

# Persistence-based Segmentation of Deformable Shapes

Primož Skraba  
INRIA-Saclay  
Orsay, France  
`primož.skraba@inria.fr`

Maks Ovsjanikov  
Stanford University  
Stanford CA, USA  
`maks@stanford.edu`

Frédéric Chazal  
INRIA-Saclay  
Orsay, France  
`frédéric.chazal@inria.fr`

Leonidas Guibas  
Stanford University  
Stanford CA, USA  
`guibas@stanford.edu`

## Abstract

In this paper, we combine two ideas: persistence-based clustering and the Heat Kernel Signature (HKS) function to obtain a multi-scale isometry invariant mesh segmentation algorithm. The key advantages of this approach is that it is tunable through a few intuitive parameters and is stable under near-isometric deformations. Indeed the method comes with feedback on the stability of the number of segments in the form of a persistence diagram. There are also spatial guarantees on part of the segments. Finally, we present an extension to the method which first detects regions which are inherently unstable and segments them separately. Both approaches are reasonably scalable and come with strong guarantees. We show numerous examples and a comparison with the segmentation benchmark and the curvature function.

## 1. Introduction

Given a shape, we would like to segment it into a small number of meaningful components that can then be analyzed and processed individually. Mesh segmentation has applications in a wide range of fields including reverse engineering, medical imaging (1) as well as shape retrieval (2) and partial matching (3). Although mesh-segmentation is a very active area of research with considerable history (see e.g. (4) for a survey dating from 2006 and (1) for a survey of methods for CAD applications), few methods have been proposed with theoretical guarantees on the quality of the segmentation. The fundamental problem is that the quality of a segmentation is in general ill-defined. The correct number of segments is application dependent and is often given as input by the user. The desired segmentation is then obtained through trial and error by manually inspecting segmentations produced under different parameter choices. Recently, Chen et al. proposed a benchmark for mesh segmentation (5), which is based on human segmentations of a database of ob-

jects. Thus, it reflects the (particular) human beliefs of how the objects need to be segmented. This benchmark partially alleviates the problem of the ground-truth; however, the question of *stability* of mesh segmentation algorithms is still a prominent one. To be practical, an algorithm must produce a relevant segmentation and be stable under different parameter choices.

Persistence-based clustering (PBC) (6) is a method based on the notion that relevant segments correspond to basins of attraction of some function. Hill climbing algorithms are often used to find these basins, but they are generally unstable. Topological persistence (7) computes the prominence of the basin of attraction associated to each extremal point based on a hierarchy. To compute the segmentation, we use this information to merge segments in a theoretically justified way. The topological framework ensures that under small perturbations the resulting segmentation is provably stable.

We incorporate PBC into a framework for isometry-invariant mesh segmentation by combining it with the Heat Kernel Signature (HKS) function (8). It is invariant to isometric deformations of the underlying shape, and is stable under small perturbations of the surface. The combination of the two methods results in a *stable, isometry-invariant* mesh segmentation. Furthermore, the HKS has a time parameter  $t$ , which can be interpreted as an intrinsic notion of scale. Therefore, by choosing various values of  $t$ , we can obtain segmentations of the mesh *at multiple scales*. For a single choice of scale and the associated HKS, PBC first returns the prominence of the segments in the form of an intuitive persistence diagram (PD), which allows the user to choose a merging parameter, resulting in a stable segmentation. If the number of correct segments is known *a priori*, the algorithm selects the appropriate merging parameter, as well as gives feedback on the stability of the resulting segmentation. We also introduce a randomized method for detecting regions on the mesh which are inherently unstable, and which are segmented separately.

## 2. Related Work

Mesh segmentation is a fundamental operation in geometry processing, and has been widely studied over the past several decades. For a review of existing methods we refer the reader to recent surveys on this field (e.g. (4; 1; 9)). According to the ontology presented in (1), mesh segmentation algorithms can be grouped into two general classes: volumetric and surface based. Since our method is purely surface-based, we only consider similar techniques that fall within this category.

The algorithm proposed in this paper is most closely related to watershed methods, and those based on critical points defined on the mesh. Similarly to our technique, these methods define a function on the mesh and segment the surface by associating vertices with extremal points (local maxima and minima) of this function (10; 11; 12). The majority of these methods use various notions of curvature as the watershed function (1). As we demonstrate in this paper, curvature is often not robust enough to allow for meaningful shape segmentation. Instead, we use a recently proposed Heat Kernel Signature (8), which can be interpreted as a multiscale notion of curvature, and allows for more robust segmentation.

Our method is also related to feature-based techniques that first identify feature points on the mesh, and then segment it into regions that respect the feature points, e.g. (13; 14; 15). Similarly to these methods, we use local maxima of the Heat Kernel Signature as feature points on the mesh. However, we also employ tools from persistent homology, and specifically persistence-based clustering (6) to only consider significant or persistent feature points. As we show in this paper, using only prominent features is crucial for a successful segmentation.

