

# Hybrid Surface Partitioning of Finite Element Meshes

J. Shen, Y. Song, D. Yoon

Computer & Information Science Department  
University of Michigan - Dearborn  
Dearborn, MI, USA

S. Liu

Generalety, LLC  
Livonia, MI, USA

**Abstract** – A hybrid automatic and interactive scheme is developed for partitioning the surfaces of mechanical parts represented by finite element meshes on the basis of geometric discontinuity. Given a finite element surface mesh  $M$  in three dimensions, the algorithm firstly finds a partition of  $M$  into  $k$  subregions  $M_1, \dots, M_k$  on the basis of  $G^1$  discontinuities in an automatic manner. Secondly, an interactive approach is proposed to decompose surface meshes further w.r.t.  $G^2$  discontinuity in reference to surface curvatures. The overall computational complexity is  $O(n \log n)$ , where  $n$  is the maximum between the numbers of elements and vertices in  $M$ , respectively. Numerical experiments have been conducted to show the effectiveness of our algorithm and its application in shape morphing or optimization.

**Keywords:** Geometric modeling, Finite element mesh, Surface curvature, Histogram.

## 1 Introduction

Surface partitioning of unstructured meshes is important to many problems in engineering and science. In the areas of computational geometry, virtual environment, computer games, tolerance verification, collision detection and motion planning, the partitioning problem may be abstracted as a convex solid decomposition of polyhedral or a convex surface decomposition of polygonal meshes [1, 2]. The key idea of the approaches in this group is to break the entire model into a number of convex entities. The problem of finding the minimum number of patches is NP-complete, but there are several good approximations.

In the area of computer graphics, surface partitioning has been used to assist morphing, mesh reduction, shape matching, and collision detection. A novel hierarchical mesh decomposition algorithm was recently proposed to cut up a complex object into simple sub-objects by using deep concavities as an indicator [3]. This approach is well suited in decomposing biological creatures such as animals and human beings, but is not capable of identifying features such as fillets of mechanical parts.

In the area of reverse engineering, a recover and select paradigm was proposed by Leonardis et al. [4]. An iterative process was used to recover and select specific instances of the required geometric primitives, including planes, spheres, cylinders, cones and tori. On the basis of

this approach, a non-iterative method, direct segmentation, was also studied [5, 6]. However, it is suited only for regular objects that are bounded by simple analytic surfaces.

In this paper, we focus on the surface partitioning of a special type of mesh, finite element mesh, for shape optimization. It is reasonable to assume that there is no surface noise on finite element meshes except discretization errors. Even though the CAD information can be useful for the partitioning, with the boundary representation each B-Spline patch or surface may not have any functional meaning, especially for freeform objects or irregular objects with anti-symmetry. It is therefore useful to develop a surface partitioning algorithm for finite element meshes, which is capable of decomposing a mesh into geometrically-meaningful partitions for the purpose of shape optimization. A geometrically meaningful entity herein means neither a convex entity nor an equally-sized subregion. It refers to a form feature that describes a portion of a part's nominal (or idealized) geometry.

We now state the problem that we wish to solve. It is assumed that a finite element mesh without surface noises is available. In the case of polyhedral meshes, a polygonal surface mesh needs to be extracted as a preprocessing step. The objective of our approach is to partition this mesh into fewer numbers of geometrically meaningful subregions, each of which corresponds to a form feature or a general freeform feature. In [7], we proposed a surface partitioning scheme for finite element meshes. It is a fully automatic scheme that may generate some undesirable partitions because of a lack of interactivity in determining a suitable curvature threshold, which often varies from problem to problem. In this paper, we propose a new approach with the following main contributions:

- (1) A hybrid scheme: automatic surface partitioning by  $G^1$  discontinuity and interactive surface partitioning by  $G^2$  discontinuity. This provides users a flexible way to tackle the partitioning problem for the finite element meshes without surface noises. In this paper, we define  $G^1$  discontinuity as the discontinuity of surface tangent direction, which is equivalent to the discontinuity of surface normal, and  $G^2$  discontinuity as the discontinuity of surface curvature.

