# Mesh Segmentation via
# Spectral Embedding and Contour Analysis

Rong Liu and Hao Zhang

GrUVi Lab, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

## Abstract

*We propose a mesh segmentation algorithm via recursive bisection where at each step, a sub-mesh embedded in 3D is first spectrally projected into the plane and then a contour is extracted from the planar embedding. We rely on two operators to compute the projection: the well-known graph Laplacian and a geometric operator designed to emphasize concavity. The two embeddings reveal distinctive shape semantics of the 3D model and complement each other in capturing the structural or geometrical aspect of a segmentation. Transforming the shape analysis problem to the 2D domain also facilitates our segmentability analysis and sampling tasks. We propose a novel measure of the segmentability of a shape, which is used as the stopping criterion for our segmentation. The measure is derived from simple area- and perimeter-based convexity measures. We achieve invariance to shape bending through multi-dimensional scaling (MDS) based on the notion of inner distance. We also utilize inner distances to develop a novel sampling scheme to extract two samples along a contour which correspond to two vertices residing on different parts of the sub-mesh. The two samples are used to derive a spectral linear ordering of the mesh faces. We obtain a final cut via a linear search over the face sequence based on part salience, where a choice of weights for different factors of part salience is guided by the result from segmentability analysis.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations

## 1. Introduction

The problem of segmenting a 3D shape into meaningful parts has received a great deal of attention lately due to its wide ranging applications in geometry processing [Sha06]. The challenges we face are at least two-fold. On one hand, cognitive studies on the segmentability of a shape and the process with which a human performs shape segmentation are still on-going. While from the perspective of geometry processing, translating the relevant results from cognitive research into realizable segmentation algorithms and achieving robustness and efficiency amid complexities introduced by 3D data have both proven to be non-trivial tasks.

In this paper, we propose a top-down mesh segmentation algorithm which recursively bisects a mesh piece (sub-mesh) based on *part salience*, a concept proposed from cognitive studies [HS97]. The key characteristic of our approach is the transformation of relevant 3D problems to *2D and 1D domains*, via *spectral embedding*, so that the solution to these problems is reduced to contour and sequence analysis.

Part salience combines part size, protrusiveness and cut strength to qualify the visual salience of a part on a shape. However, no discussion on how the three factors should be combined has been given so far. We stipulate that the answer would depend on the nature of the segmentation in question; this is one of the issues we address. A second consideration is how to compute a salient cut for a sub-mesh. To this end, we rely on a 1D embedding of the faces of the sub-mesh which would allow us to perform a linear search along the face sequence to detect the most salient cut.

A high-quality 1D mesh embedding should retain as much information about the original mesh geometry as possible and it should be quick to obtain. To accomplish this, we compute a geometry-aware and locality-preserving spectral embedding efficiently using the Nyström method as explained in Section 5. Instead of finding samples for Nyström approximation and determining the part *segmentability* (to serve as a stopping criterion for segmentation) on the original mesh, we spectrally project the sub-mesh, using its graph Laplacian
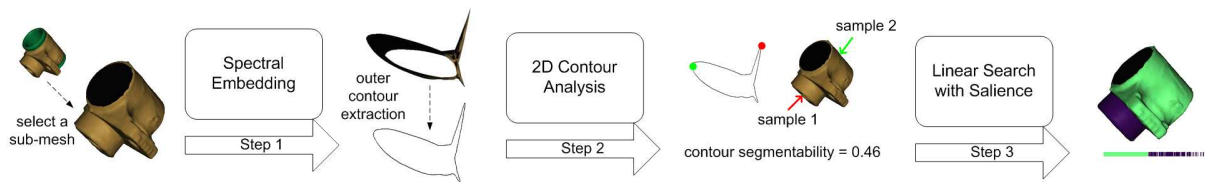
**Figure 1:** *An iteration of our algorithm consists of three steps. In Step 1, from the pool of sub-meshes to segment, the one with the largest surface area is spectrally embedded in 2D with a contour extracted from the outline of the embedding. Step 2 tests contour segmentability and, for a pass, computes two samples (shown as red and green dots), that are mapped to two faces on the sub-mesh. These two faces are fed to Step 3, where the mesh faces are linearly arranged. A search along the sequence finds the position with the highest salience score, resulting in a cut. The resulting sub-meshes are inserted back to the pool.*

$\mathcal{L}$, into a plane and perform segmentability analysis and sampling over a contour extracted from the 2D projection. If the contour does not pass the segmentability test, we compute a second spectral projection based on a geometric operator $\mathcal{M}$ that emphasizes concavity and repeat the analysis.

Using the ACE method [Kor03], the projections are efficient to compute. The extracted contours not only reveal the global shape characteristics of the given sub-mesh, which improves segmentation quality, they also simplify the shape analysis tasks. Depending on the nature of the segmentation at hand, either the $\mathcal{L}$-embedding or the $\mathcal{M}$-embedding will be applied to compute part segmentability and sample locations, as we describe in Section 3; this will also in turn guide the choice of weights applied to part salience measure.

In Section 4, we propose a method to estimate the segmentability of a contour based on a quantification of convexity. We combine the area [SHB93] and perimeter [ZR04] ratios, which have been proposed as individual convexity measures, to define segmentability, as the two ratios can complement each other. A novel feature of our method is that it can neutralize the effect of bending via multidimensional scaling (MDS) derived from *inner distance* [LJ05], which, for a given pair of contour vertices, is defined as the length of the shortest path, inside the contour, between the two vertices.

Finally, we compute two samples along the extracted contour, based on inner distance again. We show that this novel sampling strategy can robustly detect two points residing on different parts of the shape captured by the contour and in turn, when mapped back onto the sub-mesh, the two samples also reside on different parts. This would improve the quality of our spectral bisection, where the two samples are used by Nyström approximation to derive the 1D spectral embedding for our salience-guided linear search. A graphical illustration of the flow of our algorithm is given in Figure 1.

Our mesh segmentation algorithm requires no user intervention and a few tunable parameters can be fixed throughout. The main contributions we make are listed below.

• **Use of spectral embeddings for shape analysis:** A common feature of the low-dimensional embeddings used in

our work is their ability to reveal shape semantics. While MDS based on inner distances more faithfully reflects the concavity of a shape, the other two spectral embeddings emphasize global and low-frequency shape information; they also simplify the problem by reducing 3D data processing to contour (2D) and sequence (1D) analyses.

• **A novel shape segmentability measure:** Adopted for automatic stopping in our segmentation algorithm, the measure is insensitive to bending and takes advantage of complementary properties of existing convexity measures.

• **A robust sampling scheme based on inner distance:** The ability of inner distances to capture part structures [LJ05] is confirmed in our solution to the sampling problem. Our work highlights the usefulness of inner distance, as an alternative to surface-based distance measures (e.g., geodesic distance), for shape analysis.