Note that although several techniques, e.g. (16), also attempt to group feature points together in order to reduce over-segmentation, the main difference of our work is that we use provably correct techniques to extract persistent features. Indeed, despite the plethora of work in mesh segmentation virtually no practical methods come with guarantees on either the quality of the reconstructed segmentation or the stability of segmentation with respect to the necessarily present parameters. In this paper we propose a practical method for mesh segmentation but also aim to bridge the gap between existing theoretical tools for stability analysis and methods for mesh segmentation.

Isometry-invariant mesh segmentation has recently been addressed in e.g. (17; 13; 18; 19), where the goal is to produce a consistent segmentation of the shape under potential near-isometric deformations such as articulated motion. These methods often use spectral

invariants, such as the eigenfunctions of the Laplace-Beltrami operator (17; 19) to achieve isometry invariance. However, eigenfunctions can be rather unstable especially if the gap between the corresponding eigenvalues is small. Recently, more stable invariants based on the heat diffusion have been introduced in (18; 8). In particular, the Heat Kernel Signature (8) has shown promising results for feature based shape analysis. However, existing methods that employ diffusion invariants are either hierarchical (13) or use heuristics to filter unstable feature points (8). In this paper, we use theoretically justified tools from persistence-based clustering (PBC) (6) to result in a multi-scale, stable, and isometry invariant segmentation method that comes with theoretical guarantees.

## 3. Persistence-based Clustering

In persistence-based clustering (PBC) (6), the goal is to recover the *basins of attraction* of a function  $f$  on a space  $\mathbb{X}$ . PBC falls under the umbrella of topological data analysis (20) and makes heavy use of topological persistence theory (7; 21). The approach can be conceptually divided into two parts: computing a global description of the function called a *persistence diagram* (PD) and using this information to compute the segmentation.

For a function on a space  $(\mathbb{X}, f)$ , we track connected components over different superlevel sets  $\mathbb{X}^a = f^{-1}[a, \infty)$  for  $a \in (-\infty, +\infty)$ . As we sweep  $a$  from  $+\infty$  to  $-\infty$ , new connected components are either born, or previously existing connected components are merged together. Each connected component is associated with a local maximum of  $f$ , when the component is first born. Merging occurs when  $a$  is such that there is a path on  $\mathbb{X}$  between  $x_1$  and  $x_2$ , such that  $f(x) \geq a$  for all  $x$  along that path. Furthermore, persistence theory creates a hierarchy of components: when two components corresponding to local maxima  $x_1$  and  $x_2$ , s.t.  $x_1 < x_2$  are merged, we say that the component corresponding to  $x_1$  *dies*. Equivalently, the component corresponding to the smaller local maximum is always merged into the component corresponding to the larger local maximum. On the other hand a component which corresponds to a local maximum  $x_1$  is *born* at time  $x_1$ . The Persistence Diagram (PD) represents the births and deaths of all the connected components by assigning to each component a point in the extended plane  $\overline{\mathbb{R}}^2$ , where the  $x$ -coordinate represents the birth of the component and the  $y$ -coordinate represents its death.

The value of the PD is that it provides a stable representation of the structure of the function on the space. The persistence of each connected component is simply

the vertical distance of the corresponding point to the diagonal. Since each point represents a local maximum (peak) of the function, the more persistent peaks correspond to the points which are far from the diagonal. Conversely, points close to the diagonal are more likely to correspond to noisy peaks. If the peaks of the function are persistent enough, we can separate them from noisy local maxima by a line parallel to the diagonal. The distance of this line to the diagonal is called the *merging parameter*, denoted by  $\tau$ . By controlling this parameter, the user controls the number of the segments produced by our mesh segmentation algorithm.

Once  $\tau$  is chosen, we again sweep  $a$  from  $+\infty$  to  $-\infty$  and keep track of the connected components. Now, however, when we merge two components, we check to see if both are sufficiently persistent. Assuming the two maxima function values are  $x_1$  and  $x_2$  such that  $x_1 < x_2$ , if  $x_1 - a \leq \tau$ , the merge is performed as above. If, however,  $x_1 - a > \tau$  no merge is done. Once completed, the number of segments corresponds precisely to the number of points above the line in the PD.

One of the practical advantages of Persistence-based Clustering is that before the user is forced to choose the number of segments, during the first stage of the algorithm, the Persistence Diagram provides a compact visual representation of all of the local maxima of the function. By inspecting the PD, the user can not only chose a meaningful merging parameter, but also can get a sense of the stability of the number of segments under different choices of  $\tau$ .

**Implementation** Computing the PD and the segmentation can be done using the same algorithm. As input we take a mesh  $M$ , a function  $f$  defined on its vertices and the merging parameter  $\tau$ . To compute the PD, we set  $\tau = \infty$ . We then process the vertices in decreasing value of  $f$ . For a processed vertex, we maintain the segmentation,  $C()$ , which returns the current maximum over its connected component.