- (2) A local histogram<sup>1</sup> of nodal surface curvatures and a corresponding region growing process are used for achieving the surface partitioning of polygonal meshes by  $G^2$  discontinuity. Users have an option to choose one of five different types of curvatures.

The rest of this paper is organized as follows. The algorithm is introduced in Section 2, and numerical experiments and discussions are presented in Section 3. Finally, some conclusions are given in Section 4.

## 2 Algorithm for Surface Partitioning

As stated in Section 1, the main goal of this paper is to develop a surface partitioning scheme that is based on geometric characteristics of polygonal mesh surfaces. Surface discontinuity is a major mark that can be used to fulfil this objective, especially in the area of shape optimization on finite element models in which no surface noises exist and the main requirement to the partitioning is to obtain geometrically meaningful partitions (i.e., form features) rather than convex entities, equally-sized entities, or a minimum number of entities. There are a variety of surface discontinuity such as  $G^1$ ,  $G^2$ , ...,  $G^n$  among which only the first two are practically useful, i.e., any  $G^i$  ( $i > 2$ ) discontinuity has an insignificant influence on the shape of the surface if  $G^1$  and  $G^2$  continuities are guaranteed. Since  $G^2$  discontinuity  $\supset$   $G^1$  discontinuity in a sense that some  $G^2$  discontinuities do not imply  $G^1$  discontinuity, we can divide the entire task into two main steps in a sequential manner as follows:

Algorithm 1: Hybrid Surface Partitioning

1. Automatic surface partitioning by  $G^1$  discontinuity on the base of surface normal
2. Interactive surface partitioning by  $G^2$  discontinuity with respect to curvatures

In this way, our scheme approximately covers the surface partitioning under any possible range of geometric discontinuity. In addition, the homogeneity of nodal surface normal on each of the resulting surface patch by  $G^1$  discontinuity partitioning is guaranteed such that a further region growing for surface partitioning by  $G^2$  discontinuity with respect to curvatures becomes supported (see Section 2.2.2 for details).

### 2.1 Surface partitioning by $G^1$ discontinuity

The treatment to  $G^1$  discontinuity depends mainly upon the existence of surface noises. In the case of no surface noises, surface normal of each finite element can be used as an indicator to guide a surface partitioning process. However, with surface noises, surface curvature

information is normally needed. In engineering analysis and optimization, it is reasonable to assume no surface noises on finite element meshes. Thus, in this paper the angle formed by surface normals of two adjacent elements is used as an indicator to determine when a  $G^1$  discontinuity will occur. In [7], we developed an automatic scheme for surface partitioning by  $G^1$  discontinuity. It is basically a breath-first search for region growing.

The difference between our scheme and existing approaches lies in the fact that we first partition the surface by  $G^1$  discontinuity independently. In contrast, existing approaches blindly handle the discontinuities caused by both  $G^1$  and  $G^2$  discontinuities (i.e., both sharp edges and high-curvature transitions) [8, 9].

After the completion of  $G^1$  partitioning, planar sub-regions may be identified by another round of breath-first searches over  $G^1$ -continuous sub-surfaces with a second angular threshold for surface normal change,  $\beta$ . Its default value is  $5^\circ$  and can be adjusted by users. In the past, finding planar sub-regions was proposed by several researchers [6, 10-12]. In this paper, we use a curvature threshold to prevent the case in which a low-curvature region is mistakenly considered as a planar sub-surface. Our numerical experiment indicates that the identification of planar sub-regions is well suited to mechanical parts with many planar surfaces, but may cause over-segmentations on parts with freeform surfaces. Thus, we set the generation of planar sub-surface as an option for users to choose in a surface partitioning process.

### 2.2 Surface partitioning by $G^2$ discontinuity

Two crucial questions about  $G^2$  partitioning are whether we really need it and whether it leads to over-segmentation. The answers to the questions are context-dependent. In most application areas mentioned in Section 1, surface noises exist such that the  $G^2$  partitioning becomes useless and likely produces over-segmentation. However, in shape morphing or optimizations with finite element meshes there exist only approximation or discretization errors. A careful usage of  $G^2$  partitioning can generate additional useful information about the underlying shape of the model for an effective shape morphing or optimization. Therefore, we propose to handle  $G^2$  partitioning interactively by letting users to select a curvature threshold dynamically with respect to a curvature histogram and to select one of five types of curvatures for generating a desirable result.

