

# Three-Dimensional Surface Mesh Segmentation Using Curvedness-Based Region Growing Approach

Anupama Jagannathan, *Member, IEEE*, and Eric L. Miller, *Senior Member, IEEE*

**Abstract**—A new parameter-free graph-morphology-based segmentation algorithm is proposed to address the problem of partitioning a 3D triangular mesh into disjoint submeshes that correspond to the physical parts of the underlying object. Curvedness, which is a rotation and translation invariant shape descriptor, is computed at every vertex in the input triangulation. Iterative graph dilation and morphological filtering of the outlier curvedness values result in multiple disjoint maximally connected submeshes such that each submesh contains a set of vertices with similar curvedness values, and vertices in disjoint submeshes have significantly different curvedness values. Experimental evaluations using the triangulations of a number of complex objects demonstrate the robustness and the efficiency of the proposed algorithm and the results prove that it compares well with a number of state-of-the-art mesh segmentation algorithms.

**Index Terms**—Mesh segmentation, shape descriptor, curvedness, graph morphology.

## 1 INTRODUCTION

THREE-DIMENSIONAL mesh segmentation refers to the problem of partitioning a given 3D triangular mesh into large homogeneous disjoint submeshes usually based on certain geometric properties of the vertices that comprise the mesh. Typically, such meshes are obtained from unstructured point clouds of underlying 3D objects, which in turn are generated by range sensors [4]. Examples of applications that benefit from mesh segmentation include vertex simplification, object recognition, and scene understanding [1]. The success of several existing mesh segmentation algorithms, judged based on their ability to provide meaningful partitions, can be attributed to the specific applications for which they have been designed. For the segmentation results to be useful for higher level tasks such as object recognition, it would be advantageous to develop an algorithm that best mimics the human visual system in terms of isolating different physical parts in an object [27], [25]. In this work, this challenge is addressed by considering a graph-morphology-based region growing algorithm that uses curvedness [5], [10], [11], computed at every vertex in the mesh as the similarity metric for segmentation purposes.

Fig. 1 illustrates the need for a new segmentation algorithm that outputs sufficiently large disjoint submeshes. In Fig. 1a, due to noise or inaccuracies in the input data, the traditional segmentation approaches will fail to extract different parts of the object, that is, homogeneous regions of connected vertices corresponding to a certain intensity of

curvature. This is because noise causes perturbation of the vertices, resulting in inaccurate shape descriptor values associated with certain vertices. The algorithm proposed in this work allows for the extraction of homogeneous submeshes, as shown in Fig. 1b.

The input to the proposed algorithm is a 3D manifold surface triangulation [19] that represents the underlying object of interest. Specifically, we focus on objects that consist of subregions of homogeneous texture. Since manifold meshes have at most two triangles incident along any edge in the mesh, the computation of the shape descriptor such as the curvedness at a vertex is fairly straightforward. The input mesh is then characterized as an attributed graph, and the curvedness values are used to adaptively determine the curvedness thresholds for segmentation. Essentially, a submesh consists of a connected set of vertices whose curvedness values are in a *desired* interval, as specified by a pair of curvedness thresholds. The extraction of a certain submesh is an iterative morphological process involving graph dilation and morphological filtering of outlier vertices. A fundamental advantage of the proposed algorithm is that the selection of curvedness thresholds is done without any user intervention, and the algorithm does not require the specification of the desired number of submeshes. The details of the algorithm are provided in Section 5.

The performance of our algorithm was tested on triangulations of various complex 3D objects, which are widely accepted in the fields of computer graphics and machine vision. It is observed that the adaptive selection of curvedness thresholds leads to submeshes that correspond to the physical parts of the underlying object. Additionally, the incorporation of morphological filtering ideas with graph dilation during every iteration of the submesh growing process offers the desired robustness in the presence of noise.

We contribute to the existing state of the art by

- formulating a new submesh extraction algorithm that combines graph morphology and signal filtering ideas and

• A. Jagannathan is with Motorola Inc., 1250 North Tustin Avenue, Anaheim, CA 92807. E-mail: ajaganna@ieee.org.

• E.L. Miller is with the Department of Electrical and Computer Engineering, Tufts University, Halligan Hall, 161 College Avenue, Medford, MA 02155. E-mail: elmiller@ece.tufts.edu.

Manuscript received 7 Aug. 2006; revised 17 Jan. 2007; accepted 18 Jan. 2007; published online 6 Mar. 2007.

Recommended for acceptance by R. Klette.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0583-0806. Digital Object Identifier no. 10.1109/TPAMI.2007.1125.

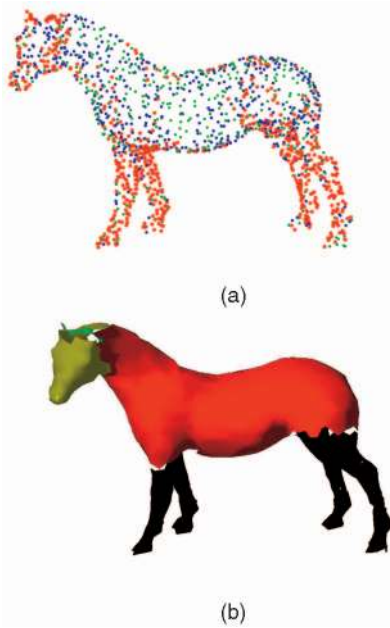


Fig. 1. In (a), the vertices have been partitioned into three categories corresponding to three different curvedness threshold intervals, computed using the threshold selection technique described in this work. A straightforward application of a standard region growing algorithm will result in small fragments of connected vertices as opposed to regions representing physical parts. The segmentation results obtained using the proposed graph-morphology-based algorithm is shown in (b). (a) Colors on the vertices correspond to different curvedness thresholds intervals. (b) Segmentation using the proposed algorithm.

- developing a parameter-free approach through the adaptive selection of curvedness thresholds that leads to disjoint submeshes that seem to match the human visual segmentation of the underlying object.

This paper is organized as follows: In Section 3, we provide the background required for computation of the shape descriptor at every vertex in the input triangular mesh. Sections 4 and 5 provide a complete description of the algorithm and highlight its capabilities. In Section 6, we compare our algorithm with an existing state-of-the-art approach and provide results that demonstrate the effectiveness of our approach.

## 2 RELATED WORK

Over the past decade or two, the problem of mesh segmentation has received significant attention and numerous algorithms [33], [34], [20], [21], [22], [23], [1], [2] have been proposed to solve the problem. We have grouped the related work into three categories.

The first category covers methods that use Reeb graph ideas [20], [21], [22], [23], based on Morse theory, for segmentation of meshes. A Reeb graph, or a contour tree, represents the topological skeleton of the underlying 3D object and uses height functions for the determination of level-set curves. Each such curve represents a vertex in the graph. Segmentation is achieved by extracting edges that link different pairs of vertices. The main drawback of the basic a Reeb graph approach is the determination of appropriate height functions that would lead to good segmentations. Also, the approach is highly sensitive to noise [21]. Various extensions to this approach have been proposed, which

include the formulation of application-dependent height functions [20], discrete Reeb graphs [22], extended Reeb graphs [23], and affine-invariant Reeb graphs [21]. The affine-invariant Reeb graphs [21] provide a rotation-and-translation-invariant segmentation. However, to obtain good results, Mortara and Patanè [21] suggest that the input mesh be uniform. Also, the work in [21] does not specifically address the perceptual aspect of segmentation.

