective on general polygonal meshes. First, they assume a Solid Model (BREP) as input data and therefore, well-defined geometry and topology including smooth surfaces. Polygonal meshes are anything but smooth. Second, since the data is derived from a Solid Modeler, Gaussian curvature is easily computed. Computing Gaussian (discrete or continuous) curvature is problematic for polygonal meshes. Also, Gaussian curvature is not the best property to determine shape behaviour[33].

We first present two approaches that have been used for segmentation of polygonal meshes and then present our method.

3 The Watershed Algorithm

A value λ_i could be associated with each vertex $\mathbf{v}_i$ in the data set which somehow encapsulates the characteristics of the locality of the vertex. The definition of segmentation is one in which regions consist of connected vertices which have the same λ (within a tolerance). Curvature is selected as the mathematical basis for region separation, i.e. the scalar value λ. Curvature estimation from 3D meshes is dealt with in Hamann [11], Calladine[29] and Kobbelt [14], who extract curvature from a locally fitted quadratic polynomial approximant; and Besl [3], Hoschek [30], and Mangan and Whitaker [16], who describe various discrete curvature approximation schemes. Vertices having the same curvature value would be grouped into regions, separated by vertices with high curvature (which serve as region boundaries). An improved curvature estimation scheme is presented in [33].

The segmentation scheme used is derived from the *Watershed* algorithm for 3D meshes described in [16]. However, there has been considerable research work relevant to this problem based on various other techniques (see ang and Lee [28], Fan et al. [8], Sapidis and Besl [19], Besl [3], and Hoffman and Jain [12]).

The Watershed algorithm for image segmentation described in Serra [20] is a classic in the 2D image segmentation field. Eberly [7] has a good discussion on ridges and watersheds, with emphasis on practical definitions. Sijbers [22], in his PhD thesis, uses a 3D analog of Vincent and Soille's [26] watershed algorithm by assuming that the crest surfaces of the gradient magnitude of the 4D relief correspond to structure surfaces of the original 3D data.

Mangan and Whitaker [16] generalize the watershed method to arbitrary meshes by using either the discrete Gaussian curvature or the norm of covariance of adjacent triangle normals at each mesh vertex as the *height field*, analogous to the pixel intensity on an image grid which drives the 2D watershed segmentation algorithm.

This section briefly outlines the watershed algorithm. For details we refer to the literature cited in the original publications. Once the curvature κ_i at each vertex $\mathbf{v}_i$ in the mesh has been computed and stored, the segmentation can begin. κ_i here represents a generic curvature metric and can be Gaussian, absolute, mean, RMS, etc. During segmentation, a label will be assigned to each vertex $\mathbf{v}_i$ in the mesh indicating the region to which the vertex belongs.

Convex areas (elevations) on the object have a high magnitude of curvature with a positive sign, while concave areas (depressions) have a high magnitude of curvature with a negative sign (except for curvatures which are positive by definition). High positive curvature would appear as a ridge on the curvature distribution function, while a high negative curvature would manifest itself as a valley. Since the original watershed segmentation algorithm segments the surface into regions separated by areas of high curvature, the valleys would not act as region separators as would be expected (see Fig. 1(a)). We consider only the magnitude of curvature (Fig. 1(b)) to solve this problem.

