Mesh Segmentation with Concavity-aware Fields

Oscar Kin-Chung Au   Youyi Zheng   Menglin Chen   Pengfei Xu   Chiew-Lan Tai

Abstract—This paper presents a simple and efficient automatic mesh segmentation algorithm that solely exploits the shape concavity information. The method locates concave creases and seams using a set of concavity-sensitive scalar fields. These fields are computed by solving a Laplacian system with a novel concavity-sensitive weighting scheme. Isolines sampled from the concavity-aware fields naturally gather at concave seams, serving as good cutting boundary candidates. In addition, the fields provide sufficient information allowing efficient evaluation of the candidate cuts. We perform a summarization of all field gradient magnitudes to define a score for each isoline and employ a score-based greedy algorithm to select the best cuts. Extensive experiments and quantitative analysis have shown that the quality of our segmentations are better than or comparable with existing state-of-the-art more complex approaches.

Index Terms—Concavity-aware Field, Mesh Segmentation, Isolines.

1 Introduction

Segmentating 3D models into meaningful parts is a fundamental problem in geometric processing. Many mesh algorithms such as mesh editing, shape retrieval, and object rigging, require shape segmentation as a preprocessing step. However, automatically segmenting 3D models into components that are consistent with human perception is an extremely difficult task due to the lack of semantic information.

As humans generally perceive desirable segmentations at concave creases and seams, which have negative minima principal curvature, this minima rule serves as a basis for many previous automatic segmentation approaches. Most of these approaches extensively use concavity information as the basic setup data in the underlying mathematical model for segmentation [1], [2], [3], [4], [5]. Normal variation, dihedral angle, complex fitting approaches are often employed to measure the concaveness of the shape geometry.

Recently, statistics-driven automatic segmentation algorithms have been proposed, attempting to solve the segmentation problem by combining existing approaches or learning from training data. Promising results have been obtained, but with high computational cost and sophisticated techniques. For instance, the randomized method of Golovinskiy and Funkhouser [4] explicitly leverages several existing segmentation methods to generate a random set of segmentations and measure the likelihood of each edge lying on a segmentation boundary in the randomized set. This approach aims at identifying the most consistent cuts obtained by different automatic segmentation methods and thus eliminate any potential undesired bias of individual method. Disregarding the complexity of any individual methods and the parameters needing to be tuned, combining different approaches already introduces computation and implementation overhead. The data-driven approach of Kalogerakis et al. [5] simultaneously segments and labels parts of 3D models using an objective function learned from a collection of labeled training meshes. This method produces segmentations of better quality than other existing automatic methods. However, the requirement of training data and the complex features used in their method leads to long computation overhead (typically a couple of minutes for moderate sized models.

In this paper, we present a simple, fully automatic mesh segmentation method that exploits the shape concavity information as the key information to achieving high-quality segmentation. Our method relies on a set of concavity-aware scalar fields with large field variation at the concave regions where desirable segmentation boundaries lie. We call these fields *segmentation fields* as they provide concise information for segmentation.

The fields are computed by solving a Laplacian system with a novel concavity-sensitive weighting scheme. We place small capacities on edges with high curvature and concaveness, making the resulting fields exhibit large variation at concave seams located along the propagation paths, thus differentiating these regions from flat and convex ones. To discover all possible concave seams (as segmentation boundaries) on the surface, we compute multiple segmentation fields that propagate along different directions of the model's surface. We design a robust approach to set boundary constraints of the multiple fields such that every concave seam can be discovered by at least one segmentation field.

We consider the isolines of the fields as possible boundary cuts since along an isoline the variation of the field is minimized, thus obeying the minima rule.

• O. K.-C. Au is with the School of Creative Media, The City University of Hong Kong.E-mail: {kincau}@cityu.edu.hk
• Y. Zheng, M. Chen, P. Xu, and C.-L. Tai are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. E-mail: {youyi, menglin, xupengfei, taicl}@cse.ust.hk

Multiple isolines lying along different concave regions are selected from each segmentation field to form a set of candidate cuts. The gradient magnitude of the fields serve as important information to identify the candidate isolines. Specifically, since our fields are designed to exhibit large variations in the concave regions, such regions have large field gradients. This enables the design of a simple selection scheme for choosing the isolines with local maximum gradient magnitudes as candidate cuts. To choose the best final cuts from the candidate set, we again exploit the segmentation fields. Since each field only discovers the concavity information of some parts of the shape, we summarize the gradient magnitudes of all the segmentation fields to determine the quality of each candidate cut. A score is assigned to each cut based on the summarized gradient field and local shape variation. Finally, a simple greedy algorithm is applied to select the set of segmentation cuts based on their assigned scores.

The proposed method is sufficiently easy to implement, involving only solving Laplacian systems and using simple procedures to identify the isoline candidate cuts and select the final cuts. It is computationally efficient because all the segmentation fields are in fact solved from the same linear system with different boundary conditions, thus factorization needs to be performed only once. Further, the method is fully automatic, not requiring the users to adjust any parameters.

We have tested our method extensively and compared our segmentation results with those of the state-of-the-art methods using the Princeton Segmentation Benchmark (PSB) [6]. In terms of segmentation quality, our method outperforms the more complex state-of-the-art randomized cuts method [4] and is comparable to the learning approach of Kalogerakis et al. [5].

2 Related Work

Existing mesh segmentation algorithms can be categorized into two classes based on the types of models they aim to segment. The first class includes methods designed to segment CAD models, mainly for reverse engineering purposes. They segment models into patches, with each patch as the best fit choice from some predefined mathematical surfaces, such as cylinders, spheres and planes. The second class of segmentation algorithm aims at segmenting organic shapes into semantic part components, respecting human understanding of object components. Our method falls into the second class.

Guidance for mesh segmentation algorithms has predominantly been the minima rule proposed by Hoffman et al. [7], [8], which states that human perception tends to cut a shape along concave regions in the direction of minimum principal curvature. Many existing mesh segmentation methods leverage the concavity information as a key measure for the underlying algorithms. For example, the K-mean clustering [9], graph cut-based fuzzy clustering [1], random walk algorithm [3], the primitive-fitting-based method of Attene et al. [10] and spectral clustering methods [11], [12] use normal variance, dihedral angle or pair-wise distance to design the concavity measurements. Note that all these methods use a single set of seeds / cluster centers to directly find the segmentation cuts, thus they depend heavily on the number of seeds / clusters and the initial settings. In addition, these methods [1], [3], [9], [11], [12] often formulate the segmentation problem as optimization problems that directly seek a single segmentation solution, without considering or reusing other potential segmentation solutions. However, single optimization-based methods usually do not suffice to generate high quality segmentation results for all kinds of models. Quantitative experimental results using the Princeton benchmark [6] show that, the randomized cuts method [4] which combines multiple optimization based segmentation algorithms with randomized initial settings can generate better segmentation results.