The second category covers methods that extend classical segmentation approaches used in image analysis to three dimensions. Mangan and Whitaker [1] propose a watershed algorithm for segmentation. They compute the total curvature at every vertex and identify the local curvature minima that represent thresholds. Adjacent vertices with uniform curvatures between two minima are labeled as belonging to the same region. The algorithm is designed to provide good segmentations only for uniform meshes. Rössl et al. [3] propose a boundary extraction algorithm by extending binary morphology to three dimensions. They compute curvatures at every vertex in the mesh and subsequently threshold them to obtain a binary feature vector. Further, they apply various binary morphological operators to obtain the skeleton of the feature region. Each skeleton is then postprocessed to obtain a graph-based characterization. The Medial Axis Transformation (MAT) [24] provides an affine-invariant segmentation but is sensitive to noise [21]. In [32], an object is segmented into parts by combining Markov Random Field ideas with shape index and curvedness metrics.

The third category covers algorithms that perform perceptual segmentation. These algorithms are based on the minima theory that defines a framework for how human perception will decompose an object into its constituent parts [2]. Essentially, this theory defines boundaries as lines of negative minima curvature. Wu and Levine [27] address the perceptual aspect by formulating an algorithm based on the simulated distribution of electrical charge across the surface of a mesh. Page et al. [2] use the principal curvatures as shape descriptors and implement a variation of the watershed algorithm to identify regions that are bounded by lines of negative minima curvatures. In this work, the proposed morphology-based algorithm uses curvedness as a similarity metric to partition a 3D mesh into constituent submeshes.

## 3 DEFINITIONS AND NOTATIONS

### 3.1 Shape Descriptor: Curvedness

In our segmentation framework, the geometric aspect of an object is defined using curvedness, which is also known as the bending energy [10], [5]. It measures the intensity of the surface curvature and describes how gently or strongly curved a surface is. Mathematically, it is defined as

$$C_v = \sqrt{(\kappa_{max}^2(v) + \kappa_{min}^2(v))/2}, \quad (1)$$

where  $\kappa_{max}(v)$  and  $\kappa_{min}(v)$  are the principal curvatures of the surface at vertex  $v$  and are defined as

$$\begin{aligned} \kappa_{max}(v) &= H(v) + \sqrt{H^2(v) - K(v)}, \\ \kappa_{min}(v) &= H(v) - \sqrt{H^2(v) - K(v)}. \end{aligned} \quad (2)$$

The mean curvature and the Gaussian curvatures, denoted by  $H(v)$  and  $K(v)$ , respectively, are estimated by considering

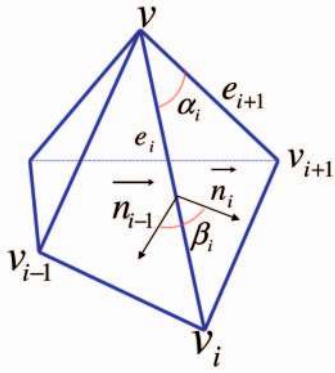


Fig. 2. Definition of various parameters with respect to the triangular faces incident on vertex  $v$ .  $\alpha_i$  is the angle subtended by the triangular face  $f_i$  at the vertex  $v$  and is computed as the angle between the corresponding edge vectors  $e_i$  and  $e_{i+1}$ ;  $\beta_i$  is the dihedral angle between adjacent triangular faces and is computed as the angle between the corresponding normals.

the triangular mesh as a piecewise linear approximation of an unknown smooth surface [9], [31]. Therefore,

$$K(v) = \frac{2\pi - \sum_{i=1}^k \alpha_i}{A/3}, \quad (3)$$

$$H(v) = \frac{\frac{1}{4} \sum m_j \|e_j\|}{A/3},$$

where

$$m_j = \begin{cases} \beta_j & \text{if edge } e_j \text{ is convex} \\ 0 & \text{if edge } e_j \text{ is planar} \\ -\beta_j & \text{if edge } e_j \text{ is concave,} \end{cases} \quad (4)$$

$A = \sum_{i=1}^k f_i$  is the sum of areas of all triangular faces incident on the vertex  $v$ , and  $\alpha_i$  denotes the angle subtended by a triangular face  $f_i$  at vertex  $v$ .  $\beta_j$  is the dihedral angle between two adjacent triangular faces  $f_j$  and  $f_{j+1}$  and is computed as the angle between the corresponding normals. Here,  $\|\cdot\|$  denotes the euclidean norm. These quantities are illustrated in Fig. 2.

Curvedness will be used as the shape feature in the proposed segmentation algorithm described in Section 5.

### 3.2 Graphs

In this work, an input mesh is defined as an attributed graph  $G[(V, C), E]$ , where  $V(G) = \{v_1, v_2, \dots, v_n\}$  is the set of vertices comprising the mesh,  $C(G) = \{C_{v_1}, C_{v_2}, \dots, C_{v_n}\}$  is the set of curvedness values associated with the vertices in the mesh, and  $E(G)$  is a set of edges connecting the vertices in  $V(G)$ . Vertices  $v_i$  and  $v_j$  are *adjacent* and are neighbors if there exists an edge  $e_{ij}$  connecting them [17]. The neighborhood  $N(v_i)$  of a vertex  $v_i$  consists of a set of vertices that are adjacent to vertex  $v_i$ .

Given the graph  $G$  and a threshold interval  $[t_k, t_{k+1})$  with threshold values  $t_k, t_{k+1} \in [C_{min}, C_{max}]$ , where  $C_{min}$  and  $C_{max}$  are the minimum and the maximum curvedness values, respectively, a *maximally connected attributed subgraph* (MCASG)  $Y$  is defined as follows:

1.  $V(Y) \subset V(G)$ .
2.  $E(Y) = E(G) \cap (V(Y) \times V(Y))$ , that is,  $E(Y)$  contains edges from the naturally generated edge set.
3.  $\forall v_i \in V(Y), C_{v_i} \in [t_k, t_{k+1})$ .
4. There exists a path  $p$  from  $v_i$  to  $v_j$  containing distinct vertices  $v_0, v_1, \dots, v_m \in V(Y)$  such that Condition 3 holds true for every vertex along the path  $p$ .

The details of the MCASG extraction algorithm are presented in Section 5.

In the following section, we explain the need for new graph-based morphological operations and provide an overview of the proposed algorithm's characteristics.