To process a vertex  $x$ , we first determine if it is a local maximum in the mesh by comparing  $f(x)$  with  $f(y)$  for all  $y$  in a one-ring neighborhood of  $x$ . If  $x$  is a local maximum, a new component is born and the vertex is assigned to itself in the segmentation,  $C(x) = x$ . If  $x$  is not a local maximum, we assign it to the neighbor with the highest function value. If the vertex is adjacent to two or more existing components, we check the persistence of the components and merge them only if they are not  $\tau$ -persistent. To merge two segments with maxima  $x_1$  and  $x_2$  such that  $f(x_1) < f(x_2)$ , we set  $C(x_1) = x_2$ .

When all of the vertices are processed, the segment of each vertex can be found by iterating  $C$  until we reach a fixed point (i.e.  $C(x) = x$ ). When computing

the PD, every time we merge two components, we output the pair  $(f(x_1), f(x))$ , where  $x_1$  is the maximum with the smaller value of  $f$  and  $x$  is the point currently being processed. These are precisely the points of the PD. This procedure is equivalent the standard persistence algorithm for 0-dimensional homology (7).

If we use the Union-Find data structure, the algorithm requires linear storage and runs in  $O(n\alpha^{-1}(n))$  time where  $n$  is the number of edges in the mesh, and  $\alpha^{-1}(\cdot)$  is the inverse Ackermann function. For surface meshes, the number of edges is linear in number of vertices, making our algorithm highly scalable.

## 4. Heat Kernel Signature

Note that Persistence-based Clustering, gives the user a choice of function defined on the mesh. As mentioned in the introduction, we use the Heat Kernel Signature (HKS) introduced by Sun et al. (8) to obtain an isometry-invariant multi-scale segmentation.

The HKS can be defined on any Riemannian manifold  $\mathbb{M}$ , via the heat kernel of  $\mathbb{M}$ . The heat kernel  $k_t(x, y) : \mathbb{R}^+ \times \mathbb{M} \times \mathbb{M} \rightarrow \mathbb{R}$ , is the minimal fundamental solution of the heat equation, which, intuitively measures the amount of heat transferred from point  $x \in \mathbb{M}$  to point  $y \in \mathbb{M}$  in time  $t$ , given a point source at  $x$  at time 0 (see (22) for a thorough discussion of the heat kernel and its properties).

Following (8), the HKS of a point  $x$  is defined as  $\text{HKS}(x) : \mathbb{R}^+ \rightarrow \mathbb{R}$ ,  $\text{HKS}(x, t) = k_t(x, x)$ . The relevant properties of the Heat Kernel Signature are (see (8) for the discussion): (i) Invariance under isometric deformations of the shape, (ii) Stability under near-isometric perturbations of the surface, and (iii) Multi-scale:  $k_t(x, x)$  is closely related to Gaussian curvature for small  $t$  and can be interpreted as a multi-scale notion of curvature *at the scale* defined by  $t$ .

Sun et al. (8) used HKS as an isometry-invariant descriptor of points on the shape. Here we extend their construction to segment the shape into stable clusters. In particular, we fix a time  $t$  and consider the function  $f : \mathbb{M} \rightarrow \mathbb{R}$ ,  $f(x) = k_t(x, x)$ . The additional time parameter  $t$  allows us to control *the scale* at which the segmentation is done. We illustrate the dependence of our method on this scale parameter  $t$  in Section 7.

To compute the Heat Kernel Signature function, we follow the procedure of Sun et al. (8) who define the HKS on the mesh  $M$  as  $k_t(x, x) = \sum_{i=0}^k e^{-\lambda_i t} \phi_i^2(x)$ , where  $\phi_i$  and  $\lambda_i$  are the first  $k$  eigenvalues and eigenvectors of the Laplace-Beltrami operator of  $M$ . We use the discretization of the Laplace-Beltrami operator by Belkin et al. (23), and compute the eigenvalues and eigenvectors using the sparse eigen-solver in MATLAB.

Note that due to exponential decay of the influence of individual eigenvalues, only a few eigenpairs are necessary to estimate HKS for large values of  $t$ . For all of the experiments in this paper, we used  $k = 300$ .

## 5. Theoretical Guarantees

**Persistence-Based Clustering** At the heart of theoretical guarantees of PBC is the fact that PDs can be proven to be stable. We say two PDs,  $Df$  and  $D\hat{f}$ , are close, if their bottleneck distance is small. The bottleneck distance is the  $\ell^\infty$ -distance over all one-to-one matchings between the points of the diagrams. In a seminal result, it was shown for two close tame, continuous functions on a space, the bottleneck distance is bounded by the sup-norm of the difference of the functions (24).