There are other segmentation methods which use more complex information and analysis tools. For example, methods based on core extraction [2] and critical point analysis [13] use centricity information to determine the prominent parts of the models. The shape diameter function [14] and the part-aware metric [15] are used to compute local visibility information and determine the diameter of the parts of objects. Other methods that use tubular analysis [16] or skeleton-based methods [17] have also been proposed. Kalogerakis et al. [5] employ a conditional random field classifier for simultaneous segmentation and labeling of parts in 3D meshes. It integrates hundreds of informative features such as curvature, shape diameter function, etc, in the training process. Most of these alternative methods involve complex procedures, leading to high computation cost.

Our method resembles the randomized method [4] in that we also select the final cuts from a set of candidate cuts. However, in [4] several existing segmentation methods are used to compute the candidate cuts, thus it is time consuming and depends on the selected underlying segmentation methods. Our method is based only on different segmentation fields solved from the same linear system with different boundary conditions. Multiple potentially desirable candidate cuts on different concave regions can be directly extracted from a single segmentation field, making the process efficient and scalable. In addition, instead of considering a large set of randomized candidate cuts, our method selects the candidate isolines based on maximum concavity variation. Most selected isolines are good candidates for segmentation boundaries and the final selection process picks the best ones among them.

The isolines of smooth scalar fields have been used in many geometric applications. In particular, feature preserving harmonic fields with different Laplacian weighting schemes have been introduced for interactive mesh segmentation [18], [19], [20]. However, to the best of our knowledge, concavity-aware fields have not been exploited. We show in this paper how concavity-aware fields can be applied to automatic segmentation.

3 Our Approach

The key idea of our approach is to use a set of *concavity-aware* segmentation fields to capture concave regions for segmentation. We introduce *inverse Gaussian curvature weighting* to design *concavity-aware* fields that exhibit large field variations along concave and high curvature regions. To faithfully explore the entire model surface for all potential concave regions, we identify a small set of extreme points on the model surface where we set boundary conditions of the Laplacian system defining the fields such that they have with sufficient variation along paths between the extreme points. From each computed segmentation field, we select a set of isolines as potential cuts. All these selected cuts form a pool of candidate cuts, from which, we evaluate, summarize and select the final segmentation cuts.

The key ingredient of our approach is the concavity-aware segmentation fields. This shape analysis tool captures the model's global shape information and serves two purposes. First, isolines sampled from the fields are potential cuts. Second, the field provides sufficient information for evaluating the quality of isolines as good cutting boundaries.

3.1 Concave-aware Segmentation Field

Our method relies on several desirable properties of the segmentation fields. Specifically, we require the fields to be sufficiently smooth and are curvature-sensitive such that denser isolines of the fields can be sampled at the concave regions (where desired segmentation cuts lie). We define the segmentation field as a variant of the surface harmonic field, which is computed by solving the Poisson equation $\Delta\Phi = \mathbf{0}$, where Δ denotes the discrete Laplacian operator [21].

Solving the Poisson equation in the least-squares sense leads to the matrix equation $\mathbf{A}\Phi = \mathbf{b}$, with

$$\mathbf{A} = \begin{bmatrix} \mathbf{L} \\ \mathbf{C} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B} \end{bmatrix}. \tag{1}$$

Here $\mathbf{B}$ and $\mathbf{C}$ are the vector and matrix expressing the boundary conditions, respectively, and $\mathbf{L}$ is the Laplacian matrix:

$$\mathbf{L}_{ij} = \begin{cases} -1 & \text{if } i = j \\ \frac{\omega_{ij}}{\sum_{(i,k)\in\mathbf{E}} \omega_{ik}} & \text{if } (i,j) \in \mathbf{E} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where E denotes the mesh edge set. The behavior of the computed field depends on the weighting scheme for the Laplacian operator, i.e., the w_{ij}. For example, the cotangent-weighting scheme leads to a smooth transiting harmonic field which is well-suited for applications such as deformation and shape approximation. However, such fields can not identify the local shape variation, making them not suitable for segmentation purposes.

Our goal is to induce a field that faithfully captures the concave seams of the given shape. To achieve this, we put small capacities on edges that lie in the concave regions to allow the field to have large variations in these regions. Specifically, we introduce a concave-sensitive weighting scheme defined as follows:

$$\omega_{ij} = \frac{|e_{ij}| \cdot \beta}{G_{ij} + \epsilon}, \tag{3}$$

where $|e_{ij}|$ is the length of the edge (i,j), G_{ij} is the sum of the absolute Gaussian curvature at vertex i and vertex j, ϵ is a small constant to prevent zero division and β is the concave weight of edge (i,j). We use the Gaussian curvature so that the computed fields are propagated following the underlying geometry. We take the inverse of Gaussian curvature such that high curvature regions will have a relatively larger field variation than flat regions.

The concave weight β is the key component in the design of the weighting scheme. The intuition is that smaller values (capacities) will lead to smaller field throughput, hence larger field variation. In our experiments, we set β to a small constant (0.01) when either vertex i or vertex j is a concave vertex, and 1 otherwise. A vertex i is considered concave if there exists at least one adjacent vertex j such that the following inequality holds

$$\frac{(v_i - v_j)}{\|v_i - v_j\|} \cdot (n_j - n_i) > \varsigma. \tag{4}$$

Here n_i and v_i are the normal and position of vertex i respectively and ς is a small constant (1e-3 in our experiments). This weighting scheme ensures that the field has large variation in concave regions, hence leading to the effect of denser isolines gathering at these regions when sampled uniformly (see Figure 1). Note that inverse Gaussian curvature weighting produces segmentation fields that may contain values outside the range of the predefined boundary conditions. Therefore the solved segmentation fields are generally not harmonic fields.

We use the one-ring neighbors of a vertex to determine its concavity. This implies that noise or fine details can influence the determination of the concave vertices and hence cause local vibration in the field propagation. (see Figure 2). A more sophisticated curvature computation method [22] or geometry fitting [23] may be adopted to alleviate the influence of noise and local detail. We took an alternative approach of adding a preprocessing step to smooth the mesh normals before computing the concave vertices (Figure 2 right). We use a fixed 5 iterations of Gaussian smoothing with kernel 0.1 throughout our experiments and find this simple approach to work well.

3.2 Multiple Segmentation Fields

Our segmentation fields have strong differentiating power on components. A single segmentation field can identify multiple components of a given model, as shown in Figure 1. However, a single field cannot discover

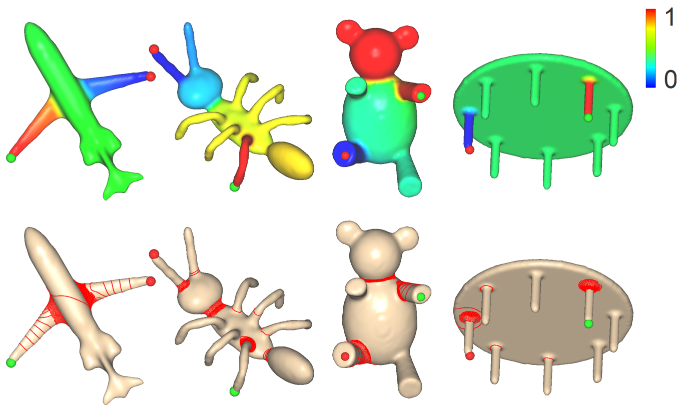


Fig. 1. (Top row) Curvature-aware segmentation fields. The fields are defined by setting boundary conditions at the highlighted extreme points (0 at red point and 1 at green point). Observe that the visualized fields have clear boundaries which coincide with the concave seams. Furthermore, one segmentation field can identify multiple concave seams. (Bottom row) The isolines (50) sampled from the field. Uniform sampling leads to dense isolines at the concave seams.

