

Hierarchical Shape Segmentation and Registration via Topological Features of Laplace-Beltrami Eigenfunctions

Martin Reuter

Preprint: July 21 2009

Abstract This work introduces a method to hierarchically segment articulated shapes into meaningful parts and to register these parts across populations of near-isometric shapes (e.g. head, arms, legs and fingers of humans in different body postures). The method exploits the isometry invariance of eigenfunctions of the Laplace-Beltrami operator and uses topological features (level sets at important saddles) for the segmentation. Concepts from persistent homology are employed for a hierarchical representation, for the elimination of topological noise and for the comparison of eigenfunctions. The obtained parts can be registered via their spectral embedding across a population of near isometric shapes. This work also presents the highly accurate computation of eigenfunctions and eigenvalues with cubic finite elements on triangle meshes and discusses the construction of persistence diagrams from the Morse-Smale complex as well as the relation to size functions.

Keywords Laplace-Beltrami Operator · Hierarchical Mesh Segmentation · Eigenfunctions · Persistence · Morse-Smale Complex

1 Introduction

Shape segmentation has gained much importance in computer graphics for the purpose of modeling, reverse engineering of CAD objects, meshing, compression, shape analysis, shape retrieval, mesh parametrization, texture mapping, and morphing. As more and more 3D mesh



Fig. 1 Segmentation and registration across different poses

models become available the demand for shape analysis and understanding is constantly growing. Shape segmentation, the task to subdivide a shape into meaningful subparts, is at the core of many intelligent automatic algorithms to compare, analyze and understand shape.

The human visual system is extremely efficient in identifying significant features, e.g. by identifying concave regions or regions of negative principal curvature [1], ignoring insignificant noise. In spite of methods that focus on mesh segmentation with respect to geometric properties (i.e. equal area, planarity ...) this work therefore tries to mimic the human visual system and focuses on a hierarchical segmentation of a surface into subparts by identifying features (such as protrusions) on different significance levels. Furthermore, for a population of near isometric shapes our method can guide the registration of the segmented parts across shapes, enabling the transfer of labels or other descriptors, necessary, for example, for distributing semantics.

These goals are achieved by two main steps. First, the segmentation is guided by topological features of a non-constant eigenfunction of the Laplace-Beltrami op-

Martin Reuter (reuter@mit.edu)
 Massachusetts Institute of Technology,
 Department of Mechanical Engineering.
Presently: Massachusetts General Hospital / Harvard Medical
 A.A. Martinos Center for Biomedical Imaging

erator (LBO), whose extrema describe interesting protrusions and whose saddle points denote the corresponding concave regions. Using the concept of topological persistence we can then remove topological noise and describe the shape hierarchically. Therefore, it is possible to automatically detect meaningful features and segment the object at different significance levels. Second, for a group of near-isometric articulated shapes (in different poses) the segmentations are registered by exploiting the isometry invariance of the LBO, thus transferring information from one shape to the next (see Fig. 1). The possible switching and sign flips of the eigenfunctions are compensated by measuring their topological distance with the help of their persistence diagrams. This step can involve user interaction, if the shapes are very symmetric (usually they are, as humans and animals show a left/right symmetry). After possible reordering and/or negating of the functions, the shapes are projected into their spectral domain where the segments can be robustly registered.

2 Related Work

This paper touches many fields from computational geometry, spectral mesh processing and manifold learning [2] as well as computational topology (Morse-Smale complex and persistence diagrams [3]). However, it is most closely related to shape segmentation methods, in particular pose invariant segmentations. Recent overviews on segmentation algorithms can be found in [4] and [5].

Generally shape segmentation methods can roughly be grouped into methods that (1) focus primarily on geometric aspects of the segmentation (equal area, curvature, planarity, angles, normals ...) and methods that (2) try to segment shape into meaningful subparts. The first group consists of algorithms designed for computer aided design (CAD) objects (to detect planar faces, cylinders, tubular parts, etc.) or to guide mesh compression or remeshing procedures. Here specifically [6] should be mentioned as it describes the use of the Morse-Smale (MS) complex of a user selected higher eigenfunction of the mesh Laplace operator and is therefore closely related to this work. However, their aim is to remesh the shapes with quadrangular elements for which they use the integral lines of the MS complex to create initial quad patches, while our work tries to segment the shape into meaningful parts by taking the level sets at important saddles of one of the first eigenfunctions as cutting curves. Therefore, the presented work is relatively invariant with respect to pose, while the higher eigenfunctions used in [6] are very likely to change significantly for different poses of the shape.

Furthermore, their discrete Laplace operator is not invariant with respect to local density changes of the input mesh (as it is missing the mass matrix).

We will focus on algorithms from the second group mentioned above, as they are usually designed for articulated objects and therefore include the method described in this paper. A recent example is [7] which extends a random walk method to meshes for automatic segmentation (see also the related work section for a good overview). The authors propose a seed based method, that tries to align to concave regions (or convex edges for CAD type objects). Very often region growing algorithms are used, however, it can be difficult to automatically select seed points in an optimal or consistent manner and to terminate the growth at meaningful locations. Post processing is usually needed to smoothen the borders of the final regions. Using level sets of low frequency eigenfunctions, as proposed in this work, yields smoothly bounded segments and avoids seed point selection.

Quite a few segmentation methods, e.g. [8,9], are hierarchical (as is the one in this paper). These methods construct segmentations into parts and subparts, though at smaller scales the parts are not necessarily meaningful anymore. The method described in this work will not construct sub-segmentations if no meaningful subparts exist. A mesh segmentation that can be used to construct a mapping between two shapes is given in [10]. It is not designed for near isometric shapes and does not align segments to features, but can deal very well with higher genus surfaces by identifying the handles. To create a meaningful correspondence for the mapping feature/surgery points have to be placed by the user.

Recently methods for pose invariant shape segmentation have become popular. They are usually based on intrinsic metrics (e.g. geodesic distances) and often exploit some kind of spectral embeddings of various operators/matrices. A pose invariant shape representation based on geodesic distances and multidimensional scaling (as proposed in [11] with application to shape classification) has been used for shape segmentation in [12] by extracting feature points and cores. The method can produce consistent segmentations, but the algorithm to localize feature points is expensive and limits the complexity of models. Furthermore, the use of geodesic distances is unstable with respect to topology changes of the meshes.

Similar methods, but without the isometry invariance, include [13] who use a spectral embedding of an affinity matrix containing geodesic distances plus non-intrinsic angular distances of faces to perform a k -means clustering. The authors later remove the dependency

on geodesic distances in [14] and instead propose to use the graph Laplacian and another more geometric operator M to embed the shape into the plane via the first two non constant eigenvectors. The graph Laplacian is based only on mesh connectivity and therefore not robust w.r.t. mesh connectivity and resolution while M is based on minimal principal curvatures and thus not intrinsic. That is why both these spectral methods were not utilized and are not suited for pose invariant settings. The Laplace-Beltrami operator in this paper on the other hand is invariant w.r.t. meshing and isometries.

In [15] a very similar Laplace operator is used, however, constructed by only linear finite elements with a lumped (diagonal) mass matrix (see also [16] for a comparison of different Laplace Beltrami discretizations). Also a simple segmentation example is presented in [15] again based on k -means clustering of the spectral embedding of near isometric shapes. These spectral embeddings have been described in the past in manifold learning [2, 17] and are also used in this work for the final registration step, but not for the segmentation. Note that the segmentations in [15] depend on the number of desired clusters and do not generally align to any shape features.

Finally another more recent work [18] needs to be mentioned. It describes the use of diffusion distances for pose invariant hierarchical shape segmentation. As the diffusion distance is just the Euclidean distance in the spectral embedding space, the fundamental idea is similar to [15] mentioned above (however with a differently scaled embedding space and additional medial structures to guide the segmentation). As we will explain later in more depth, spectral embeddings suffer from the fact, that sign-flips and switching can disturb the embedding. Furthermore, these methods need to compute several eigenfunctions until a cut-off is reached and it remains unclear how accurately these functions can be computed with the linear discretizations of the Laplace operator used. Nevertheless, [18] shows some impressive hierarchical segmentations. Even though they detect the features on coarser resolutions, they do not necessarily align to the concavities and also segment the shape at locations without specific features on higher resolution levels. Even though their segmentations stay approximately fixed across near isometric deformations, they are not registered as is done in this work.

Note that a closely related field is shape correspondence. Especially [19, 20] and [21] employ very similar methods. [19] again use an embedding space of the affinity matrix filtered with a Gaussian kernel, while the other two methods rely on a graph Laplacian (borrowed from manifold learning) for the embedding of the 2D

surface meshes and 3D voxel-based shapes respectively. The difficulties of sign flips and eigenvector switching are discussed and approached differently. We will describe these problems in more detail in Section 3.2 and the different approaches to deal with them in Section 4.6. Note that [22] employs level sets of eigenfunctions to implicitly construct correspondence of brain structures for statistical shape analysis. Other work employing spectral entities of the Laplace Beltrami operator includes retrieval [23, 24], medical shape analysis [25–27, 22], filtering/smoothing [28–30] of which specifically [28] mentions the use of zero level sets of specific eigenfunctions for mesh segmentation. This idea is later analyzed in more detail in [16], where also a comparison of different common discrete Laplace Beltrami operators is given. Zero level sets, however, do not generally align to shape features, therefore we will not use them here. A recent overview on spectral mesh processing can be found in [31].

From the above discussion it becomes clear that the main improvements of this method and contributions of this paper include:

- the smoothness of bounding curves of the segments,
- the fact that no sub-segmentations are created if no geometric features are available there,
- the high accuracy of the eigenfunctions via cubic FEM as opposed to many mesh Laplacians (see [16, 32])¹,
- the capability to do hierarchical pose invariant segmentations and part registrations at the same time,
- the robustness to different poses,
- the robustness with respect to mesh quality, density and noise
- the detailed treatment of the difficulties (sign flips, switching)
- and the detailed overview on the background material (FEM, Morse-Smale complex, size functions, separated persistence diagrams) and description of the implementation.

After describing the necessary background on the Laplace-Beltrami operator and the problem of ordering and negating its eigenfunctions (Section 3), we will focus on the Morse-Smale (MS) complex and persistence diagrams as a tool to analyze the eigenfunctions and propose the “separated persistence diagram”. We develop a method to construct the persistence diagram from the MS complex and discuss connections to other topological structures (such as the Reeb graph and size functions). Especially the comparison between persistence diagrams and size functions seems to be novel

¹ Our software “shapeDNA-tria” can be obtained from <http://reuter.mit.edu>

and discovers that they are not quite the same even for real-valued functions. Section 4 presents the proposed method with a detailed description of the implementation employing the cubic finite element method (FEM) on triangular meshes. We show how the eigenfunctions can be reordered and negated, using persistence diagrams, to allow for a spectral projection and registration of the segmented parts. Finally, we present examples of segmentations and registrations and results on robustness in Section 5.

3 Background

3.1 Riemannian Manifolds

A geometric shape can be described by its 2D (boundary) surface or if it is closed also by its 3D volume. In this work we choose to concentrate on the surface case only (2D Riemannian manifold). A Riemannian manifold is a manifold M endowed with a Riemannian metric $I(v, w)$, also called the first fundamental form. The **Riemannian metric** is an inner product in the tangent space and can be defined for two tangent vectors v and w with the help of the first fundamental matrix G . For example for a given local surface parametrization $P(u_1, u_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ with partial derivatives $P_1 := \partial P / \partial u_1$ and $P_2 := \partial P / \partial u_2$ the metric is given by:

$$I(v, w) = v^T G w \quad (1)$$

$$\text{with } G := (g_{ij}), \quad g_{ij} = P_i P_j \quad (2)$$

The first fundamental form is used to compute lengths, angles and areas on the manifold. The surface element $d\sigma$, for example, can be computed as:

$$d\sigma = |P_1 \times P_2| du_1 du_2 = \sqrt{\det G} du_1 du_2 \quad (3)$$

Since it is possible to measure lengths, we can compute shortest distances between two points $x, y \in M$ along the surface via $\text{geo}_M(x, y)$, called the **geodesic distance**. All quantities that depend on the metric only are **intrinsic** (also ‘‘isometry invariant’’) as they do not depend on the actual isometric embedding of the manifold within the embedding space (usually the \mathbb{R}^3). For example, the LBO spectrum and eigenfunctions discussed below are intrinsic properties.