### 4 GRAPH-MORPHOLOGY-BASED SEGMENTATION: OVERVIEW

Fig. 3 illustrates the need for a new graph theoretic formulation. The color on the vertices indicates the range of curvedness values assigned to them. The segmentation of  $G$  using the known algorithms will result in four MCASGs, as shown in Fig. 3b. Such partitions are not acceptable for our purposes because the MCASGs correspond to small discontinuous regions (arising from the noisy data). On the other hand, the proposed morphology-based processing of the graph  $G$  forces the outlier vertices  $v_1$  and  $v_2$  in Fig. 3a to behave like their neighbors, thereby resulting in two MCASGs, as indicated in Fig. 3c. The proposed algorithm has the following components:

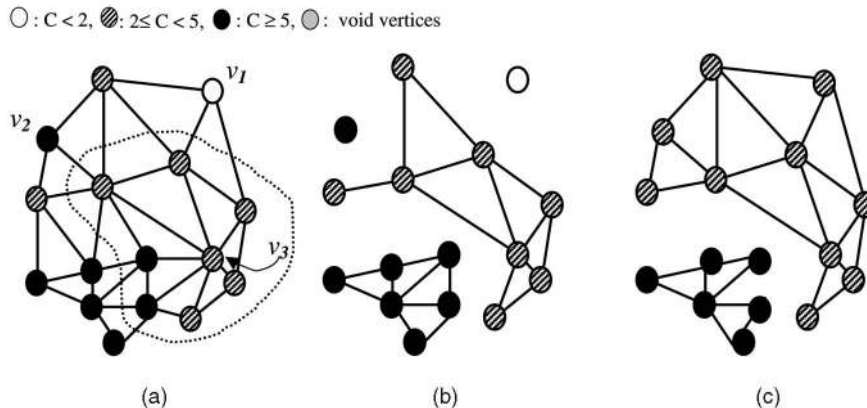


Fig. 3. The implication of not transforming the geometric properties of outlier vertices is shown by way of Output1 (b). Output2, shown in (c), is obtained by our algorithm. The proposed algorithm performs graph dilation by identifying and morphologically filtering outlier vertices such as  $v_1, v_2$ . (a) Input mesh  $G[(V, C), E]$ . (b) Output1: four disjoint MCASGs. (c) Output2: two disjoint MCASGs.



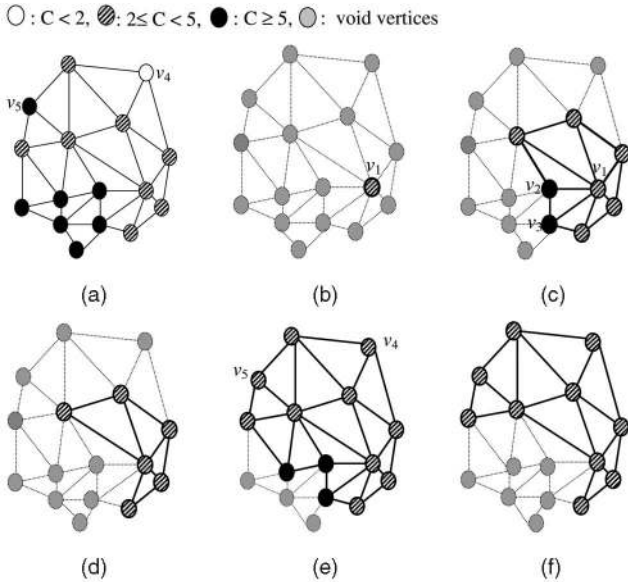


Fig. 4. The steps involved in the extraction of two disjoint MCASGs for the given input mesh  $G$ . The curvedness thresholds are specified as before. The MCASG corresponding to a certain curvedness threshold interval is obtained when the algorithm converges, that is, when  $M^k = M^{k-1}$ . The MCASG then is given by  $M^k$ . (a) Input graph  $G[(V, C), E]$ . (b) Initialization of  $V(M^0)$ ; initial submesh for first iteration. (c) Dilated graph  $G_d^1$ . (d) Matched graph  $M^1$ ; initial submesh for second iteration. (e) Dilated graph  $G_d^2$ . (f) Matched graph  $M^2$ .

- *Adaptive threshold selection.* Submeshes are obtained using adaptively determined curvedness threshold intervals. The details of the threshold selection process are presented in Section 5.3.
- *Submesh growing.* MCASG extraction process is initialized by identifying a vertex from the input mesh with a curvedness value in the specified threshold interval. For example, given the input mesh shown in Fig. 4a and the curvedness threshold interval  $[2, 5)$ , the segmentation initialization is shown in Fig. 4b. At any iteration, the initial submesh is expanded by graph dilation, which also implicitly identifies and filters the outliers in the expanded submesh. Specifically, this work exploits the idea that the geometric behavior of a vertex is influenced by its neighbors so that an outlier vertex is transformed to be a part of the MCASG by

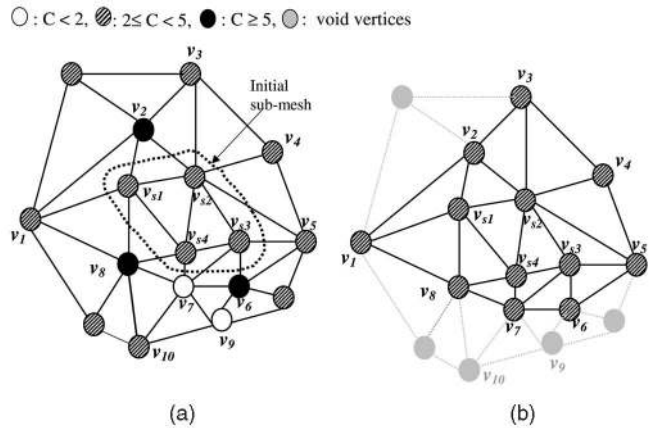


Fig. 5. (a) Dilated graph extraction process involves expansion of the initial submesh, identification, and morphological filtering of outliers in the expanded submesh. The idea exploited here is that a vertex geometrically behaves like its neighbors. Thus, it is possible to replace the curvedness value of an outlier by the median curvedness computed over its one-connected neighbors. For example,  $C_{v_7} = \text{median} \{C_{v_{s1}}, C_{v_{s2}}, C_{v_{s3}}, C_{v_{s4}}, C_{v_{s5}}, C_{v_{s6}}, C_{v_{s7}}, C_{v_{s8}}, C_{v_{s9}}, C_{v_{s10}}\}$ . (a) Input:  $G[(V, C), E]$ , initial submesh is shown in dotted lines; curvedness threshold range  $[2, 5)$ . (b) Dilated graph: curvedness values of certain vertices are modified in the expanded submesh.

- replacing its curvedness value by the median computed over the curvedness values of its neighbors. The resulting expanded submesh is scooped out of the input mesh to form the *dilated graph* for the iteration. As an example, graph dilation around the initialized vertex (Fig. 4b) followed by the extraction of the expanded submesh results in the dilated graph for the first iteration, as shown in Fig. 4c. Observe that the morphological filtering during this iteration has not transformed the outlier neighbors ( $v_2, v_3$ ) of the initialization vertex  $v_1$ . On the other hand, during the second iteration, the dilation of  $M^1$  morphologically filters the outliers  $v_4$  and  $v_5$  (Fig. 4e).
- *Removal of outliers.* After filtering, the vertices in the dilated graph that are still outside the desired threshold interval are discarded. For example, the graph  $M^1$ , shown in Fig. 4d, which is used as the submesh for expansion during the second iteration, is obtained by discarding the outliers from the dilated graph  $G_d^1$ . It may be noted that the MCASG corresponding to a

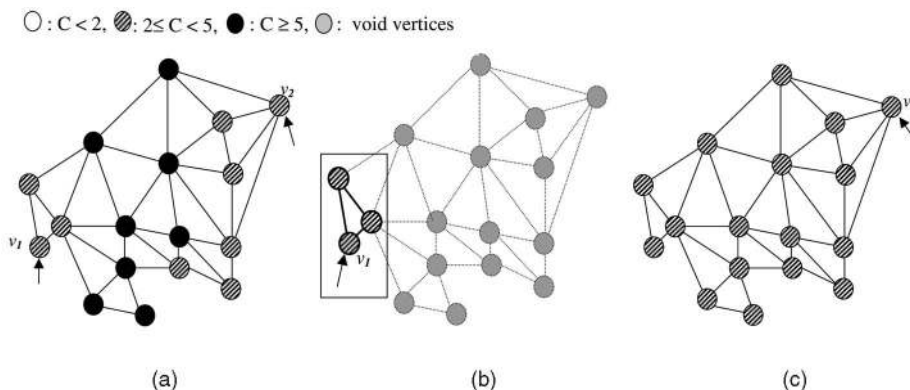


Fig. 6. Algorithm exhibits robustness to bad initializations. Selection of vertex  $v_1$  as the starting vertex does not allow the submesh to grow as much. The vertices in such MCASGs are considered unprocessed. Vertex  $v_2$  is a good choice for initialization. (a) Input graph  $G$ . (b) Initialization: Choice 1. (c) Initialization: Choice 2.