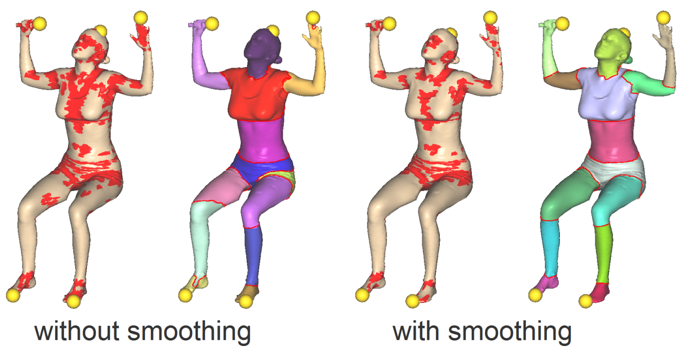


Fig. 2. We apply normal smoothing to more robustly distinguish the concave vertices, leading to better segmentation results.

all components of a model. Therefore it is necessary to define multiple fields with values propagating in different directions in order to detect all possible concave creases and seams on the shape.

Each segmentation field is defined by boundary conditions specified at two surface points. They are mesh vertices located at prominent parts so that the paths connecting them can pass through all possible surface regions where desirable cutting boundaries lie. We call such vertices *extreme points*. A possible way to identify these extreme points is to employ the sophisticated method used in [24] which first computes an average squared geodesic distance field, followed by using an iterative poisson disk sampling scheme to select the shape extremities. However, this method is time consuming and requires tuning of thresholds to control the feature size. We opt for a simpler method for this task. In particular, we leverage the field propagation to detect the extreme points. We observe that by propagating

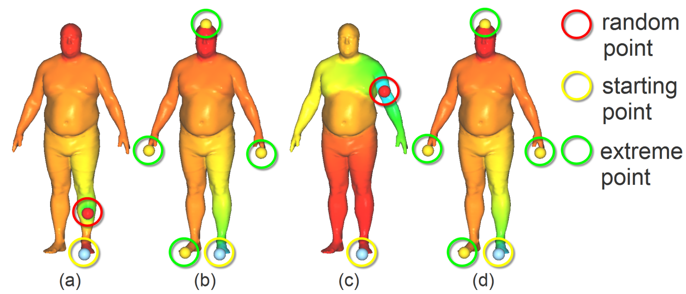


Fig. 3. Automatic identification of shape extremities. (a) From the field computed with a randomly selected point, we select the most prominent point as the start point for computing another field whose local minima and maxima are identified as the shape extreme points (b). (d) shows that a set of similar extreme points are found when a different random point is used (c).

a field from a single point to the rest of the shape, all the prominent parts of an object will have local maximum/minimum field values.

To start, we randomly select a surface point p and compute a field that propagates from p to the rest of the shape by constraining the value at p to be 0, and the values at its one-ring neighbors to be 1. This makes the field propagate from the selected point in a breadth-first manner. We consider the point q with maximum field value as the most prominent point of the shape. To make the selection robust against the random point selected, we then use q as the new starting point and compute a field again based on it. All the points with local maximum/minimum field values (i.e., those with the largest or smallest field value among its one-ring neighbors), including point q, are then selected as the extreme points. These selected points are mostly located at prominent parts of the shape due to the propagation behavior.

The above process typically suffices to identify all shape extremities for most models. However, for shapes without obvious extruding components (e.g., the fish model in Figure 10), the above process may not capture all the necessary extremities. To ensure that sufficient number of extreme points are discovered as boundary constraints, we repeat the above selection process with another random starting point, and filter out any newly detected point if it is close to an existing one measured by the geodesic distance. In particular, we ignore the newly found extreme point if it is within $0.15L$ of any existing extreme point, where L is the longest geodesic distance between any two detected extreme points. The iteration stops if no new extremity is found. From our experience, 1 or 2 such iterations suffice to discover all extreme points in most cases. In Figure 3, we show that the extremity identification process is robust against different random starting points.

Denote the set of extreme points as $\mathbf{U}$. For each pair of extreme points (u, v) in $\mathbf{U}$, we compute a corresponding

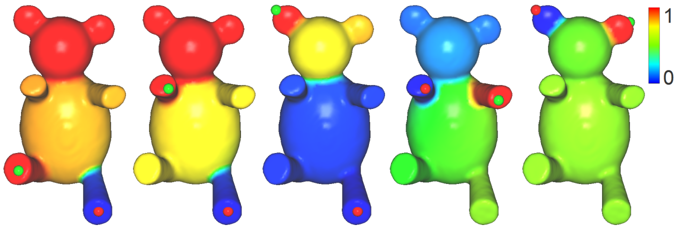


Fig. 4. Five (out of a total of C_2^6) segmentation fields for the teddy model. Each field is defined by the two highlighted extreme points (0 at red point and 1 at green point). Note that all components can be faithfully differentiated by these fields.

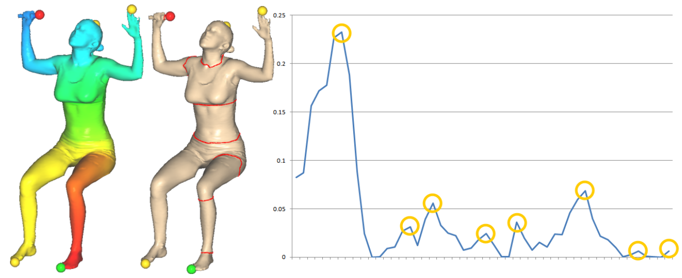


Fig. 5. (Left) The segmentation field computed with the extreme points at the left foot and the right hand. (Middle) The candidate isolines found from that field. (Right) The candidate isolines correspond to local maxima (circled) in the histogram of the gradient magnitudes of the sampled isolines.

segmentation field Φ_{uv} by constraining the value of one point to be 1 and the other 0. These segmentation fields serve as the basis for the subsequent segmentation process. Note that there are a total of C_2^n fields, where n is the number of extreme points. Since n is usually a small number, the field computation is very efficient using the factorization updating scheme of [25].

Figure 4 shows some segmentation fields of a model. Observe that, within each component of the model, the color is mostly constant. Moreover, there are clear boundaries between the different colored regions and these boundaries coincide with the concave creases, demonstrating the effectiveness of our weighting scheme in identifying concave creases.

3.3 Candidate Isolines Selection

Our next goal is to locate candidate cutting boundaries at concave regions (Figure 1, 4). Following the minima rule, we consider the isolines of the fields as the potential boundary cuts. Along an isoline, the field variation is minimized, so is the curvature variation.