Two Riemannian manifolds M_1 and M_2 are **isometric**, if a bijective map $\phi : M_1 \rightarrow M_2$ exists, preserving geodesic distances of any two points: $\text{geo}_{M_1}(x, y) = \text{geo}_{M_2}(\phi(x), \phi(y))$. The function ϕ is called an isometry and can be understood as a bending (inelastic deformation) that does not contain any stretching. A piece of paper, for example, can be bend into the third dimension without stretching. Common deformations (e.g. a

person in different body postures) will of course include some local stretching, but as this effect is relatively small, such shapes will be called **near-isometric**.

To better define this term, we shall use the **Gromov-Hausdorff distance** [33, 34], which is a useful measure of distance between two metric spaces (here bounded Riemannian manifolds M_1, M_2). For the two embedding maps $\phi : M_1 \rightarrow M_2$ and $\psi : M_2 \rightarrow M_1$ the mutual embedding error can be measured by taking two pairs from the set $\mathcal{G}(\phi, \psi) = \{(x, \phi(x)), x \in M_1\} \cup \{(\psi(y), y), y \in M_2\}$ and then computing the maximal deviation of the distances of the corresponding two points within each of the manifolds:

$$\mathcal{E}(\phi, \psi) = \sup_{\substack{(x_1, y_1), \\ (x_2, y_2) \\ \in \mathcal{G}(\phi, \psi)}} |\text{geo}_{M_1}(x_1, x_2) - \text{geo}_{M_2}(y_1, y_2)|. \quad (4)$$

The Gromov-Hausdorff distance basically tries to find the embedding maps ϕ and ψ that minimize the embedding error:

$$d_{\mathcal{GH}}(M_1, M_2) = \frac{1}{2} \inf_{\phi, \psi} \mathcal{E}(\phi, \psi). \quad (5)$$

$d_{\mathcal{GH}}(M_1, M_2)$ will be zero if and only if the manifolds are isometric. If $d_{\mathcal{GH}}(M_1, M_2) \leq \epsilon/2$ we call the manifolds near-isometric (with embedding error ϵ). This concept of distance does not allow large metric deformations, even if they are very local. The reason for this restrictive setup is not, that the first few eigenfunctions will be influenced much by such a local change (in fact they are very robust w.r.t. such high frequency features), but rather numerical issues that arise in the FEM computation if too many ill-shaped triangles with nearly zero area are involved.

For a real-valued function $h : M \rightarrow \mathbb{R}$ defined on the manifold ($h(u_1, u_2)$ with partial derivatives h_1 and h_2) the **gradient** is

$$\text{grad } h = \nabla h = (P_1, P_2) G^{-1} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \quad (6)$$

where (P_1, P_2) is the matrix containing the partial derivatives of the parametrization P as columns. Note that the product of the gradients of two functions f and h is then given by

$$\nabla(f, h) := I(\text{grad } f, \text{grad } h) = (f_1, f_2) G^{-1} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \quad (7)$$

3.2 Laplace-Beltrami

The **Laplace-Beltrami operator** Δ (LBO) is the generalization of the Laplace operator to real-valued (twice

differentiable) functions f on any Riemannian manifold M (here M is always compact). It can be defined

$$\Delta f := \text{div}(\text{grad } f) \quad (8)$$

with $\text{grad } f$ the **gradient** of f (Eq. (6)) and div the **divergence** both defined on the manifold (see Chavel [35]).

The LBO is the one operator that fulfills the following equation for any two (twice) differentiable functions f and h :

$$\int_M h \Delta f \, d\sigma = - \int_M \nabla(f, h) \, d\sigma \quad (9)$$

with the surface (or volume) element $d\sigma$ (Eq. (3)). Eq. (9) is still true for manifolds with boundary, if a Dirichlet or Neumann boundary condition (see below) is applied.

The **spectrum** of the LBO consists of **eigenfunctions** f_i each with a corresponding **eigenvalue** λ_i , namely the solutions to the Helmholtz equation:

$$\Delta f + \lambda f = 0. \quad (10)$$

If the manifold has a boundary Γ_M , we need to specify a **boundary condition**, in order to be able to solve this equation. We have basically two options: the Dirichlet boundary condition fixes the function $f \equiv 0|_{\Gamma_M}$ to be zero on the boundary, while the less restrictive Neumann boundary condition fixes the derivative of f into the normal direction of the boundary to be zero $\partial f / \partial n \equiv 0|_{\Gamma_M}$. If not stated explicitly, we have a closed surface or we use the Neumann condition.

Many properties of the LBO and its spectrum are known. Here we will focus on the properties that are important in the context of this paper. For more details we refer the reader to our earlier work [23,36] and to Courant and Hilbert [37]. We will assume the manifold M to be connected.

1. The (ordered) spectrum $\lambda_0 < \lambda_1 \leq \lambda_2 \leq \dots$ is a diverging sequence of positive real valued numbers. It is discrete for a compact manifold.
2. The LBO has an orthonormal base of eigenfunctions f_i . If we have a higher dimensional eigenspace the eigenvalue will be repeated in the ordered sequence above and we can construct an orthonormal basis of that eigenspace.
3. If the manifold is closed or if the Neumann boundary condition is used, the first eigenvalue λ_0 will be zero and its eigenfunctions f_0 are the constant functions. We will ignore it and start with f_1 in this work.
4. Embedding the manifold into its spectral domain by the map

$$\Psi_n(x) = \left(\frac{f_1(x)}{\lambda_1}, \frac{f_2(x)}{\lambda_2}, \dots, \frac{f_n(x)}{\lambda_n} \right) \quad (11)$$

where $x \in M$ is often done in manifold learning [2], because this leads to an optimal embedding. In particular the first eigenfunction (in the discrete graph Laplacian setting also called the Fiedler vector) gives an optimal projection of the mesh onto the real line. This can be seen by looking at the Dirichlet energy $E(f) := \int_M \|\text{grad } f\|^2 \, d\sigma$ that measures the smoothness of a map f . By setting f to a normed eigenfunction we get (with Eq. 9)

$$E(f) = \int_M \nabla(f, f) \, d\sigma = - \int_M f \Delta f \, d\sigma = \lambda_i \quad (12)$$

thus, the first non-constant eigenfunction gives the smoothest map (orthogonal to the constant functions and complying with the boundary conditions) as it has the smallest eigenvalue. Nevertheless, cutting off the projection at some higher n might lead to difficulties without prior processing of the functions (reordering) as proposed below.

An example of eigenfunctions on a Riemannian manifold are the spherical harmonics. They represent an orthonormal basis of the eigenfunctions on the sphere. They are given by:

$$Y_l^m(\theta, \varphi) := N e^{im\varphi} P_l^m(\cos \theta) \quad (13)$$

with a normalization factor $N(l, m)$, the order $l \geq 0$ and degree $|m| \leq l$ and the associated Legendre polynomials P_l^m . Note, by $e^{iy} = \cos(y) + i\sin(y)$ the complex spherical harmonics can be separated into their real and imaginary parts which differ by a rotation. We plot $\text{real}(Y_l^m)$ for $m > 0$ and $\text{img}(Y_l^m)$ for $m < 0$ to obtain orthogonal functions. The polar angle θ ranges from $0 \leq \theta \leq \pi$ and the longitude φ from $0 \leq \varphi \leq 2\pi$. All functions with the same order l are within the same eigenspace, so the eigenspaces have the dimensions $2l + 1$, i.e. 1, 3, 5... .

Only in very rare cases do we know the exact spectrum, therefore it usually needs to be approximated numerically (we use a cubic finite element method - FEM, in order to obtain highly accurate results). But even if we were able to compute the eigenfunctions and eigenvalues with a very high accuracy there would still be three problems when comparing or working with them:

1. *Sign flips* can occur, as all scalar multiples of an eigenfunction are contained in the same eigenspace, we might obtain the negative function (after normalization).
2. *Higher dimensional eigenspaces* can theoretically occur. The numerical solver will construct an arbitrary basis of such an eigenspace. Even worse, due to numerical errors the corresponding eigenvalues will not

be exactly the same, so it is not even clear that we really have a higher dimensional eigenspace. In these cases cubic FEM can add stability in the numerical computations, yielding highly accurate eigenvalues. In most settings, higher dimensional eigenspaces are rare. They occur in highly symmetric shapes, e.g. the sphere or square, but even in the case of the rectangle with side length a and b they can only exist if a/b is rational². Nevertheless, being close to a higher dimensional eigenspace can already be problematic.

3. *Switching of eigenfunctions* can occur because of several reasons. The eigenfunctions are ordered according to their eigenvalue. Two close eigenvalues can switch their order because of numerical instabilities or because of geometry deformations. As we will see at the example of the sphere, this can happen, if a rotational symmetry is destroyed through a geometry deformation in two different ways, thus splitting a two-dimensional eigenspace into two one-dimensional spaces. But in general it does not need to be an unstable symmetry that causes the problem.

The sphere, for example, has high dimensional eigenspaces, due to its symmetries. Actually the first eigenspace (ignoring the constant eigenspace) is three dimensional. A basis can be given by three orthogonal eigenfunctions, each dividing the sphere along their zero level sets into north-south, east-west and front-back hemispheres. So the zero level set (nodal line) of the first (Y_1^0) is the equator while the nodal lines of the later two (Y_1^{-1} and Y_1^1) are the great circles through the poles, rotated by 90 degrees. In general the nodal lines are m great circles passing through the poles and $l - m$ circles of equal latitude. Eigenfunctions that have only circles with constant latitude as their nodal lines (if and only if $m = 0$) will be called “*latitudinal eigenfunctions*”, eigenfunctions with only great circles through the poles (if and only if $|m| = l$) “*pole eigenfunctions*” and eigenfunctions with both properties “*mixed*”. When stretching the sphere into the direction of the poles, it becomes clear that the pole eigenfunctions Y_1^{-1} and Y_1^1 will remain in a two-dimensional eigenspace (due to the remaining symmetry) while the latitudinal eigenfunctions Y_1^0 will separate. Now depending on the transformation (either stretching or compressing) the order of the eigenfunctions will switch.

When stretching the sphere the latitudinal eigenfunction Y_1^0 will have a lower eigenvalue. This is due

² Proof: WLOG let $a = 1$ and $b \in \mathbb{R} \setminus \mathbb{Q}$ be irrational. The eigenvalues are $\lambda_i = \pi^2(M^2 + N^2/b^2)$ with $M, N \in \mathbb{N}_{\geq 0}$ therefore we need to find two pairs $(M_1, N_1) \neq (M_2, N_2)$ such that $b^2(M_1^2 - M_2^2) = (N_2^2 - N_1^2)$ which is impossible.

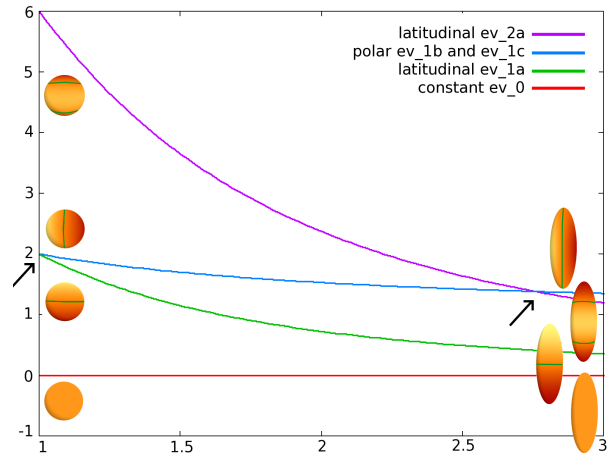


Fig. 2 Trace of the first lowest eigenvalues for different r_z . The higher dimensional eigenspaces of the sphere are split as soon as $r_z > 1$. There is another switch at $\tilde{r}_z \approx 2.76$.

to the optimal embedding property. As the first eigenfunction (with non-zero eigenvalue) is the optimal embedding map of the manifold onto the real line (of all the eigenfunctions) it follows the main direction of the elongated object. Figure 2 shows how the corresponding eigenvalues behave when stretching the ellipsoid even further. The eigenvalues have been computed numerically with the FEM method for smooth surfaces with known parametrization as presented in [23,36] and are highly accurate. It can be seen that another eigenvalue (it is the one of the latitudinal function Y_2^0) from the next higher eigenspace drops quickly and collapses with the one of the pole eigenfunctions from the first space at a $\tilde{r}_z \approx 2.76$ (radius in the pole direction). So for radii above this \tilde{r}_z we have two latitudinal functions as the first eigenfunctions with lowest eigenvalues. This demonstrates that small changes in geometry can lead to switching of the eigenfunctions even without destroying any symmetries.