Let  $S \subset \mathbb{R}^3$  be a compact surface and  $f : S \rightarrow \mathbb{R}$  be a  $c$ -Lipschitz function. Assume that we have a mesh,  $M \subset \mathbb{R}^3$  of  $S$ , whose vertex set  $\mathcal{P}$  is contained in  $S$  and that there exists an homeomorphism  $h : M \rightarrow S$  satisfying the following properties: (i) The diameter of any triangle of  $M$  is bounded by some  $\epsilon > 0$ , (ii) For any  $p \in \mathcal{P}$ ,  $h(p) = p$ , and (iii) For any  $p \in S$ ,  $d(p, h(p)) < \epsilon$ . The following then holds:

**Corollary 1.** *For a  $c$ -Lipchitz function  $f$  on a compact surface  $S$ , and for a mesh  $M$  satisfying (i)-(iii), we can define a function  $\hat{f}$  on  $M$ , such that the bottleneck distance is bounded by  $d_B(Df, D\hat{f}) \leq 2c\epsilon$ .<sup>1</sup>*

*Proof.* Define the (non continuous) function  $\hat{f} : M \rightarrow \mathbb{R}$ :  $\hat{f}$  is equal to  $f$  on the vertices  $\mathcal{P}$ . On each cell (edge, triangle) of  $M$ ,  $\hat{f}$  is equal to the maximum of the values of  $f$  at the vertices of the cell. Let  $g = \hat{f} \circ h^{-1}$ . From (i) and (iii), it follows that for any  $p \in S$ ,  $|f(p) - g(p)| < 2c\epsilon$ . Note that  $\hat{f}$  is tame (see (25)) and so  $g$  is also tame. Furthermore,  $g$  has the same persistence diagram as  $\hat{f}$  as the persistence of a function is invariant under homeomorphism. The proposition then follows directly from the stability theorem (24).  $\square$

If the vertex set lies off the surface, the result holds as long as we have estimates of the error in the function value at the vertices through the homeomorphism.

With the stability of the PD assured, we require the PD be decomposable into two disjoint regions: the prominent peaks (persistence greater than  $d_2$ ) and the topological noise (persistence less  $d_1$ ). Intuitively, if the gap,  $d_2 - d_1$ , is sufficiently large compared to the mesh granularity  $\epsilon$ , the number of persistent segments is stable. This follows directly from Corollary 1: with  $\epsilon$

<sup>1</sup>Note that the diagrams for  $f$  and  $\hat{f}$  are defined over different spaces, namely  $S$  and  $M$ .

small enough, the bound on the distance the points can move implies that the two regions will remain disjoint. If HKS is used as the filtration function, because of its isometry invariance, this result immediately extends to the case of two meshes which are respectively close to isometric shapes.

Finally, there is an approximation result on the basins of attraction (Spatial Stability Theorem - Theorem 4.9 (6)). Here we only recount the idea: for all the  $\tau$ -persistent points in  $D\hat{f}$ , the trace of a corresponding segment coincides with the basin of attraction on the underlying surface (almost)-until the first time it gets connected to another  $\tau$ -persistent segment. This region is guaranteed to be stable because it is related to the underlying object, rather than to our measurements. Furthermore, because the function is  $c$ -Lipschitz and sufficiently persistent, the stable region is always non-empty (and the algorithm will return it correctly).

**Heat Kernel Signature** In this section we look at the stability guarantees for the HKS. We first look at the stability of HKS with respect to time. Theorem 3 in (26) bounds the derivatives of the heat kernel function with respect to time as a consequence of the fact that it is a real analytic function for all  $t > 0$ , directly implying that the function is stable in the choice of  $t$ .

For the guarantees of the PBC to apply, we require the function to be well-behaved with respect to space as well as  $t$ . For this we use Theorem 3.3 from (22), which states that the heat kernel is a  $C^\infty$  function in space. Since we are working only on compact surfaces, the continuity condition implies that it is Lipschitz for some finite constant  $c$ .

The final condition is that the function we compute on vertices is close to the HKS on the surface. Ideally the approximation result should hold for all values of  $t$ . The study of the heat kernel is an active area of research, and only partial results exist. The method we use to compute the HKS, the Mesh Laplace operator defined in (23), is known to converge to the surface Laplace-Beltrami operator. This result however, only holds for small values of  $t$ . We conjecture that the Mesh Laplace operator does converge for all relevant values of  $t$  for a fine enough mesh.

These are some preliminary results of the heat kernel and by equivalence the HKS. There is still significant work to be done, including computing explicit bounds on the Lipschitz constant and proving convergence of the operator for a large enough value of  $t$ .

## 6. Regions without Features

In many cases, due to large plateaus in the function value, though the segments are guaranteed to have a