Since concave regions have large field variation, we exploit the field gradient magnitudes for selection of candidate cuts. In each field Φ, we first uniformly sample a set of isolines $\mathscr{I} = \{I_1, I_2, \cdots, I_\tau\}$. and compute the gradient magnitude $\mathcal{G}_{I_i}$ of each isoline.

We assume that the segmentation field has linear variation on each face, i.e., each face has a constant gradient magnitude. Denote the gradient field of the segmentation field Φ as $\mathcal{G} = \{g_i | f_i \in \mathbf{F}\}$, where g_i is the gradient magnitude of the face f_i in the segmentation field. We define the gradient magnitude $\mathcal{G}_{I_i}$ for an isoline I_i as the normalized average gradient magnitude of all the faces through which it passes:

$$\mathcal{G}_{I_i} = \frac{\sum_{j \in \hat{F}_i} \hat{\ell}_j g_j}{\sum_{j \in \hat{F}_i} \hat{\ell}_j}, \tag{5}$$

where $\hat{F}_i$ is the set of faces that isoline I_i passes through, $\hat{\ell}_j$ is the length of the isoline segment within the face f_j, and g_j is the gradient magnitude of face f_j of the segmentation field Φ.

Isolines lying on concave regions can now be easily identified by their large gradient magnitudes. However, since there are denser isolines at concave regions, there will be multiple isolines in the same region with large gradient magnitudes. Our idea is to pick the isoline with the local maximum gradient magnitude in each concave region as a candidate cut. We introduce a simple histogram filtering algorithm for this purpose. Specifically, we obtain a histogram of the gradient magnitude of all isolines of each field (see Figure 5 for an example) and consider an isoline as a candidate cut if its gradient magnitude is a local maximum, that is, we select the isoline I_i if $\mathcal{G}_{I_i} > \mathcal{G}_{I_k}, k \in \{i-2, i-1, i+1, i+2\}$.

For models with several nearby concave regions forming a consecutive concave seam, lengthy isolines may pass through all the concave regions (see the insect model in Figure 8 left). Such isolines are not good boundary cuts but they may have large local gradient magnitude due to the high concavity of the region on which they lie. To filter out these loose and lengthy isolines, we reject an isoline as a candidate cut if it is η (=1.5) times longer than any other isoline that passes through any common face.

For models with no clear prominent parts, such as the cup model in Figure 8 and some CAD models, the improper location of shape extremities may introduce candidate isolines on some flat patch-type regions (Figure 8). Recall that our method is mainly designed for detecting concave regions of part-type components, so candidates on flat patches should be rejected. We employ a simple heuristic approach to eliminate them. We measure the flatness of an isoline by analyzing the portion of face normals of the isoline which lie on the same semi-sphere and reject the isoline if the portion exceeds a certain range τ (=80%).

The selection process identifies a set of candidate isolines from each segmentation field (Figure 6). However, the isolines selected from different fields may overlap and lie in the same regions. In the next subsection, we introduce a summarization process to select the best final cuts from the candidate cuts.

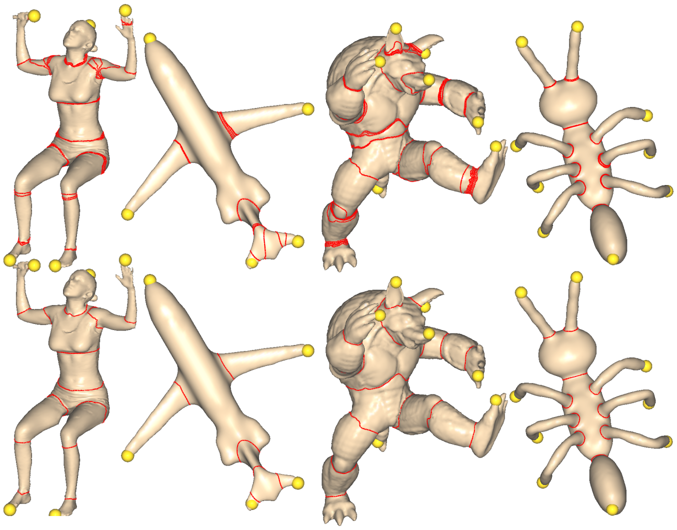


Fig. 6. (Top row) Our algorithm successfully locates the candidates isolines at desired boundary regions. (Bottom row) The final cuts selected from the candidates isolines.

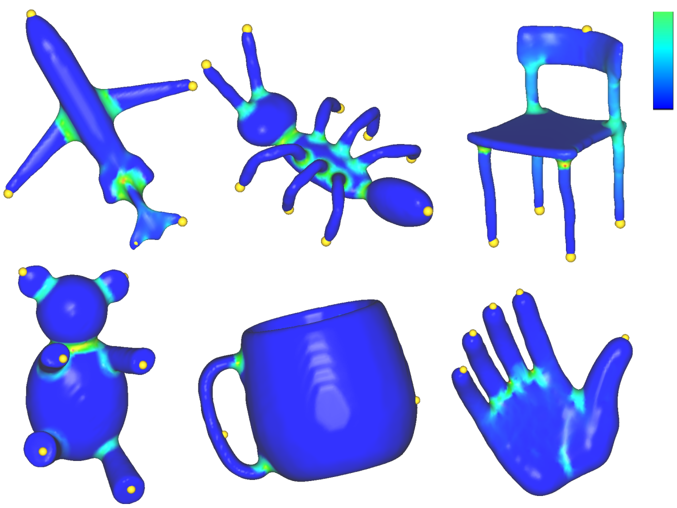


Fig. 7. Summing up the normalized gradients of all fields reveals a set of regions with clear boundaries.

3.4 Final Cuts Selection

Denote the collection of candidate cuts selected from all fields as $C = \{c_1, c_2, \cdots, c_M\}$. We assign a score to each of the candidates representing its quality as a cut boundary. For this purpose, we again exploit the field gradient magnitude. A face with a large gradient magnitude in any segmentation field means it is likely to lie in a concave region between two components of the object. Hence, we perform a summarization process on all the field gradients for each face. Specifically, we define a score S_i for each face f_i as the sum of the face's normalized gradient magnitudes in all the segmentation fields:

$$S_i = \sum_{(u,v)} (\bar{g}_i^{uv}) \tag{6}$$

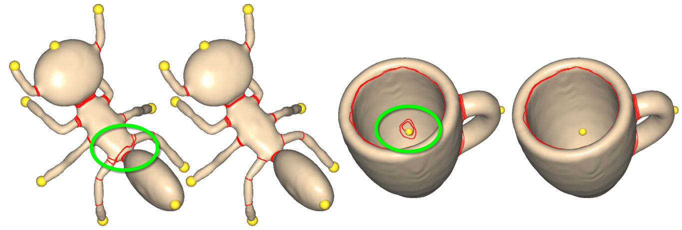


Fig. 8. Filtering candidate isolines. (Left) reject loose and lengthy candidate isolines. (Right) reject candidate isolines lying on flat patches.