Finally, we will analyze the behavior of the spectral embedding Ψ_n around \tilde{r}_z . Figure 3 depicts the spectral embeddings into 3D and 2D space of two ellipsoids with nearly the same z -radius. Because of the two latitudinal eigenfunctions for $r_z > \tilde{r}_z$ the embedding collapses to a curve in 2D (a flat surface in 3D), while for $r_z < \tilde{r}_z$ the embedding is egg-shaped without any collapsing dimensions. These difficulties are often ignored, but happen quite frequently when dealing with similar, possibly stretched objects. We will approach them with the help of Morse theory.

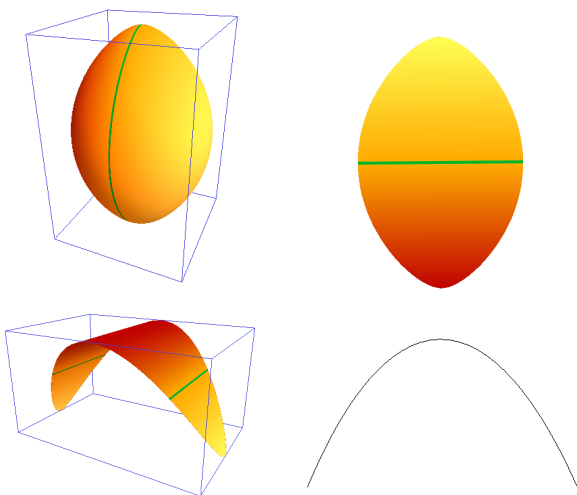


Fig. 3 Spectral projection onto the first 3 (left) and first 2 (right) eigenvectors. Top row: embedding of ellipsoid with $z_r = 2.74$ and bottom row with $z_r = 2.76$. A slight difference in the stretching factor changes the embedding drastically (a whole dimension collapses). The colors show eigenfunction 3 (left) and 2 (right).

3.3 Morse-Smale

Here we want to give a short overview on the concepts needed for the topological analysis of eigenfunctions. As eigenfunctions can be considered quite tame (always bounded when defined on compact manifolds) they can be studied well with Morse theory. The reader is referred to [38, 39] for more background on Morse theory and on the Morse-Smale complex. A good overview also treating the evolution of the used concepts (size functions, Morse-Smale complex, Reeb graph and persistent homology) can be found in [40].

Definition 1 The **critical points** of a real-valued function h on M are the locations where the gradient vanishes ($\nabla h = 0$). A critical point is called **non-degenerate**, if the Hessian (the matrix containing the second partial derivatives) is non-singular at that point. A **critical value** is a value y of h where $h^{-1}(y)$ contains a critical point.

For example, the saddle point in the center of a monkey saddle is a degenerate critical point (it is a 2-fold saddle). In the 2D surface case the non-degenerate critical points are the extrema (minima, maxima) and simple saddle points (1-fold saddles).

Definition 2 A **Morse function** is a smooth function whose critical points are non-degenerate.

In general any twice differentiable function can be unfolded to a Morse function by unfolding its k -fold saddles ($k > 1$) into lower saddles. This holds for the

eigenfunctions of the LBO, which are not necessarily Morse functions (e.g. recall the specific spherical harmonics where many nodal lines pass through the poles as great circles, thus creating k -fold saddles with arbitrary high k).

Definition 3 The **level set** L_x of a real-valued function h on the manifold M consists of the points $p \in M$ where $h(p) = x$ (so it is the pre-image $h^{-1}(x)$). The **lower level set** $L_{<x}$ are the points p where $h(p) < x$ and the upper level set can be defined in similar manner.

The critical points and level sets of one of the first eigenfunctions will be used as important features later. In this context it will be essential to identify important critical points and filter unimportant “topological noise” (see Fig. 8, left, for an example of a minimum-saddle pair that needs to be canceled). This can be achieved by constructing the Morse-Smale complex [3] and using persistence to establish a hierarchical representation as well as a persistence diagram as a descriptor of the analyzed function h . For that we need the following definitions.

Definition 4 An **integral line** $l(s) : \mathbb{R} \rightarrow M$ is a maximal path on the manifold M whose tangent vectors agree with the gradient of h . The origin of l is $orig(l) = \lim_{s \rightarrow -\infty} l(s)$ and the destination $dest(l) = \lim_{s \rightarrow \infty} l(s)$. Then the stable manifold $S(a)$ and the unstable manifold $U(a)$ of a critical point a are defined:

$$S(a) = \{a\} \cup \{p \in M : p \in l, dest(l) = a\} \quad (14)$$

$$U(a) = \{a\} \cup \{p \in M : p \in l, orig(l) = a\} \quad (15)$$

The limits of the integral lines (origin and destination) are critical points of h . Integral lines are either disjoint or the same and cover all non-critical points of M . The stable and unstable manifolds decompose M into regions of similar flow.

The **Morse-Smale complex** is formally constructed by intersecting the stable and unstable manifolds to obtain the Morse-Smale cells that build the complex. It can be thought of as following integral lines from the saddles to the extrema. Therefore, the MS complex can be represented as a graph, whose nodes are the critical points and whose edges are the paths (integral lines) connecting each saddle with the corresponding extrema. The MS complex divides the manifold into patches whose integral lines reach the same minimum and maximum³. Figure 4 depicts the MS complex of the 12th eigenfunction on the torus. The level sets of

³ Note that the subdivision induced by the Morse-Smale complex in conjunction with a user selected eigenfunction of the mesh-Laplacian (a discrete version of the LBO, in fact a sim-

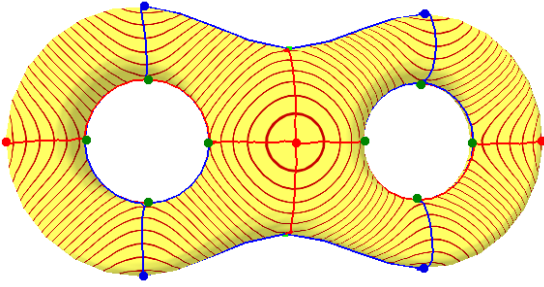


Fig. 4 MS Graph of the 8th eigenfunction (level sets) on the bitorus.

the eigenfunctions are shown together with the critical points (4 minima blue, 4 maxima red, 10 saddles green) and edges of the MS graph.

By pairing saddles with neighboring extrema the Morse-Smale graph provides information on the level set of h . In fact, when sweeping through the function h into the positive direction (starting at minus infinity) and observing the development of the lower level set components and holes (non-bounding 1-cycles), we see that at each minimum a component is created while at each maximum we close a hole. There exist two types of saddles (positive and negative). For genus zero surfaces the negative saddles connect two components (deleting one component) and the positive saddles create a new hole.

The MS complex guides the analysis of the topology by incorporating **persistence**. The concept of persistence pairs each negative saddle with a preceding minimum and each maximum with a preceding positive saddle in a way that minimizes the differences of the function values for each pair. That is, a negative saddle is paired with the higher of the two minima whose components it connects. Similarly a positive saddle is paired with the smaller of the two maxima that close the created holes. The persistence at a critical point is the absolute height difference $ps(a) = |h(a) - h(b)|$ if a and b are paired. For an unpaired critical point it is $ps(a) = \infty$. For a genus g surface the global maximum and minimum and the $2g$ positive saddles at the handles (starting the $2g$ cycles that never die) will remain unpaired in ordinary persistence. We will describe the construction of the persistence pairs from the MS complex in Section 4.3.

A **persistence cancellation** removes a saddle/extremum pair and can be used to remove topological noise or to represent the MS complex on different res-

olved linear FEM without mass matrix) has been used in [6] to remesh shapes with quad-meshes. We are not interested in this subdivision, as it is not aligned with shape features, instead we propose to use the specific curves given by the level sets as cutting curves (more later).

olutions. Imagine, we want to cancel a positive saddle s and its paired local maximum m . This is done by extending the paths from the other saddles that reach the maximum m down through the saddle s up to the higher maximum on the other side of s . The saddle s and maximum m and all paths starting or ending at s are removed from the complex. A saddle/minimum cancellation is done in a similar fashion. Usually the paths are not completely deleted, but only hidden and stored in a hierarchical representation (see also [41]) making it possible to vary the desired level of detail freely. If the persistence pairs are ordered by their persistence, then the i -th pair is connected by an edge in the MS graph after i persistence cancellations.

A good way to represent the topology of the function h is given by the **persistence diagram** [42, 43], where the persistence pairs are plotted as a set of points $(x, y) = (h(a), h(b))$ for each minimum/saddle or saddle/maximum pair (a, b) . It is possible to have points with higher multiplicities and the diagonal is also considered as additional points (with infinite multiplicity, representing components that are created and immediately destroyed). In order to be able to easily distinguish between a creation by a minimum and the creation of a hole by a positive saddle we suggest to plot the two types of pairs on different sides of the diagonal in the persistence diagrams, i.e. to plot $(x, y) = (h(a), h(b))$ for each minimum/saddle or maximum/saddle pair (a, b) . Thus the maximum/saddle pairs are below the diagonal. By doing so we can easily compute the persistence diagram of the negated function $-h$ by mirroring at the origin, as negation exchanges the roles of minima and maxima as well as positive and negative saddles (see Figure 5). We will call these diagrams **separated persistence diagrams (SPD)**.

It should be noted that the older concept of **size functions** is closely related to the persistence computation above (see e.g. [44] for pairing of minimum points and saddles). Size functions [45, 46] are defined on the half-plane $H^+ := \{(x, y) \in \mathbb{R}^2 : x < y\}$ and count the components of the lower level sets $L_{\leq y}$ that contain points with $h(p) \leq x$. More formally on a manifold M with a real valued function $h : M \rightarrow \mathbb{R}$ the size function

$$l_{(M,h)} : H^+ \rightarrow \mathbb{N} \quad (16)$$

is defined, so that for each $(x, y) \in H^+$ the function $l_{(M,h)}(x, y)$ is equal to the number of connected components of the set $L_{\leq y} := \{p \in M : h(p) \leq y\}$ that contain at least one point where the function h is smaller or equal to x . So y cuts off the top part of h and we count the connected components of lower level sets $L_{\leq y}$ that contain points with values that go deep enough (at

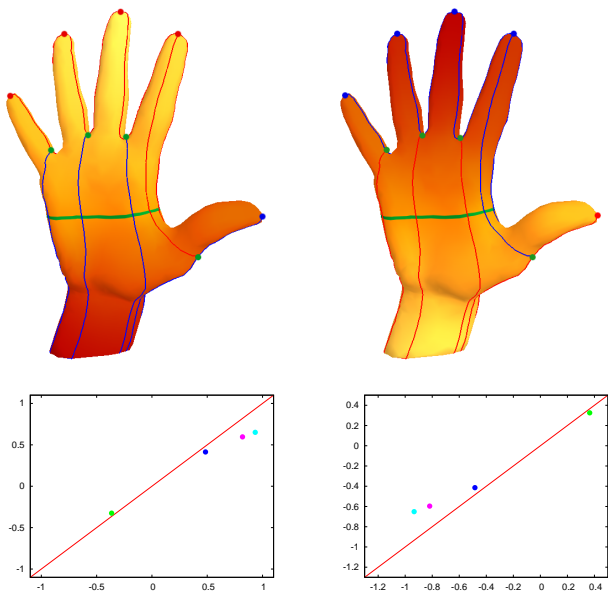


Fig. 5 MS-Graph and SPD of the 1st EF on the hand model (left) and negated 1st EF (right).