There are two important issues that need to be addressed in order to make the surface partitioning by  $G^2$  discontinuity successful. Firstly, an accurate estimation of discrete curvatures on polygonal mesh surfaces is required. Secondly, a robust decomposition of curvatures over

<sup>1</sup> "Local" refers to the region of each surface patch resulted from partitioning by  $G^1$  discontinuity.

polygonal mesh surfaces is needed. As to the estimation of discrete curvature, since the algorithm in [7] is used to identify all the  $G^1$  discontinuities such that all the partitioned sub-surfaces are  $G^1$  continuous within their interior regions, differential geometry can then be applied on each sub-surface to estimate the discrete curvatures, even at the boundary on the side of the interior. For the curvature decomposition, a local histogram of a special curvature and a corresponding region growing process are used to produce a surface partitioning by  $G^2$  discontinuity.

### 2.2.1 Estimation of discrete nodal curvatures

In general, either nodal or element curvatures can be used as a representation of discrete surface curvatures. Since the nodal curvature is closer to the curvature definition in differential geometry and the element curvature provides only a kind of average information over each surface element, nodal curvature, discrete estimate of curvatures at each vertex, is adopted in region growing, while the element curvature is used in the generation of planar surfaces described in Section 2.1. There is a vast amount of literature related to different approaches for estimating discrete curvatures. However, no report is available for the comparison among these schemes with respect to accuracy, convergence and computational efficiency. In reference [7], we also proposed an accurate estimation of discrete nodal curvature with a mathematically-proven convergence. The basic idea is to use a quadric surface patch to approximate a vicinity of each surface vertex of a surface patch that is already partitioned by  $G^1$  discontinuity. Principal components analysis is used to determine the orientation of this local quadric patch, and a linear least-squares fitting and formula in differential geometry are used to determine the nodal curvatures. We have proven that our scheme is sufficient to describe all major types of surfaces: elliptic, hyperbolic, parabolic and planar, at the neighborhood of a vertex on each surface, and the discrete curvature will converge to the corresponding continuous curvature correctly if the mesh density increases.

### 2.2.2 Local multimodal histogram of nodal surface curvatures

One way to carry out the partitioning by  $G^2$  discontinuity is to use a histogram of curvatures like the usage of histogram in image segmentation. However this approach does not work well with multiple curvature clusters, especially when they overlap each other in the histogram. The following special treatments make this approach a more viable way in  $G^2$  discontinuity partitioning:

(1) we use a local histogram of nodal curvatures. The word “local” means that a histogram is created on each

surface patch obtained by  $G^1$  partitioning. In such a local histogram, the probability for multiple curvature clusters to overlap each other is significantly reduced.

- (2) we use a region growing process as a means to perform the surface partitioning by  $G^2$  discontinuity. This process is partially justified by the homogeneity of surface normal on each surface patch obtained by  $G^1$  discontinuity partitioning. Such a surface propagation can handle the overlapped curvature clusters properly as long as these clusters are not spatially connected to each other over the surface.
- (3) Any curvature cluster with a small population is neglected as a way to suppress the influence of curvature noises and extremely fine details on the surface.
- (4) An interactive tool is constructed to perform the  $G^2$  partitioning selectively and recursively. Users have an option to choose one of five curvature types for the construction of a curvature histogram and to select a particular surface patch to undergo a  $G^2$  partitioning in an interactive way. Users can dynamically change the curvature threshold with respect to a histogram and preview the result of  $G^2$  partitioning. Such a process is repeated interactively until a desirable result is obtained.

In this paper, we divide the abscissa of the curvature histogram into  $N_{cb}$  curvature bins, while an ordinate is used to represent a count of vertices that fall into each curvature interval or bin on the abscissa, as shown in Figure 1. Figure 2 shows a user interface used for displaying a curvature histogram and letting users to specify curvature markers that separate different curvature clusters. In this figure, one curvature marker is created and translated to an appropriate position by using a mouse drag along the abscissa of the curvature histogram. Users are allowed to create as many curvature markers as necessary to separate multiple curvature clusters and to preview a tentative partitioning result. Each type of curvatures may play a significant role in different situations, as indicated by a feature map in Figure 3. We propose to provide five types of curvatures: Gaussian, mean, maximum principle, minimum principle, and  $k_{12}$  [7] for users to select.