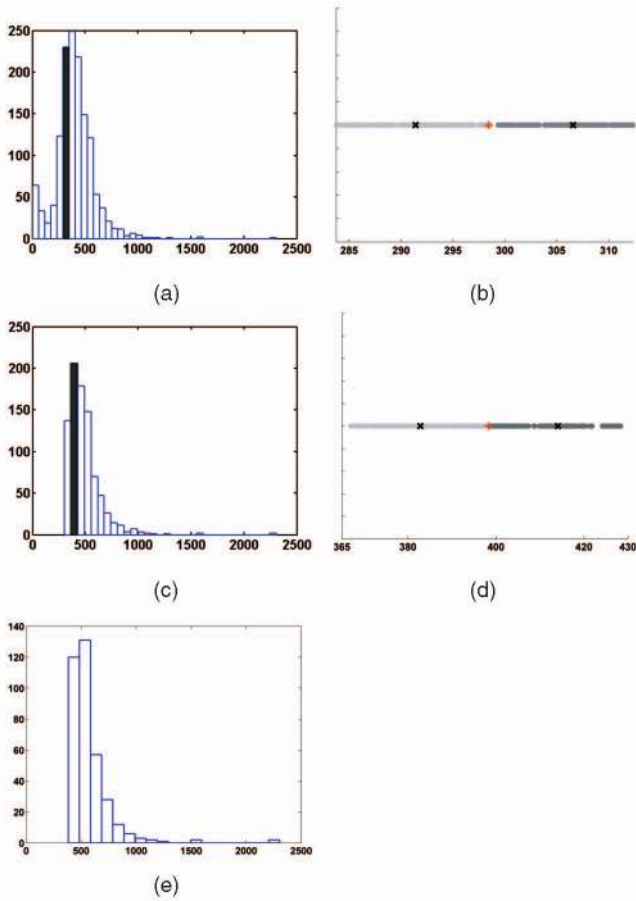


Fig. 7. Selection of curviness threshold for the simplified horse consisting of 1,548 vertices. The gray-colored bin in (a) and (c) indicates the first bin of appropriate peakedness. All curviness values in this bin are clustered into two classes using K-Means with cluster centers  $cen_1$  and  $cen_2$ . In (b) and (d) “x” and “+” correspond to the cluster centers and the bin center, respectively. The cluster center closest to the bin center is assigned as the threshold  $t_{i+1}$ . (a)  $i = 1$ ,  $W_1 = 56.02$ ,  $k = 6$ ,  $t_1 = 0$ . (b)  $cen_1 = 292.04$ ,  $cen_2 = 306.84$ ,  $t_2 = 292.04$ . (c)  $i = 2$ ,  $W_2 = 57.8$ ,  $k = 2$ ,  $t_2 = 292.04$ . (d)  $cen_1 = 382.84$ ,  $cen_2 = 414.09$ ,  $t_3 = 382.84$ . (e)  $i = 3$ ,  $W_3 = 93.9$ ,  $t_3 = 382.84$ .

certain threshold interval is obtained when the algorithm converges, that is, when  $M^k = M^{k-1}$ .

In the following section, we formally describe the basic segmentation algorithm and then extend the approach for the extraction of multiple disjoint MCASGs.

## 5 ALGORITHM FOR EXTRACTION OF MCASG

### 5.1 Basic Segmentation Algorithm

Given an input mesh  $G$  and a curviness threshold range  $[t_i, t_{i+1})$ , the extraction of a certain MCASG, say,  $M$ , involves

1. *Initialization step.* Select an arbitrary vertex  $v$  from  $G$  such that its curviness value  $C_v \in [t_i, t_{i+1})$ . Set  $V(M^0) = v$ ,  $E(M^0) = \emptyset$ , and  $C(M^0) = C_v$ .
2. *Iteration step.* For  $k = 1, 2, \dots$ , perform the following:
  - a. *Dilated graph extraction.* Determine the neighbors of the vertices in  $V(M^{k-1})$  and perform median filtering on their curviness values if necessary. Extract the dilated graph  $G_d^k$  such that

$$\begin{aligned}
 V(G_d^k) &= \{V(M^{k-1}) \cup v' | v' \in N(v), \exists v \in V(M^{k-1})\} \\
 C(G_d^k) &= \{C_v'\}, \text{ where} \\
 \forall v \in V(G_d^k) \\
 C_v' &= \begin{cases} \text{median}\{C_{v'} | v' \in N(v)\}, & C_v \notin [t_i, t_{i+1}) \\ C(v), & \text{otherwise} \end{cases} \\
 E(G_d^k) &= \{e_{uv} \in E(G) | u, v \in V(G_d^k)\}.
 \end{aligned} \tag{5}$$

The dilated graph extraction process is illustrated in Fig. 5.

- b. *Extraction of MCASG.* The removal of the outlier vertices in the dilated graph will result in a graph  $M^k$ , where i)  $V(M^k) = \{v \in V(G_d^k) | C_v \in [t_i, t_{i+1})\}$ , ii)  $E(M^k) = (V(M^k) \times V(M^k)) \cap E(G_d^k)$ , and iii)  $C(M^k) = \{C_v, \forall v \in V(M^k)\}$ . All the vertices in  $M^k$  have their curviness value in the desired curviness threshold interval.

The algorithm converges when  $M^k = M^{k-1}$ . The MCASG corresponding to the threshold range  $[t_i, t_{i+1})$  is then given by  $M^k$ .

It is observed that this approach causes the smoothing of a local surface shape, by modifying outlier curviness values. We list below certain modifications to the basic algorithm that provide practical and robust segmentations.

### 5.2 Modified Algorithm

In Section 5.1, the segmentation initialization was done by arbitrarily selecting a vertex having its curviness in the desired interval. However, if such a vertex has its neighbors outside the desired interval, then a reasonably sized MCASG may not be guaranteed. The implications of a bad initialization are illustrated in Fig. 6. If we set  $V(M^0) = v_1$  and implement our segmentation algorithm, we observe that the MCASG cannot grow as much as would have been expected, as shown in Fig. 6b. Since such MCASGs do not really represent *reasonably* large segmented regions, the vertices that comprise such MCASGs are considered “unprocessed” from a segmentation point of view. This leads to the selection of another vertex for segmentation initialization. As shown in Fig. 6c, vertex  $v_2$  is definitely a good starting point. It may be noted that the term *reasonable* is subjective, and for our problem, we drop MCASGs with fewer than 15 vertices and consider these vertices as unprocessed. As we show in the examples, such an approach works well for quite a broad range of objects. Therefore, as far as the final segmentation is concerned, we feel that the algorithm is robust against bad initializations. For a given curviness threshold range  $[t_i, t_{i+1})$ , due to the propagating nature of the segmentation algorithm, there will be a single MCASG at the output. In general, there may be several parts in any object with a similar intensity of curvature, which are otherwise disconnected. We take into account such situations and modify the basic segmentation algorithm to obtain multiple disconnected similar MCASGs.