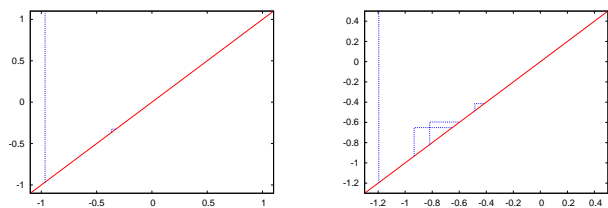


Fig. 6 Size function of the 1st EF on the hand model (left) and negated EF1 (right).

least down to x). See Figure 6 for the plot of a size function. In fact the same information can be represented by only storing the corner points of the small triangles where the size function is discontinuous (for now ignoring the vertical line at the global minimum x_0 of h). The discontinuities of the size function can only occur if the level set at x or y contains a critical point, so at the corners both x and y are critical values⁴.

Let us sweep y from the absolute maximum downwards. If the level set at y sweeps across a maximum on a closed manifold, we cut off a cap but the connectivity of the lower level sets $L_{\leq y}$ is not changed, so nothing happens. The same is true when passing a positive saddle. Here two of the caps we cut out before connect and create a bigger hole but the connectivity of the lower level sets is not changed. A change can only happen when passing a negative saddle. A negative saddle merged two level set components when coming from below so now (coming from above) we split them, cre-

ating a new component. When passing a minimum we definitely destroy a component, but as $x < y$ this component has not been counted anyway. So as y shrinks, the number of components can only increase at negative saddles.

Similarly looking at the x direction. As x shrinks we have less components that have values equal or below x , so we can only reduce the number of counted components. This can only happen when passing a local minimum, because then that component might be removed from the list, if it does not contain other minima that lie below x .

Note that saddle/maximum pairs are not considered at all. The question is: are the corner points (x, y) that represent a size function a subset of the points in the persistence diagrams? This is actually true. When sweeping through h from above and splitting a component at a negative saddle then the component that is deleted first at a minimum at $x < y$ is the one that has the smallest absolute difference in function values and thus this saddle and minimum are indeed a persistence pair. After cancellation we can continue this argument. The size function of the hand model in Figure 5, for example, can be obtained from the separated persistence diagram by removing the points below the diagonal⁵. So the size function contains a subset of the information in the persistence diagrams. It is possible to compute both the size function of h and $-h$ (see also Fig. 6) to obtain all persistence pairs.

Let us briefly look at the **Reeb graph** as it is often used in shape analysis applications for the same purpose, to analyze the topological behavior of a function. The Reeb graph [48] captures the evolution of the components of the level sets (see Fig. 19 for two examples). The Reeb graph of a function h is obtained by contracting the connected components of the level sets to points. Thus the branching points and leaves (end points) in a Reeb graph correspond to level set components that contain a critical point of h . The leaves are the extrema while the branching points are the saddles (either splitting a level set component or merging two components). The other points can be considered to lie on the edges between leaves and branching points. Note that the Reeb graph is a one-dimensional topological structure (a graph) with no preferred way of drawing it in the plane or space or attaching it to M (unlike the MS complex). Nevertheless, for higher genus surfaces ($g > 0$), the Reeb graph can be used to additionally pair so called “essential” homology classes (see [49, 50])

⁴ See also Frosini [47] who proves that it is sufficient to only analyze the Morse graph for the construction of the size function.

⁵ Note that additionally a vertical line needs to be drawn at the global minimum as that is the location in the size function where the number of components changes from 1 to 0.

that represent important features such as handles (this is called extended persistence).

4 Method

We propose a method to segment shapes into parts (hierarchically) while registering the parts across shapes. The proposed method consists of the following steps:

1. Computation of the first eigenfunctions
2. Construction of the Morse-Smale complexes
3. Topological noise removal on the first non-zero eigenfunction
4. Hierarchical subdivision of the shape based on the level sets at saddles
5. Construction of the persistence diagrams of the first eigenfunctions
6. Reordering and flipping of eigenfunctions
7. Registration using the spectral embedding

We will describe these steps in the following sections and focus on the implementation.

4.1 Computation of Eigenfunctions

The eigenfunctions are computed with a Finite Element method (FEM) using cubic elements. The advantage of the FEM approach over other discrete mesh Laplacians is the easy extension to higher degree approximations. These higher degree elements yield very accurate results as they are known to converge quickly to the exact solution (for a shrinking mesh size the convergence is of order 6 for the eigenvalues and order 4 for the eigenfunctions in the L_2 norm [51]). See [16] for a comparison of some common discrete Laplace operators.

Probably the first to employ a linear FEM approach on triangular meshes was [52]. The higher order approach used here is already described in [53, 23, 36] for triangle meshes, NURBS patches, other parametrized surfaces and (parametrized) 3D tetrahedra meshes and in [26] for voxels. Here we will explain the computation for the specific case of piecewise flat triangulations and give closed formulas in the Appendix. Note that [54] also contains a description of higher order FEM on triangle meshes. Note that our software “shapeDNA-tria” for the computation of eigenvalues and eigenfunctions via up to cubic FEM on triangle meshes is available at <http://reuter.mit.edu>.

The FEM computation is based on a variational formulation (which holds for any test/form function φ):

$$\begin{aligned} \iint \varphi \Delta F \, d\sigma &= -\lambda \iint \varphi F \, d\sigma \\ \Leftrightarrow \iint \nabla(F, \varphi) \, d\sigma &= \lambda \iint \varphi F \, d\sigma \end{aligned} \quad (17)$$

By approximating $F = \sum f_i \varphi_i$ (using a basis of form functions φ_i as described below) and setting the matrices

$$\begin{aligned} A &= (a_{lm}) := \left(\iint (\sum_{j,k} (\partial_j \varphi_l) (\partial_k \varphi_m) g^{jk}) d\sigma \right) \\ B &= (b_{lm}) := \left(\iint \varphi_l \varphi_m d\sigma \right) \end{aligned} \quad (18)$$

(with the surface element $d\sigma = \sqrt{\det(G)} \, du_1 \, du_2$), we obtain the following system of equations (one equation for each function φ_i):

$$A\mathbf{f} = \lambda B\mathbf{f} \quad (19)$$

involving two large, sparse, symmetric, positive (semi) definite matrices A and B . This approach can be employed for parametrized smooth surfaces, as well as for tetrahedra meshes or voxel data describing 3D solid objects. Here we focus on the special case of triangle meshes. Due to their piecewise flatness, the necessary integrals over the triangular elements can be linearly transformed to the unit triangle and can therefore be pre-computed. This speeds up the whole process immensely. Note, that the B matrix is not a diagonal matrix as usually encountered with discrete mesh Laplacians. In fact, the B matrix represents the **correct inner product** for functions of the same degree as the form functions on a specific geometry⁶.

We will give a detailed description and the local integrals so that a reimplementaion is easily possible. A linear parametrization P of a triangle T with vertices p_1, p_2, p_3 over the unit triangle can be given by:

$$P(\xi, \eta) = p_1 + \xi(p_2 - p_1) + \eta(p_3 - p_1) \quad (20)$$

$$P_1 = \frac{\partial P}{\partial \xi} = p_2 - p_1 \quad (21)$$

$$P_2 = \frac{\partial P}{\partial \eta} = p_3 - p_1 \quad (22)$$

$$G = (g_{ij}), \quad g_{ij} = P_i P_j \quad (23)$$

where we also show the partial derivatives and the corresponding first fundamental matrix G . Note P_1, P_2 and G are constant within each triangle.

The matrices A and B can be constructed by working on each triangle at a time. Within each triangle small matrices A' and B' (the so-called element matrices) are constructed whose entries are then added into the corresponding entries in the large matrices A and B after looking up the global index of each node. In the linear FEM case the nodes are identical with the 3 vertices, as 3 values are sufficient to specify a linear function over the triangle. For the cubic FEM approach, we need 10 values to specify a cubic function

⁶ The product of two functions $F = \sum_i f_i \varphi_i$ and $G = \sum_i g_i \varphi_i$ given by their values at the nodes is simply $\iint FG \, d\sigma = (f_1, \dots, f_n) B (g_1, \dots, g_n)^T$.

completely. These nodes are the 3 vertices, 2 additional nodes equally spaced along each edge and one node in the center of the triangle (see Fig. 20 in the Appendix for the location and the local index of each node). Thus the element matrices A' and B' will be 10×10 matrices.

The next step is to define local cubic basis functions φ_i having the value 1 at node i and value 0 at the other 9 nodes. For the entries of the matrix A' we need to compute the integral over the products of the gradients, and for the matrix B' simply the integral over the products of the basis functions themselves for each combination of nodes. So we construct the element matrices:

$$A' = (a'_{lm}) := \left(\iint (D\varphi_l)^T G^{-1} (D\varphi_m) d\sigma \right) \\ = \left(\iint \sum_{j,k} (\partial_j \varphi_l) g^{jk} (\partial_k \varphi_m) d\sigma \right) \quad (24)$$

$$B' = (b'_{lm}) := \left(\iint \varphi_l \varphi_m d\sigma \right).$$

With the partials P_1, P_2 (Eq. 20) and the pre-computed 10×10 matrices $I_1 \cdots I_4$ containing the integrals of the products of the local basis functions and their derivatives over the unit triangle (given in the Appendix, Table 1) we rewrite Eq. (24):

$$A' = \frac{1}{\|P_1 \times P_2\|} \left((P_1)^2 I_1 + (P_2)^2 I_2 - P_1 P_2 I_3 \right) \quad (25)$$

$$B' = \|P_1 \times P_2\| I_4 \quad (26)$$

To add the entries of A' and B' into A and B we only need to look-up the global index of each node. This algorithm can be nicely parallelized as different processes can work on different triangles at the same time. Nevertheless, this is not really necessary even for large meshes, as the construction of the matrices is very fast for flat triangles (very different from working on e.g. NURBS patches, where the evaluation of values and derivatives is slow and where a numerical integration method is needed due to the curved geometry).

Once A and B are constructed, the eigenvalue problem Eq. (10) can be solved. For this generalized symmetric eigenvalue problem $Ax = \lambda Bx$ with A positive semi-definite and B positive definite we can find non-negative eigenvalues λ_i and eigenvectors x_i such that (by setting the diagonal matrix of eigenvalues $\Lambda := \text{diag}(\lambda_i)$ and the matrix $X := (x_1, x_2, \dots, x_n)$) of eigenvectors we always have:

$$X^T A X = \Lambda \text{ and } X^T B X = I \quad (27)$$

where I is the identity (see for example [55]). Therefore one obtains a base of orthonormal eigenfunctions with respect to the inner product B . For higher dimensional eigenspaces the eigenfunctions can be chosen to be orthogonal. They can always be scaled so that $x_i^T B x_i = 1$.

In this work, only a small number of eigenvalues and eigenfunctions is needed. For the computation we follow [53, 23]. As we deal with a large, but sparse, problem an iterative Krylov methods (such as Lanczos) can be used very efficiently. We use the shift-invert method as implemented in ARPACK [56] to obtain the smallest eigenvalues and eigenfunctions. Because iterative methods work best on large eigenvalues, the problem is first inverted into $Bx = \frac{1}{\lambda} Ax$. Then for the iterative algorithm only the product $z = Bx$ needs to be computed and $Ay = z$ needs to be solved repeatedly, which is done by computing a sparse LU decomposition of A once. We use ARPACK for solving the shift-inverse problem and SuperLU [57] for the sparse factorization. The function calls are managed through the C++ wrapper Arpack++ [58]. Matlab also implements a similar method.

4.2 Construction of Morse-Smale Complexes

The construction of Morse-Smale complexes of piecewise linear functions over triangulated manifolds is described in [3]. The first step is to compute the critical points. In order to apply the following methods to the eigenfunctions as computed above, we globally refine our mesh (each triangle is split into 9 similar sub-triangles) and approximate the cubic functions by piecewise linear functions over this finer mesh.

For a piecewise linear real valued function h given by the values at the vertices (h_i) of a triangle mesh, the critical points have to lie at a vertex (if all the function values at the vertices are different). In case two or more values are exactly the same, we can use the vertex index to order the values. A maximum (minimum) is a vertex i whose function value h_i is larger (smaller) than the values at all of his neighbors. A saddle is a vertex i where the value of $h_j - h_i$ at the neighbors j switches sign at least 4 times. If it switches sign only twice, the vertex is called regular. In general, if $h_j - h_i$ switches sign x times we speak of a k -fold saddle where $k := x/2 - 1$ (if $k = 0$ we have a regular point). It is possible to unfold k -fold saddles into lower k saddles (see [3]), but this is rarely necessary. Such a situation never occurred in any of our experiments.