Even though the interactive manipulation of curvature markers is a necessity for robust  $G^2$  partitioning, it is still desirable if there is an automatic scheme to provide an initial estimate to the curvature markers. Our approach to determine the initial curvature markers is described below.

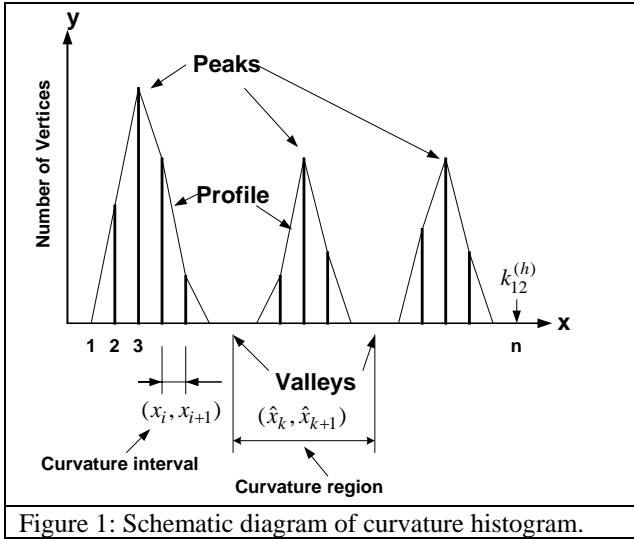


Figure 1: Schematic diagram of curvature histogram.

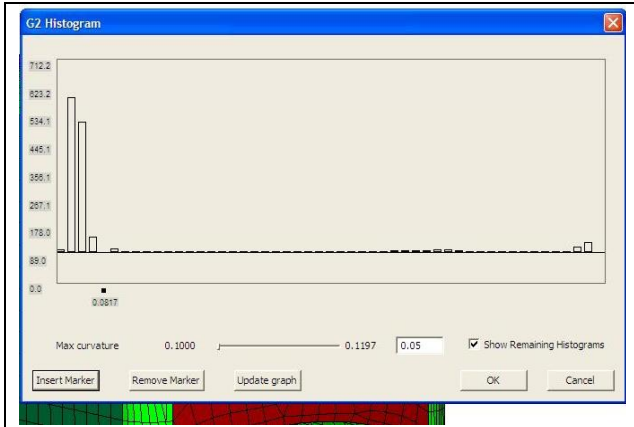


Figure 2: The working window for interactive manipulation of curvature thresholds.

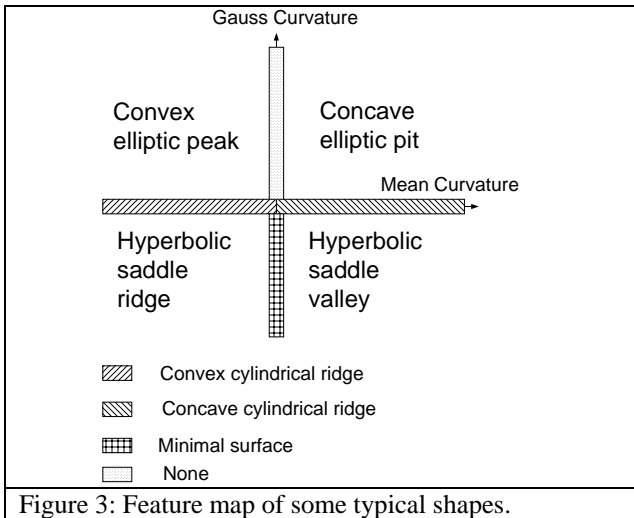


Figure 3: Feature map of some typical shapes.