## 2. Related works

One of the earliest mesh segmentation algorithms, based on watershed, is due to Mangan and Whitaker [MW99]. Since then several different approaches have been developed, as thoroughly covered in the recent survey by Shamir [Sha06]. In addition, a comparative study of several representative techniques is given by Attene et al. [AFS06]. Thus we shall not attempt to be exhaustive in discussing related works in this section. For example, we will only focus on automatic algorithms, although semi-automatic approaches, e.g., [JLCW06], have also been gaining popularity.

We refer to the mesh segmentation problem we wish to solve as *part-based* [Sha06], where the aim is to appeal to human perception and segment a given shape into its semantic parts. In contrast, surface-type segmentation decomposes a mesh into patches that fulfill certain application-dependent criteria, e.g., planarity [Sha06], constant curvature [MW99], or convexity [KS06, LA06]. Results based on convexity do tend to agree with those from perceptual approaches due to the connection between convexity and the minima rule, which stipulates that part boundaries should be at negative minima of principal curvatures [HR84].

Although mesh segmentation via explicit boundary extraction has been proposed, e.g., feature grouping [LA06] and mesh scissoring [LLS*05], region-based methods appear to be more common. Roughly speaking, a region-based segmentation algorithm is either bottom-up, typically via region growing, e.g., [MW99, KS06, Pag03, ZTS02], or top-down via recursive clustering, e.g., [KT03, KLT05, STK02, ZL05, LJZ06]. Due to its local nature, region growing is susceptible to over-segmentation. Thus a merging step in post-processing is often necessary; but this may still lead to counterintuitive results. Top-down approaches based on global shape characteristics and optimization can offer an effective alternative, but they are often more expensive to execute.

Regardless of the type of a mesh segmentation algorithm, the questions of whether a part should be further segmented and which cut or obtained part is of higher quality are always inevitable. The notion of part salience [HS97] has been utilized by several algorithms, e.g., to reject counter-intuitive cuts [LLS*05], to find a locally optimal cut [ZL05], or to derive a patch merging heuristic [Pag03]. The main difficulty with applying part salience is that there is no easy way to determine an appropriate weighting between its three contributing factors. This has led to the consideration of convexity as a measure of segmentability and part quality.

The best known convexity measure for a 2D shape $P$ is the ratio between the area (area is replaced by volume for a closed mesh) enclosed by $P$ and the area of the convex hull $CH(P)$ of $P$ [SHB93]. The asymmetry of this measure with respect to particular symmetric protrusion and intrusion pairs has motivated Rosin and Mumford [RM04] to replace the convex hull by a convex polygon $Q$ that best fits the shape $P$, in the sense of maximizing the overlap between $P$ and $Q$. However, the polygon $Q$ is non-trivial to compute.

As boundary defects on a shape which do not impact shape area can compromise the performance of the area ratio as a convexity measure, boundary-oriented approaches have been considered. A natural choice would be to use the ratio between the perimeter of the convex hull of a shape $P$ and the perimeter of $P$ to measure convexity [ZR04]. Boxer [Box93] proposes to use either the maximum or average distance between the boundary of a shape and its convex hull instead. Krayevoy and Sheffer [KS06] extend this idea to measure the convexity of a mesh patch in their decomposition algorithm and achieve automatic stopping by thresholding a measure which combines convexity and compactness. It is worth noting however that convexity and segmentability are related but not quite the same concept. While convexity depends heavily on the degree of concavities along the boundary of a shape, segmentability is more structural and should ideally be *invariant to shape articulation*.

Although most mesh segmentation algorithms operate on the original mesh, a few transform-domain techniques have been proposed. Katz et al. [KLT05] perform MDS based on geodesic distances to help extract points near extremities of articulated shapes. Spectral clustering via *k*-means [LZ04] and recursive spectral bisection by searching through 1D mesh face sequences [ZL05] have also been proposed. The main advantage of using 1D embeddings is that any efficiently computable (but hard to optimize) segmentation quality criteria, e.g., part salience [HS97], can be incorporated into a linear search. A downside however is that the 1D embeddings may not provide sufficiently rich information for shape analysis. Moreover, the naive furthest point sampling [LJZ06] or sampling schemes based on distances to a small set of uniformly distributed mesh faces [ZL05] are too ad-hoc and can produce non-robust results. These issues are both addressed by our new approach.

## 3. 2D projections based on mesh Laplacians

A key observation we make is that there are two primary factors which can render a shape segmentable. The first is the *branching* factor. That is, if a shape has branches, then it is segmentable. We refer to segmentability caused by branching as *structural segmentability*. The second factor is due to concavity over the boundary of a shape in the absence of branching, where the induced segmentability is referred to as *geometrical segmentability*.

To facilitate our segmentability analysis and sampling tasks in terms of robustness and efficiency, we first project a sub-mesh embedded in 3D into a plane and then extract a 2D contour from the projection. It is no surprise that a significant amount of information may be lost when going from 3D to 2D. However, by carefully choosing the projection operators, we are able to preserve, and even enhance, semantic shape information useful for subsequent shape analysis. In our approach, the structural and geometrical segmentability information can be extracted from the input 3D mesh, separately, by the two operators we have chosen.

### 3.1. Spectral projection via graph Laplacian $\mathcal{L}$

Given a mesh with $n$ vertices, we define its graph Laplacian $\mathcal{L} \in \mathbb{R}^{n \times n}$ as $\mathcal{L} = \hat{D} - \hat{W}$, where $\hat{W}_{ij} = 1$ whenever $(i, j)$ is an edge, otherwise $\hat{W}_{ij} = 0$; $\hat{D}$ is a diagonal matrix of the row sums of $\hat{W}$. Let $\mathcal{L} = \hat{\Xi}\hat{\Lambda}\hat{\Xi}^T$ be the eigenvalue decomposition of $\mathcal{L}$, where the eigenvalues in $\hat{\Lambda}$ are in ascending order. It is well known that $\mathcal{L}$ is positive semi-definite, its smallest eigenvalue $\hat{\lambda}_1 = 0$, and its corresponding eigenvector $\hat{\Xi}_1$ is a constant vector. Since $\mathcal{L}$ is symmetric, its eigenvectors are orthogonal and span the whole space $\mathbb{R}^{n \times n}$.