#### Modified Segmentation Algorithm

1. Obtain the list of all vertices, say,  $L$ , satisfying the curviness threshold criterion.
2. Select an arbitrary unprocessed vertex from  $L$  and implement the basic segmentation algorithm.
3. Drop the MCASG obtained in Step 2 if it is not reasonably large (fewer than 15 vertices) and

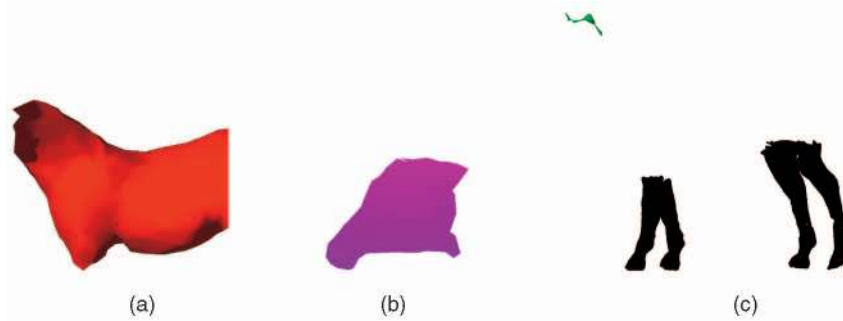


Fig. 8. Segmentation of the simplified [30] horse with 1,548 vertices into MCASGs corresponding to the threshold intervals shown in Fig. 7. (a) Curvedness interval  $[0, 292.04]$ . (b) Curvedness interval  $[292.04, 382.84]$ . (c) Curvedness interval  $[382.84, 2304.4]$ .

consider the corresponding vertices as unprocessed; else, proceed to Step 4.

4. Repeat Step 2 until all vertices in  $L$  have been processed. This step ensures that all possible sets of MCASGs satisfying the given curvedness threshold criterion have been extracted.

**Stopping Condition.** The algorithm terminates either when all vertices in  $L$  have been processed or when only isolated regions with fewer than 15 vertices (which are not reachable by any propagations) are left.

### 5.3 Adaptive Selection of Thresholds

The proposed segmentation algorithm is driven by the assumed knowledge of a pair of curvedness thresholds  $[t_i, t_{i+1}]$  that identifies the range of curvedness values allowed for the vertices in an MCASG. A threshold pair corresponds to the representative curvedness values for an MCASG. The motivation for the threshold selection process described here is derived from K-Means clustering [28] and histograms [18]. For our problem, the use of cluster centers resulting from a straightforward application of the K-Means algorithm to the set of curvedness values does not result in the desired MCASGs. Also, the optimal number of classes needs to be specified. We have found that the use of curvedness histogram peaks as thresholds leads to oversegmentation. Such oversegmentations can be avoided by selecting a curvedness value (threshold) that is close to the identified peak.

We propose an iterative threshold determination technique based on the subbin processing of the histograms. Specifically, during an iteration  $i$ , we construct a curvedness histogram with an optimal bin width over the set of curvedness values in the interval  $[t_i, C_{max}]$ . Here,  $t_i$  is the minimum of the  $N_i$  curvedness values available for processing at level  $i$ . When  $i = 1$ ,  $t_i = C_{min}$ . Next, we identify the *first bin* with an appropriate peak, that is, the bin with at least 10 percent of the total number of vertices in the input mesh. Subbin processing of this bin involves 1) partitioning of the curvedness values in the bin into two classes using K-Means and 2) assigning the cluster mean that is closer to the bin center as the threshold  $t_{i+1}$  for the extraction of the MCASG in the curvedness interval  $[t_i, t_{i+1}]$ . The repetition of this process for  $i = 2, 3, \dots$  results in the determination of all the curvedness intervals  $[t_i, t_{i+1}]$  that are required for the extraction of the corresponding MCASGs. The threshold determination process is illustrated in Fig. 7 and is formally described below.

For  $i = 1, 2, \dots$ , the selection of a threshold pair  $[t_i, t_{i+1}]$  involves

1. *Construction of a curvedness histogram.* During iteration  $i$ , a histogram is constructed over the curvedness values in the interval  $[t_i, C_{max}]$ . The set of curvedness values is partitioned into  $m$  bins such that the  $j$ th bin is the open interval  $[t_i + (j - 1)W_i, t_i + jW_i)$ . Here,  $W_i$  is the optimal histogram bin width [29], computed as

$$W_i = 3.49\sigma_i N_i^{-1/3}, \quad (6)$$

where  $\sigma_i$  is the standard deviation of the  $N_i$  curvedness values.

In Fig. 7, for  $i = 1$ , the histogram is constructed by using *all* the curvedness values, whereas for  $i = 2$  and  $i = 3$ , the corresponding histograms are constructed over the curvedness values in the intervals  $[292.04, C_{max}]$  and  $[382.84, C_{max}]$ , respectively.

2. *Identification of the first bin of interest.* The  $k$ th bin is identified as the *first bin of interest*, where  $k = \min\{j | n_j \geq 10\%N\}$ . Here,  $n_j$  is the number of vertices whose curvedness value falls in the  $j$ th bin. In Fig. 7a,  $k = 6$ , as indicated by the gray-colored bin.
3. *Subbin processing for the determination of  $t_{i+1}$ .* While there exists the *first bin of interest* with bin center  $X_i$ , perform Steps a, b, and c:
  - a. Using K-Means, partition the curvedness values in this bin into two classes and identify the corresponding cluster centers  $cen_1$  and  $cen_2$ ;  $cen_1 < cen_2$ . The cluster centers in Figs. 7b and 7d are obtained by partitioning of the gray-colored bins shown in Figs. 7a and 7c, respectively.
  - b. Set  $t_{i+1} = cen_l$ , where  $l = \arg \min_{y=1,2} |X_i - cen_y|$ . Thus, the cluster center that is closer to the bin center  $X_i$  is selected as a threshold. In Fig. 7b,  $t_2 = cen_1$ .
  - c. Extract MCASG corresponding to the threshold interval  $[t_i, t_{i+1}]$  using the segmentation algorithm described in Section 5.1. Fig. 8 illustrates the various MCASGs obtained using the threshold intervals indicated in Fig. 7.
4. *Stopping Condition.* Subbin processing terminates when none of the bins in the curvedness histogram satisfy the peakiness constraint. Then, the threshold interval for the remaining unprocessed mesh is  $[t_i, C_{max}]$ . Such a histogram is shown in Fig. 7e.

