Principal Curvature-Driven Segmentation of Mesh Models: A Preliminary Assessment

Joaquim Madeira
DETI / IEETA — Universidade de Aveiro
jmadeira@ua.pt

Samuel Silva
IEETA — Universidade de Aveiro
sss@ua.pt

André Stork
Fraunhofer IGD — TU Darmstadt
andre.stork@igd.fhg.de

Sebastian Peña Serna
Fraunhofer IGD
sebastian.pena.serna@igd.fhg.de

Abstract

Three methods for triangle mesh segmentation, based on precomputed principal curvature values and using a region growing algorithm to label the vertices defining distinct surface regions, were developed, aiming at supporting the later manipulation of mesh models. Examples are presented, using different models, to illustrate their behavior. Results are promising but, in some cases, there is a clear need for a further post-processing step to refine the boundaries between adjoining regions and eliminate segmentation artifacts.

Keywords

Mesh Models, Segmentation, Principal Curvatures, Region Growing.

1. INTRODUCTION

Segmentation, the partition of a 3D mesh model (or its surface) into meaningful components, is one important step in many mesh processing or manipulation methods, as well as in various applications, such as shape feature recognition and semantic modeling, and several mesh segmentation techniques have been developed in recent years (e.g., see [Attene 06] and [Shamir 06] and their references). Such a segmentation is usually guided by mesh properties (e.g., surface area or curvature), which are extracted prior to the proper segmentation process.

Aiming at the later interactive edition and manipulation of mesh model features, in particular for models representing mechanical parts, we decided to evaluate the results of relatively simple region growing approaches to triangle mesh segmentation, using principal curvature values previously computed at each mesh vertex as guiding properties. Our goal was, for such models, to avoid costly (e.g., optimization) procedures of existing methods (e.g., see [Lavoué 05] or [Vieira 05]).

Three segmentation methods were developed and tested:

1. simple mesh vertex labeling through region growing, directly using the value of one of the principal curvatures computed at each vertex;

2. as an improvement to the former, the range of obtained curvature values is partitioned into labeled *bins*, as a first step, and each mesh vertex is labeled accordingly; disconnected surface regions having the same label are then properly re-labeled through region growing;

3. to better define the boundary of each surface region, sharp model edges are first detected, using principal curvature values, and the inner vertices of each enclosed region are then identified and labeled by region growing.

In the next section, a brief overview of existing approaches for mesh model segmentation, as well as methods for the computation of principal curvature values, is presented.[1] Afterwards, the developed segmentation methods are described and their preliminary results illustrated by some examples, using different models. Some conclusions and ideas for further work are presented next.

2. OVERVIEW

Mesh segmentation is aimed at either partitioning a mesh into a number of surface regions that are (relatively) uniform with respect to some properties (e.g., distance to a fitting plane or curvature values) — geometric segmentation —, or identifying parts that correspond to relevant shape features — semantic segmentation. Several approaches to mesh segmentation are based on region growing: first, starting from seed vertices, an initial set of regions is identified by (recursive) neighbor inspection, selection, and

[1] A detailed enumeration of the various existing approaches is out of the scope of this short paper.

clustering; then, from that initial set, the regions partitioning a model are defined through assignment and/or merging operations. Other existing approaches are based on spectral analysis or graph-based techniques. Of particular importance for any segmentation method are the criteria (e.g., planarity, allowed differences in the directions of normal vectors, or features of fitting surfaces) for deciding which elements belong to the same region, or particular constraints (e.g., minimum diameter or area of a region) conditioning the segmentation process. For further details see the survey in [Shamir 06], as well as the comparison of results for different segmentation methods in [Attene 06].

Curvature values estimated at mesh vertices can be used to identify mesh features, such as ridges or convex and concave shapes, providing an indication of how a mesh surface behaves in the neighborhood of a vertex. Mesh curvature computation methods are usually classified into three categories [Gatzke 06]: (1) fitting methods, which determine an analytic function that locally best-fits the mesh and whose curvature function is well-defined (e.g., [Goldfeather 04]); as an alternative, discrete methods either (2) directly estimate curvature values and directions (e.g., [Meyer 03]) or (3) approximate the curvature tensor, from which curvature information is then found (e.g., [Rusinkiewicz 04, Theisel 04]). For further details see the survey and the comparison of results for different curvature computation methods in [Gatzke 06].