Let us represent the coordinates of the mesh vertices by $X \in \mathbb{R}^{n \times 3}$. Then $X$ can be written as a linear combination of the eigenvectors of $\mathcal{L}$. Projecting $X$ into the subspace spanned by the $k$ leading eigenvectors of $\mathcal{L}$, we get

$$\tilde{X} = \hat{\Xi}_{(1 \dots k)} \hat{\Xi}_{(1 \dots k)}^T X,$$

where $\hat{\Xi}_{(1 \dots k)}$ contains the first $k$ columns of $\hat{\Xi}$. This projection process can also be seen as applying an ideal low-

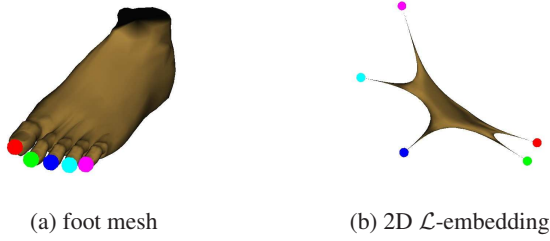(a) foot mesh          (b) 2D $\mathcal{L}$-embedding

**Figure 2:** *2D spectral embedding of a foot model with branching. Coloring of the dots implies correspondence.*



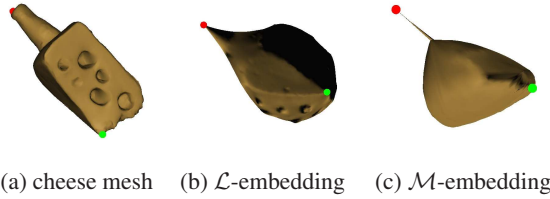(a) cheese mesh    (b) $\mathcal{L}$-embedding    (c) $\mathcal{M}$-embedding

**Figure 3:** *Segmentability of the cheese mesh is more strongly revealed in the embedding via the geometric operator $\mathcal{M}$.*

pass filter (only allowing the leading $k$ frequencies to "pass") to the mesh coordinate signal, akin to Laplacian smoothing [Tau95]. Choosing $k = 3$, we obtain a planar shape, since the first eigenvector is constant.

As a result of Laplacian smoothing, high-frequency information, e.g., surface fluctuations, in the original shape is removed while low-frequency (high-level) shape characteristics, e.g., branching, is retained and even enhanced by the 2D embedding; the latter is able to capture structural segmentability. Figure 2 illustrates this phenomenon. The five toes of the foot mesh, shown in (a), form branches. Plot (b) shows its spectral 2D embedding via $\mathcal{L}$. As we can observe, the "toe branches", as low-frequency structures, are extracted and enhanced in the embedding space.

### 3.2. The geometric operator $\mathcal{M}$

Although smoothing using $\mathcal{L}$ is able to reveal the structural segmentability of a shape, it may fail to capture geometrical segmentability information since $\mathcal{L}$ is defined by mesh connectivity only. Figure 3 depicts such a situation. We see that although the original mesh (a) is clearly segmentable, its 2D $\mathcal{L}$-embedding is not and this could introduce difficulty to our contour analysis. To resolve this issue, we design a new Laplacian operator, $\mathcal{M} = D - W$, which is geometric. To capture local surface geometry, we define for edge $e = (i, j)$

$$W_{ij} = \begin{cases} (|\kappa_i| + |\kappa_j|) \cdot |\langle \vec{e}, \vec{z} \rangle| \cdot l & \text{if } \kappa_i < 0 \text{ or } \kappa_j < 0 \\ \varepsilon & \text{otherwise,} \end{cases}$$

where $\kappa_i$ and $\kappa_j$, computed as in [CSM03], are the minimal principal curvatures at vertices $i$ and $j$, $\vec{z}$ is the normalized

average of the principal curvature directions for $\kappa_i$ and $\kappa_j$, $\vec{e}$ is the (normalized) direction of $e$, and $l$ is the length of $e$ normalized by the average length of all edges in the sub-mesh. As before, $D$ is a diagonal matrix of $W$'s row sums. When $\kappa_i \geq 0$ and $\kappa_j \geq 0$, implying local convexity at $e$, we set $W_{ij} = \varepsilon = 0.1$ to maintain vertex connectivity. If $(i, j)$ is not an edge, $W_{ij} = 0$. We see that if an edge has vertices with negative minimal curvature and it roughly aligns with the local minimal curvature direction, it carries a larger weight. We now examine the effect of this scheme on the embedding.

Consider any matrix $Y \in \mathbb{R}^{n \times k}$, it can be shown that

$$\frac{1}{2} \sum_{ij} W_{ij} ||y_i - y_j||^2 = \text{tr}[Y^T(D - W)Y] = \text{tr}(Y^T \mathcal{M} Y), \quad (1)$$

where $Y = [y_1^T | y_2^T | \dots | y_n^T]^T$ and each $y_i$ is seen as a $k$-dimensional point. Let $\mathcal{M} = \Xi \Lambda \Xi^T$ be an eigenvalue decomposition. The first $k$ eigenvectors of $\Xi$ can be denoted, in column or row vector format, by $\Xi_{(1 \dots k)} = [\xi_1 | \xi_2 | \dots | \xi_k] = [u_1^T | u_2^T | \dots | u_n^T]^T$. It is known that $Y = \Xi_{(1 \dots k)}$, i.e., $y_i = u_i, i = 1 \dots k$, minimizes (1), under the requirement that the columns of $Y$ are of unit length. Thus given a large weight $W_{ij}$, $||u_i - u_j||^2$ tends to be relatively small.

As in Section 3.1, via Laplacian smoothing, we obtain the embedding $\tilde{X} = \Xi_{(1 \dots k)} \Xi_{(1 \dots k)}^T X$ of the mesh given by coordinates $X$. It follows that the squared distance between two embedded vertices, $\tilde{x}_i$ and $\tilde{x}_j$ (two rows of $\tilde{X}$), is

$$||\tilde{x}_i - \tilde{x}_j||^2 = ||(u_i - u_j)\Xi_{(1 \dots k)}^T X||^2 \leq ||u_i - u_j||^2 ||\Xi_{(1 \dots k)}^T X||^2.$$

As $||u_i - u_j||^2$ is relatively small for large $W_{ij}$, $||\tilde{x}_i - \tilde{x}_j||^2$ should also be relatively small, since $||\Xi_{(1 \dots k)}^T X||^2$ is a constant. Thus for each pair of mesh vertices $i$ and $j$, a larger $W_{ij}$ suggests that their embeddings will be closer to each other. With the way the weights in $W$ are defined, mesh vertices from a continuous concave region will be pulled close to each other in the embedding space, forming a constriction. This is illustrated in Figure 3(c), where we see that the segmentability of the cheese model is greatly enhanced in the $\mathcal{M}$-embedding, facilitating subsequent contour analysis.

For efficiency, we apply ACE [Kor03], a multi-level method, to compute the second and third smallest eigenvectors of $\mathcal{L}$ and $\mathcal{M}$, which are necessary for the embedding.

## 4. Contour analysis

After a sub-mesh is embedded in 2D, we extract its outline by rendering the mesh faces in black against a white background and tracing the boundary of the resulting binary image. As the mesh is connected, so is the rendered region. Ignoring interior holes, we obtain a single outer contour of the region. Since the traced contour is jaggy, we smooth it using truncated Fourier descriptors (FDs) [ZR72]. In our implementation, low-frequency FDs are added one by one until the normalized reconstruction error is below a threshold, which is set to be 0.5% of the contour scale in all cases.

The next preprocessing step is to simplify the smoothed contour, thus reducing the cost of computing the inner distances (defined in Section 4.2). Many contour simplification algorithms are available. We adopt the greedy vertex removal scheme of [LL99] based on a curvature cost and stop when the contour has 50 vertices. In subsequent discussions, we denote the smoothed and simplified contour by $\zeta$.