$\bar{g}_i^{uv}$ denotes the normalized gradient magnitude (by the largest gradient value in that field) of face i in the segmentation field Φ_{uv}. The normalization gives each field equal importance. Figure 7 visualizes the summed gradient field. We can see that large values appear consistently at the desired cutting regions, which again demonstrate the differentiating power of our concavity-aware segmentation fields.

With the scores of all faces, we next compute the *gradient score* $s_{g,i}$ for each candidate cut c_i as the average weighted sum of the scores of all faces through which the candidate cut passes:

$$s_{g,i} = \frac{\sum_{j \in \hat{F}_i} \hat{\ell}_j S_j}{\sum_{j \in \hat{F}_i} \hat{\ell}_j}. \tag{7}$$

The gradient score $s_{g,i}$ only considers the local concavity *along* the candidate cut, but not the local shape variation *across* the cut. To make the measure more robust against local shape variation, we introduce a *shape variation score* $s_{v,i}$ that measures the local shape variation across the candidate cut. Specifically, for each candidate cut c_i, we extend its face strip (the faces that it passes through) on both sides by k steps. This leads to k face strips on each side. Denote the length of a candidate isoline i as l_i, we compute the shape variation score $s_{v,i}$ in a multi-scale manner similar to [20]:

$$s_{v,i} = \sum_{j=1}^{k} e^{-\frac{j^2}{2\sigma^2}} \Delta_j^i \tag{8}$$

where Δ_j^i is defined as follows:

$$\Delta_j^i = \frac{l_{ij}^l + l_{ij}^r - 2l_i}{l_{max}}, \tag{9}$$

with l_{ij}^l and l_{ij}^r denoting the length of the jth strips to its left and right, respectively, and l_{max} is the maximum length of the candidate cuts in C. The multi-scale convolution using Gaussians makes the measure insensitive to the choice of k [20]. In practice, we select similar parameters as in [20] ($k = 5$ and $\sigma = 2$). Although more accurate computation with geodesic rings propagating from each side can be employed, we found this simple setting to work well in experiments. Note that we do not directly use the approach in [20] for s_v because in

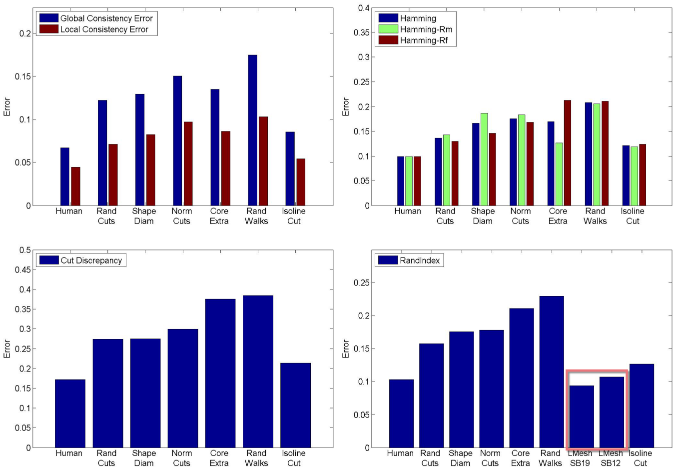


Fig. 9. Evaluation of segmentation. Evaluations using the protocols of [6] and the human segmentations in the Princeton Segmentation Benchmark. Our method outperforms existing state-of-the-art non data-driven automatic segmentation methods. The bottom right graph includes a comparison with the approach of Kalogerakis et al. [5] in terms of Rand Index error. SBx refers to x number of models in the training set per category.

	Human	Rand Cuts	Shape Diam	Norm Cuts	Core Extra	Rand Walks	SB19	SB12	Isoline Cut
Average	10.3	15.7	17.6	17.8	21.1	22.9	9.4	10.7	12.7
Human	13.5	15.8	17.9	18.2	22.5	29.5	11.9	12.9	12.3
Cup	13.6	22.4	35.8	23.6	30.7	33.4	9.9	9.9	21.1
Glasses	10.1	9.7	20.4	14.2	30.1	31.6	13.6	14.1	9.8
Airplane	9.2	11.5	9.2	18.6	25.6	26.1	7.9	8.2	12.7
Ant	3.0	2.5	2.2	4.7	6.5	6.8	1.9	2.2	3.9
Chair	8.9	18.9	11.1	9.3	18.7	16.7	5.4	5.6	12.1
Octopus	2.4	6.7	4.5	6.3	5.1	6.9	1.8	1.8	4.1
Table	9.3	37.4	18.4	9.8	24.4	13.9	6.2	6.6	6.5
Teddy	4.9	4.5	5.7	12.1	11.4	12.7	3.1	3.2	5.3
Hand	9.1	9.7	20.2	15.6	15.5	22.2	10.4	11.2	11.5
Plier	7.1	10.9	37.5	18.3	9.3	23.0	5.4	9.0	7.3
Fish	15.5	29.7	24.8	39.9	27.3	40.6	12.9	13.2	24.3
Bird	6.2	11.4	11.5	21.2	12.4	28.0	10.4	14.8	9.7
Armadillo	8.3	8.1	9.0	12.0	14.1	10.7	8.0	8.4	10.6
Bust	22.0	25.1	29.8	33.2	31.5	33.5	21.4	22.2	24.4
Mech	13.1	28.3	23.8	17.5	38.7	24.4	10.0	11.8	12.2
Bearing	10.4	12.9	11.9	17.9	39.8	27.1	9.7	17.6	17.7
Vase	14.4	16.0	23.9	26.8	22.6	28.7	16.0	17.1	16.8
FourLeg	14.9	17.7	16.1	18.9	19.1	20.8	13.3	13.9	18.1

TABLE 1
Per-category Rand Index errors of segmenting the PSB using our method and previous methods

our case the nearby isolines in the same field may be too close or too far away depending on the local concavity.

The final score assigned to each of the candidate cuts is defined as

$$s_i = s_{g,i} \times s_{v,i}. \tag{10}$$

Note that this score could be positive or negative. A higher score means a higher chance that the candidate cut is a desirable final cut. Next, we iteratively select a candidate cut c_i with the highest score from the candidate set C to form the final boundary set Γ. To avoid selecting nearby or overlapping cuts in the same boundary region, we reject a new cut if it introduces a new segment whose area compared to the object total area is smaller than a certain ratio χ. That is, this parameter controls the minimum scale of cut-out segments. Since it is mainly designed to filter out the intersecting or overlapping candidate cuts, we simply set χ to 0.02.

The process stops either when C is empty or the score of the newly selected candidate cut shows a sudden decrease compared to the previous one, i.e., when the score compared to the previous cut is smaller than a ratio (0.1 in our experiments). The sudden decrease in score means that the newly found one has very low concavity, hence it is safe to reject it and stop the iteration.