Observe that a new curvedness histogram is constructed during every iteration. This is because segmentation (at the previous iteration) modifies the curvedness values of



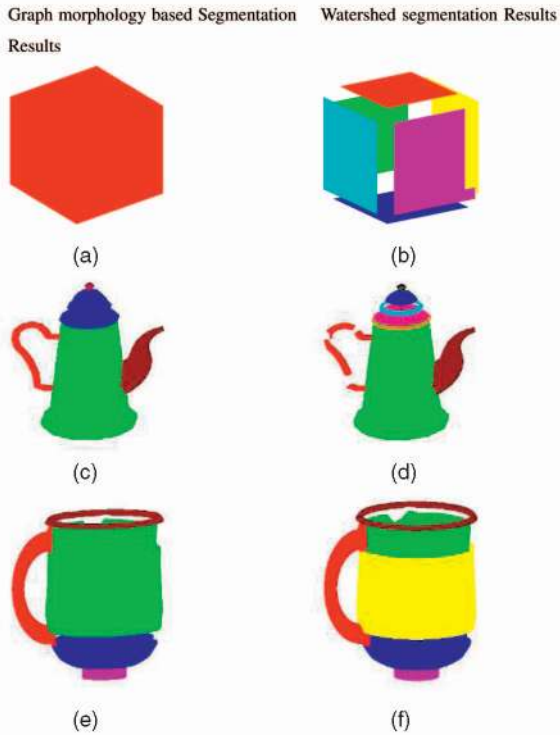


Fig. 9. Proposed segmentation algorithm partitions the input mesh into submeshes corresponding to the physical parts of the underlying object. Since a cube consists of planar faces, the algorithm outputs only one MCASG, whereas the watershed algorithm results in six submeshes. (a) One MCASG. (b) Six submeshes. (c) Five MCASGs. (d) Eleven submeshes. (e) Five MCASGs. (f) Six submeshes.

certain vertices. A fundamental advantage of such an approach is that the thresholds are selected without any user intervention, and it does not require the specification of the desired number of submeshes.

## 6 EXPERIMENTS AND DISCUSSION

### 6.1 Comparison with the State of the Art

A comprehensive comparison of our algorithm with all the existing state-of-the-art approaches is beyond the scope of

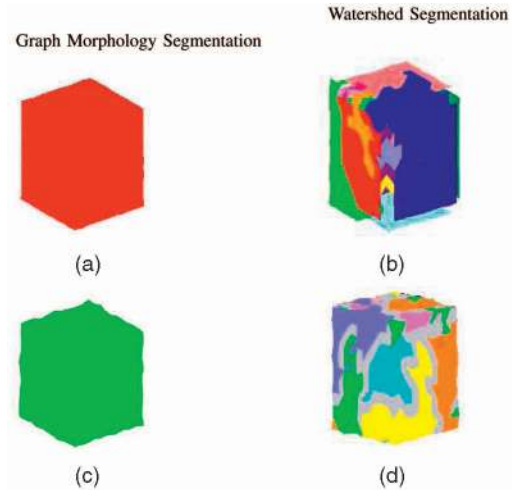


Fig. 10. For reasonable noise levels, our proposed algorithm partitions the cube into exactly one MCASG. On the other hand, the watershed segmentation algorithm does not provide the desired results. (a) SNR = 55 dB, 1 MCASG. (b) SNR = 55 dB, 13 submeshes. (c) SNR = 45 dB, 1 MCASG. (d) SNR = 45 dB, 34 submeshes.

this paper. From the mesh segmentation literature, we selected the watershed algorithm [1] for evaluation purposes, primarily because it is also built on the ideas borrowed from morphology. Our algorithm differs from the watershed algorithm [1] in the way the morphological operators are defined and applied.

#### 6.1.1 Perceptual Aspect

To qualitatively analyze the segmentations, we simulated a uniformly sampled point cloud of a cube, which was subsequently triangulated using the commercially available *Points2Polys* software. Our algorithm segments the cube into exactly one MCASG, as shown in Fig. 9a. The watershed algorithm [1], on the other hand, segments the mesh into six submeshes (Fig. 9b), wherein each submesh corresponds to a face in the cube. The input to both the algorithms is a surface mesh. For a cube, the curvatures and, hence, the curvedness values associated with the vertices that lie along the edges are starkly different from the values associated with the interior vertices. The vertices along the edges are treated as

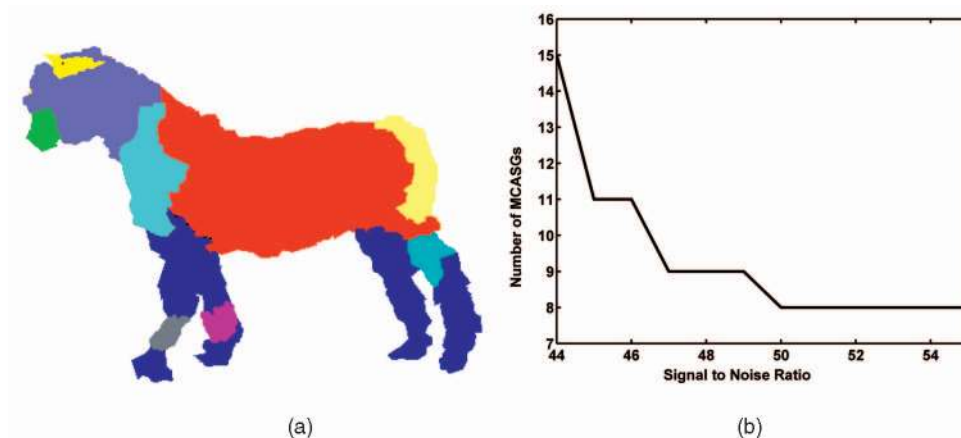


Fig. 11. The point cloud of the horse was subjected to varying amounts of Gaussian noise (SNR between 44 dB and 55 dB). A considerable amount of noise is required for the algorithm to generate patchy segmentations. (a) SNR = 44 dB; 15 MCASGs obtained for a noisy horse with 59,547 vertices. (b) Plot of SNR versus the number of MCASGs for the horse.

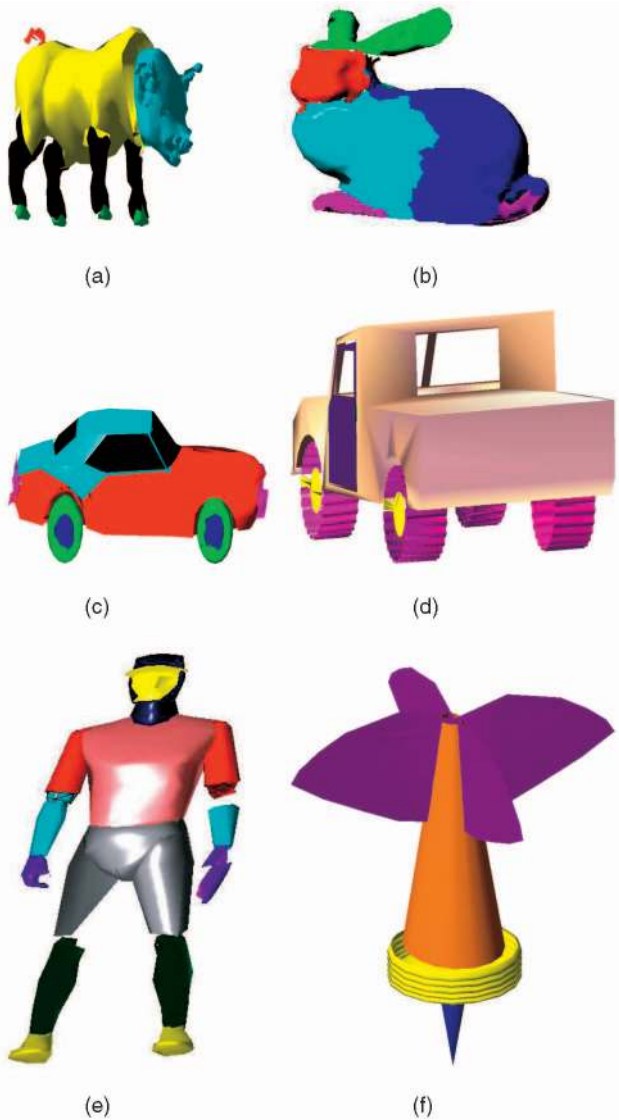


Fig. 12. Segmentation results for complex surfaces. (a) Pig: 11 MCASGs. (b) Bunny: 6 MCASGs. (c) Car: 19 MCASGs. (d) Pickup truck: 15 MCASGs. (e) Human: 15 MCASGs. (f) Dart: 4 MCASGs.