We construct the Morse-Smale (MS) graph simply by following the gradient field from each saddle up to the maxima (called UP-paths) and down to the minima of h (called DOWN-paths). The gradient inside a triangle with vertices p_1, p_2, p_3 and function values at the vertices f_1, f_2, f_3 can be computed as in Eq. (6) with $P_1 = p_2 - p_1$, $P_2 = p_3 - p_1$, $h_1 = f_2 - f_1$, and $h_2 = f_3 - f_1$. We follow the gradient inside each triangle or walk along an edge, if the gradient points outside



Fig. 7 MS complex of the 1st EF on the dancing lady. Left: full complex, right: simplified complex (only 2 min, 3 max and 3 saddles)

and the projection along that edge is positive. If we have several options (this can happen when starting at or hitting a vertex), we follow the steepest slope. The intersection points at the edges of the mesh are not inserted into the mesh, but only stored for the paths. Figure 7 (left) shows the MS complex of the first eigenfunction of a female in dancing posture.

4.3 Topological Noise Removal

To be able to remove topological noise, and later also to specify the level of the hierarchy for the mesh segmentation, we use the concept of persistence and cancellations as described above.

After the MS graph has been constructed, the persistence pairs can be found by the following algorithm:

1. Find the path (saddle s / extremum a pair) with the smallest persistence (absolute difference of function values).
2. If the other extremum b of the same type, connected to the involved saddle s , is not the same extremum ($a \neq b$), then the saddle must have the opposite sign. This is because no other saddle connected to a can lie in-between s and a therefore s either destroys the component created in the minimum a or the maximum a fills the hole created at s . Pair this saddle/extremum (s, a) and cancel it (as described above, by extending the paths reaching a to the other extremum b through the saddle s).
3. If both extrema are the same ($a = b$) this cannot be a persistence pair (the saddle either represents an “essential” homology and remains unpaired, or it has the same sign as the extremum). Remove the two paths from s to a and to b from the graph.
4. As long as we have paths, go to step 1.

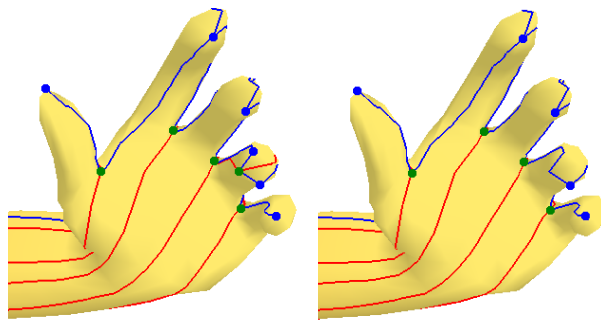


Fig. 8 Close up of a cancellation, left: topological noise at the ring finger, right: removed

Always selecting the shortest paths (smallest persistence) ensures, that no other saddles connected to the extremum can lie in-between. This makes it possible to classify the saddle and depending on the result, either pair and cancel the path, or remove the inappropriate edges.

Once the pairing is complete we can restore all original edges of the MS graph. It is now possible to hierarchically simplify the MS graph by canceling persistence pairs (starting with the smallest) until a user specified threshold T is reached. In fact, running the above algorithm until the smallest persistence is larger than T gives the same result. Such a simplification can be used to guide the hierarchical segmentation or to remove topological noise. As the low frequency eigenfunctions are very smooth, topological noise is only introduced through the triangulation locally and thus can be well distinguished from “real” extrema. The noisy saddle-extrema pairs have a very low persistence and can be filtered out automatically (usually by setting a threshold of 0.01 percent of the range of the function). Figure 8 shows a close-up of such a cancellation. A further simplified complex can be seen in Figure 7 (right), only showing the most significant topological features.

4.4 Hierarchical Subdivision Based on Level Sets

The idea behind the segmentation is to use the level sets at the saddles to partition the shape. The saddles of the first eigenfunction are positioned at important locations, identifying features (protrusions) of the shape. Cutting the corresponding level set will cut off the respective feature. Due to the hierarchical representation of the function by the persistence pairs of the MS graph, we are able to identify features on different persistence levels. Depending on how strongly we simplify the complex using cancellations we can guide the level of shape segmentation. This leads to a hierarchical representa-

tion of the shape. The following algorithm describes the computation of the parts:

At a pre-specified simplification level, for each extremum:

1. First find the closest adjacent saddle (using the difference in function values)
2. Compute the level set touching that saddle (without inserting it into the mesh) and obtain at least two loops.
3. Identify the loop that cuts out the submanifold containing the extremum.
4. Insert this loop into the mesh.
5. Starting at the extremum, colorize triangles until reaching the loop.

The computation of a level set (step 2) at a saddle is straight forward. We start at the saddle and find the four intersections at edges (or neighboring vertices). We follow one intersection and find the next, until we return to the saddle and close that loop. Now we start into another free direction (one of the remaining intersections adjacent to the saddle) and close another loop.

Step 3 can be achieved by selecting a non-critical point on each loop and following the gradient field up or down depending on the type of the current extremum. When we reach the current extremum we found the corresponding loop⁷. This loop can be inserted into the mesh (step 4) by splitting intersecting edges and inserting new triangles locally. Finally the whole submanifold can be found by starting a flooding algorithm at the extremum without crossing the loop. The color is still arbitrary until we match the parts. When these steps have been completed for all extrema, we end up with submanifolds homeomorphic to the disk (the protrusions) and with a base manifold. See Figure 9 for a coarse segmentation (including only most significant topological features) and a detailed segmentation (where even the fingers are partitioned).

4.5 Construction of Persistence Diagrams

For the segmentation we used only the first eigenfunction so far. In order to be able to register the parts, we need to employ higher eigenfunctions. These can be organized with the help of persistence diagrams. Not much needs to be said on the construction of these diagrams. Once the persistence pairs have been created we can understand each pair as a point in the set $D_h = \{(x, y) = (h(a), h(b))\}$ for each minimum/saddle

⁷ For higher genus surfaces, it is possible to reach the same maximum from several loops. Then they are all boundaries of the same segment.

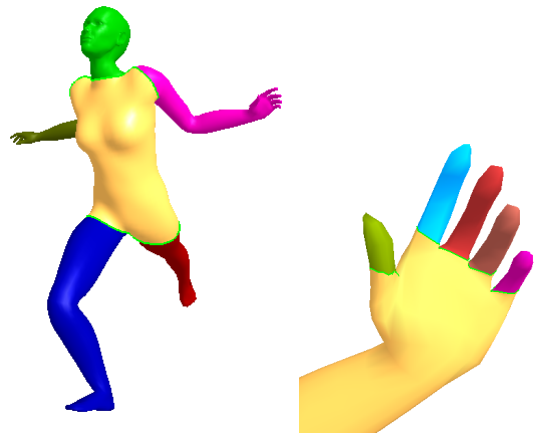


Fig. 9 Segmentation on different persistence levels. Left: using only the most significant critical points, right: closeup of the hand using all (except noise).

and maximum/saddle pair. Again, this choice is different from the persistence diagrams described in the literature [42, 59] as we mirror the maximum/saddle pair so that they lie below the diagonal. The reason for this action is that a sign change of the function h can simply be represented by mirroring the diagram about the origin. If the separated persistence diagram is given by the set D_h then D_h^{-1} is defined to be the set containing the mirrored points $(-x, -y)$. With this definition we have $D_{-h} = D_h^{-1}$.

4.6 Reordering and Flipping Eigenfunctions

As described in Section 3.2 the order and sign of eigenfunctions can be unstable. For the registration in the next section we need to correct for sign flips and switching. This can be done by computing the MS graphs and corresponding persistence diagrams for the eigenfunctions across shapes and then measuring the distance of the diagrams as follows.

Given two shapes \mathcal{A} and \mathcal{B} with eigenfunctions a_i and b_i (up to a specific index, usually $i < 6$) we compute D_{a_i} and D_{b_i} as well as $D_{b_i}^{-1}$ and compare each D_{a_i} with the diagrams of shape \mathcal{B} . By comparing to the mirrored diagrams $D_{b_i}^{-1}$ we ensure that sign flips will be found. Since a persistence diagram can be seen as a multiset, containing the persistence points (possibly with multiplicities) and the points on the diagonal, the difference of two diagrams D_1 and D_2 can be computed by the bottleneck distance d_b or by the Hausdorff distance d_h

$$d_b(D_1, D_2) = \inf_{\varphi} \sup_x \|x - \varphi(x)\|_{\infty}$$

$$d_h(D_1, D_2) = \max\left\{\sup_x \inf_y \|x - y\|_{\infty}, \sup_y \inf_x \|y - x\|_{\infty}\right\}$$

where $x \in D_1$ and φ varies among all bijections from D_1 to D_2 ⁸. The Hausdorff distance is easier to compute and has been used here.

We can select for each eigenfunction of \mathcal{A} the (possibly negated) eigenfunction in \mathcal{B} with the closest distance. It is usually sufficient to compare only with just a few neighboring functions as more distant switching is very unlikely. The described algorithm can detect sign flips in most cases. However if the mesh shows almost perfect intrinsic symmetries some sign flips cannot be detected intrinsically. In this case a negative eigenfunction might be equal to its reflected version and the corresponding persistence diagrams will be symmetric. Here the user needs to manually select the correct sign. Actually this problem is nicely exploited in [60] who use it to detect the intrinsic symmetries. As explained above, another problem might be the occurrence of higher dimensional eigenspaces that can be detected by almost identical eigenvalues. However, higher dimensional eigenspaces are extremely rare and usually occur in the first eigenfunctions only in highly symmetric shapes (sphere, square) or by stretching exactly by the right amount (recall the ellipsoid example). Possible switching and rotation in the spectral domain as observed in [20] occur only in cases of larger deformation (alien vs. human or a stretched shape). For near isometric shapes everything stays quite tame and one mainly has to deal with the sign flips. We noticed switching only in one case (lion) where global stretching of the torso and front legs is involved.

Note that Jain et al. [20] solve the switching and sign flip problem of their operator (Gaussian affinity matrix, a graph Laplacian) by constructing an optimal vertex correspondence of the two shapes in the spectral domain and by choosing the eigenfunction of the other shape that minimizes the sum of Euclidean distances of all vertex pairs. This is a computationally very intensive global optimization problem, but yields a full vertex correspondence in the end (if no sign flips due to intrinsic symmetries exist). Mateus et al. [21] propose a different approach to solve these difficulties. In their setup they compute a linear graph Laplacian on point cloud data of a volumetric shape representation where switching seems to be more prominent. To identify switching and some sign flips (they also do not discuss the undetectable symmetric sign flips) they propose to compare the full eigenfunction histograms. In our case the persistence diagrams (a set of points) are already available from the segmentation and can easily be compared.

⁸ For a point $x = (x_1, x_2) \in \mathbb{R}^2$ the maximum norm is: $\|x\|_\infty := \max\{x_1, x_2\}$.

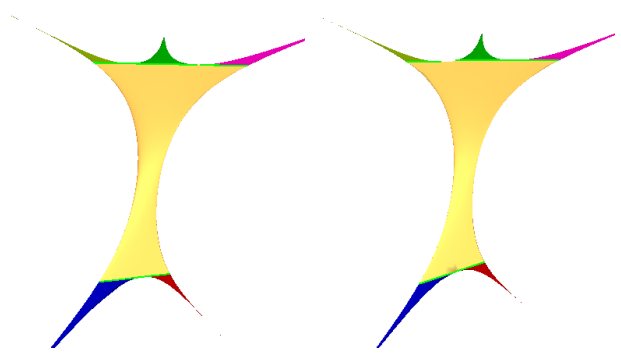


Fig. 10 Spectral projection onto the first 3 eigenfunctions of the lady in dancing (left) and walking position (right).