## 4.1. Convexity and segmentability

The segmentability of a 2D shape is clearly related to its convexity/concavity: the more concave a shape is, the more segmentable it tends to be. Denote by $A(\zeta)$ the area enclosed by a contour $\zeta$; the perimeter and the convex hull are denoted by $P(\zeta)$ and $CH(\zeta)$, respectively. The following convexity measures are well-known and simple to compute:

$$C_1(\zeta) = \frac{A(\zeta)}{A(CH(\zeta))}, \qquad C_2(\zeta) = \frac{P(CH(\zeta))}{P(\zeta)}.$$

It is not hard to see that $C_1, C_2 \in [0,1]$ and they are equal to 1 if and only if $\zeta$ is convex. Past research has shown that neither measure provides a completely satisfactory measure of convexity: while $C_1$ struggles with boundary defects that do not impact on shape area, $C_2$ is sensitive to noise. However, they appear to *complement* each other [ZR04]. Motivated by this, we propose to use the following simple extension,

$$C(\zeta) = \max\{C_1(\zeta), C_2(\zeta)^\alpha\},$$

where the exponent $\alpha \geq 1$ not only ensures that $C \in (0,1]$ and $C = 1$ if and only if $\zeta$ is convex, but also addresses possible differences in scale between $C_1$ and $C_2$. Throughout our experiments, we have set $\alpha = 4$.

For a telling example which illustrates the advantage of using the new measure, consider rough outlines of the letters **I** and **R**, shown in Figure 4. By $C_1$, **I** will be deemed to be more concave and thus more segmentable than **R**; this is counterintuitive. With the new measure $C$, $C_2^\alpha$ would prevail over $C_1$ for **I** and return a more reasonable result. The new measure can also properly rank the convexity between the outline of the letters **I** and **C**. The potentially large discrepancies between $C_1$ and $C_2$, e.g., for **I**, explains our preference of using the max operator over convex combination. The inability of either $C_1$ or $C_2$ to properly capture the convexity of a shape should not diminish the utility of the other.
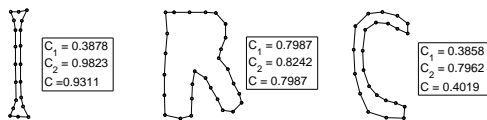


**Figure 4:** *Three letter shapes and their respective area- and perimeter-based convexity measures.*

Despite a correlation between convexity and segmentability, they are not exactly the same concept. Naturally, the re-
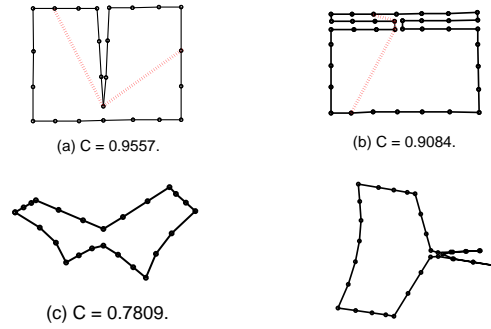


(a) C = 0.9557.  (b) C = 0.9084.

(c) C = 0.7809.

**Figure 5:** *The shapes in (a) and (b) can compromise the performance of C for measuring segmentability. However, they can be "stretched out" via MDS using inner distances, as shown in (c) and (d), to lead to more intuitive results.*

sult of part-based or semantic segmentation should be invariant to shape articulation and this should also hold for a segmentability measure. However, the same cannot be said about convexity/concavity. The effect of bending can compromise the performance of $C$ as a measure of segmentability, e.g., for shapes shown in Figure 5(a) and (b). It is worth noting that the distance-based convexity measure [Box93] would also give unreasonable result for the shape in (b).

Our proposed solution to the above problem involves a normalization with respect to shape articulation. Specifically, we perform MDS based on a distance measure between contour vertices that is insensitive to bending. Unlike the case for manifold meshes [LZ04], geodesic distances, though bending-invariant, are no longer applicable for our purpose in the contour setting. We thus resort to the use of inner distances [LJ05], described in the next section. Since the computation of inner distances and MDS involve extra work, we will only go through these steps when the original contour $\zeta$ is deemed not to be segmentable using $C$.

## 4.2. Inner distance and MDS embedding

The inner distance between two vertices $u$ and $v$ on the contour $\zeta$ is the length of the shortest path between $u$ and $v$ which lies interior to $\zeta$. As shown in Figure 5 (a) and (b), the red dotted lines depict the shortest paths between the corresponding contour vertices inside the shape. Ling and Jacob [LJ05] first propose the use of inner distances for retrieval of articulated shapes and they have shown that inner distances are insensitive to shape articulation.

A natural way to compute pairwise inner distances between $n$ contour vertices is to first construct a visibility graph $G = (V(G), E(G))$, where $(u, v) \in E(G)$ if and only if $u$ is visible to $v$ inside the contour, and then run a shortest path algorithm; this procedure has $O(n^3)$ complexity. There is an optimal $O(n)$ scheme for finding inner distances from a

point, which would lead to a quadratic-time procedure to compute all pairwise inner distances, but this scheme relies on the linear-time polygon triangulation algorithm of Chazelle [Cha91]. We have opted to implement the simplistic $O(n^3)$ approach since each contour $\zeta$ we need to analyze has only 50 vertices. Note that the concept of inner distances can be easily extended to closed polygonal meshes, but unfortunately computing inner distances within a 3D polyhedron becomes an NP-hard problem [dBvKOS97].

After constructing an $n \times n$ matrix of pairwise inner distances, we perform classical MDS [CC01] and embed the contour into the plane. The inner distances are approximated by Euclidean distances via MDS and as such, the contours are normalized against bending. Two results of MDS using inner distances are shown in Figure 5(c) and (d). Our convexity measure $C$ can then be applied to the new embedding.

### 4.3. Segmentability analysis procedure

To take into consideration of both the structural and geometrical nature of segmentation, we conduct contour analysis on embeddings derived from the graph and geometric Laplacian in order. A pseudo code for our analysis procedure is given below, where $\zeta_{\mathcal{L}}$ and $\zeta_{\mathcal{M}}$ are the contours obtained from the $\mathcal{L}$- and $\mathcal{M}$-embeddings, respectively, $S(\zeta) = 1 - C(\zeta)$ is defined to be the *segmentability score* of a contour $\zeta$, and $\eta$ is a user-selected segmentability threshold value.

1. If $S(\zeta_{\mathcal{L}}) > \eta$, then perform sampling on $\zeta_{\mathcal{L}}$ and return.
2. Compute embedding $\zeta_{\mathcal{L}}^*$ of $\zeta_{\mathcal{L}}$ via MDS based on inner distances. If $S(\zeta_{\mathcal{L}}^*) > \eta$, then sample $\zeta_{\mathcal{L}}$ and return.
3. If $S(\zeta_{\mathcal{M}}) > \eta$, then sample $\zeta_{\mathcal{M}}$ and return.
4. Compute embedding $\zeta_{\mathcal{M}}^*$ of $\zeta_{\mathcal{M}}$ via MDS based on inner distances. If $S(\zeta_{\mathcal{M}}^*) > \eta$, then sample $\zeta_{\mathcal{M}}$ and return.