outliers, and the process of graph dilation forces the outliers to behave like their neighbors by modifying their curvedness values. Hence, segmentation results in exactly one MCASG. On the other hand, the watershed algorithm [1] treats the vertices along the edges of the cube as points of minima. Hence, the gradient descent from vertices lying on the interior of the cube toward the minima results in six connected components. It may well be argued that partitioning a cube into six submeshes (obtained using watershed algorithm) is more *meaningful* than segmentation into a single MCASG. As stated in the beginning of the paper, the definition of *meaningful* is highly application dependent. For the purposes of recognition, we think that it is reasonable to partition a cube into a single MCASG. We would like to point out that our results are similar to the perceptual segmentation results presented in [2].

Figs. 9c and 9e represent the segmentation results on a tea pot and tea cup using the proposed graph-morphology-based segmentation algorithm, whereas Figs. 9d and 9f represent the corresponding segmentations obtained using

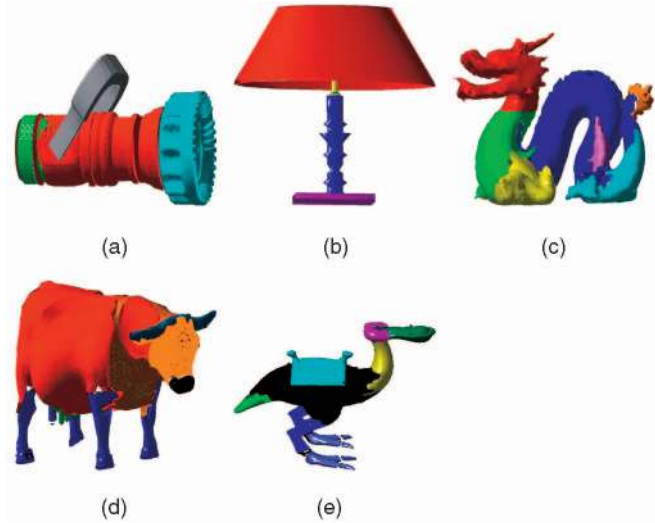


Fig. 13. Segmentation of some more complex surfaces. (a) Fire hose nozzle: 7 MCASGs. (b) Lamp: 5 MCASGs. (c) Dragon: 7 MCASGs. (d) Cow: 11 MCASGs. (e) Dinopet: 10 MCASGs.

the watershed algorithm. For the tea pot, our segmentation algorithm results in five MCASGs, whereas the application of the watershed algorithm results in 11 submeshes. We would like to draw attention to the fact that the segmentation results of the tea cup and tea pot, as provided by our algorithm, are comparable to the perceptual segmentation results obtained by the fast marching watershed algorithm presented in [2] (Figs. 2a, 2b, 2c, 3c, and 9b in [2]).

### 6.1.2 Effect of Noise

The point cloud of the cube was subjected to two different levels of Gaussian noise resulting in signal-to-noise ratio (SNR) = 55 dB and 45 dB. SNR is the ratio of signal-to-noise energy on a logarithmic scale and is mathematically expressed as  $SNR = 20\log(S/\sigma_n^2)$ , where  $\sigma_n^2$  is the variance of the Gaussian noise, and  $S$  is the maximal signal strength. Since the noise causes a perturbation of the vertices that constitute the point cloud, the curvature estimates are not accurate, resulting in many more outlier vertices or minima (as compared to the noise-free point cloud). As illustrated in Figs. 10a and 10c, due to median filtering during dilation, our algorithm segments the noisy input mesh into exactly one MCASG. The watershed algorithm results in 13 and 34 submeshes for SNR = 55 dB and 45 dB, respectively, as shown in Figs. 10b and 10d.

The effect of noise on more complex surfaces was analyzed by subjecting the point cloud of the horse to varying amounts of white Gaussian noise (SNR varying between 44 dB and 55 dB). Fig. 11a indicates the segmentation obtained at SNR = 44 dB. From Fig. 11b, we conclude that considerable noise is required for the algorithm to result in patchy MCASGs.

## 6.2 Complex Data Sets

In order to demonstrate the effectiveness of the proposed algorithm, acceptable test cases from the realms of computer graphics and object recognition/machine vision were segmented into corresponding MCASGs. Multiple similar yet disjoint MCASGs are obtained using the modified segmentation algorithm described in Section 5.2.



TABLE 1  
Timing Performance on Various Data Sets

object	Number of vertices	Number of sub-meshes	Time (in seconds)
human	2097	15	13.98
dart	1122	4	6.23
tea pot	2220	5	15.11
tea cup	11241	5	75.08
simplified horse	1548	8	11.59
horse	59547	8	451.31
Car	7401	19	43.30
Pickup Truck	4902	15	29.97
Bunny	35947	6	238.61
Pig	4332	11	27.89
Lamp	1954	5	14.25
Fire-hose nozzle	5885	7	38.78
Cow	2902	11	18.34
Dinopet	8145	10	45.22
Dragon	22998	7	189.56

The results in Figs. 12 and 13 confirm the robustness of the adaptive threshold selection process and its applicability in a wide context. As these results demonstrate, the proposed algorithm provides coarse yet clean segmentations for objects such as a car, pickup truck, dart, and lamp, which consist of regions of homogeneous texture. On the other hand, the quality of segmentation for highly textured surfaces such as the bunny and the dragon points to the limitations of the approach. Specifically, for such objects, the algorithm seems to oversegment certain regions of the mesh. A plausible solution to dealing with such object models is the inclusion of texture features in addition to the curvedness values. Further, the problem of undersegmentation of certain regions of the mesh can be addressed by casting the solution within a multiscale framework.

Table 1 shows the timing performance of the proposed algorithm on various data sets. The algorithm was coded in Matlab and tested on a Pentium IV processor at 1.5 GHz with a 256-Mbyte memory.

## 7 CONCLUSIONS

In this paper, a graph-morphology-based 3D mesh segmentation algorithm was presented to classify vertices into different categories based on their intensities of curvatures. Results indicate that 1) graph dilation together with morphological filtering of outliers can effectively deal with the noise, thereby resulting in robust segmentation and 2) the algorithm compares well with the existing state-of-the-art segmentation approaches. As part of future work, we will extend the approach to deal with textured objects by incorporating texture features in addition to the shape features considered in this work. Also, the algorithm will be extended to perform multiscale segmentation.

## ACKNOWLEDGMENTS

This work was supported by a grant from Textron Systems. The authors would like to thank all the anonymous reviewers whose feedback led to significant improvements in the explanation of the ideas presented in this paper.