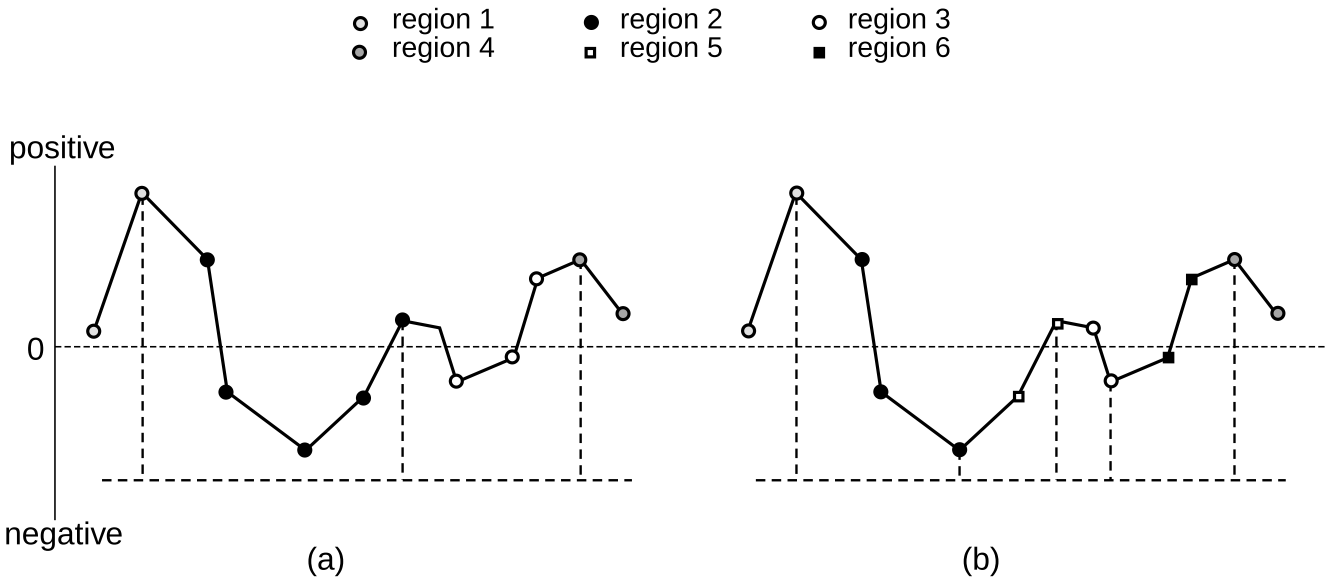

Figure 1: Considering the curvature sign may yield an incorrect segmentation.

3.1 Surface Curvature

The quality of results from the watershed segmentation algorithm depend substantially on the accuracy and stability of the estimated curvature values. Curvature at the mesh vertices should faithfully reflect the local properties of the underlying surface. This section therefore briefly describes the various terms in the theory of surface curvature. The following are sources of the material covered below: Besl [3], DeRose [6], Farin [9], Hyde et al. [13], Stoker [23], Struik [24], and Willmore [27].

The *parametric* surfaces considered are of the form

$$\mathbf{x} = \mathbf{x}(\mathbf{u}); \qquad \mathbf{u} = (u, v) \in [\mathbf{a}, \mathbf{b}] \subset \mathbb{R}^2, \tag{1}$$

where u and v are parameters which take real values and vary freely in the domain $[\mathbf{a}, \mathbf{b}]$. The functions $\mathbf{x}(u, v) = (x(u, v), y(u, v), z(u, v))$ are single valued and continuous, and are assumed to possess continuous partial derivatives.

The *first fundamental form*, denoted by I is given by

$$I = \dot{\mathbf{x}}.\dot{\mathbf{x}} = Edu^2 + 2Fdudv + Gdv^2 \tag{2}$$

where

$$E = \mathbf{x}_u^2 = \mathbf{x}_u.\mathbf{x}_u, \quad F = \mathbf{x}_u.\mathbf{x}_v, \quad G = \mathbf{x}_v^2 = \mathbf{x}_v.\mathbf{x}_v. \tag{3}$$

The *second fundamental form*, denoted by II is given by

$$II = Ldu^2 + 2Mdudv + Ndv^2, \tag{4}$$

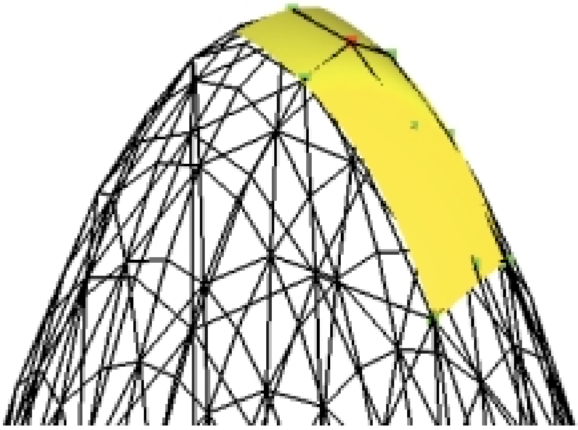

Figure 2: Figure showing local surface fit for curvature estimation.

where

$$L = \mathbf{N}\mathbf{x}_{uu}, \quad M = \mathbf{N}\mathbf{x}_{uv}, \quad N = \mathbf{N}\mathbf{x}_{vv}. \tag{5}$$

and $\mathbf{N}$ is the surface normal at point $\mathbf{x}$.

The *normal curvature* of the surface at point $\mathbf{x}$ in the direction of tangent $\mathbf{t}$ is given by

$$\kappa_0 = \kappa_0(\mathbf{x};\mathbf{t}) = \frac{II}{I} = \frac{(Lu')^2 + 2Mu'v' + (Nv')^2}{(Eu')^2 + 2Fu'v' + (Gv')^2} \tag{6}$$

Since the normal curvature is based on direction, it attains maximum and minimum values, called the principal curvatures. The principal curvatures, κ_1 and κ_2, can be combined to give us *Gaussian* curvature, given by

$$K = \kappa_1\kappa_2 = \frac{LN - M^2}{EG - F^2} \tag{7}$$

3.2 Other Types of Curvature

The principal curvatures can be combined in other useful and geometrically meaningful ways such as mean, RMS and absolute curvatures.

Mean curvature is given by

$$H = \frac{(\kappa_1 + \kappa_2)}{2} = \frac{1}{2}\frac{NE - 2MF + LG}{EG - F^2}. \tag{8}$$

The mean curvature, being the average of the principal curvatures, is less sensitive to noise in numerical computation than the principal curvatures.