3. PRINCIPAL CURVATURE-BASED SEGMENTATION

The developed mesh segmentation methods aim at being faster and less demanding alternatives to existing approaches, and are intended to be applied to mesh models of mechanical parts/objects, with distinct shape features and sharp edges. In addition to using region growing to identify successive mesh regions, due to its computational simplicity, we decided to use principal curvature values as guiding properties for the segmentation, since mechanical models possess distinctive principal curvature distributions, which can also be used for model classification (e.g., [Ip 06]).

For a point on a planar curve, curvature is defined as the reciprocal of the radius of the osculating circle at that point. Given a point on a surface, the surface normal vector at that point and an intersecting plane containing the point and the normal vector, normal curvature is defined as the curvature of the defined intersection curve. The principal curvatures k_1 and k_2 are the largest and smallest normal curvature values at that point. Note that, for a given surface point, if all normal curvature values are equal, no principal curvatures are defined (e.g., for a point on a planar or spherical surface).

The principal curvature values used by the segmentation methods are previously computed using the method by Theisel et al., for an efficient estimation of the curvature tensor, from which the principal curvature values can be easily extracted [Theisel 04]: first, given the estimated normal vectors at each mesh vertex, the face curvature tensor is computed separately for each mesh triangle; then, for each mesh vertex, the principal curvature values are computed by averaging the contributions of the curvature tensors defined for the neighboring triangles.

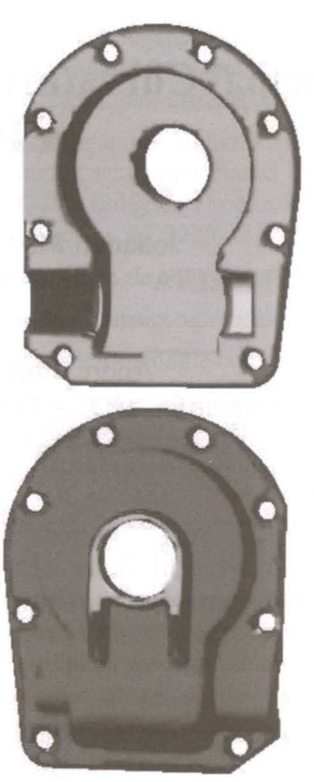


Figure 1. Simple region growing using k_1: front and back of the colored CASTING **model depicting the segmented regions.**

3.1. Simple Region Growing

In this method only the values computed at each mesh vertex, for one of the principal curvatures (k_1 or k_2), are used. The vertex labeling process starts by choosing and labeling one (start or seed) vertex; then the curvature values of its direct (1-ring) neighbors are inspected: if they are within the range $[pCurv - \delta, pCurv + \delta]$, where $pCurv$ is the principal curvature value for the start vertex and δ a tolerance value defined by the user, they are marked with the same label as the start vertex, thus considering them part of the same surface region. The neighborhood analysis proceeds recursively until no more (neighboring) vertices can be labeled and included in the same region. The region growing process then restarts by selecting an unlabeled mesh vertex, if any, and proceeding in the same way.

Two issues influence the segmentation results obtained through this method: (1) the user defined tolerance δ and (2) the (unlabeled) vertices chosen as starting points for region growing. In the test examples, the starting vertices have been chosen with no particular criterion. An example prototype, with a simple user interface, was developed which allows varying the δ values and visualizing the segmentation results.

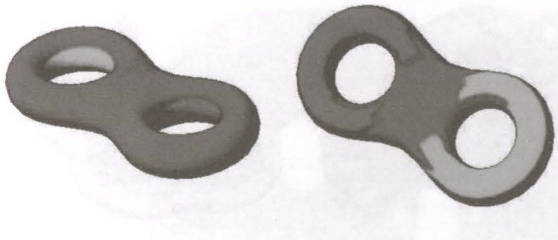


Figure 2. Simple region growing: segmented EIGHT **model using principal curvatures κ_1 (left) and κ_2 (right).**

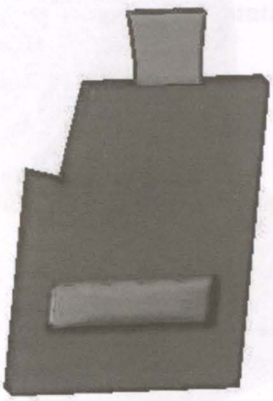


Figure 3. Simple region growing, using k_1: protruding regions in the BLOCKFS **model have been segmented.**

Figure 1 shows a model segmented using this method: the center hole is not segmented correctly, but a smaller value of δ would have resulted in several small regions appearing on the back side. Principal curvature κ_1 allowed, in general, better segmentation results than κ_2; such different segmentation results are depicted in Fig. 2 for the EIGHT model: notice how the segmentation created using κ_1 produces a visually pleasing result, which is semantically correct. Finally, notice how this method allowed segmenting protruding regions of a model, as shown in Fig. 3.