4.7 Registration by Spectral Projection

The actual registration of the components is done in the spectral domain. Each mesh point v_i is mapped into the spectral domain by the embedding

$$\Psi_n(v_i) = \left(\frac{f_1(v_i)}{\lambda_1}, \frac{f_2(v_i)}{\lambda_2}, \dots, \frac{f_n(v_i)}{\lambda_n} \right) \quad (28)$$

(where f_i are the eigenfunctions with eigenvalues λ_i). This ensures an alignment of the projected meshes independently of the original embedding in three space. In our examples it was sufficient to compute the Euclidean distance between extrema in the spectral domain (using the first 4-6 eigenfunctions) to find the correct correspondence on a low persistence resolution. If the resolution is higher, small features might not be accurately mapped by the first n eigenfunctions. In these cases the registration can first be done on a low resolution. Then the registered regions can be cut off and the whole analysis can be continued on the subparts. Figure 10 shows the spectral projections of the dancing and walking lady (only the first three dimensions, rotated and of course projected onto the paper). It can be nicely seen how the extrema stand out and are well aligned.

5 Examples

5.1 Segmentation

The segmentation of meshes can be done hierarchically by simplifying the Morse-Smale complex to a specific persistence level. The initial function does not necessarily need to be the first eigenfunction. However, usually the first eigenfunction is the best choice as it nicely represents the main trend of the object. Figure 11 shows the first and fourth eigenfunction of a bird mesh⁹ together with the level sets. It can be seen that in this case

⁹ Provided by the Aim@Shape Repository.

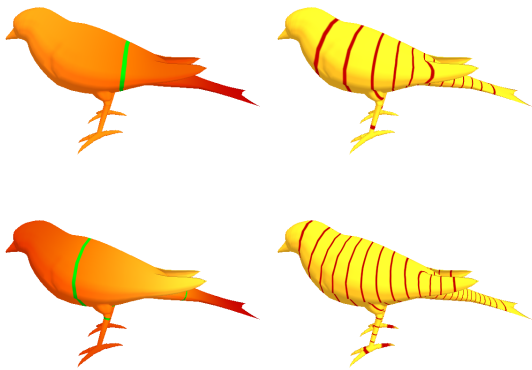


Fig. 11 Top row: eigenfunction 1 of the bird with level sets. Bottom row: eigenfunction 4 with level sets.

the fourth eigenfunction follows the same trend, but its zero level set has four components instead of only one. It is therefore also a candidate function to be used for the segmentation. In this case the obtained result with the fourth function seems to produce a slightly better segmentation as the function does not push the level set at the saddle into the protrusions as can be sometimes observed with the first eigenfunction.

To demonstrate the hierarchical segmentation we computed the MS complex on the bird mesh on different persistence levels and constructed the corresponding segmentations in Figure 12. Figure 13 show the MS complex and two segmentation levels of the cow model¹⁰. It can be seen that different parts of the shape are highlighted according to the selected level. Segmentations on different levels can be overlaid to segment a given shape. This approach will only segment the shape at the relevant features. Parts without any significant features will not be segmented (e.g. the torso). However it is possible to cut a shape into two nearly equally sized parts along the zero level set of the first eigenfunctions which is usually located right at the torso (e.g. the green line in Fig. 7 or around the belly of the cow Fig. 13). The feline model¹¹ is an example of a genus 2 surface. The saddle at the handles will not be canceled in ordinary persistence, therefore their level set components might border segments even on higher persistence levels as can be seen in 14 where one leg is cut off at the tail handle. There are principally two solutions: either ignore all unpaired saddles (making the segmentations invariant w.r.t. handles and topological noise) or by incorporating the pairs of essential homology classes into the persistence framework (extended persistence). Fig-

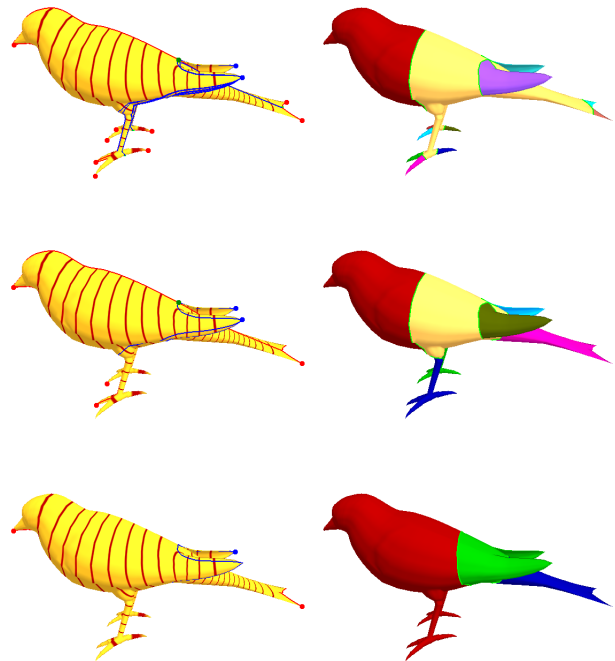


Fig. 12 Left: Morse-Smale complex for three different hierarchies. Right: corresponding segmentations.

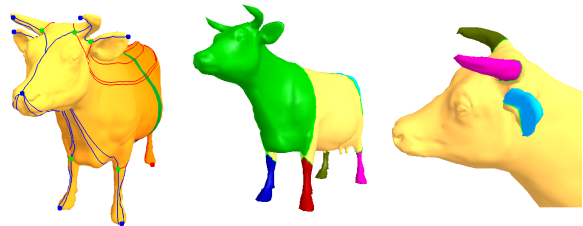


Fig. 13 Left: Morse-Smale complex of the cow, the saddles mark the features. Segmentation on different levels.



Fig. 14 The saddle at the tail handle of the feline terminates the leg segment even on higher persistence levels.

ure 15 shows more segmented shapes¹², where for the memento figure¹³, the camel and the centaur¹⁴ different hierarchy levels are presented.

¹⁰ Provided by Aim@Shape.

¹¹ Caltech Multi-Res Modeling Group

¹² Octopus provided by Aim@Shape

¹³ University Utrecht, Artihof Art Center (Bergschenhoek and Benschop) provided by the Aim@Shape Repository.

¹⁴ TOSCA Technion Israel

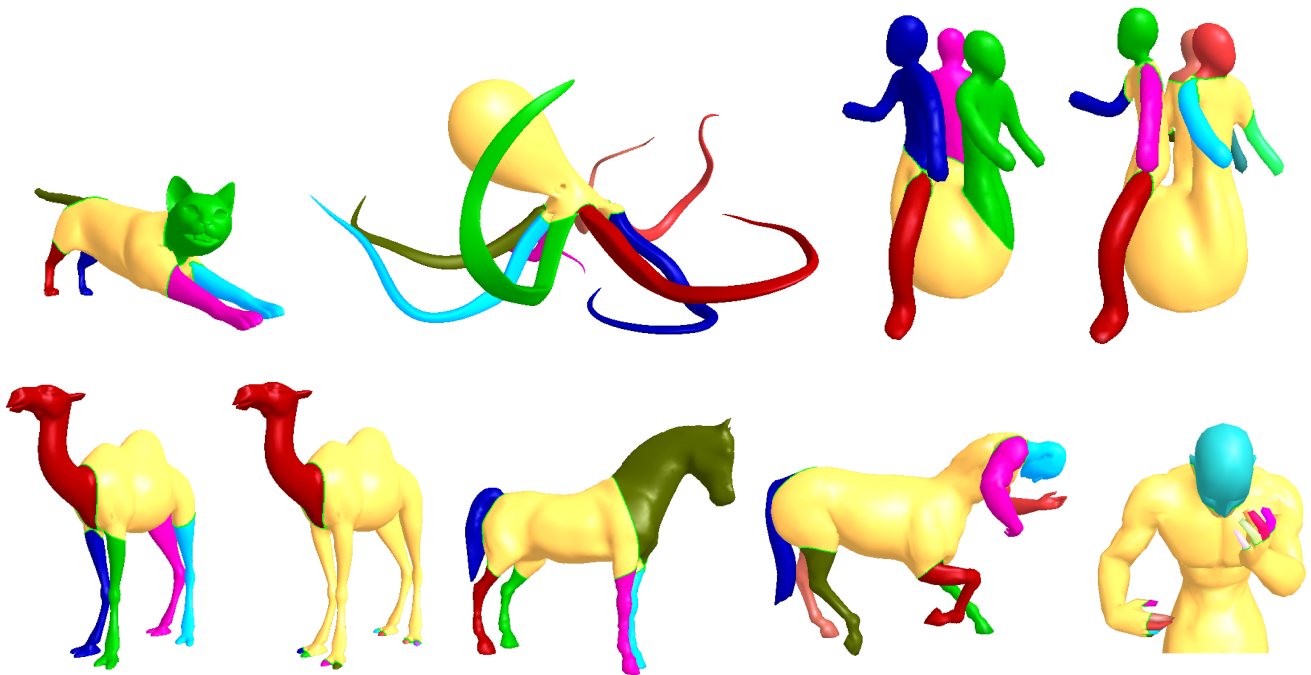


Fig. 15 Segmentation of different meshes. For the camel, centaur and memento figure we show segmentations on different hierarchies.

5.2 Registration

In this section we will present some examples where the segmented parts are registered across a population of near isometric shapes using the spectral projection as described above. The four ladies in different postures were depicted as an example in Figure 1 at the beginning of the paper¹⁵. Further examples are the centaur [61], the lion and the horse family [62]. It should be noted that due to the left/right symmetry of the meshes user interaction was necessary to correct the sign flip of specific eigenfunctions, whose zero level set corresponds with the symmetry. The negation of such a function will be identical to the mirrored function (left to right), and therefore the persistence diagrams are identical. In these cases it is not possible to automatically determine the sign. Actually no intrinsic method, neither eigenfunction histograms [21] nor correspondence in the spectral domain [20], can solve this. Furthermore, linear FEM and other mesh or graph Laplacians can be inaccurate [16] which might lead to switching of eigenfunctions if the mesh quality is low. When numerical difficulties are the reason the much higher accuracy of cubic FEM can help to remove some of the permutations. In this work, however, we noticed switching only in the case of the lion where larger global deformations at the torso and front legs are involved. Once the correct relation between the eigenfunctions was established the

spectral projection always registered the correct parts in all examples. It is also remarkable that the segments remain very stable across the shapes, even though the segmentations are performed independently on each individual shape.

5.3 Robustness

To demonstrate the robustness of the proposed method with respect to noise, mesh quality and density the algorithm was tested on different meshes of the gorilla¹⁶. Figure 17 shows the gorilla with different noise levels. Gaussian noise was added by off-setting each vertex into its normal direction with different values of standard deviation (here from left to right: 0, 0.5, 1). Even with a high noise level, the segmentation and registration stays accurate. Figure 18 depicts the gorilla in a different pose. Also here the registration pairs the correct parts (color). The three close-ups show the original (low quality), the smoothed (high quality) and a dense (highest quality) mesh. The segmentation and registration results are very robust and remain independent of the chosen mesh quality and connectivity.

¹⁵ Meshes from SHREC dataset 2007.

¹⁶ TOSCA Technion Israel

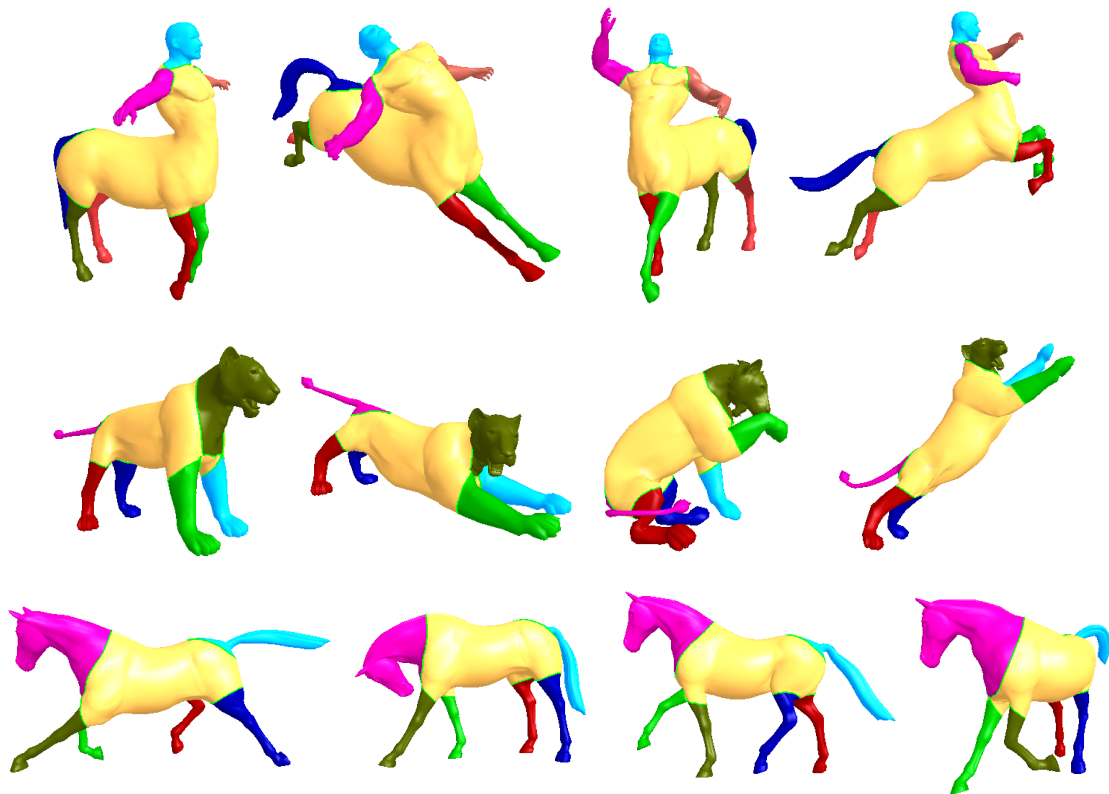


Fig. 16 Registration of segments across near-isometric shapes. Same color indicates registered parts across poses of the same model.