If all the thresholding tests above fail, then the sub-mesh corresponding to the contours is deemed not to be segmentable and our algorithm stops for that sub-mesh. Generally speaking, finding a robust stopping criterion for recursive mesh segmentation is not easy. Previous attempts, e.g., [KT03,ZL05], do not distinguish between structural and geometrical segmentability. Although the latter accounts for part salience, no weighting scheme for the three salience factors was given. An additional advantage of our approach is efficiency, as the 3D problem is reduced to contour analysis. Next, we describe our sampling procedure.

### 4.4. Sampling based on inner distance

Given a segmentable contour $\zeta$, we wish to obtain two sample points from different parts of $\zeta$. Before describing our scheme, we first introduce the concept of *integrated bending score* or *IBS* at a point along a contour with respect to the whole contour. Given a point $f$ on $\zeta$, its IBS is defined as

$$\mathcal{R}_f = \int_{t \in \zeta_f} \frac{I_{ft}}{E_{ft}} dt,$$

where $\zeta_f = \zeta - \{f\}$ is formed by the union of the line segments of $\zeta$ with the absence of point $f$, and $E_{ft}$ and $I_{ft}$ denote the Euclidean and inner distances between two points $f$ and $t$, respectively. We compute the above integral per line segment $\overline{ij}$ in $\zeta$, namely, $\mathcal{R}_f = \sum_{\overline{ij} \in \zeta} \mathcal{R}_f(\overline{ij})$.

To compute $\mathcal{R}_f(\overline{ij})$, we first calculate the inner and Euclidean distances from $f$ to $i$ and $j$; these are $I_{fi}, I_{fj}, E_{fi}$, and $E_{fj}$. Note that we are only interested in IBS's at contour vertices, thus $f$, $i$, and $j$ are all contour vertices. Due to the complexity of inner distance computations, we have chosen not to over-sample the line segment $\overline{ij}$ to compute $\mathcal{R}_f(\overline{ij})$. Instead, we rely on linear interpolation to estimate $I_{ft}$ and $E_{ft}$. Let $l$ be the length of $\overline{ij}$. We estimate $\mathcal{R}_f(\overline{ij})$ as

$$\mathcal{R}_f(\overline{ij}) = \int_0^l \frac{I_{fi} + \frac{t}{l}(I_{fj} - I_{fi})}{E_{fi} + \frac{t}{l}(E_{fj} - E_{fi})} dt.$$

After algebraic derivation, we arrive at

$$\mathcal{R}_f(\overline{ij}) = \frac{dbl + (ad - bc)[ln(c + dl) - ln(c)]}{d^2},$$

where $a = I_{fi}l$, $b = I_{fj} - I_{fi}$, $c = E_{fi}l$, and $d = E_{fj} - E_{fi}$.

Note that ideally, we would want to compute the IBS at a point with respect to the original contour. Our approach is aimed at achieving efficiency and it can introduce approximation errors from both contour simplification and linear interpolation. However, our experiments verify that these errors are negligible for the purpose of sampling.

Intuitively, the IBS $\mathcal{R}_f$ measures the extent a point $f$ is isolated inside the region enclosed by the contour $\zeta$. Thus our first sample $s_1$ is simply selected as the contour vertex which maximizes $\mathcal{R}_f$. This way $s_1$ tends to reside on a peripheral part that is most isolated from the core of the shape enclosed by $\zeta$. The second sample $s_2$ is then chosen as the contour vertex with the largest inner distance from $s_1$.

Past sampling schemes aimed at finding points from different parts of a shape rely on measures which combine geodesic and curvature-related distances, e.g., [KT03, ZL05]. Using ratios as in our approach, we avoid the introduction of a weight parameter. Quality-wise, we compare our scheme with those based on farthest distances in Figure 6 using three increasingly harder test cases, to show the advantage of our approach. Similar to the centricity measure of Lee et al. [LLS*05], the IBS utilizes more global information, which we expect to produce more robust results.

## 5. Salience-guided spectral cut

Recall that given a sub-mesh, we first spectrally embed it into the plane and perform contour analysis. If the sub-mesh passes the segmentability test, two sample faces on the sub-mesh, corresponding to the sample points computed along a planar contour, are returned. Now the two sample faces are used in Nyström approximation [FBCM04, ZL05] to derive a linear sequence for the faces of the sub-mesh.
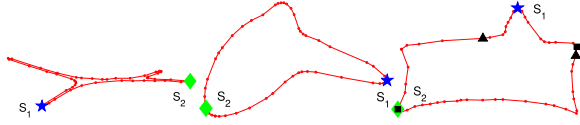
**Figure 6:** *Results of our sampling scheme: samples marked by $s_1$ and $s_2$. In the right-most figure, we compare against two variations of farthest-point sampling. The first finds two points, marked by squares, that are mutually farthest away in inner distance. The second scheme picks two points, shown as triangles, that have the largest distance, computed as the sum of magnitude of negative curvatures, along the contour. Note that a point can traverse along a contour in two directions; the one giving the smaller distance is considered.*

### 5.1. Spectral embedding using Nyström method

We follow the approach from [ZL05], which is inspired by the normalized cut idea for image segmentation [SM97]. Instead of using the Fiedler vector of the normalized graph Laplacian to compute a 1D embedding, as in [SM97], a normalized affinity matrix $\mathcal{N} = Q^{-1/2} A Q^{-1/2}$ is constructed and the 1D embedding is given by a *component-wise ratio* between the second and first largest eigenvectors of $\mathcal{N}$, where we order the eigenvectors by the magnitude of the corresponding eigenvalues. The matrix $A$ is called the affinity matrix where $A_{ij}$ encodes a relation between faces $i$ and $j$ of the sub-mesh; the matrix $Q$ is a diagonal matrix of $A$'s row sums. As in [ZL05], we model affinity $A_{ij}$ by an exponential kernel applied to a distance measure [KT03] which combines geodesic and angle distances between the two faces over the mesh surface. For a large sub-mesh, computing $A$ and its eigenvectors is too time consuming. Therefore, we resort to subsampling techniques using the Nyström method.

Using Nyström approximation, it is sufficient to utilize two carefully chosen samples and compute only affinities from the two samples. The eigenvectors will be computed approximately via extrapolation. The computational complexity is reduced from $O(m^2 \log m)$ to $O(m \log m)$, where $m$ is the size of the sub-mesh. Details on a derivation and use of the Nyström method can be found in [FBCM04, ZL05].

### 5.2. Finding a salient cut

Having obtained a 1D embedding of the mesh faces, we perform a linear search over the face sequence. Each bisection of the sequence corresponds to a cut in the sub-mesh, resulting in two parts. We compute a salience measure for the part having a smaller surface area. The salience score for a part is given as a weighted sum of appropriate measures for part size, cut strength, and part protrusiveness, as given in [ZL05]. These measures can all be updated in constant time during the linear search. The cut which achieves the highest salience score is chosen as the segmentation of the sub-mesh and the resulting parts may be further processed.