Root mean square curvature(RMS) is a good measure of surface flatness and is given by

$$\kappa_{\mathrm{rms}} = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}}, \tag{9}$$

and can easily be computed as

$$\kappa_{\text{rms}} = \sqrt{4H^2 - 2K}. \tag{10}$$

A value of $\kappa_{\text{rms}} = 0$ at a point indicates a perfectly flat surface in the neighborhood of that point.

Absolute curvature is given by

$$\kappa_{\text{abs}} = |\kappa_1| + |\kappa_2|. \tag{11}$$

Mean and RMS curvatures have a closed form as given by equations (8) and (10) and these do not require actual computation of principal curvatures. κ_0 and κ_1 are expensive to compute and hence the cost of computing absolute curvature is considerably higher than computing mean and RMS counterparts.

Among continuous methods, mean curvature was found to be more resilient to noise in numerical computation since it averages the principal curvatures. Being an extrinsic property, mean curvature produced results closer to the expected results which were based on user perception of shape. RMS curvature has its advantages when dealing with specialized segmentation (see [2]). However, absolute curvature resulted in segmentations that generally outperformed all others. Like mean, the absolute curvature is the summation of (the absolute values of) κ_1 and κ_2 giving it greater noise resilience. It is positive by definition unlike mean and Gaussian curvatures.

3.3 No Hard Boundary Problem

A major drawback of the vertex-based method is that no hard boundaries are created for the features or regions. Each vertex of an object has its own region information. Therefore, triangles on boundaries have multi-region information. Figure 3(a) shows the boundary triangles in white. The three vertices of a triangle can be part of three different regions, whereas the triangle itself would be a "gray" area, i.e. it would not belong to any one region. This means the regions will not have hard boundaries or edges. We call this "*no hard boundary*" problem. To solve this problem, we create new triangles by adding points (mid-points) on the edges which have different region labels. Then, each new vertex contains multiple labels (Figure 4). This is an extension of the original algorithm[16]. For the selection of region label of a vertex which has multiple labels, the common label of the vertices of the triangle is selected, Figure 3(b).

Polygon meshes representing mechanical (CAD) objects are frequently sparse in some area, with just enough vertices to define each area. This is usually a result of the optimal triangulation from a CAD program or decimation process. In this case, the Watershed method may not segment the object properly or may lose even important regions on the objects. In Figure 5, the main regions of the object are treated as boundaries because there are not enough vertices in the regions. The boundary solution mentioned above will not solve this problem because the method does not create new regions from the boundary regions and the boundary regions will be lost. Moreover, some regions of this object are

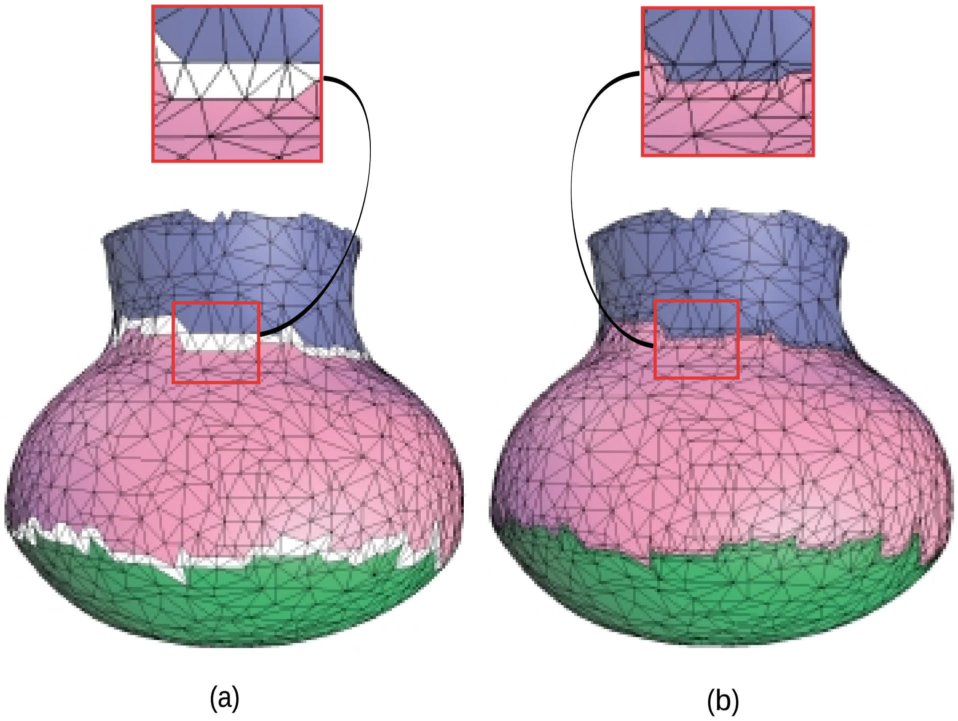

Figure 3: Segmentation using Watershed method : (a) with no hard boundaries and (b) with boundary solution.