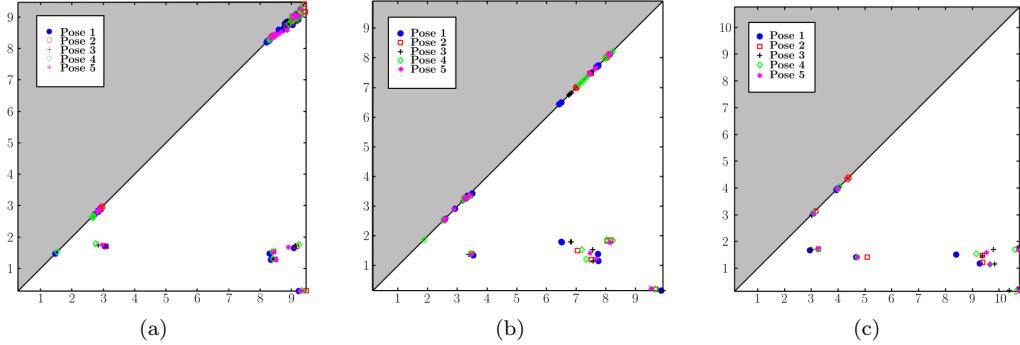


Figure 1. The PD over 5 approximately isometric deformations for (a) the human, (b) the dog, (c) the horse

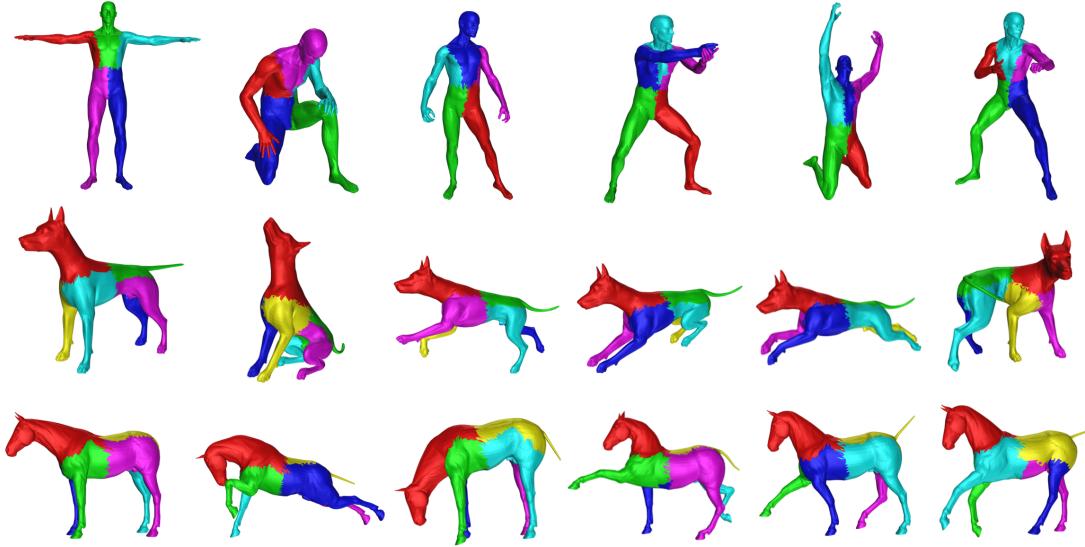


Figure 2. Segmentation with HKS with  $t = 0.1$ , for near isometric deformations of three models.

stable region, an arbitrarily large part of the segments can be unstable (6). Restricting the class of functions does little to alleviate the problem. Using the results of Section 5, we propose an extension of the algorithm which allows us to detect the unstable regions and segment them separately.

Each segment not only has a stable region, but the vertex with maximum value in the segment must also lie in this stable region. This is true under function perturbation, resampling, etc. This is a simple consequence of the Spatial Stability Theorem (6). This implies that there is a unique bijection between stable segments under perturbation. Furthermore, the bijection can be found by simply comparing the segment assignment of the maxima. Hence to find the unstable regions, we simply perturb the function using bounded random noise and see how the segmentation changes. Running this over several times, we detect which points are assigned to different segments.

The extended algorithm then naturally follows. We

begin again by computing the PD of the function. For a selected merging parameter  $\tau$ , we compute  $\beta$  such that no point in the PD has a persistence of  $[\tau - \beta, \tau + \beta]$ . We then run the segmentation algorithm  $m$  times, each time perturbing the original function values by some bounded random noise,  $\sigma$ , such that  $|\sigma| \leq \beta$ . In practice we add uniform noise, however the algorithm will work for any bounded distribution. By Corollary 1, the number of segments,  $k$  is constant over all runs. To find the bijection of the segments from runs  $i$  and  $i+1$ , for each maximum in run  $i$ , we identify which segment the maximum is assigned to in run  $i+1$ . This defines a map between segments which is guaranteed to be bijective. For each point we then store a  $k$ -dimensional vector indicating how many times the point has been assigned to each segment. This vector can be turned into a probability distribution by normalizing over the number of runs. We can then define the notion of  $\omega$ -stable by considering points stable if they assigned to a segment at least  $\omega$  fraction of the time.