To obtain high-quality segmentation, it is essential that the two sample faces reside on different parts of the shape. As we cannot define what a part is, we would not be able to argue for the correctness of our sampling scheme. However, experimentally, we have found our sampling scheme to work robustly for a variety of shapes. The other key ingredient of our scheme is the part salience score, which requires an appropriate weighting between the three factors. Denote the weights applied to part size, cut strength, and part protrusiveness by $\omega_s$, $\omega_c$, and $\omega_p$, respectively. Through experimental study, we observe that part size should take the least weight. Depending on which operator, $\mathcal{L}$ or $\mathcal{M}$, has passed the segmentability test, either $\omega_p$ or $\omega_c$ should be emphasized more. Specifically, if $\mathcal{M}$ ($\mathcal{L}$) is used, it implies that the current segmentability, if any, is caused by concavity (protrusion), therefore $\omega_c$ ($\omega_p$) should take more weight.

## 6. Experimental results

Except for the case shown in Figure 8, all the free parameters of our algorithm are fixed: We set $\eta = 0.1$ as our threshold value for segmentability tests and $\alpha = 4$ for the convexity measure $C$. The weights for part salience are: $\omega_s = 0.1$, $\omega_c = 0.3$, $\omega_p = 0.6$, when $\mathcal{L}$ is applied, and $\omega_s = 0.1$, $\omega_c = 0.6$, $\omega_p = 0.3$ when $\mathcal{M}$ is applied.

We first show our algorithm at work in steps for the bunny model. In the first five columns of Figure 7, the top row shows the obtained segments in color. A brown part is the current sub-mesh to segment, and its embedding, either via $\mathcal{L}$ or $\mathcal{M}$, is shown in the second row. Plots (k) and (l) are the $\mathcal{L}$- and $\mathcal{M}$-embeddings of the bunny's left ear. As neither of them passes the segmentability check, the ear is not segmented. The final result is shown in Figure 11(g).
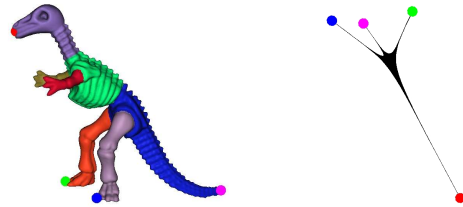


**Figure 8:** *Low-frequency or structural segmentation of the dinosaur with 112,384 faces, using $\mathcal{L}$ only. Shown on the right is the first spectral embedding, where coloring of the dots imply correspondence between 2D and 3D. Note that two hands of the model are "absorbed" by the body in the first embedding, but they will stand out in later iterations.*

**Properties of $\mathcal{L}$ and $\mathcal{M}$ :** The ability of the operator $\mathcal{L}$ to extract "low-frequency" structural information is desirable in situations such as in Figure 8. It is counterintuitive to segment the dinosaur model along all the concavities over its surface. If we only apply $\mathcal{L}$ for structural segmentability detection, the segmentation stops as shown in the figure. Note
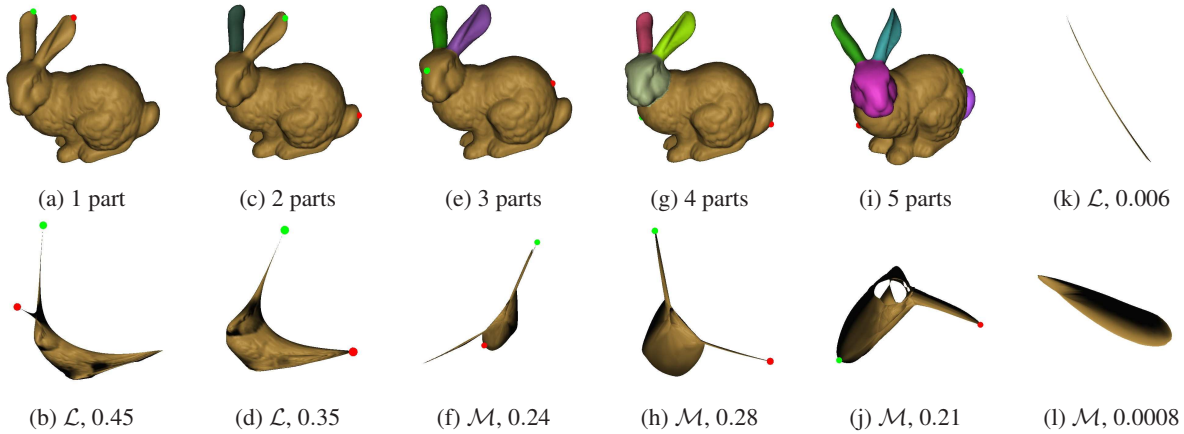
| (a) 1 part | (c) 2 parts | (e) 3 parts | (g) 4 parts | (i) 5 parts | (k) $\mathcal{L}$, 0.006 |

| (b) $\mathcal{L}$, 0.45 | (d) $\mathcal{L}$, 0.35 | (f) $\mathcal{M}$, 0.24 | (h) $\mathcal{M}$, 0.28 | (j) $\mathcal{M}$, 0.21 | (l) $\mathcal{M}$, 0.0008 |

**Figure 7:** *(a-j): four iterations of our algorithm on the bunny. The colored dots represent the obtained samples, over the part to segment. Segmentation results are given in top row and the second row shows the embeddings, via $\mathcal{L}$ or $\mathcal{M}$, as indicated, along with the final segmentability score. (k-l): embeddings of the left ear using $\mathcal{L}$ and $\mathcal{M}$, both failing the segmentability test.*
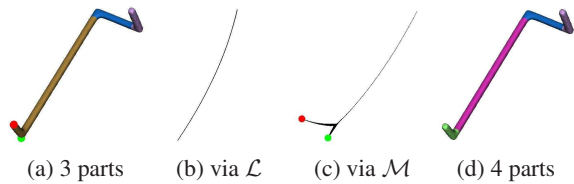


| (a) 3 parts | (b) via $\mathcal{L}$ | (c) via $\mathcal{M}$ | (d) 4 parts |

**Figure 9:** *The brown part in (a) is not segmentable via the $\mathcal{L}$-embedding (b). When the $\mathcal{M}$-embedding (c) is considered, the part is segmented, leading to the final result in (d).*

that the protrusion and cut strength salience factors, $\omega_p$ and $\omega_c$, need to be tuned to obtain such a result. This is the only case where the fixed parameters are modified. On the other hand, $\mathcal{M}$ has its own importance too. Figure 9(a) shows the segmentation result using $\mathcal{L}$. The brown part fails the segmentability test since its embedding, shown in (b), is quite convex. However, the embedding produced by $\mathcal{M}$ enhances bending and manifests itself with strong segmentability, resulting in a better segmentation in (d).