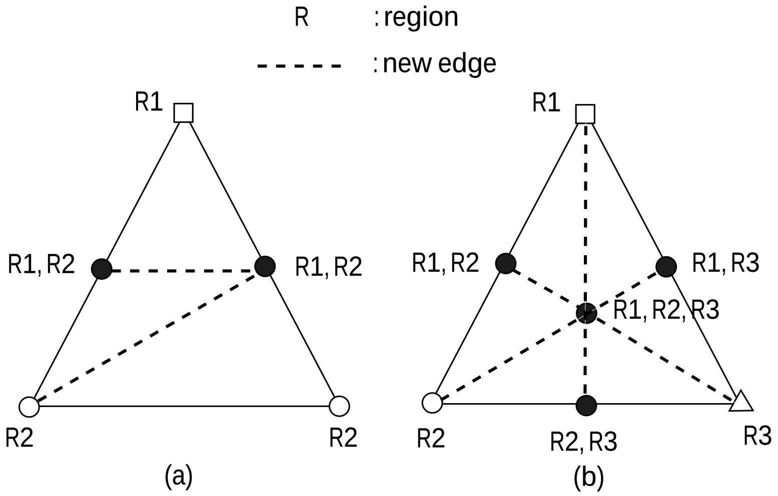

Figure 4: Creation of triangles on boundaries of an original mesh : (a) for a triangle shared by two regions and (b) for a triangle shared by three regions.

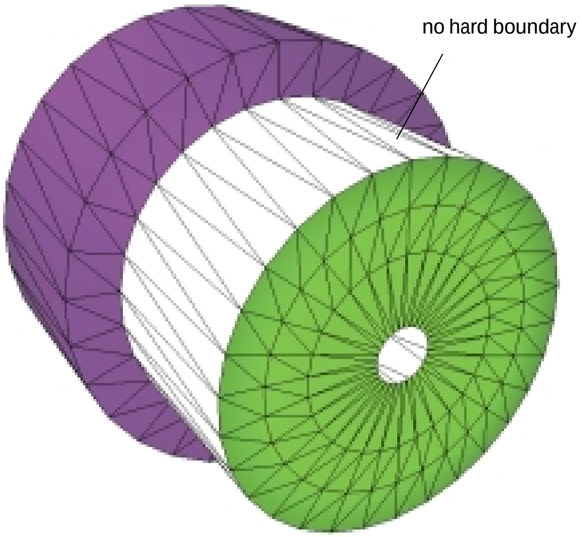

Figure 5: The mesh segmented using the Watershed method

not segmented properly. This problem is be caused by the vertices on feature edges having similar curvatures and those vertices may be treated as a same region. We solve this problem with our proposed approach in Section 5.

4 Feature Segmentation Based on Dihedral Angle

This method uses an edge-based method for defining feature boundaries. A Feature Edge is defined as follows.

Feature Edge : an edge shared by two planes whose normal vectors make an angle greater than a certain threshold.

The edges obtained are integrated into curves, and these curves are classified as jump boundaries, folds (roof edges) and ridge lines [43]. Jump boundaries and folds are used to segment the mesh into several regions. The boundary lines are also treated as Feature Edges.

4.1 Shortcomings of the Method

The main disadvantage of the Feature Edge-based method is that this results in many disconnected Feature Edges and thereby incomplete Feature Loops. Figure 6 shows this problem. Feature Edges are shown in brown color.

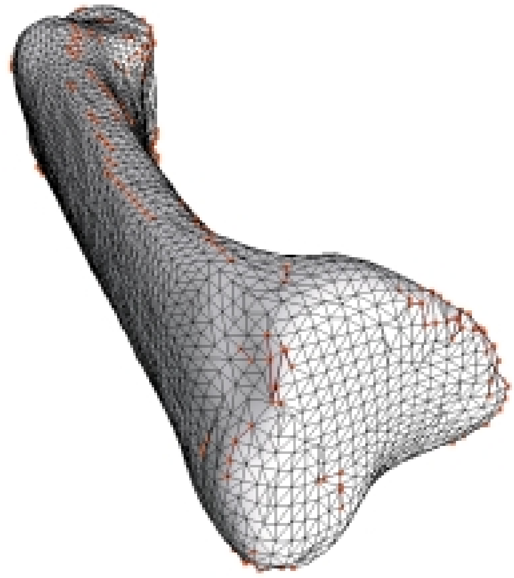

Figure 6: The feature vertices and edges of the mesh.