3.5 Parameter Analysis

In this section, we analyze the parameters used in our automatic segmentation method. We did not try to optimize the parameters, instead we fixed them at specific values that we observe to produce fairly good results. For example, for the parameters used in the normal smoothing part, we could have used more sophisticated approaches as in [26] to adaptively optimize the Gaussian kernel. Instead, for simplicity and efficiency, we opt for a fixed kernel approach which works well for a large variety of models tested in PSB (Figure 10) or other common data sets (Figure 12). For the parameters used in candidate cuts selection (i.e., the parameter to reject a lengthy isoline, the parameter to measure the flatness of an isoline) and in the final cut selection (i.e., the parameter to reject overlapping isolines and the parameter to stop the selection), we tested the following parameter ranges and found that the results are largely the same: τ in 0.8-0.9, χ in 0.02-0.05, η in 1.5-2.0.

We evaluated two important parameters in our experiments. The first one is the concave weight β which is designed to increase the segmentation field variation in concave regions. Smaller β values cause the isolines to appear more densely in the concave regions (see Figure 14). However, too small values (e.g. β = 0.001) lead to an ill-conditioned linear system. For all examples in this paper, we set β = 0.01.

The second important parameter is the number of isolines extracted from each field. Our concavity-aware segmentation field induces denser isolines at concave seams, however sampling more isolines does not necessarily improve the quality of the segmentation results. It suffices to set the number of isolines to be large enough to distribute isolines on all concave regions along paths between extreme points. We tested 20 through 200 isolines per field for the models in Figure 10 and observed that sampling >100 isolines per field leads to undesirable cuts included as candidate cuts (due to obscure local maximum gradients caused by shape noise and variation) and degraded segmentation results (e.g., Figure 15). We fix our sampling at 50 isolines for all our experiments.

4 Results and Discussion

Results. We have applied our system to segment a wide variety of models, including the whole set of models in the Princeton Segmentation Benchmark (see Figure 10) and other models with complex structures (see Figure 12). For most models, our automatic segmentation

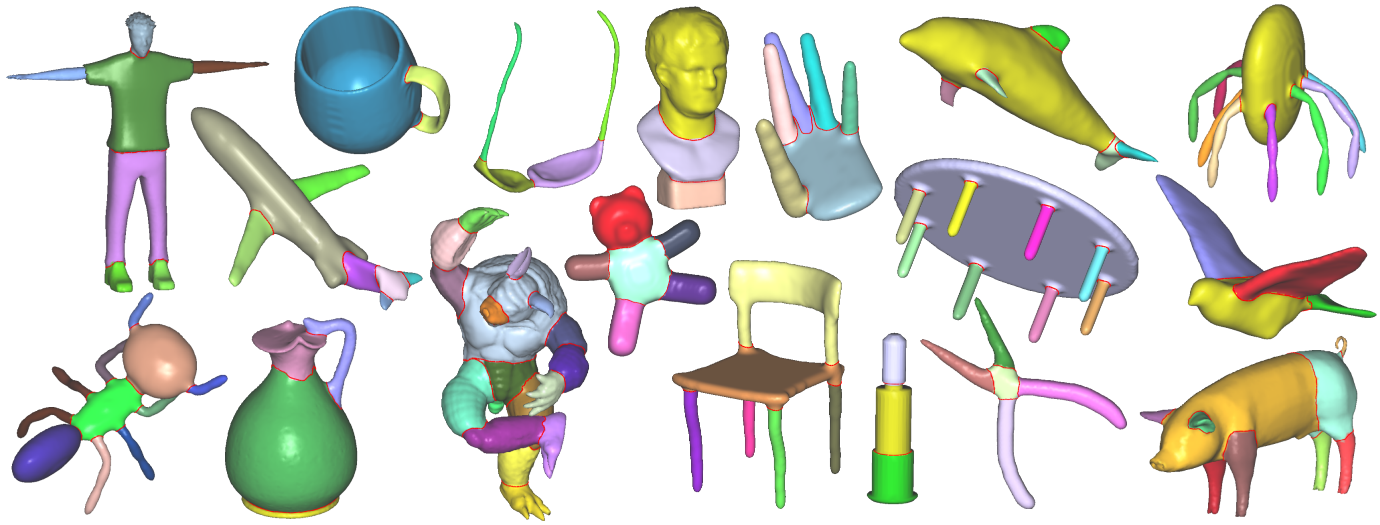


Fig. 10. Segmentation results of representative models in the Princeton Segmentation Benchmark. All results are obtained fully automatically. It can be observed that our segmentation boundaries follow naturally the concave geometry.

method is able to generate boundary cuts that follow the shape geometry well.

We tested all 19 categories of 380 models in the Princeton Segmentation Benchmark. Our method performs well in all categories, especially for the categories that contain models with obvious protruding components, such as bear, ant, table, birds and octopus. We evaluated the segmentation results using the proposed protocol in [6]. Figure 9 shows the quantitative results of our method compared with other automatic segmentation methods. The results show that our simple and efficient method consistently outperforms existing state-of-the-art methods, including complex methods like randomized-cuts [4]. We also compared the Rand Index error of our results with those reported in [5] (see Figure 9, bottom right). Our method performs slightly worse than theirs, however note that their approach involves complicated training-based learning which integrates numerous informative features and user labeling input. In terms of computation requirements, our approach is much more efficient. For a typical model of 100K triangles, the computation of shape extremities and the updating of all segmentation fields takes less than 10 seconds, and the rest of the steps take less than 2 seconds, on an Intel 2.2GHZ laptop with 2GB memory.

Table 1 reports the Rand Index errors of our method and previous methods across all categories in the PSB. Note that our method has similar errors across all categories, demonstrating its stability and general segmentation ability. Other methods have variant performance in different categories, for example, note the large Rand Index error in the table category for the randomized cut method [4] and the plier category for the shape-diameter method [14].

Sensitivity to mesh tessellation. Since our concavity-aware segmentation fields are built upon local curvature

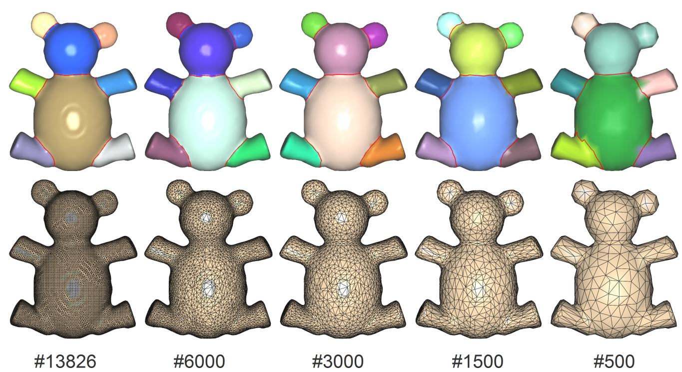


Fig. 11. Our approach is largely insensitive to mesh tessellation. However, very coarse tessellations could significantly influence the estimation of concavity and curvature which will affect the segmentations (see one ear of the rightmost teddy).

and concaveness, it is interesting to see how different mesh tessellations affect the segmentation results. We have tested different examples and found that our method is largely insensitive to different tessellations. We conjecture that this insensitivity is due to the fields being solved for in a global sense. Figure 11 shows an example. It can be seen that the results for different tessellations are almost the same until the mesh becomes very coarse and significantly affects the estimation of curvature and shape concavity. Applying a more robust estimation scheme that builds on multi-scale surface fitting may solve this problem.