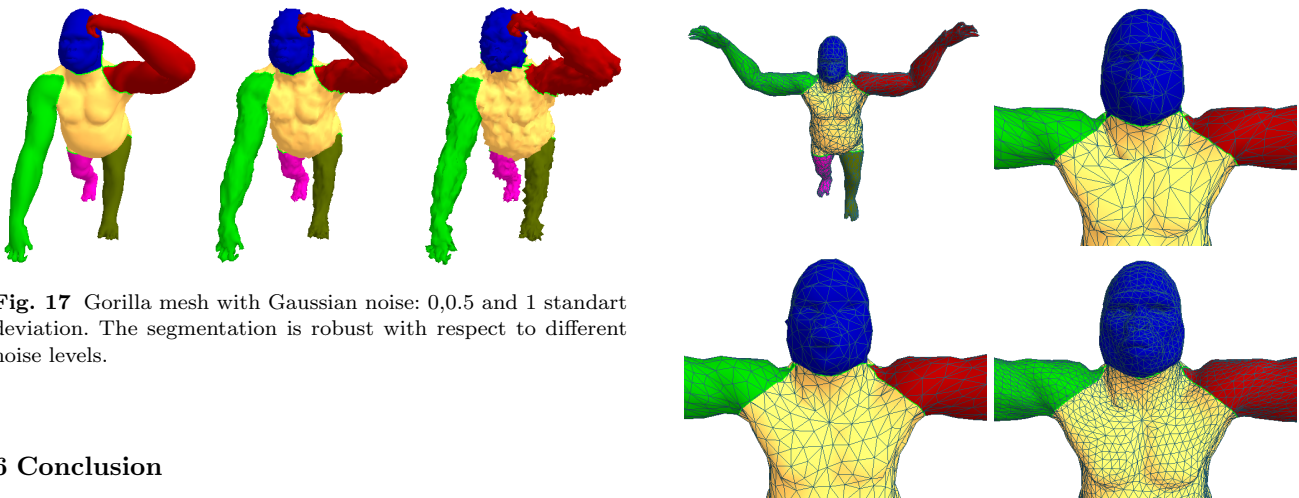


Fig. 17 Gorilla mesh with Gaussian noise: 0, 0.5 and 1 standart deviation. The segmentation is robust with respect to different noise levels.

6 Conclusion

This paper presents a method to efficiently construct consistent shape segmentations by employing eigenfunctions of the Laplace-Beltrami operator. Furthermore, with topological persistence it is possible to construct hierarchical segmentations that remain very stable across near-isometric populations of shapes and with respect to noise or mesh quality/density. We demonstrate that a spectral projection of the shapes can register the segments.

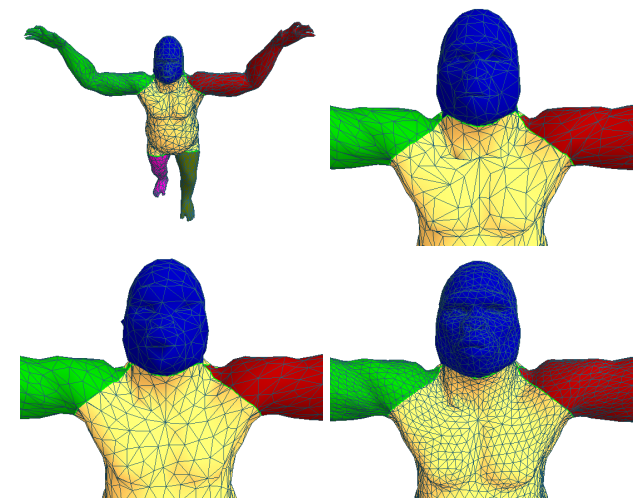


Fig. 18 Gorilla mesh with different qualities: low, higher (smoothed), highest (dense mesh). The results are stable with respect to mesh quality and density.

Moreover, this work describes the numerical computation of the Laplace-Beltrami spectrum via cubic FEM on triangular meshes. Detailed background is given on the difficulties that arise when comparing eigenfunctions (sign-flips, switching). A method employing sepa-

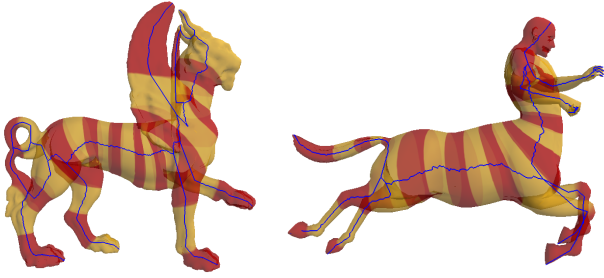


Fig. 19 Skeletal representation of the level sets (Reeb graph) for the feline and centaur model (first eigenfunction).

rated persistence diagrams to detect and solve these difficulties is proposed. Also an algorithm to construct separated persistence diagrams directly from the Morse-Smale complex is presented.

Further research will make use of the parts' correspondence to construct a bijection between the meshes useful for texture transfer, morphing and shape analysis. It is also possible to construct a skeleton, for example for animating shape, with the help of the Reeb graph that can easily be computed from the level sets (see Fig. 19). The Reeb graph can also be helpful to improve this approach for surfaces of higher genus by incorporating the pairs of essential homology classes to include handles (extended persistence). The presented method for shape segmentation can also be extended to 3D solids using the 3D Laplacian on tetrahedra or voxel representations as described in [23, 26].

Acknowledgements This work has been funded by a Feodor Lynen fellowship of the Alexander von Humboldt Foundation. The author would like to thank Prof. Nicholas M. Patrikalakis, MIT, for hosting this research and Prof. Herbert Edelsbrunner, Duke University, for his valuable comments.

A Appendix

Here we describe the element matrices for cubic finite elements on triangle meshes, necessary for the implementation. The corresponding software “shapeDNA-tria” can be found at <http://reuter.mit.edu>. For a given parametrization of a triangle (see Eq. 20) we have $W := \sqrt{\det(G)} = \|P_1 \times P_2\|$ (Lagrange’s identity) and $d\sigma = W d\xi d\eta$ where ξ and η are the parameter of the unit triangle. Therefore Equation (24) becomes:

$$\begin{aligned} (a'_{lm}) &= \iint (\sum_{j,k} (\partial_j \varphi_l) (\partial_k \varphi_m) g^{jk}) W d\xi d\eta \\ &= \iint [(P_2)^2 \partial_\xi \varphi_l \partial_\xi \varphi_m + (P_1)^2 \partial_\eta \varphi_l \partial_\eta \varphi_m \\ &\quad - P_1 P_2 (\partial_\xi \varphi_l \partial_\eta \varphi_m + \partial_\eta \varphi_l \partial_\xi \varphi_m)] \frac{1}{W} d\xi d\eta \\ &= \frac{1}{\|P_1 \times P_2\|} [(P_1)^2 \iint \partial_\eta \varphi_l \partial_\eta \varphi_m d\xi d\eta \\ &\quad + (P_2)^2 \iint \partial_\xi \varphi_l \partial_\xi \varphi_m d\xi d\eta \\ &\quad - P_1 P_2 \iint (\partial_\xi \varphi_l \partial_\eta \varphi_m + \partial_\eta \varphi_l \partial_\xi \varphi_m) d\xi d\eta] \end{aligned} \quad (29)$$

$$\begin{aligned} (b'_{lm}) &= \iint \varphi_l \varphi_m W d\xi d\eta = \|P_1 \times P_2\| \iint \varphi_l \varphi_m d\xi d\eta \\ &= \|P_1 \times P_2\| I_4 \end{aligned}$$

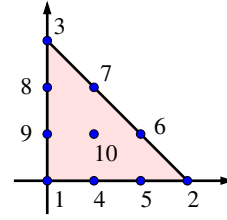


Fig. 20 Local numbering of the nodes for cubic FEM

The four symmetric matrices I_1 to I_4 contain integrals of products of form functions (φ_l) or their derivatives over the unit triangle. They can therefore be computed once for the entire process. The vectors $P_1 = p_2 - p_1$ and $P_2 = p_3 - p_1$ are constant for each triangle. The form function φ_l is the cubic function on the unit triangle with the values one at node l and zero at the 9 other nodes. The values at these 10 locations determines the cubic function in two variables completely, therefore $l, m \in \{1, \dots, 10\}$. To explicitly identify an entry in the 10×10 matrices I with an index combination (l, m) we need to specify the location of the nodes within the unit triangle. This is done in Figure 20.

Instead of stating the cubic form functions and their derivatives explicitly, we only present the integrals as needed for the computation. More details on constructing form functions can be found in [63]. The local matrices I_1 to I_4 of the integrals are shown in Table 1. Together with the geometry information of each triangle (as shown by Equation 29) they are used to construct the symmetric element matrices $A' = (a'_{lm})$ and $B' = (b'_{lm})$ per triangle. Their entries are then added into the global matrices A and B at location (i, j) where i and j are the global indices of the local node indices l and m . Hence, A and B are symmetric and sparse. Note that the linear FEM case leads to the mesh Laplace operator also known as cotangent weights with area adjustment [64] after lumping the mass matrix B (adding each off-diagonal entry to the diagonal).

References

- Hoffmann, D., Singh, M.: Saliency of visual parts. *Cognition* **63**, 2978 (1997)
- Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computations* **15**(6), 1373–1396 (2003)
- Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry* **30**(1), 87–107 (2003)
- Attene, M., Katz, S., Mortara, M., Patanè, G., Spagnuolo, M., Tal, A.: Mesh segmentation - a comparative study. In: *International Conference on Shape Modeling and Applications (SMI 06)*, pp. 14–25. Matsushima, Japan (2006)
- Shamir, A.: A survey on mesh segmentation techniques. *Computer Graphics Forum* **27**(6), 1539–1556 (2008)
- Dong, S., Bremer, P.T., Garland, M., Pascucci, V., Hart, J.C.: Spectral surface quadrangulation. *ACM SIGGRAPH 2006* pp. 1057–1066 (2006)
- Lai, Y.K., Hu, S.M., Martin, R.R., Rosin, P.L.: Fast mesh segmentation using random walks. In: *SPM 08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pp. 183–191. ACM, New York, NY, USA (2008). ISBN 978-1-60558-106-2
- Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graphics* **22**(3), 954–961 (2003)

$$I_1 = \frac{1}{80} \begin{pmatrix} 34 & 0 & -7 & 3 & 3 & -3 & -3 & 27 & -54 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7 & 0 & 34 & -3 & -3 & 3 & 3 & -54 & 27 & 0 \\ 3 & 0 & -3 & 135 & -27 & 27 & 27 & 0 & 0 & -162 \\ 3 & 0 & -3 & -27 & 135 & -135 & 27 & 0 & 0 & 0 \\ -3 & 0 & 3 & 27 & -135 & 135 & -27 & 0 & 0 & 0 \\ -3 & 0 & 3 & 27 & 27 & -27 & 135 & 0 & 0 & -162 \\ 27 & 0 & -54 & 0 & 0 & 0 & 0 & 135 & -108 & 0 \\ -54 & 0 & 27 & 0 & 0 & 0 & 0 & -108 & 135 & 0 \\ 0 & 0 & 0 & -162 & 0 & 0 & -162 & 0 & 0 & 324 \end{pmatrix}$$