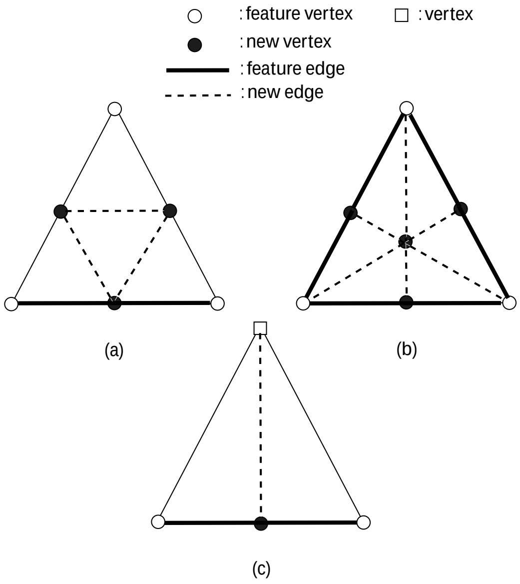

Figure 7: Creation of triangles on feature edges of an original mesh.

5 Hybrid Approach

Our proposed method takes advantages of both methods mentioned in the previous sections by using them in conjunction and hence is called the hybrid method. This creates regions with complete Feature Loops. It is described as follows.

As explained before, optimally triangulated meshes pose problems for the Watershed segmentation method. To over come this we identify all the feature edges in the mesh using a threshold (angle). First we give definition of a Feature Vertex.

Feature Vertex : Vertices that make up a Feature Edge are defined as Feature Vertices. The reverse is not necessarily true. It is important to note that if both vertices of an edge are Feature Vertices, it does not automatically qualify the edge as Feature Edge.

Step 1 - Identification of Feature Vertex

Identity all Feature Vertices.

Step 2 - Addition of Vertices

The next step in the process is to add new vertices. Vertices are added to edges of all triangles which have the property that all three vertices are Feature Vertices. See Figure 7(a). The new vertex is added at the mid point of each edge. We then connect them as shown to create four new triangles. If the triangle has three Feature Edges, then the center point of the triangle is added and six new triangles are created as shown in Figure 7(b). Addition of vertices requires fixing of the topology as illustrated in Figure 7(c). The triangle shown is a neighbor of the triangle to which we added vertices. This can lead to a hanging vertex problem. To fix this, we connect the new vertex with the vertex on the opposite edge to create two new triangles. As a result of the above, we have two kinds of new vertices; those that lie on the Feature Edges (labeled FV_{high}) and those that do not (labeled as FV_{low}). The reason for labeling is explained below.

Step 3 - Watershed Segmentation

Next, we apply Watershed segmentation to our modified mesh. Feature Vertices FV_{high} are assigned the label of maximum curvature. Since they lie on a Feature Edge, assigning them high curvature ensures that the Feature Edge will be preserved as a hard edge. The rest of the vertices in the mesh follow the same procedure as described in the Watershed algorithm for computing curvatures at the vertices. The Feature Vertices contain their own region labels as well as labels of the neighboring vertices. The addition of vertices has an impact on the Descent and Region Merge operations of the Watershed process [16]. This is done to solve the "*'no hard boundary*" problem which has been described and discussed before in this paper.

Step 4 - Removing Vertices Added in Step 2

To restore the mesh to its original form we must remove the vertices added to the mesh in step 2 above. This process restores the topology of the mesh also.

Step 5 - Collating Triangles into Regions

The goal of this step is to assign triangles and not vertices to different regions. This is achieved as follows:

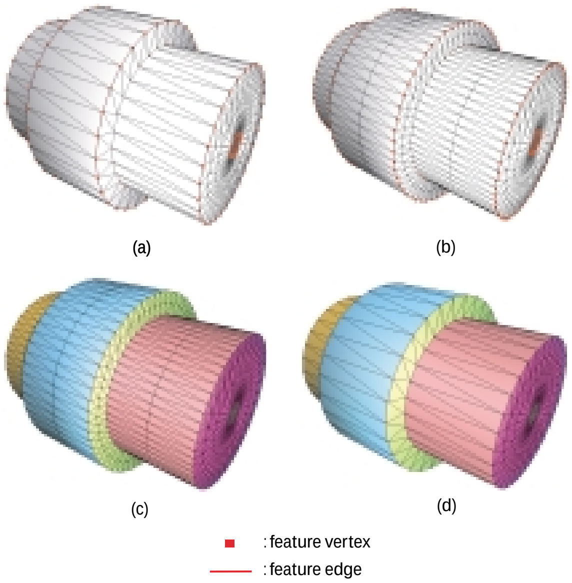

Figure 8: Various steps of the Hybrid method : (a) original mesh with feature edges and vertices, (b) modified mesh with feature edges and addition of vertices, (c) segmentation using Watershed method, (d) mesh after deleting additional vertices and triangles.

1. **Case: All vertices have the same label**
   This is simplest of the cases. The triangle is assigned the region number of its vertices.

2. **Case: One vertex has a single label**
   This is the case when one vertex has a unique label but the other two vertices have multiple labels. The triangle is assigned the region of the vertex with single label.

3. **Case: Multiple labels but only one common label**
   The three vertices of the triangle have multiple labels each, however, there is only one label that is common. The triangle is assigned the region label that is common to the vertices in the triangle.