More segmentation results are available in Figure 11, where only the bunny, hand and claw models have boundaries. Models in the first row are only segmented using $\mathcal{L}$, where the goal is to show structural segmentability at work. Note that for the children model, the skirt in green would have been segmented by any scheme relying on concavity, e.g., when geometrical segmentability using $\mathcal{M}$ is considered; this is not wrong from a geometric point of view but it does not appeal to our intuition. With our algorithm, there is the option of focusing on structural segmentability only which may lead to a stopping configuration that better reflects the shape semantics. Note also that the trident tines in the Neptune model are not considered segmentable as they are relatively short compared with the whole trident, thus not

detected as a significant "low-frequency" information. For the man model however, the toes and fingers are all detected properly, as they are relatively large in their own subparts.

Models in the second row are segmented by combining $\mathcal{L}$ and $\mathcal{M}$. As our algorithm is top-down, it is less likely to produce over-segmentation, which is typically present for schemes based on local optimization, e.g., region growing. This can be understood from the torso of the bunny and the cheese in the mouse model, where many local concavities exist. In practice, we have not observed that a high genus would pose a problem for our algorithm. In Figure 11, the children, oil pump and Neptune models all have genus greater than 0 and our algorithm works robustly on them.

**Timing:** Our algorithm is quite efficient. Taking away the computation of the eigenvectors of $\mathcal{L}$ and $\mathcal{M}$, the asymptotic complexity of our algorithm in each iteration is $O(m \log m)$ for a sub-mesh of size $m$; this is necessary for computing distances from the sample faces to the remaining faces. The ACE method [Kor03] is designed to efficiently compute a few eigenvectors of the graph Laplacian. In our experiments, we find it to scale well for both $\mathcal{L}$ and $\mathcal{M}$. Overall, on an AMD Opteron 2.4GHz CPU machine with 8GB RAM, segmenting the frog model with 100K faces into 14 parts takes a total of 86 seconds, where 47 seconds are spent computing eigenvectors for spectral embedding. The hand model with 273K faces, the largest in our experiments, takes about 52 seconds to segment into 6 parts. Since only $\mathcal{L}$ is used, which consists of integers only, ACE operates more quickly.

**Current limitations:** Unsatisfactory results may still be generated by our algorithm. For example, in the oil pump model in Figure 11(f), the big brown part is not adequately segmented. Similarly, the eyes of the frog, in Figure 11(h), do not pass the segmentability test either. However, by re-
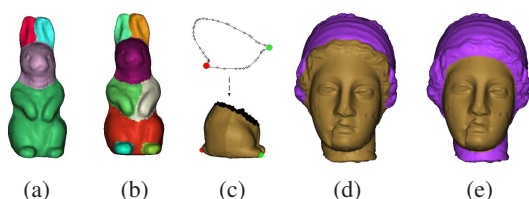
**Figure 10:** *With the default segmentability threshold* $\eta = 0.1$*, we obtain the segmented rabbit in (a). With proper sampling (c) and setting* $\eta = 0.03$*, we obtain a refined segmentation (b). Linear search over the Igea (268,686 faces) results in a suboptimal cut (d) with our current salience measure, while an optimal cut (e) is present along the sequence.*

laxing the segmentability threshold $\eta$, our algorithm will be able to produce the desirable segmentation. One such example is given in Figure 10. In (a), the default parameters are applied; while (b) is obtained by relaxing $\eta$ from 0.1 to 0.03. We believe the use of segmentability score, the abstraction of segmentation strength, provides a better control than simply specifying a desirable number of parts, which is model dependent. But the segmentability measure still requires further study as we aim for total autonomy.

The current salience measure also leaves room for improvement. An example is the Igea model, which is hard to segment due to the bumpy hair region. With the fixed weighting scheme for part salience, the algorithm obtains the result shown in Figure 10(d), while the optimal one, manually searched along the sequence, is shown in (e).

**Robustness of sampling and 1D mesh embedding:** Despite the limitations, we believe our sampling scheme and the resulting 1D mesh embeddings used for linear search are both robust. The successful segmentation of the rabbit in Figure 10(b) with a relaxed $\eta$ relies on proper sampling. As shown in (c), even though the foot and tail are relatively small features, our algorithm can correctly extract two samples from them. On the contrary, our experiment shows that farthest point sampling based on combining geodesic and angle distances would fail in this particular case. Our algorithm is also able to segment out the eyes of the frog model in Figure 11(h) nicely as two samples on the eyes can be identified. Finally, refer to the Igea models in Figure 10. Although the current result, as shown in (d), is not satisfactory, the 1D spectral embedding obtained does possess the optimal cut as shown in (e); this is found via manual traversal.

## 7. Conclusion and future work

We present an automatic mesh segmentation algorithm which iteratively bisects a sub-mesh. Although there are a few free parameters to set, a fixed set of values have enabled us to produce stable, meaningful segmentation results for a variety of mesh models. Our algorithm is based on spectral embeddings and 2D contour analysis. The main idea lies in utilizing appropriately defined projection operators to retain and even enhance structural and geometric information in a low-dimensional space, 1D or 2D in our case, where our segmentability analysis, sampling, and cut extraction problems can be more efficiently and more effectively solved.

From results shown in Section 6, we see that our algorithm excels at revealing structural composition of shapes and identifying more prominent parts, and does so in a stable manner. Other segmentation paradigms, e.g., approximate convex decomposition [KS06, LA06], would produce results of a different nature. Relaxing our thresholds for the segmentability tests will allow for more refined shape partitioning, but at the expense of introducing less meaningful parts. This motivates further study on segmentability, perhaps with the incorporation of training and prior knowledge, which should also be useful in selecting the weights for evaluating part salience. Our sampling scheme based on inner distance is not only robust but also involves no free parameters, we would like to extend it to 3D where an efficient approximation of inner distances would be essential.

Finally, more objective, or at least more systematic, ways to evaluate mesh segmentation results are still lacking. In our work, we have chosen test models that are well known to facilitate comparison of our results to those obtained by others. However, it is still difficult to quantify or rank the quality of the different approaches at this point. To address this issue, we plan to develop a mesh segmentation benchmark.

## References

[AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical segmentation based on fitting primitives. *The Visual Computer 22* (2006), 181–193.

[Box93] BOXER L.: Computing deviations from convexity in polygons. *Patt. Recog. Let. 14* (1993), 163–167.

[CC01] COX M. F., COX M. A. A.: *Multidimensional Scaling.* Chapman and Hall, 2001.

[Cha91] CHAZELLE B.: Triangulating a simple polygon in linear time. *Discrete Computational Geometry 6* (1991), 485–524.

[CSM03] COHEN-STEINER D., MORVAN J. M.: Restricted delaunay triangulations and normal cycle. In *Proc. 19-th Annual Symposium of Computational Geometry* (2003).