Since a profile of the histogram may be zigzag due to the discretization of curvature bins, a cubic spline is used herein to interpolate the profile with two reasons. First, the

spline interpolation is adopted to provide second-order derivatives of the profile, which will be used to determine multiple curvature markers of the curvature histogram. Secondly, the spline provides a kind of smoothing to the zigzag pattern of the profile. If  $x$  and  $y$  are used to represent the abscissa and ordinate of the curvature histogram respectively, a cubic spline interpolation of a profile of the histogram can be constructed by a piece-wise polynomial function:

$$y = c_1 y_i + c_2 y_{i+1} + c_3 y_i'' + c_4 y_{i+1}'', \quad x_i < x \leq x_{i+1} \quad \text{and} \quad i = 1, \dots, N_{cb} \quad (1)$$

where  $x_i < x \leq x_{i+1}$  represents a curvature bin or interval of the histogram and  $N_{cb}$  refers to the total number of curvature bins.  $y_i$  is calculated by an integer count of the number of vertices that fall into the curvature interval  $(x_i, x_{i+1})$ .

With the information of  $y_i$  and  $y_i''$  available, thresholds of the local curvature histogram can be determined by the following procedures:

**Algorithm 2.1: Determination of Multiple Markers of the Local Curvature Histogram**

(1) loop over each curvature point,  $x_i$ ,  $i = 1, \dots, m_c$ .

Here, curvature point refers to a certain curvature bin on the abscissa of the curvature histogram.

(1.1) if ( $y_i'' < 0.0$ )

(1.1.1) if the following conditions are satisfied:

$$-y_i'' / \min\{y_i'', y_i'' < 0.0\} > \text{threshold}, \quad (2)$$

$$y_i / \sum_{i=1}^n y_i > \text{threshold}, \quad (3)$$

$$y_i > y_{i-1} \quad \text{and} \quad y_i > y_{i+1}, \quad (4)$$

then

add  $y_i$  as  $\hat{y}_j$  into a histogram\_peak\_list

(2) loop over each  $\hat{y}_j$  in the histogram\_peak\_list,

$j = 1, \dots, m-1$

(2.1) find a valley between  $\hat{y}_j$  and  $\hat{y}_{j+1}$

(2.1.1) create a lowest\_histogram\_list for all  $x_i$ s that are located between  $\hat{y}_j$  and  $\hat{y}_{j+1}$ , and have a lowest value in the direction of y-axis.

(2.1.2) determine which  $x_i$  has a closest distance to the midpoint of two curvature points that correspond to  $\hat{y}_j$  and  $\hat{y}_{j+1}$

(2.1.3) insert this  $x_i$  as  $\hat{x}_k$  into a histogram\_valley\_list

In Algorithm 2.1, Equation (2) is used to find out significant valleys in terms of negative  $y_i''$ , while Equation

(3) makes sure that the corresponding vertex count is significant. Equation (4) filters out all non-peak candidates. The threshold used in Equations (2) and (3) is chosen as 0.05 in this paper, which means that any curvature bin with less than 5% of total contribution will be neglected.

One exception to Equation (4) is a curvature plateau, which corresponds to a ruled surface constructed from a clothoid curve. In this case, the entire plateau is considered as a single peak located at its middle point. However, if a curvature plateau is not a local maximum, it cannot be considered as a peak. Therefore, an additional step should be inserted before Step (1) in Algorithm 2.1 to identify all curvature plateaus that are local maxima, i.e., the plateaus that have a greater value of curvature than adjacent curvature bins at their both ends. Since a plateau is normally not a perfect one, a threshold (5% variation in this paper) is needed to take account of small variation of curvatures at an imperfect plateau.

Note that the output of Algorithm 2.1 is a `histogram_valley_list` that represents a list of curvature markers for different curvature clusters. A precondition for this algorithm is that  $\{\hat{x}_i | i=1, \dots, N_{cb}\}$  should be ordered in the magnitude of curvature. As a result, a postcondition for this algorithm is that  $\hat{x}_k$  in the `histogram_valley_list`  $\{\hat{x}_k | k=1, \dots, m-1\}$  is also ordered. Normally,  $m$  is much smaller than  $N_{cb}$ , and  $m-1$  represents the number of curvature markers for the surface patch.