4. **Case: All edges are Feature Edges**
   The triangle qualifies as a region by itself and gets assigned a unique region identifier.

5. **Case: Multiple labels and multiple common labels**
   It is possible that the each vertex of the triangle has multiple labels and there is more than one common label. In this case, the neighboring triangle that shares the common feature edge is selected. Then, the common vertex labels of the targeted triangle are compared with the common vertex label(s) of the neighboring triangle. The label that does not belong to the set of common vertex label(s) of the neighboring triangle is assigned to the targeted triangle. If the targeted triangle has more than one feature edge, then the common labels for each neighboring triangle of each feature edge side

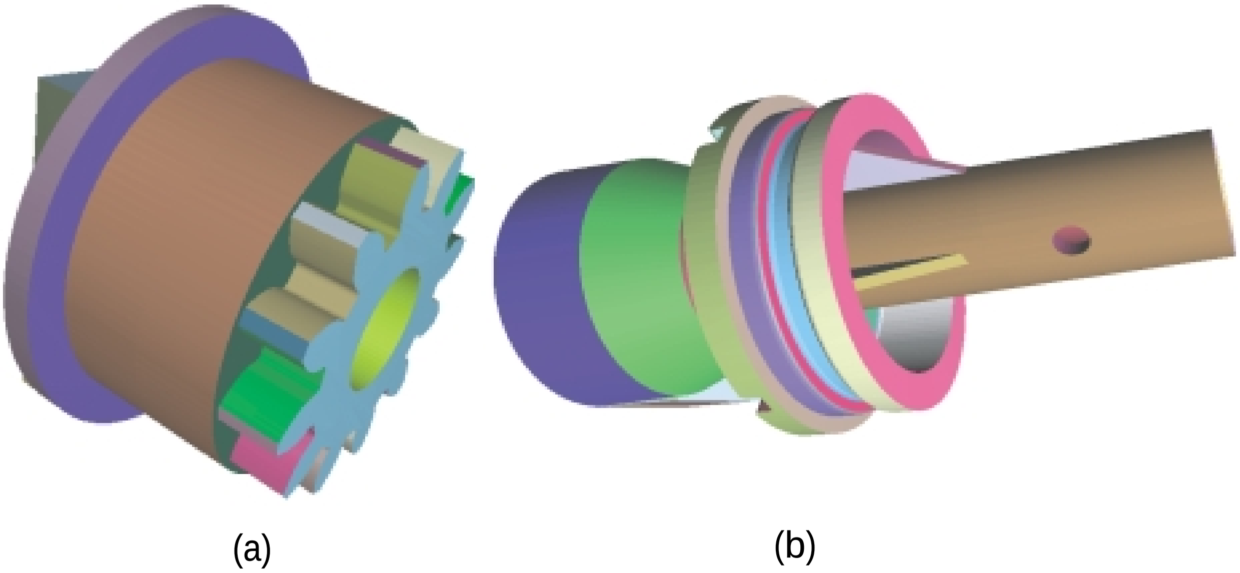

Figure 9: Segmentations of mechanical parts using the Hybrid method. Threshold for Feature edge: (a) 40° and (b) 30°, Threshold for curvature: both (a) and (b) 0.1. Both parts were scaled to a unit cube.

using the method mentioned above are selected. Then, their common label is assigned to the region of the targeted triangle. The targeted triangle may have no feature edge. It means that the triangle is in the same region as the regions of the neighboring triangles. So, the region of any neighboring triangle is selected as a region of the targeted triangle.

Figure 8 shows the steps of the Hybrid method.

6 Results and Conclusions

The hybrid method relies on the Watershed method and additionally uses advantages of dihedral angle method. The hybrid method does segmentation of smooth shaped objects and mechanical objects. It also solves the "*no hard boundary*" problem. Figure 9 shows two mechanical parts. Figure 9(a) is a gear and Figure 9(b) is a housing. Figure 10(a) shows a turbine. Notice each blade has been segmented into a different region. Figure 10(b) is a wheel (rim and tire). It is segmented into the following regions i.e. the tire, rim, bolt holes and a hole for mounting the wheel. These are correctly segmented. All examples are from public domain data sets.

We have pointed out the shortcomings of the most popular segmentation approaches and devised a hybrid method that over comes these shortcomings. Future work would involve automatic selection of threshold values both for watershed and dihedral angles.