3.2. Curvature Bins

As mentioned above, the *Simple Region Growing* segmentation method is affected by the choice of the seed vertices. As we did not want to introduce an optimization step for such choice, we devised a different strategy for an initial vertex labeling: for κ_1 or κ_2, the range of obtained curvature values is partitioned into labeled *bins* and each mesh vertex is labeled accordingly. Thus, all mesh vertices with curvature values belonging to the same bin are marked with the same label, even if they do not belong to the same surface region. Disconnected surface regions whose mesh vertices have the same label are then properly re-labeled through region growing.

Figure 4 shows the model from Fig. 1 but now segmented with this new method. Notice that the center hole has now

Figure 4. Curvature bins: segmented regions (using k_1) for the CASTING **model.**

been correctly identified. For this particular model, the initial labeling resulted from subdividing the curvature range into just four bins.

Finally, Fig. 5 shows several segmentation examples using this method. For the two models on the right, some very small regions in the middle of larger surface areas can be visually identified. This is probably due to the topology of some 1-ring neighborhoods on those regions. A post-processing step would certainly remove such relatively small regions, by vertex re-labeling.

An example prototype, with a simple user interface, was developed which allows dividing the range of selected principal curvature values (κ_1 or κ_2) into a maximum number of 10 bins, with varying amplitudes. The main problem with this segmentation method is that it can be difficult to manually define an optimum amplitude for each bin.

4. EDGE-BASED SEGMENTATION

To avoid the erroneously labeled small regions in the middle of larger regions, resulting from the previous segmentation method, and to better define the boundary of segmented regions, we decided to perform a first initial step to detect sharp edges on the model surface, again based on previously computed principal curvature values. The inner mesh vertices belonging to each enclosed surface region are then identified and labeled by region growing.

Sharp edges are detected using a method described by Vieira and Shimada [Vieira 05]. First, for each mesh vertex v_i, its neighborhood size is determined by averaging the length of the edges incident to it:

$$I_{avg,i} = \frac{1}{N} \sum_{j \in N(i)} \|v_j - v_i\| \tag{1}$$

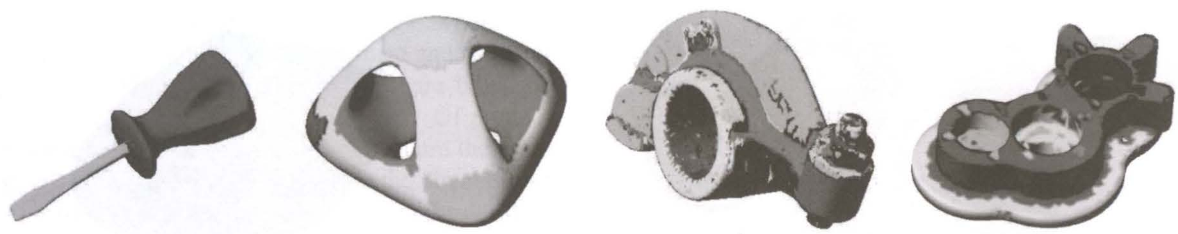
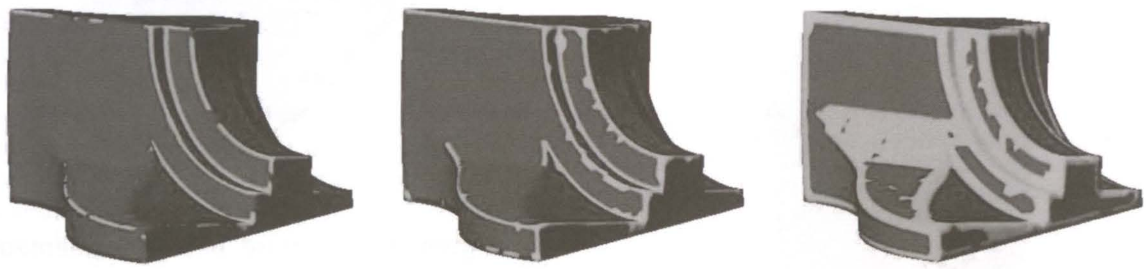


Figure 5. Curvature bins: several segmented models (using k_1). From left to right, SCREWDRIVER, GENUS3, ROCKERARM **and** PUMP.