### 2.2.3 Surface partitioning by $G^2$ discontinuity

The basic strategy in implementing surface partitioning by  $G^2$  discontinuity is to let breath-first searches (i.e., region growing processes) find out partitions on a surface in a sequential manner. Suppose that there are  $m-1$  curvature markers in a `histogram_valley_list` produced by Algorithm 2.1, these markers can be used to divide the abscissa of the curvature histogram into  $m$  regions. For each curvature region in turn from left to right  $\{\hat{x}_k, \hat{x}_{k+1} | k=0, \dots, m-1\}$ , as illustrated in Figure 1, breath-first searches are conducted to start from an element with the nodal curvatures at its all vertices being within the current curvature region  $(\hat{x}_k, \hat{x}_{k+1})$ . During the search propagation across different edges of each element over the surface, the directional nodal curvature in [7] is used as an index to determine if the propagation should continue across the current element edge. If the directional curvature is within the current curvature region  $(\hat{x}_k, \hat{x}_{k+1})$ , propagation continues. Otherwise, it terminates at this element edge.

Now we are ready to introduce an algorithm for multiple partitioning of a surface by using nodal curvatures as follows:

### Algorithm 2.2: Surface Partitioning by Discontinuity of Curvatures

- (1) loop over each surface patch generated by  $G^1$  discontinuity partitioning
  - (1.1) calculate nodal curvatures
  - (1.2) construct a local curvature histogram
  - (1.3) determine makers of the histogram
 
$$\{\hat{x}_k | k=1, \dots, m-1\}$$
  - (1.4) loop over each curvature region
 
$$\{\hat{x}_k, \hat{x}_{k+1} | k=0, \dots, m-2\}$$
 except for the last one  $(\hat{x}_{m-1}, \hat{x}_m)$ 
    - (1.4.1) perform breath-first searches
      - (1.4.1.1) initiate only from an element with directional curvatures at its all vertices along its edges being within the region  $(\hat{x}_k, \hat{x}_{k+1})$
      - (1.4.1.2) propagate across different element edges over the surface until a termination condition is satisfied: directional curvature is outside the region  $(\hat{x}_k, \hat{x}_{k+1})$
      - (1.4.1.3) If there are still some elements unprocessed, go back to (1.4.1.1) to initiate another breath-first search.
  - (1.5) group the remain elements, which do not belong to regions formed by breath-first searches in Step (1.4), into one or more different surface patches by means of their connectivity.

## 3 Numerical Experiments and Discussions

The algorithms introduced in this paper were implemented in VC++ and tested on a Pentium III HP PC. Table 1 shows the execution time and rate on different test meshes. The last column of this table demonstrates the effectiveness of our approach in partitioning a polygonal mesh with a significant number of elements into a much smaller number of geometrically meaningful sub-surfaces by  $G^1$  discontinuity and  $G^1 + G^2$  discontinuities, respectively.

Table 1: Execution time and rate on different test mesh models.

Model name	Vertex	Element	Time (sec)	Rate (elem/s)	Sub-surfaces	
					$G^1$	$G^2$
Knob	2678	5210	2.04	2553.9	7	14
Part06	15675	31294	77.53	403.64	43	46*
Part08	5451	10742	12.27	875.47	17	51*
Bumper	473	432	0.17	2541	4	8

\* refers to the cases where sub-surfaces are obtained from  $G^1$  partitioning plus the generation of planar surfaces.

$G^1$  discontinuity partitioning uses breath-first searches to traverse over the input mesh  $M$  with a time complexity  $O(n)$ , where  $n = \max(N_e, N_v)$ ,  $N_e$  and  $N_v$  are the numbers of elements and vertices in  $M$ , respectively. In the estimation of discrete curvatures at each surface vertices [7], an expensive single value decomposition is applied on each vertex in  $M$ , which costs about  $6^3$  operations for each vertex. Even though the complexity for all vertices is still  $O(n)$ , we have a big coefficient, 216, here. The complexity of Algorithm 2.1 can be represented by  $O(n + N_{cb} * N_s)$ , where  $N_{cb}$  refers to the number of curvature bins of each histogram, and  $N_s$  is the number of sub-surfaces produced by using  $G^1$  discontinuity partitioning. Normally  $N_s$  is much smaller than  $n$  and  $N_{cb}$  is 50 in this paper such that the complexity of Algorithm 2.1 is essentially  $O(n)$ . As to Algorithm 2.2, it costs  $O(n * N_t)$ , where  $N_t$  is the average number of markers of all curvature histogram, and normally ranges from 1 to 3 such that it also takes  $O(n)$ . Thus, overall time cost of the proposed approach is linear time  $O(n)$ .