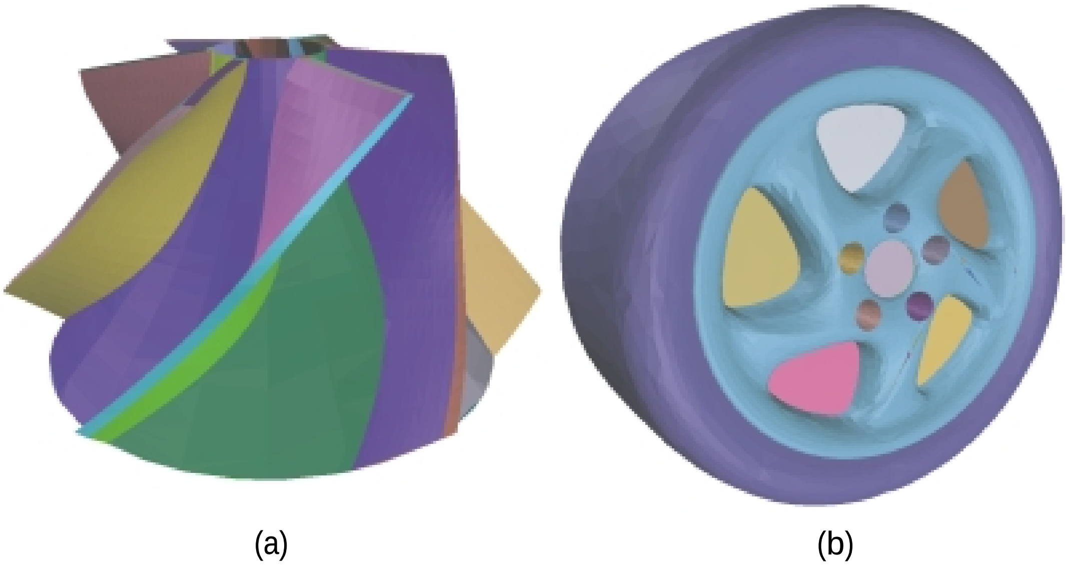

Figure 10: Segmentation of (a) a turbine and (b) a wheel using the Hybrid method. The threshold for Feature edge: (a) 25° and (b) 30°. Threshold for curvature: both (a) and (b) 0.1. Both parts were scaled to a unit cube.

7 Acknowledgments

This work was supported in part by the National Science Foundation (grant IIS-9980166). For more information on the 3D Knowledge project, visit *http://3dk.asu.edu.* The authors would like to thank various members of the Partnership for Research In Stereo Modeling (PRISM) team at Arizona State University for their support.

References

[1] A. Amresh. *Adaptive Subdivision Schemes for Triangular Meshes.* M.S. Thesis. Arizona State University, Dec 2000.

[2] G.A. Ateshian, M.P. Rosenwasser, and V.C. Mow. Curvature Characteristics and Congruence of the Thumb Carpometacarpal Joint: Differences between Female and Male Joints. *Journal of Biomechanics.* Vol. 23, No. 6, 591-607, 1992.

[3] P.J. Besl. *Surfaces in Range Image Understanding.* Springer-Verlag, 1988.

[4] W. Boehm and H. Prautzsch. *Geometric Concepts for Geometric Design.* A. K. Peters Ltd., 1994.

[5] W. Degen and V. Milbrandt. The Geometric Meaning of Nielson's Affine Invariant Norm. Computer Aided Geometric Design. Vol. 15, No. 1, 19-25, 1997.

[6] T.D. DeRose. Differential Geometry (with Applications to CAD). *SIGGRAPH '87 Tutorial Lecture.* Course 19, Geometry for Computer Graphics and Computer Aided Geometric Design, 1987.

[7] D. Eberly. *Ridges in Image and Data Analysis.* Kluwer Academic Publishers, 1996.

[8] J-T. Fan, G. Medioni, and R. Nevatia. Segmented Descriptions of 3D surfaces. *IEEE Journal of Robotics and Automation.* Vol. 3, No. 6, 527-538, 1987.

[9] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, 1996. Fourth Edition.

[10] K.F. Gauss. *General Investigations of Curved Surfaces.* Raven Press Books Ltd., 1965.

[11] B. Hamann. Curvature Approximation for Triangulated Surfaces. *Computing Suppl.* 8, 139-153, 1993.

[12] R. Hoffman and A.K. Jain. Segmentation and Classification of Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 9, No. 5, 608-620, 1987.

[13] S. Hyde, S. Andersson, K. Larsson, Z. Blum, T. Landh, S. Lidin, and B. Ninham. *The Language of Shape.* Elsevier Science B.V., 1997.

[14] L. Kobbelt. Discrete Fairing and Variational Subdivision for Free-Form Surface Design. *Visual Computer.* Vol. 16, No. 3/4, 142-158, 2000.

[15] A.W.F. Lee, W. Sweldens, P. Shroder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution Adaptive Parametrization of Surfaces. *Computer Graphics (Proc. SIGGRAPH 98).* pp. 95-104, 1998.

[16] A. Mangan and R. Whitaker. Partitioning 3D Surface Meshes Using Watershed Segmentation. *IEEE Transactions on Visualization and Computer Graphics.* Vol.5, No. 4, Oct-Dec 1999.

[17] H. Mueller and R. Jaeschke. Adaptive Subdivision Curves and Surfaces. *Proceedings of Computer Graphics International '98.* pp. 48-58, 1998.