[dBvKOS97] DE BERG M., VAN KREVELD M., OVERMARS M., SCHWARZKOPF O.: *Computational Geometry: Algorithms and Applications.* Springer, 1997.
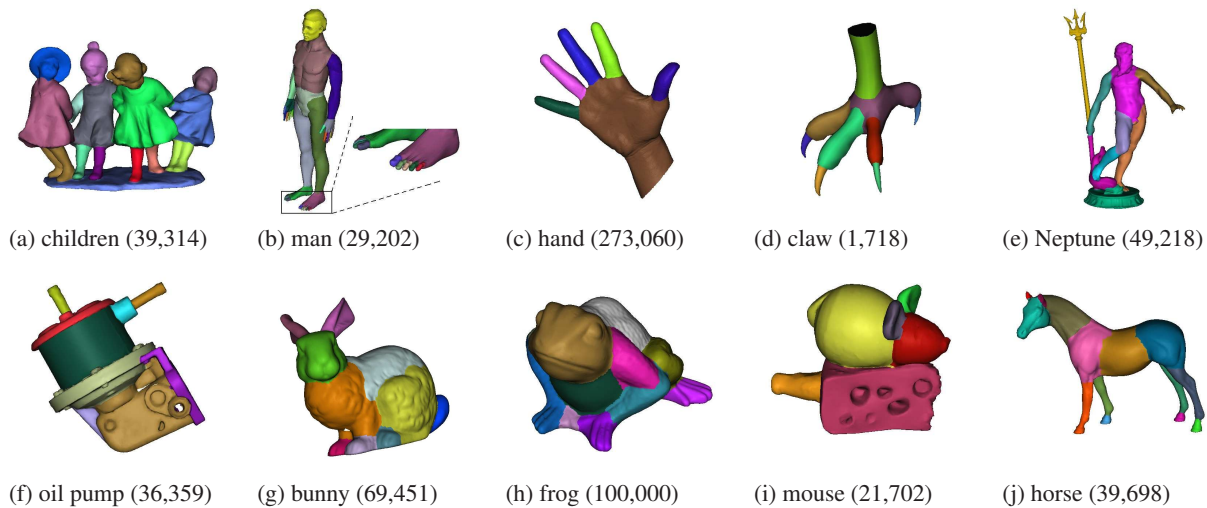
(a) children (39,314)  (b) man (29,202)  (c) hand (273,060)  (d) claw (1,718)  (e) Neptune (49,218)

(f) oil pump (36,359)  (g) bunny (69,451)  (h) frog (100,000)  (i) mouse (21,702)  (j) horse (39,698)

**Figure 11:** *More segmentation results from our algorithm (numbers shown in brackets are the face counts).*

[FBCM04]  FOWLKES C., BELONGIE S., CHUNG F., MALIK J.: Spectral grouping using the Nyström method. *IEEE Trans. Pat. Ana. & Mach. Int. 26* (2004), 214–225.

[HR84]  HOFFMAN D. D., RICHARDS W. A.: Parts of recognition. *Cognition 18* (1984), 65–96.

[HS97]  HOFFMAN D. D., SINGH M.: Salience of visual parts. *Cognition 63* (1997), 29–78.

[JLCW06]  JI Z., LIU L., CHEN Z., WANG G.: Easy mesh cutting. *Computer Graphics Forum 26*, 3 (2006), 283–292.

[KLT05]  KATZ S., LEIFMAN G., TAL A.: Mesh segmentation using feature point and core extraction. *The Visual Computer (Pacific Graphics) 21*, 8-10 (2005), 649–658.

[Kor03]  KOREN Y.: On spectral graph drawing. In *Proc. of Int'l Conf. Compu. and Comb. (COCOON)* (2003), pp. 496–508.

[KS06]  KRAEVOY V., SHEFFER A.: Variational, meaningful shape decomposition. In *Technical Sketch of SIGGRAPH* (2006).

[KT03]  KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. On Graphics 22*, 3 (2003), 954–961.

[LA06]  LIEN J. M., AMATO N.: *Approximate Convex Decomposition of Polyhedra*. Tech. Rep. TR06-002, Texas A&M, 2006.

[LJ05]  LING H., JACOBS D. W.: Using the inner-distance for classification of articulated shapes. In *Proc. of Computer Vision and Pattern Recognition* (2005), vol. 2, pp. 719–726.

[LJZ06]  LIU R., JAIN V., ZHANG H.: Subsampling for efficient spectral mesh processing. In *Proc. of Computer Graphics International* (2006), pp. 172–184.

[LL99]  LATECKI L. J., LAKÄMPER R.: Convexity rule for shape decomposition based on discrete contour evolution. *Comput. Vis. Image Underst. 73*, 3 (1999), 441–454.

[LLS*05]  LEE Y., LEE S., SHAMIR A., COHEN-OR D., SEIDEL H.: Mesh scissoring with minimal rule and part salience. *Computer Aided Geometric Design 22* (2005), 444–465.

[LZ04]  LIU R., ZHANG H.: Segmentation of 3D meshes through spectral clustering. In *Pacific Graphics* (2004), pp. 298–305.

[MW99]  MANGAN A. P., WHITAKER R. T.: Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. on Visualization and Computer Graphics 5*, 4 (1999), 308–321.

[Pag03]  PAGE D. L.: *Part Decomposition of 3D Surfaces*. PhD thesis, University of Tennessee, 2003.

[RM04]  ROSIN P., MUMFORD C. L.: A symmetric convexity measure. In *Proc. of ICPR* (2004), vol. 4, pp. 11–14.

[Sha06]  SHAMIR A.: Segmentation and shape extraction of 3D boundary meshes. In *Eurographics STAR* (2006).

[SHB93]  SONKA M., HLAVAC V., BOYLE R.: *Image Processing, Analysis, and Machine Vision*. Chapman and Hall, 1993.

[SM97]  SHI J., MALIK J.: Normalized cuts and image segmentation. In *Proc. of Computer Vision and Pattern Recognition* (1997), pp. 731–737.

[STK02]  SHLAFMAN S., TAL A., KATZ S.: Metamorphosis of polyhedral surfaces using decomposition. *Computer Graphics Forum 21*, 3 (2002), 219–228.

[Tau95]  TAUBIN G.: A signal processing approach to fair surface design. In *ACM SIGGRAPH* (1995), pp. 351–358.

[ZL05]  ZHANG H., LIU R.: Mesh segmentation via recursive and visually salient spectral cuts. In *Proc. of Vision, Modeling, and Visualization* (2005), pp. 429–436.

[ZR72]  ZAHN C., ROSKIES R.: Fourier descriptors for plane closed curves. *IEEE Trans. Computers 21*, 3 (1972), 269–281.

[ZR04]  ZUNIC J., ROSIN P.: A new convexity measure for polygons. *IEEE Trans. Pat. Ana. & Mach. Int. 26*, 7 (2004), 923–934.

[ZTS02]  ZUCKERBERGER E., TAL A., SHLAFMAN S.: Polyhedral surface decomposition with applications. In *Computer and Graphics* (2002), pp. 733–743.