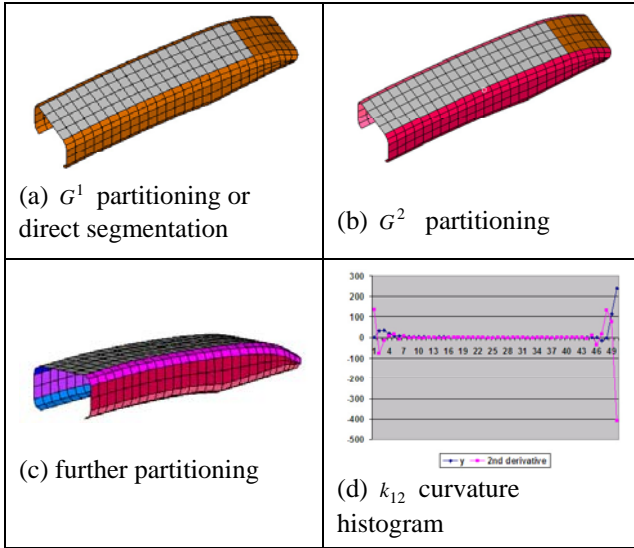


Figure 4:  $G^2$  partitioning of a bumper with irregular shape. (a) two patches generated by  $G^1$  discontinuity partitioning plus the generation of planar surfaces; (b) four sub-surfaces produced by  $G^2$  partitioning; (c) further decomposition by the interactive tool; (d)  $k_{12}$  curvature histogram.

Figure 4(a) shows two sub-surfaces generated by  $G^1$  discontinuity partitioning plus the generation of planar surfaces. It is also the result by using direct segmentation. One main problem with the direct segmentation is that

it can not handle irregular freeform surfaces quite well. Figure 4(b) is the partitioning result as four sub-surfaces by using  $k_{12}$  curvatures for  $G^2$  partitioning. The local curvature histogram is illustrated in Figure 4(c), in which  $y$  refers to the number of vertices in each curvature bin and  $2^{\text{nd}}$  derivative is the one of  $y$  with respect to  $x$  in Equation (1). By means of Equations (2) through (4), there are two significant peaks (one at 2 and the other at 50) and one valley at 25. Figure 4(c) is a result of further  $G^2$  decomposition of two side faces in Figure 4(b) by using an interactive tool. The partitioning result in Figure 4(c) is better than that in Figure 4(a) in terms of facilitating shape morphing or specification of perturbation for shape optimization. In order to maintain the smoothness between patches during shape morphing or perturbation, freeform deformation technique of 3D B-spline solid bounding box should be used.

Figure 5(a) shows the result of  $G^1$  partitioning or the direct segmentation [6], while Figure 5(b) represents a further  $G^2$  partitioning on the basis of Gauss curvature, which is well suited in separating a cylindrical surface from a spherical surface. The concept of feature map [13] can be used to guide our interactive tool to separate different types of surfaces by using the histogram of Gauss or mean curvatures, as illustrated in Figure 3. It is really desirable to have the partitioning result in Figure 5(b) instead of that in Figure 5(a) because it provides more flexibility in the shape morphing or perturbation by using the freeform deformation technique.

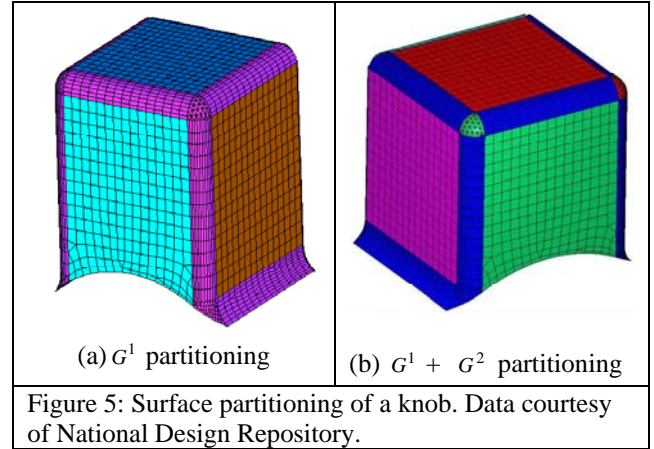


Figure 5: Surface partitioning of a knob. Data courtesy of National Design Repository.