[18] G.M. Nielson and T. Foley. A Survey of Applications of an Affine Invariant Norm. In T.Lyche and L.Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design.* pp. 445-467, Academic Press, 1989.

[19] N.S. Sapidis and P.J. Besl. Direct Construction of Polynomial Surfaces from Dense Range Images through Region Growing. *ACM Transactions of Graphics.* Vol. 14, No. 2, 171-200, April 1995.

[20] J.P. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[21] W.J. Shroeder, J.A. Zarge and W.E. Lorensen. Decimation of Triangle Meshes. *Computer Graphics.* Vol. 26, No. 2, July 1992.

[22] J. Sijbers, P. Scheunders, M. Verhoye, A. Van der Linden, D. Van Dyck, and E. Raman. Watershed-based Segmentation of 3D MR Data for Volume Quantization. *Magnetic Resonance Imaging.* Vol. 15, No. 6, 679-688, 1997.

[23] J.J. Stoker. *Differential Geometry.* Wiley-Interscience, 1969.

[24] D.J. Struik. *Lectures on Classical Differential Geometry.* Addison-Wesley Press, 1950.

[25] P.H. Todd. Intrinsic Geometry of Biological Surface Growth. *Lecture Notes in Biomathematics.* Springer-Verlag, 1986.

[26] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 13, No. 6, 583-593, 1991.

[27] T.J. Willmore. *An Introduction to Differential Geometry.* Oxford University Press, 1959.

[28] M. Yang and E. Lee. Segmentation of measured point data using a parametric quadric surface approximation. *Computer Aided Design.* Vol. 31, 449-457, 1999.

[29] C. Calladine Gaussian curvature and shell structures. *The Mathematics of Surfaces. Clarendon Press, 1986, Editor J. Gregory"* pp 179-196.

[30] J. Hoschek and D. Lasser Grundlagen der Geometrischen Datenverarbeitung B.G. Teubner, Stuttgart, 1989 English translation: *Fundamentals of Computer Aided Geometric Design* AK Peters, 1993

[31] U. Schurmans, A. Razdan, M. Marzke, J .Rowe, G. Jones, D. Collins Advances in Geometric Modeling and Feature Extraction on Pots, Rocks and Bones for Representation and Query via the Internet Archaeological Informatics Computer Applications and Quantitative Methods in Archaeology Conference 2001 April 25-29 Visby, Gotland, Sweden

[32] G. Farin and H. Hagen. A local twist estimator. In H. Hagen, editor, *Topics in Surface Modeling*, pages 79–84. SIAM, Philadelphia, 1992.

[33] S.Pulla, A. Razdan and G. Farin. Improved Curvature Estimation for Watershed Segmentation of 3-Dimensional Meshes Submitted to *IEEE Transactions on Visualization and Computer Graphics.*

[34] Y.S. Kim. Recognition of Form Geatures Using Convex Decomposition. *Computer Aided Design.* Vol. 24, 461-476, 1992.

[35] M.R. Henderson, G. Srinath, R. Stage, K. Walker and W. Regli. Boundary Representation-based Identification. In *Advances in Feature Based Manufacturing,* editors Shah, Mäntylä and Nau. Manufacturing Research and Science, Elsevier B.V., 1994.

[36] T.C. Woo. Feature Extraction by Volumetric Decomposition *Proceedings of Conference on CAD/CAM in Mechanical Engineering.* pp 39-45, MIT Cambridge, MA March 24-26 1982.

[37] R. Gadh, F.B. Prinz. A Computationally efficient Approach to Feature Abstraction in Design Manufacturing integration. *Journal for Manufacturing for Industry.* 17(1) pp 16-27, January 1995.

[38] R. Gadh, F.B. Prinz. Automatic Determination of Feature Interactions in Design-for-Manufacturing Analysis in Design Manufacturing integration. *ASME Journal of Mechanical Design* ASME Transactions, Vol 117 No 1, March 1995.

[39] R. Gadh, F.B. Prinz. Recognition of Geometric Forms using the Differential Depth Filter in Design Manufacturing integration. *Computer Aided Design* pp 583-598 Vol 24 Number 11, November 1992.

[40] R. Sonthi, G. Kunjur and R. Gadh. Shape Feature Determination using the Curvature Region Representation. *Proceedings of Conference on Solid Modeling.* pp 285-296, Atlanta 1997.

[41] A. Razdan, M.R. Henderson, P. Chavez and P.E. Erickson Feature-Based Object Decomposition for Finite Element Meshing. *Visual Computer.* pp. 291-303, January 1990.

[42] A. Razdan and M.R. Henderson. Feature Based Neighborhood Isolation Techniques for Automated Finite Element Meshing In *Geometric Modeling for Product Engineering*, Editors Josh Turner, Ken Priess. Oct. 1989, North Holland.

[43] R. Srikantiah. *Multi-Scale Surface Segmentation and Description.* M.S. Thesis. Ohio State University, 2000.