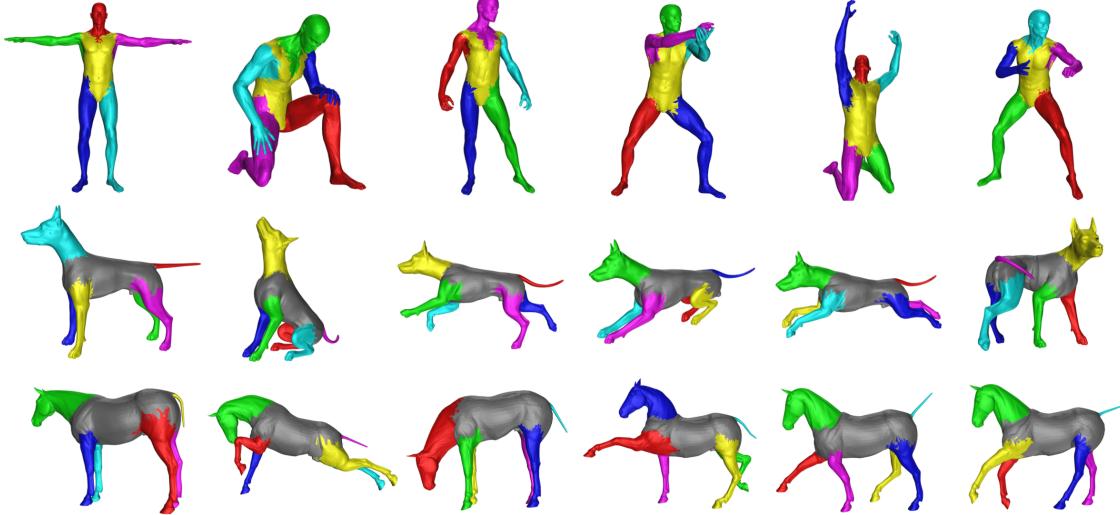


Figure 3. Segmentation with extended algorithm and HKS with  $t = 0.1$ , for near isometric deformations of three models.

A nice property of this approach is that given a bounded distribution on the function perturbations, the segmentation induces a unique measure on the vertices though a push forward from a  $n$ -dimensional hypercube. The probability that a vertex is assigned to the  $i$ -th segment is equivalent to the probability that a point chosen according to the perturbation distribution lands in a certain part of the hypercube. We leave as future work, deriving explicit bounds for the speed of convergence of our method under  $m$ , which can be done using standard results from sampling theory.

## 7. Experiments

We implemented both the standard method (which we refer to as the basic method), and the extended method to segment unstable regions separately. We present a study of 3 models from the TOSCA dataset (27) under different deformations and then show results for the 3D Segmentation Benchmark (5).

**Isometric Deformations** In this section, we show results for experiments with 3 models from (27): a human figure, a dog and a horse. The dataset provided a number of isometric deformations for each model. The dataset also contains a number of different types of perturbations of varying strengths, including: additive, topological, sampling, and shot noise as well as adding holes.

Setting  $t = 0.1$ , the PDs of the corresponding models are shown in Figure 1. The key characteristic of all 3 PDs is that have a small number of points far from the diagonal (5 for the human, 6 for the animals), and varied points along the diagonal.

For the corresponding segmentations (Figure 2), we recover the extremities (arms, legs, head and tail in the latter two) of the models. Note that the body is

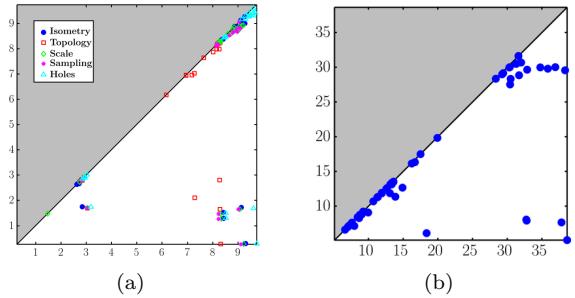


Figure 4. (a) PD comparing different types of noise for the human. (b) PD of human with HKS  $t = 0.001$

generally inconsistent. This is where the HKS function is small over a large area, making the border unstable. Using the extended approach with small additive perturbations, 30 runs and a threshold of 90%<sup>2</sup>, the results are shown in Figure 3. Additive noise on the function, rather than on  $t$  was chosen because they were faster to compute and the results were indistinguishable. In each case, we recover a segment representing the torso. In all of these cases shown, the unstable region was always one segment but in general it can add many segments. We lose some control over the number of segments we obtain in the end, since we do not now a priori how many unstable segments there will be.

Beyond isometric deformations, the PDs for the human with the different deformations are shown in Figure 4(a). The dataset provided 5 strengths of deformations; the results of the middle strength are shown. The results of “shot noise” are not shown, as outliers cause the difference in the  $\infty$ -norm to be large, resulting in a poor approximation. Of the other deformations, the error was due to topological noise. False connections

<sup>2</sup>Every point which did not fall into one segment in 90% or the runs was segmented separately.