In comparison with existing approaches introduced in Section 1, our approach not only deals with regular shapes like the direct segmentation, but also handles the antisymmetric freeform surfaces as in Figures 4 and 5. Through a careful linkage between the results from  $G^1$  and  $G^2$  partitioning, no over-segmentation is necessarily introduced. Instead, the  $G^2$  partitioning can be used to provide a second layer of information on the basis of the primary layer of  $G^1$  partitioning. This second layer of

information provides extra geometric constraints that should be obeyed from the viewpoint of feature-based design.

## 4 Conclusions

In this paper, we present a new hybrid approach to surface partitioning by an automatic  $G^1$  partitioning and an interactive  $G^2$  partitioning on the basis of feature map with respect to Gauss, mean, maximum principal, minimum principal or  $k_{12}$  curvatures. In addition, an automatic initial estimation of curvature markers for  $G^2$  partitioning is proposed. Overall, we use breath-first searches to conduct surface propagation during a partitioning process with a linear time complexity of  $O(n)$ , where  $n = \max(N_e, N_v)$ ,  $N_e$  and  $N_v$  are the numbers of elements and vertices in  $M$ , respectively.

The results of numerical experiments on various mechanical parts indicate the effectiveness of our approach. The resulting partitions are ready for the applications such as shape morphing or perturbation for optimization. One implicit assumption of this paper is that an input polygonal mesh does not contain a significant amount of noises, which is valid for finite element meshes in engineering analysis.

## Acknowledgement

This study was supported in part by NSF DMI 0514900.

## 5 References

- [1] C. Bajaj and T. K. Dey. Convex decomposition of polyhedra and robustness. *SIAM J.Comp.* 21, 339-364. 1992.
- [2] B. Chazelle, D. Dobkin, N. Shouraboura and A. Tal. Strategies for polyhedral surface decomposition: An experimental study. *Comp.Geom.Theory Appl.* 7, 327-342. 1997.
- [3] S. Katz and A. Tal. Hierarchical Mesh Decomposition using Fuzzy Clustering and Cut. *ACM Transactions on Graphics.* 22[3], 954-961. 2003.
- [4] A. Leonardis, A. Gupta and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision* 14, 253-277. 1995.
- [5] P. Benko, G. Kos, T. Varady, L. Andor and R. R. Martin. Constrained fitting in reverse engineering. *Computer Aided Geometric Design* 19[3], 173-205. 2002.
- [6] T. Varady, P. Benko and G. Kos. Reverse engineering regular objects: simple segmentation and surface fitting procedures. *International Journal of Shape Modeling* 4[3 & 4], 127-141. 1998.
- [7] J. Shen and D. Yoon, *A New Scheme for Efficient and Direct Shape Optimization of Complex Structures Represented by Polygonal Meshes*. *International Journal for Numerical Methods in Engineering* 58, 2201-2223 (2003).
- [8] A. P. Mangan and R. T. Whitaker, *Partitioning 3D surface meshes using watershed segmentation*. *IEEE Transactions on Visualization and Computer Graphics* 5, 308-321 (1999).
- [9] T. Fan, G. Medioni and R. Nevatia. Recognizing 3D objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11[11], 1140-1157. 1989.
- [10] R. B. Fisher, A. W. Fitzgibbon and D. Eggert. Extracting surface patches from complete range descriptions. 148-154. 1997. Ottawa, Canada. Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling.
- [11] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon and R. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18[7], 673-689. 1996.
- [12] C. Robertson, R. Fisher and A. Ashbrook. An improved algorithm to extract surfaces from complete range descriptions. 592-598. 1999. Proceedings of World Manufacturing Conference.
- [13] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys* 34[2], 211-262. 2002.