$$I_2 = \frac{1}{80} \begin{pmatrix} 34 & -7 & 0 & -54 & 27 & -3 & -3 & 3 & 3 & 0 \\ -7 & 34 & 0 & 27 & -54 & 3 & 3 & -3 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -54 & 27 & 0 & 135 & -108 & 0 & 0 & 0 & 0 & 0 \\ 27 & -54 & 0 & -108 & 135 & 0 & 0 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 & 0 & 135 & -27 & 27 & 27 & -162 \\ -3 & 3 & 0 & 0 & 0 & -27 & 135 & -135 & 27 & 0 \\ 3 & -3 & 0 & 0 & 0 & 27 & -135 & 135 & -27 & 0 \\ 3 & -3 & 0 & 0 & 0 & 27 & 27 & -27 & 135 & -162 \\ 0 & 0 & 0 & 0 & 0 & -162 & 0 & 0 & -162 & 324 \end{pmatrix}$$

$$I_3 = \frac{1}{80} \begin{pmatrix} 68 & -7 & -7 & -51 & 30 & -6 & -6 & 30 & -51 & 0 \\ -7 & 0 & 7 & 24 & -57 & 57 & -24 & 0 & 0 & 0 \\ -7 & 7 & 0 & 0 & 0 & -24 & 57 & -57 & 24 & 0 \\ -51 & 24 & 0 & 135 & -108 & 27 & 27 & -27 & 135 & -162 \\ 30 & -57 & 0 & -108 & 135 & -135 & 27 & -27 & -27 & 162 \\ -6 & 57 & -24 & 27 & -135 & 135 & 54 & 27 & 27 & -162 \\ -6 & -24 & 57 & 27 & 27 & 54 & 135 & -135 & 27 & -162 \\ 30 & 0 & -57 & -27 & -27 & 27 & -135 & 135 & -108 & 162 \\ -51 & 0 & 24 & 135 & -27 & 27 & -108 & 135 & -162 & 0 \\ 0 & 0 & 0 & -162 & 162 & -162 & 162 & -162 & 162 & 324 \end{pmatrix}$$

$$I_4 = \frac{1}{13440} \begin{pmatrix} 76 & 11 & 11 & 18 & 0 & 27 & 27 & 0 & 18 & 36 \\ 11 & 76 & 11 & 0 & 18 & 18 & 0 & 27 & 27 & 36 \\ 11 & 11 & 76 & 27 & 27 & 0 & 18 & 18 & 0 & 36 \\ 18 & 0 & 27 & 540 & -189 & -135 & -54 & -135 & 270 & 162 \\ 0 & 18 & 27 & -189 & 540 & 270 & -135 & -54 & -135 & 162 \\ 27 & 18 & 0 & -135 & 270 & 540 & -189 & -135 & -54 & 162 \\ 27 & 0 & 18 & -54 & -135 & -189 & 540 & 270 & -135 & 162 \\ 0 & 27 & 18 & -135 & -54 & -135 & 270 & 540 & -189 & 162 \\ 18 & 27 & 0 & 270 & -135 & -54 & -135 & -189 & 540 & 162 \\ 36 & 36 & 36 & 162 & 162 & 162 & 162 & 162 & 162 & 1944 \end{pmatrix}$$

Table 1 The four matrices containing integrals on the unit triangle for cubic FEM

9. Attene, M., Falcidieno, B., Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer* **22**(3), 181–193 (2006)
10. Li, X., Gu, X., Qin, H.: Surface matching using consistent pants decomposition. In: SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling, pp. 125–136. ACM, New York, NY, USA (2008). ISBN 978-1-60558-106-2
11. Elad, A., Kimmel, R.: On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(10), 1285–1295 (2003)
12. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. *The Visual Computer* **21**(8), 649–658 (2005)
13. Liu, R., Zhang, H.: Segmentation of 3D meshes through spectral clustering. In: Proc. of Computer Graphics and Applications, pp. 298–305 (2004). ISBN 0-7695-2234-3
14. Liu, R., Zhang, H.: Mesh segmentation via spectral embedding and contour analysis. *EUROGRAPHICS 2007* **26**, 385–394 (2007)
15. Rustamov, R.M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In: SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing, pp. 225–233. Eurographics Association, Aire-la-Ville, Switzerland (2007)
16. Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., Spagnuolo, M.: Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics* doi:10.1016/j.cag.2009.03.005 (2009)
17. Coifman, R.R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F.J., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data. part i: Diffusion maps. In: Proc. of Nat. Acad. Sci., vol. 102, pp. 7426–7431 (2005)
18. de Goes, F., Goldenstein, S., Velho, L.: A hierarchical segmentation of articulated bodies. In: Eurographics Symposium on Geometry Processing, vol. 27,5, pp. 1349–1356 (2008)
19. Jain, V., Zhang, H.: Robust 3D shape correspondence in the spectral domain. In: Proc. Shape Modeling International 2006, pp. 118–129 (2006)
20. Jain, V., Zhang, H., van Kaick, O.: Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling* **13**(1), 101–124 (2007)
21. Mateus, D., Horaud, R.P., Knossow, D., Cuzzolin, F., Boyer, E.: Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
22. Reuter, M., Wolter, F.E., Shenton, M., Niethammer, M.: Laplace-beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Computer-Aided Design* doi:10.1016/j.cad.2009.02.007 (2009)
23. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-Beltrami spectra as shape-DNA of surfaces and solids. *Computer-Aided Design* **38**(4), 342–366 (2006)
24. Peinecke, N., Wolter, F.E., Reuter, M.: Laplace spectra as fingerprints for image recognition. *Computer-Aided Design* **39**(6), 460–476 (2007)
25. Niethammer, M., Reuter, M., Wolter, F.E., Bouix, S., Peinecke, N., Koo, M.S., Shenton, M.: Global medical shape analysis using the Laplace-Beltrami spectrum. In: Conference on Medical Image Computing and Computer Assisted Intervention, Part I, LNCS 4791, Lecture Notes in Computer Science, Vol. 4791, pp. 850–857. Springer, Brisbane, Australia (2007). ISBN 978-3-540-75756-6
26. Reuter, M., Niethammer, M., Wolter, F.E., Bouix, S., Shenton, M.: Global medical shape analysis using the volumetric Laplace spectrum. In: Proceedings of the 2007 International Conference on Cyberworlds (NASAGEM), pp. 417–426. IEEE Computer Society (2007)
27. Shi, Y., Lai, R., Krishna, S., Dinov, I., Toga, A.W.: Anisotropic Laplace-Beltrami eigenmaps: Bridging Reeb graphs and skeletons. In: Computer Vision and Pattern Recognition Workshops, 2008. CVPR Workshops, pp. 1–7. IEEE Computer Society, Anchorage, AK (2008)
28. Lévy, B.: Laplace-Beltrami eigenfunctions: towards an algorithm that understands geometry. In: Proc. of the IEEE Shape Modeling and Applications, p. 13 (2006)
29. Qiu, A., Bitouk, D., Miller, M.I.: Smooth functional and structural maps on the neocortex via orthonormal bases of the Laplace-Beltrami operator. *IEEE Transactions on Medical Imaging* **25**(10), 1296–1306 (2006)
30. Vallet, B., Lévy, B.: Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proc. Eurographics 2007)* **2**(27) (2008)
31. Zhang, H., van Kaick, O., Dyer, R.: Spectral mesh processing. *Computer Graphics Forum* (2009)
32. Xu, G.: Discrete Laplace-Beltrami operators and their convergence. *Computer Aided Geometric Design* **21**, 767–784 (2004)
33. Gromov, M.: Metric Structures for Riemannian and Non-Riemannian Spaces, *Progress in Mathematics*, vol. 152. Birkhäuser Boston (1999)
34. Mémoi, F., Sapiro, G.: A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* **5**(3), 313–347 (2005)
35. Chavel, I.: Eigenvalues in Riemannian Geometry. Academic Press (1984)

36. Reuter, M.: Laplace Spectra for Shape Recognition. Books on Demand, Norderstedt, Germany (2006). ISBN 3-8334-5071-1
37. Courant, R., Hilbert, D.: Methods of Mathematical Physics, Vol I. Interscience (1953)
38. Milnor, J.: Morse Theory, *Annals of mathematics studies*, vol. 51. Princeton University Press (1963)
39. Smale, S.: On gradient dynamical systems. *Ann. of Math.* **74**, 199–206 (1961)
40. Biasotti, S., De Floriani, L., Falcidieno, B., Frosini, P., Giorgi, D., Landi, C., Papaleo, L., Spagnuolo, M.: Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.* **40**(4), 1–87 (2008). ISSN 0360-0300
41. Bremer, P.T., Edelsbrunner, H., Hamann, B., Pascucci, V.: A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics* **10**(4), 385–396 (2004)
42. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. *Discrete and Computational Geometry* **28**(4), 511–533 (2002)
43. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. In: SCG '05: Proceedings of the twenty-first annual symposium on Computational geometry, pp. 263–271. ACM, New York, NY, USA (2005). ISBN 1-58113-991-8
44. Frosini, P., Pittore, M.: New methods for reducing size graphs. *International Journal of Computer Mathematics* **70**(3), 505–517 (1999)
45. Frosini, P.: A distance for similarity classes of submanifolds of a euclidean space. *Bulletin of the Australian Mathematical Society* **42**, 407–416 (1990)
46. Frosini, P., Landi, C.: Size functions and morphological transformations. *Acta Applicandae Mathematicae* **49**(1), 85–104 (1997)
47. Frosini, P.: Connections between size functions and critical points. *Mathematical Methods in the Applied Sciences* **19**, 555–569 (1996)
48. Reeb, G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes rendus de l'Académie des sciences, Paris* **222**, 847–849 (1946)
49. Agarwal, P.K., Edelsbrunner, H., Harer, J., Wang, Y.: Extreme elevation on a 2-manifold. *Discrete Computational Geometry* **36**(4), 553–572 (2006)
50. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics* (2008)
51. Strang, G., Fix, G.J.: An Analysis of the Finite Element Method. Prentice-Hall (1973)
52. Dziuk, G.: Finite elements for the Beltrami operator on arbitrary surfaces. In: S. Hildebrandt, R. Leis (eds.) *Partial differential equations and calculus of variations. Lecture Notes in Mathematics* 1357, pp. 142–155. Springer Berlin (1988)
53. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: SPM '05: Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling, pp. 101–106. ACM Press (2005)
54. Vallet, B.: Function bases on manifolds. Ph.D. thesis, Institut National Polytechnique de Lorraine (2008)
55. Parlett, B.N.: The Symmetric Eigenvalue Problem. Prentice-Hall, Englewood Cliffs (1980)
56. ARPACK: Arnoldi package. <http://www.caam.rice.edu/software/ARPACK/>
57. Demmel, J.W., Eisenstat, S.C., Gilbert, J.R., Li, X.S., Liu, J.W.H.: A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications* **20**(3), 720–755 (1999)
58. Gomes, F.M., Sorensen, D.C.: Arpack++: A c++ implementation of arpack eigenvalue package. (1997). URL cite-seer.ist.psu.edu/67128.html
59. Edelsbrunner, H., Harer, J.: Persistent homology - a survey. In: J.E. Goodman, J. Pach, R. Pollack (eds.) *Surveys on Discrete and Computational Geometry: Twenty Years Later, Contemporary Mathematics*, vol. 453, pp. 257–281. AMS Bookstore (2008)
60. Ovsjanikov, M., Sun, J., Guibas, L.: Global intrinsic symmetries of shapes. In: *Eurographics Symposium on Geometry Processing (SGP)* (2008)
61. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing* **28**(5), 1812–1836 (2006)
62. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. In: SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, pp. 399–405. ACM, New York, NY, USA (2004)
63. Zienkiewicz, O., Taylor, R.: *The Finite Element Method - Volume 1: The Basis*. Butterworth Heinemann (2000)
64. Desbrun, M., Meyer, M., Schröder, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: *Proc. ACM SIGGRAPH '99*, pp. 317–324 (1999)