$\lambda = 1$ $\lambda = 2$ $\lambda = 10$

Figure 6. Sharp edges detected on the FANDISK **model, for different values of λ and using k_1.**

Then, vertex v_i is considered as being part of a sharp edge if

$$\frac{1}{|\kappa_{max,i}|} < \lambda I_{avg,i}, \tag{2}$$

where $\frac{1}{|\kappa_{max,i}|}$ is the minimum curvature radius for vertex v_i, and $\kappa_{max,i}$ is the largest magnitude principal curvature for that vertex.

Although Vieira and Shimada [Vieira 05] consider a value of 10 for the λ parameter in all situations, some experiments seemed to show that this parameter must vary according to a model characteristics. Figure 6 shows the sharp edges detected for different values of λ, for the FANDISK model; notice that when $\lambda = 10$ the detected sharp edges are *"exaggerated"*. Thus, λ allows adjusting the sharp edge detection sensibility.

After detecting the sharp edges and labeling their vertices, a region growing algorithm is repeatedly applied to segment the different model regions: starting in any unlabeled (inner) mesh vertex, that vertex and its neighbors will be recursively labeled until marked sharp edge vertices are reached. This process is repeated for all not yet labeled mesh vertices.

Afterwards, the vertices marked as part of sharp edges must be included in the segmented regions those edges enclose. Thus, in a further step, all sharp edge vertices are processed and included in the closest region by an appropriate labeling.

Figure 7 shows the CASTING model presented earlier, but now segmented with this method: once again it resulted in a proper segmentation. In Fig. 8 the detected sharp edges and a new segmentation for the GENUS3 model are presented: two main regions were identified, one for the interior and another for the exterior of the model. This segmentation is different from that presented in Fig. 5, which exhibits additional regions. Finally, Fig. 9 shows two more examples of segmentations obtained using this method.

Figure 8. Edge-based segmentation, using k_1: detected sharp edges (top) and segmented regions for the GENUS3 **model (bottom).**

An example prototype with a simple user interface was developed which allows: (1) defining a value for λ and visualizing the detected sharp edges and, starting from the

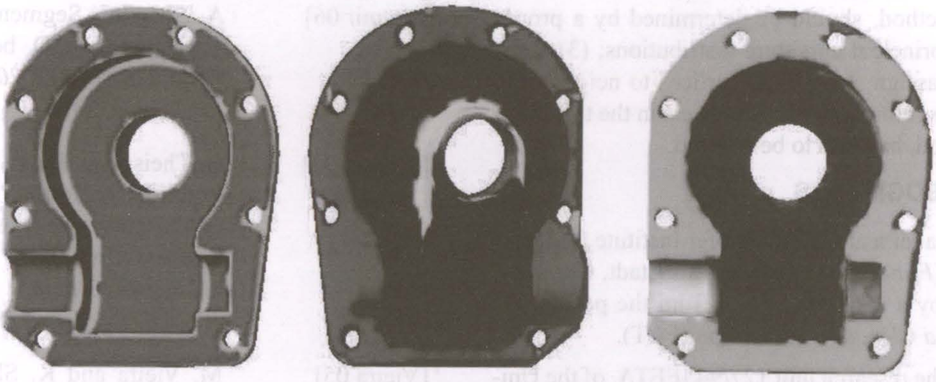


Figure 7. Edge-based segmentation, using k_1: detected sharp edges (left) and segmented regions for the CASTING model.

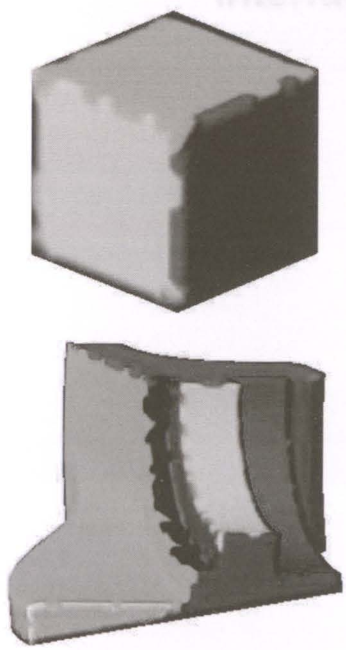


Figure 9. Edge-based segmentation, using k_1: segmented regions for the CUBE (top) and FANDISK (bottom) models.

marked vertices belonging to the sharp edges, (2) segmenting surface regions, bounded by the detected edges, by labeling the remaining vertices through appropriate region growing. Clearly, the resulting segmentation is influenced by the result of the sharp edge detection stage, i.e., by the value chosen for the parameter λ.