Comparison with other weighting schemes. We implemented other common Laplacian weighting schemes [18], [19] and tested them with our framework. Figure 13 shows the resulting isoline distributions and the final segmentations of a typical part-based model

Fig. 12. Segmentation results of complex shapes.

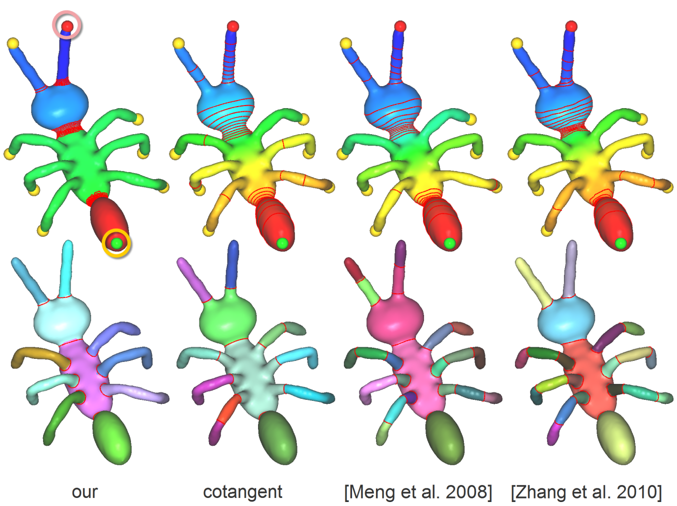


Fig. 13. Our concavity-sensitive weighting scheme produces denser isolines at concave regions compared with other weighting schemes, hence leads to better segmentation results.

using the various weighting schemes. We can see that the existing weighting schemes produce isolines that are more uniformly distributed on the mesh surface, making it difficult to extract candidate cuts simply based on the density of the isoline distribution. In contrast, our concavity-sensitive weighting scheme produces isolines that are gathered at the concave regions, leading to better segmentation results (Figure 13 bottom).

Limitations. The computation of the Gaussian curvature and the concaveness relies on local shape information. Thus our method is inappropriate for highly noisy models or models with many fine surface details (e.g., Figure 16 left). More sophisticated curvature computation methods [22], [23] could alleviate the sensitivity to noise and surface detail but will not solve the problem. For efficiency, we chose to apply a preprocessing step to smooth the vertex normals and a simple way to estimate the local concaveness.

In some cases, when a boundary region does not contain sufficient concavity to cause a sudden decrease in the throughput of the field propagation, our method

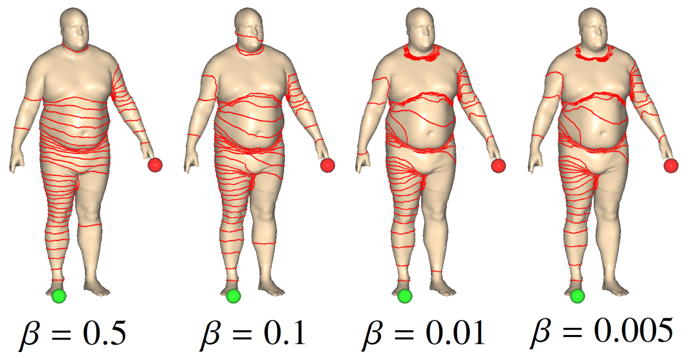


Fig. 14. The effect of concave weight parameter β. Smaller β produces denser isolines at concave seams.

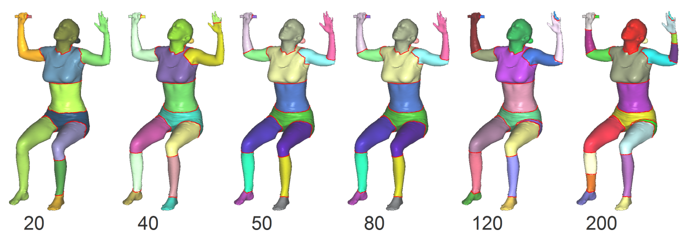


Fig. 15. Effect of different number of isolines per field on segmentation results. Increasing the number of isolines leads to more candidates being detected. However, over-sampling may cause undesirable candidates being selected due to shape noise and variation.

may fail to detect any candidate isoline in that region (Figure 16, middle). This is a limitation of our method, which may result in inconsistent segmentations for models in different poses (e.g., different models in the same category of PSB) due to variations in local geometry. For example, our method fails to detect a cut along one leg of the female model in Figure 16. Nevertheless, evaluation results show that our method still gives low per-category rand index error (Table 1).

Our method is designed for segmenting protruding components separated by concave creases and seams. It is not suitable for segmenting parts or patches with non-concave boundaries such as a diamond shape or facets in CAD models (Figure 16, right). Finally, our method relies solely on shape concavity information, without any semantic or hierarchical information, thus it cannot generate segmentation hierarchies. Generating segmentation hierarchies that truly agree with shape semantics remains a challenging problem.

5 Conclusion

We present a simple, fully automatic mesh segmentation method that directly relies on shape concavity information. The proposed concavity-aware segmentation fields capture the shape geometry along local concave creases and seams where desirable segmentation boundaries lie. Exploiting the information provided by these segmentation fields, we can easily sample a set of candidate isolines at concave regions and select the final segmentation cuts. Our method produces high-quality segmentation results better than more complex existing methods and comparable with the state-of-the-art data-driven approach. For future work, a possible extension is to improve the segmentation quality by considering more informative measures such as the Shape Diameter Function (SDF) in the design of the segmentation fields. Finally, we believe that the differentiating power of the proposed concavity-aware segmentation fields offers an effective means for understanding shapes, making the fields applicable to other geometry processing applications, such as skeleton extraction [1], region selection [27], shape tagging and shape retrieval.

Acknowledgement

We would like to thank the anonymous reviewers for their constructive comments and Shauna Dalton for proofreading the final version of the paper. This work was supported in part by grants from the City University of Hong Kong (Project No. 7200148) and the Hong Kong Research Grant Council (Project No. GRF619611).

References

[1] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts," *ACM Transactions on Graphics (Proceedings SIGGRAPH 2003)*, vol. 22, no. 3, pp. 954–961, 2003.

[2] S. Katz, G. Leifman, and A. Tal, "Mesh segmentation using feature point and core extraction," *The Visual Computer (Proceedings of Pacific Graphics 2005)*, vol. 21, no. 8-10, pp. 865–875, 2005.

[3] Y.-K. Lai, S.-M. Hu, M. R., and R. P. L., "Rapid and effective segmentation of 3d models using random walks," *Computer Aided Geometric Design*, vol. 26, no. 6, pp. 665–679, 2009.

[4] A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3D mesh analysis," *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, vol. 27, no. 5, 2008.

[5] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D Mesh Segmentation and Labeling," *ACM Transactions on Graphics*, vol. 29, no. 3, 2010.

[6] X. Chen, A. Golovinskiy, , and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Transactions on Graphics (Proceedings SIGGRAPH 2009)*, vol. 28, no. 3, 2009.