## REFERENCES

- [1] A.P. Mangan and R.T. Whitaker, "Partitioning 3D Surface Meshes Using Watershed Segmentation," *IEEE Trans. Visualization and Computer Graphics*, vol. 5, no. 4, pp. 308-321, Oct.-Dec. 1999.
- [2] D.L. Page, A.F. Koschan, and M.A. Abidi, "Perception-Based 3D Triangle Mesh Segmentation Using Fast Marching Watersheds," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 27-32, 2003.
- [3] C. Rössl, L. Kobbelt, and H.P. Seidel, "Extraction of Feature Lines on Triangulated Surfaces Using Morphological Operators," *Proc. AAAI Symp. Smart Graphics*, pp. 71-75, 2000.
- [4] M. Suk and S.M. Bhandarkar, *Three-Dimensional Object Recognition from Range Images*. Springer, 1992.
- [5] C. Dorai and A.K. Jain, "COSMOS: A Representation Scheme for 3D Free-Form Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1115-1130, Oct. 1997.
- [6] *Math. Morphology in Image Processing*, E.R. Dougherty, ed. first ed., 1993.
- [7] H. Heijmans, P. Nacken, A. Toet, and L. Vincent, "Graph Morphology," *J. Visual Comm. and Image Representation*, vol. 3, no. 1, pp. 24-38, 1992.
- [8] L. Vincent, "Graphs and Mathematical Morphology," *Signal Processing*, vol. 16, pp. 365-388, 1989.
- [9] N. Dyn, K. Hormann, S.J. Kim, and D. Levin, "Optimizing 3D Triangulations Using Discrete Curvature Analysis," *Math. Methods for Curves and Surfaces*, pp. 135-146, 2000.
- [10] J.J. Koenderink, *Solid Shape*. MIT Press, 1990.
- [11] J.J. Koenderink and A.J. Van Doorn, "Surface Shape and Curvature Scales," *Image and Vision Computing*, vol. 8, pp. 557-565, 1992.
- [12] C.C. Pu and F.Y. Shih, "Threshold Decomposition of Gray-Scale Soft Morphology into Binary Soft Morphology," *CVGIP: Graphical Models and Image Processing*, vol. 57, no. 6, pp. 522-526, 1995.
- [13] P. Kuosmanen and J. Astola, "Soft Morphological Filtering," *J. Math. Imaging and Vision*, vol. 5, no. 3, pp. 231-262, 1995.

- [14] J.M. Reinhardt and W.E. Higgins, "Efficient Morphological Shape Representation," *IEEE Trans. Image Processing*, vol. 5, no. 1, pp. 89-101, 1996.
- [15] C.D. Ruberto and A.G. Dempster, "Attributed Skeleton Graphs Using Mathematical Morphology," *IEEE Electronics Letters*, vol. 37, no. 22, 2001.
- [16] I. Pitas and A.N. Venetsanopoulos, "Morphological Shape Decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 38-45, Jan. 1990.
- [17] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, MIT Electrical Eng. and Computer Science, 1994.
- [18] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, second ed. Prentice Hall, 2001.
- [19] A. Guézic, G. Taubin, F. Lazarus, and B. Horn, "Cutting and Stitching: Converting Sets of Polygons to Manifold Surfaces," *IEEE Trans. Visualization and Computer Graphics*, vol. 7, no. 2, 2001.
- [20] Y. Shinagawa and T.L. Kunii, "Constructing a Reeb Graph from Cross Sections," *IEEE Computer Graphics and Applications*, pp. 44-51, 1991.
- [21] M. Mortara and G. Patané, "Affine-Invariant Skeleton of 3D Shapes," *Proc. Shape Modeling Int'l*, pp. 1-8, 2002.
- [22] D. Bepalov, A. Shokoufandeh, W.C. Regli, and W. Sun, "Scale-Space Representation of 3D Models and Topological Matching," *Proc. Eighth ACM Symp. Solid Modeling and Applications*, pp. 208-215, 2003.
- [23] S. Takahashi, Y. Shinagawa, and T.L. Kunii, "A Feature-Based Approach for Smooth Surfaces," *Proc. Fourth ACM Symp. Solid Modeling and Applications*, pp. 97-110, 1997.
- [24] E.C. Sherbrooke, N.M. Patrikalakis, and E. Brisson, "An Algorithm for Medial Axis Transform of 3D Polyhedral Solids," *IEEE Trans. Visualization and Computer Graphics*, vol. 2, no. 1, Jan.-Apr. 1996.
- [25] K. Siddiqi and B.B. Kimia, "Parts of Visual Form: Computational Aspects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 239-251, Mar. 1995.
- [26] M. Wertheimer, "Laws of Organization in Perceptual Forms," *A Sourcebook of Gestalt Psychology*, pp. 71-88, 1950.
- [27] K. Wu and M.D. Levine, "3D Part Segmentation Using Simulated Electrical Charge Distributions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, Nov. 1997.
- [28] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*. Wiley-InterScience, 2001.
- [29] D. Scott, "On Optimal and Data-Based Histograms," *Biometrika*, vol. 66, pp. 605-610, 1979.
- [30] W.J. Schroeder, J.A. Zarge, and W.E. Lorensen, "Decimation of Triangular Meshes," *Proc. ACM SIGGRAPH '92*, pp. 65-70, 1992.
- [31] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva, "Computing and Rendering Point Set Surfaces," *IEEE Trans. Visualization and Computer Graphics*, vol. 9, no. 1, pp. 3-15, Jan.-Apr. 2003.
- [32] A. Pichler, R.B. Fisher, and M. Vincze, "Decomposition of Range Images Using Markov Random Fields," *Proc. IEEE Int'l Conf. Image Processing*, pp. 1205-1208, 2004.
- [33] S. Katz and A. Tal, "Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts," *ACM Trans. Graphics*, vol. 22, pp. 954-961, 2003.
- [34] X. Li, T.W. Toon, T.S. Tan, and Z. Huang, "Decomposing Polygon Meshes for Interactive Applications," *Proc. Symp. Interactive 3D Graphics*, pp. 35-42, 2001.



interests are in computer vision, image processing, and pattern recognition. She is a member of the IEEE.



**Anupama Jagannathan** received the PhD degree in electrical engineering from Northeastern University, Boston, in 2005. Her doctoral work involved the formulation of algorithms for the segmentation and recognition of 3D objects within the frameworks of graph theory and thermodynamics. Since 2005, she has been with the Biometrics Business Unit, Motorola, where she is responsible for developing algorithms for fingerprint identification. Her current interests are in computer vision, image processing, and pattern recognition. She is a member of the IEEE.

**Eric L. Miller** received the SB degree in 1990, the SM degree in 1992, and the PhD degree in 1994 all in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge. He is currently a professor in the Department of Electrical and Computer Engineering, Tufts University. His research interests include physics-based tomographic image formation and object characterization, inverse problems in general and inverse scattering in particular, regularization, statistical signal and image processing, and computational physical modeling. This work has been carried out in the context of applications including medical imaging, nondestructive evaluation, environmental monitoring and remediation, land mine and unexploded ordnance remediation, and automatic target detection and classification. He is a member of Tau Beta Pi, Phi Beta Kappa, and Eta Kappa Nu. He received the Faculty Early Career Development (CAREER) Award from the US National Science Foundation in 1996 and the Outstanding Research Award from the College of Engineering, Northeastern University, in 2002. He is currently serving as an associate editor for the *IEEE Transactions on Geoscience and Remote Sensing* and was in the same position at the *IEEE Transactions on Image Processing* from 1998 to 2002. He is the cogeneral chair of the 2008 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2008) to be held in Boston. He is a senior member of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).