A further problem is how to deal with the mesh vertices initially marked as part of sharp edges: assigning them blindly to one of the closest surface regions might result in some erratic behavior on the region boundaries — as can be seen, for example, on the CUBE model in Fig 9. Also, for a narrow region, two opposing sharp edges might end up by getting connected (due to vertex proximity), thus dividing the region in two: this can be observed in the frontal part of the FANDISK model shown in Figs. 6 and 9.

5. CONCLUSIONS AND FURTHER WORK

Using principal curvature values as guiding properties, three methods for triangle mesh segmentation using region growing to label the vertices defining distinct surface regions were described. The methods are intended to be applied to mesh models of mechanical parts/objects, with distinct shape features and sharp edges. The developed methods aim at being faster and less demanding alternatives to existing approaches. A preliminary assessment of their behavior was done using different models.

The first segmentation method, using just the simple region growing, is computationally less demanding, but is rather sensitive to the choice of seed mesh vertices and the amplitude of the allowed curvature variation interval, as expected. At the moment, the second method, based on the previous partition of the range of curvature values into bins, cannot be applied without user intervention for the selection of the number of bins and their amplitudes. The third edge-based segmentation method gave very interesting results, but might not work for some types of models (e.g., the EIGHT model) due to the impossibility of detecting sharp edges in such cases.

Each one of the devised methods provides a first solution to the mesh segmentation problem and results seem to be promising, although, in most cases, there is a clear need for further post-processing of the segmentation results (i.e., the labeled mesh) to remove segmentation artifacts, as is done in similar segmentation approaches (e.g., in [Lavoué 05] and [Vieira 05]).

An interesting idea for further work is to evaluate the eventual advantage of performing the segmentation using simultaneously the two principal curvature values assigned to each mesh vertex, in all three methods. Furthermore, some additional processing operations can be envisaged for each of the devised methods. For instance, (1) for the simple region growing, an efficient criterion for choosing the seed mesh vertices should be applied; (2) the number and amplitudes of the curvature bins, in the second segmentation method, should be determined by a proper analysis of the principal curvature distributions; (3) a criterion to better assign sharp edge vertices to neighboring regions, and thus refine region boundaries, in the third segmentation method, has also to be devised.

6. ACKNOWLEDGMENTS

The stay of J. Madeira at the Fraunhofer Institute for Computer Graphics (*Fraunhofer IGD*) in Darmstadt, Germany, was supported by a sabbatical grant from the portuguese *Fundação para a Ciência e Tecnologia* (FCT).

S. Silva thanks the research unit 127/94 IEETA, of the University of Aveiro, for the grant that supports his work.

The work of A. Stork and S. Peña Serna was partially supported by the *Aim@Shape* European Network of Excellence.

References

[Attene 06] M. Attene, S. Katz, M. Mortara, G. Patané, M. Spagnuolo, and A. Tal. Mesh segmentation — A comparative study. In *Proc. Shape Modeling and Applications 2006 (SMI'06)*, pages 14–25, 2006.

[Gatzke 06] T. Gatzke and C. Grimm. Estimating curvature on triangular meshes. *Int. J. Shape Modeling*, 12:1–28, 2006.

[Goldfeather 04] J. Goldfeather and V. Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graphics*, 23:45–63, 2004.

[Ip 06] C. Y. Ip and W. C. Regli. A 3D object classifier for discriminating manufacturing processes. *Computers and Graphics*, 30:903–916, 2006.

[Lavoué 05] G. Lavoué, F. Dupont, and A. Baskurt. A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37:975–987, 2005.

[Meyer 03] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, 2003.

[Rusinkiewicz 04] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Proc. 2nd Int. Symp. 3D Data Processing, Visualization, and Transmission (3DPVT'04)*, pages 486–493, 2004.

[Shamir 06] A. Shamir. Segmentation and shape extraction of 3D boundary meshes. *EUROGRAPHICS 2006 — State of the Art Report*, 2006.

[Theisel 04] H. Theisel, C. Rössl, R. Zayer, and H.-P. Seidel. Normal based estimation of the curvature tensor for triangular meshes. In *Proc. 12th Pacific Conf. Computer Graphics and Applications (PG 2004)*, pages 288–297, 2004.

[Vieira 05] M. Vieira and K. Shimada. Surface mesh segmentation and smooth surface extraction through region growing. *Computer-Aided Geometric Design*, 22:771–792, 2005.