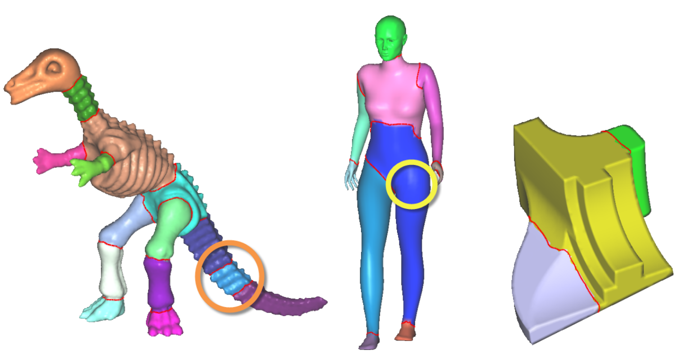


Fig. 16. Limitations of our method. (left) Our method produces over segmentations when the model contains too many concave creases and seams; (middle) Our method may not detect candidate cuts in regions where the concavity fails to inhibit the field propagation; (right) Our method is not suitable for patch-type segmentation.

[7] D. Hoffman, W. Richards, A. Pentl, J. Rubin, and J. Scheuhammer, "Parts of recognition," *Cognition*, vol. 18, pp. 65–96, 1984.

[8] D. D. Hoffman and M. Singh, "Salience of visual parts," *Cognition*, vol. 63, no. 1, pp. 29–78, 1997.

[9] S. Shlafman, A. Tal, and S. Katz, "Metamorphosis of polyhedral surfaces using decomposition," *In Eurographics*, pp. 219–228, 2002.

[10] M. Attene, B. Falcidieno, and M. Spagnuolo, "Hierarchical mesh segmentation based on fitting primitives," *Vis. Comput.*, vol. 22, no. 3, pp. 181–193, 2006.

[11] H. Zhang and R. Liu, "Mesh segmentation via recursive and visually salient spectral cuts," in *Proc. of Vision, Modeling, and Visualization*, 2005, pp. 429–436.

[12] R. Liu and H. Zhang, "Mesh segmentation via spectral embedding and contour analysis," *Computer Graphics Forum (Special Issue of Eurographics 2007)*, vol. 26, no. 3, pp. 385–394, 2007.

[13] H.-Y. S. Lin, L. H.-Y.M., and J.-C. Lin, "Visual salience-guided mesh decomposition," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 46–57, 2007.

[14] L. Shapira, A. Shamir, , and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," *Vis. Comput.*, vol. 24, no. 4, pp. 249–259, 2008.

[15] R. Liu, H. Zhang, A. Shamir, and D. Cohen-Or, "A part-aware surface metric for shape analysis," *Computer Graphics Forum (Special Issue of Eurographics 2009)*, vol. 28, no. 2, pp. 397–406, 2009.

[16] M. Mortara, G. Patan'e, M. Spagnuolo, B. Falcidieno, and J. Rossignac, "Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies," *In ACM symposium on Solid modeling and applications*, pp. 339–344, 2004.

[17] D. Reniers and A. Telea, "Skeleton-based hierarchical shape segmentation," in *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, 2007, pp. 179–188.

[18] M. Meng, Z. Ji, and L. Liu, "Sketching mesh segmentation based on feature preserving harmonic field," *Journal of Computer-Aided Design & Computer Graphics*, vol. 20, no. 9, pp. 1146–1152, 2008.

[19] J. Zhang, C. Wu, J. Cai, J. Zheng, and X. cheng Tai, "Mesh snapping: Robust interactivemesh cutting using fast geodesic curvature flow," in *Computer Graphics Forum (In Proc. of Eurographics 2010)*, vol. 29, no. 2, 2010, pp. 517–526.

[20] Y. Zheng and C.-L. Tai, "Mech decomposition with cross-boundary brushes," in *Computer Graphics Forum (In Proc. of Eurographics 2010)*, vol. 29, no. 2, 2010, pp. 527–535.

[21] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," 1999.

[22] E. Kalogerakis, P. Simari, D. Nowrouzezahrai, and K. Singh, "Robust statistical estimation of curvature on discretized surfaces," in *Proceedings of the Eurographics/ACM Siggraph Symposium on Geometry Processing*, 2007.

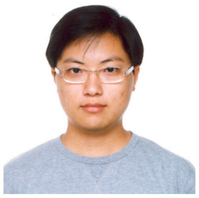
[23] Y.-K. Lai, Q.-Y. Zhou, S.-M. Hu, J. Wallner, and H. Pottmann, "Robust feature classification and editing," *IEEE Trans. Vis. Comp. Graphics*, vol. 13, no. 1, pp. 34–45, 2007.

[24] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi, "Deformation-drive shape correspondence," *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing 2008)*, vol. 27, no. 5, pp. 1431–1439, 2008.

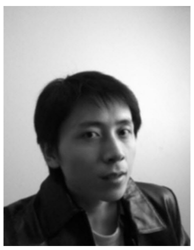
[25] T. Davis, *User Guide for CHOLMOD: a sparse Cholesky factorization and modification package*, 2008.

[26] R. Zatzarinni, A. Tal, and A. Shamir, "Relief analysis and extraction," in *ACM SIGGRAPH Asia 2009 papers*, ser. SIGGRAPH Asia '09, 2009, pp. 136:1–136:9.

[27] M. Attene, M. Mortara, M. Spagnuolo, and B. Falcidieno, "Hierarchical convex approximation of 3d shapes for fast region selection." *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1323–1332, 2008.

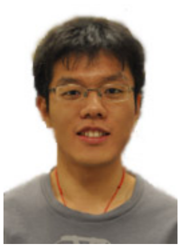


Oscar Kin-Chung Au obtained the PhD, MPhil. and BEng degrees, all from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. He is an assistant professor at the School of Creative Media, the City University of Hong Kong. His research interests include geometric modeling and processing, computer graphics, and interactive techniques.



Youyi Zheng is a Ph.D. candidate at the Department of Computer Science & Engineering, Hong Kong University of Science & Technology. He received his B.Sc. and M.Sc. in Mathematics both from the Zhejiang University. His research interests include digital geometric processing, computer graphics, and interactive techniques.

Menglin Chen obtained his MPhil degree from the Department of Computer Science & Engineering, Hong Kong University of Science & Technology. He received his B.Sc. and M.Sc. in Software Engineering from the Department of Computer Science in the Zhejiang University. His research interests include digital geometric processing and computer graphics.



Pengfei XU is a Ph.D. candidate at the Department of Computer Science & Engineering, Hong Kong University of Science & Technology. He received his B.Sc. in Mathematics from the Zhejiang University. His research interests include digital geometric processing and computer graphics.

Chiew-Lan Tai is currently an Associate Professor of Computer Science at the Hong Kong University of Science and Technology. Her research interests include geometric modeling and processing, computer graphics, and interactive techniques. She received her BSc degree in Mathematics from the University of Malaya, MSc in Computer & Information Sciences from the National University of Singapore, and DSc degree in Information Science from the University of Tokyo.