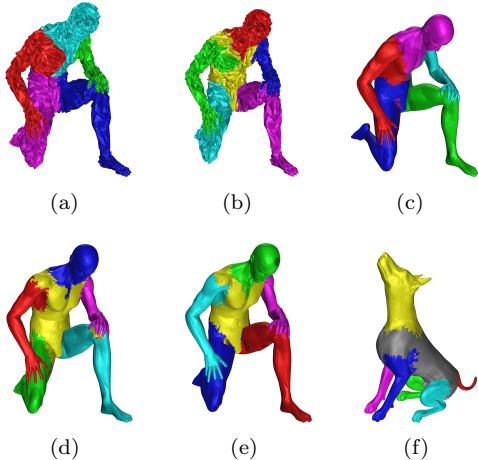


Figure 5. Segmentation of human with noise added (a) for the basic algorithm, (b) extended algorithm. Result for topological deformation with (c) the basic algorithm (d) extended algorithm; Segmentation with extended algorithm (e) with holes in the mesh; (f) Topological noise.

connect components much earlier than they would be otherwise. Both of these violate the assumptions of the proofs and dealing with them is beyond the scope of the paper. The results with additive noise, shown in Figure 5(a,b) are remarkably similar to the isometric case, despite the mesh being quite noisy. Likewise for the case where holes are added (Figure 5(e)). In the case of topological noise Figure 5(c,d), part of the knee is segmented with the hands due to the extra connections. Note however, that in the case of the dog model, extra connections do not connect distinct segments, and the segmentation is unaffected.

A large motivation for using HKS is its multiscale property. Figure 4(b) shows the PD of the human with  $t = 0.001$ , where the same 5 prominent segments are present (two points are very close). However, there are many additional persistent points. These correspond to fingers, toes, and other smaller features. As a result the segments are much noisier. However, as shown in Figures 6(a) and (c), we recover the fingers as segments. The areas such as the arms are assigned haphazardly to different fingers. Certain segments arise from the bends in arms and legs (compare Figure 6(a) and Figure 6(b)) as bending is not an isometry and HKS is not preserved at the fine scales. With the extended method, we see that outside the extremities, the segmentation is mostly unstable (Figure 7).

For the dog model, the smaller value of  $t$ , Figure 8 shows cleaner results, due to the fact that features are not at different scales. Therefore, we recover natural segments, i.e. the individual ears, lower jaw, and parts of the legs. The perturbation again recovered the torso, although it also contains part of the head.

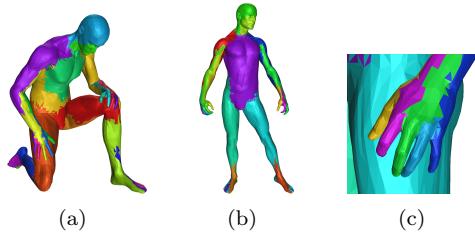


Figure 6. (a) Segmentation of human with  $t = 0.001$ . (b) Different pose (c) Close up of hand.

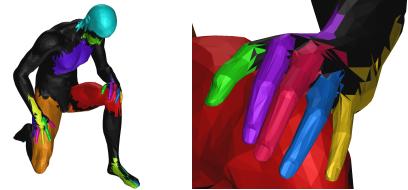


Figure 7. Segmentation with extended algorithm of human with  $t = 0.001$ . Black indicates unstable areas.

Method	CD	RI	HD	CE	
				1	2
Curvature Basic	0.348	0.218	0.257	0.255	0.253
Curvature Extended	0.322	0.221	0.220	0.232	0.201
HKS Basic	0.213	0.124	0.105	0.129	0.067
HKS Extended	0.164	0.120	0.097	0.121	0.061

Table 1. Benchmark Results

**Benchmark** We also ran the 3D Segmentation benchmark (5) on the algorithm. We used the median of the number of segments in the human segmentations for each model. Segmentations were computed with both versions of the algorithm, and using HKS over a range of  $t$ 's and the curvature magnitude provided in the benchmark.

For HKS, we generated segmentations for logarithmically spaced values of  $t \in [0.005, 10]$ . The results were compared against all the human segmentations. Due to limited space, Table 1 shows only the results averaged over the dataset. For HKS, the segmentations for all values of  $t$  were compared, but only the lowest score for each model was included in the average. For a complete description of the metrics see (5). The results are comparable with other methods described in (5). Further, HKS performed better than curvature, due to the fact that segmentations at several scales were produced. Finally, the extended algorithm in both cases has lower scores suggesting that the lack of features is something worth segmenting separately.

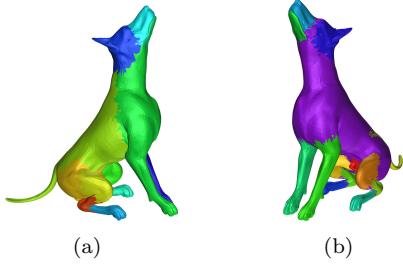


Figure 8. Segmentation of dog with  $t = 0.001$  (a) Basic method (b) Extended method.

## 8. Future Work & Conclusions

In this paper, we have presented a provably stable method for segmentation of isometric shapes, by combining the strengths of persistence-based clustering with the multiscale Heat Kernel Signature function. The use of persistence diagrams not only gives the user a convenient way to chose the proper parameters, but also provides a notion of stability, hinting at what the relevant number of segments should be. We also present a way to segment regions without features and give some theoretical guarantees on our method. The key remaining challenges consist of strengthening our analysis to provide a provably convergent scheme to compute the HKS and analyze its smoothness properties. It would be also interesting to apply our method in the case of point clouds in possibly high dimensions.

## References

- [1] A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis, “3d mesh segmentation methodologies for cad applications”, *Computer-Aided Design and Applications*, vol. 4, no. 6, pp. 827–842, 2007.
- [2] A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis, and N.S. Sapidis, “Retrieval of 3D Articulated Objects Using a Graph-based Representation”, in *Proc. 3dOR*, 2009, pp. 29–36.
- [3] R. Toldo, U. Castellani, and A. Fusillo, “Visual Vocabulary Signature for 3D Object Retrieval and Partial Matching”, in *Proc. 3dOR*, 2009, pp. 21–28.
- [4] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal, “Mesh segmentation - a comparative study”, in *SMI ’06*, 2006, p. 7.
- [5] X. Chen, A. Golovinskiy, and T. Funkhouser, “A benchmark for 3D mesh segmentation”, *ACM Trans. on Graphics*, vol. 28, no. 3, Aug. 2009.
- [6] F. Chazal, L.J. Guibas, S. Oudot, and P. Skraba, “Persistence-Based Clustering in Riemannian Manifolds”, RR 6968, INRIA, 2009.
- [7] H. Edelsbrunner, D. Letscher, and A. Zomorodian, “Topological persistence and simplification”, *Discrete Comput. Geom.*, vol. 28, pp. 511–533, 2002.
- [8] J. Sun, M. Ovsjanikov, and L.J. Guibas, “A concise and provably informative multi-scale signature based on heat diffusion.”, *Comp. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.
- [9] A. Shamir, “A survey on mesh segmentation techniques”, *Comp. Graph. Forum*, vol. 27, no. 6, pp. 1539–1556, 2008.
- [10] A. P. Mangan and R. T. Whitaker, “Partitioning 3d surface meshes using watershed segmentation”, *TVCG*, vol. 5:4, pp. 308–321, 1999.
- [11] E. Zuckerberger, A. Tal, and S. Shlafman, “Polyhedral surface decomposition with applications”, *Computers & Graphics*, vol. 26:5, pp. 733 – 743, 2002.
- [12] A. Razdan and M. Bae, “A hybrid approach to feature segmentation of triangle meshes”, *Computer-Aided Design*, vol. 35:9, pp. 783 – 789, 2003.
- [13] F. de Goes, S. Goldenstein, and L. Velho, “A hierarchical segmentation of articulated bodies”, *Comp. Graph. Forum*, vol. 27:5, pp. 1349–1356, 2008.
- [14] Y.K. Lai, Q.Y. Zhou, S.M. Hu, and R.R. Martin, “Feature sensitive mesh segmentation”, in *Proc. SPM*, 2006, pp. 17–25.
- [15] S. Katz, G. Leifman, and A. Tal, “Mesh segmentation using feature point and core extraction”, *The Vis. Comp.*, vol. 21, no. 8, pp. 649–658, 2005.
- [16] A. Agathos, I. Pratikakis, S. Perantonis, and N.S. Sapidis, “Protrusion-oriented 3D mesh segmentation”, *The Vis. Comp.*, vol. 26, no. 1, pp. 63–81, 2010.
- [17] R. Rustamov, “Laplace-beltrami eigenfunctions for deformation invariant shape representation”, in *Proc. SGP*, 2007, pp. 225–233.
- [18] K. Gebal, J. A. Bærentzen, H. Aanæs, and R. Larsen, “Shape analysis using the auto diffusion function”, *Com. Gr. Forum*, vol. 28:5, pp. 1405–1413, 2009.
- [19] M. Reuter, “Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions”, *Proc. IJCV*, vol. 89:2, pp. 287–308, 2010.
- [20] G. Carlsson, “Topology and data”, *AMS Bulletin*, vol. 46, no. 2, pp. 255–308, 2009.
- [21] A. Zomorodian and G. Carlsson, “Computing persistent homology”, *Discrete Comput. Geom.*, vol. 33, no. 2, pp. 249–274, 2005.
- [22] A. Grigor’yan, “Heat kernels on weighted manifolds and applications”, *Contemporary Mathematics*, vol. 398, pp. 93–193, 2006.
- [23] M. Belkin, J. Sun, and Y. Wang, “Discrete laplace operator on meshed surfaces”, in *SCG*, 2008, pp. 278–287.
- [24] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, “Stability of persistence diagrams”, in *SCG*, 2005, pp. 263–271.
- [25] F. Chazal, D. Cohen-Steiner, M. Glisse, L.J. Guibas, and S.Y. Oudot, “Proximity of persistence modules and their diagrams”, in *SCG*, 2009, pp. 237–246.
- [26] E. B. Davies, “Non-gaussian aspects of heat kernel behaviour”, *LMS*, vol. 55:1, pp. 105–125, 1997.
- [27] A. Bronstein, M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*, Springer, 2008.