EUROGRAPHICS 2011 / M. Chen and O. Deussen
(Guest Editors)

Volume 30 (2011), Number 2

Discovery of Intrinsic Primitives on Triangle Meshes

Justin Solomon, Mirela Ben-Chen, Adrian Butscher and Leonidas Guibas

Stanford University

Figure 1: *Localized vector fields which are an optimized linear combination of approximate Killing fields (left), a segmentation from these fields (middle), and a decomposition into intrinsic primitives with prominent intrinsic symmetry generators (right).*

Abstract

The discovery of meaningful parts of a shape is required for many geometry processing applications, such as parameterization, shape correspondence, and animation. It is natural to consider primitives such as spheres, cylinders and cones as the building blocks of shapes, and thus to discover parts by fitting such primitives to a given surface. This approach, however, will break down if primitive parts have undergone almost-isometric deformations, as is the case, for example, for articulated human models. We suggest that parts can be discovered instead by finding intrinsic *primitives, which we define as parts that posses an approximate intrinsic symmetry. We employ the recently-developed method of computing* discrete approximate Killing vector fields *(AKVFs) to discover intrinsic primitives by investigating the relationship between the AKVFs of a composite object and the AKVFs of its parts. We show how to leverage this relationship with a standard clustering method to extract k intrinsic primitives and remaining asymmetric parts of a shape for a given k. We demonstrate the value of this approach for identifying the prominent symmetry generators of the parts of a given shape. Additionally, we show how our method can be modified slightly to segment an entire surface without marking asymmetric connecting regions and compare this approach to state-of-the-art methods using the Princeton Segmentation Benchmark.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

Searching for meaningful parts of a given shape is an important step in the geometry processing pipeline, providing an initial division for parameterization, matching, correspondence, editing and other basic tasks. One way to approach this problem is to characterize the properties of a "good" part, keeping the target application in mind. For example, if the end goal is shape understanding, parts should come from meaningful categories recognized by humans. If, on the other hand, the target application is parameterization, then it would be more useful if the parts were parameterizable over simple domains like spheres and disks. Finally, a shape editing application might prefer parts that are described by a small set of parameters that the user can manipulate, such as the axis and radius of a surface of revolution.

In this paper, we define a "good" part as one exhibiting an approximate intrinsic symmetry; we name these parts *intrinsic primitives*. Such parts are near-isometric deformations of extrinsically symmetric shapes like spheres, planes and surfaces of revolution, whose symmetries can be expressed us-

ing continuous parameters. We do not seek reflectional symmetries or other discrete structures. They are useful for applications that seek parts with structure that can be described succinctly, and they often correlate well with parts perceived by humans. While extrinsically symmetric primitives represent a fairly limited set of surfaces, allowing for intrinsic and approximate symmetry considerably widens the set of observable self-similarities, capturing shapes whose structure is clear perceptually but not visible in the Euclidean setting. For instance, the tentacles of an octopus model have cylindrical structure and the head of a human model might be approximately spherical; these structures may not be visible if they are forced to conform to exact cylinders or spheres when checking for structure without allowing for bending and localized deviations from symmetry.

Fitting primitives for shape partitioning is a well-known approach in image processing [AFS06] and geometry processing [GG04]; and our work is a natural extension to the intrinsic setting. To find intrinsic primitives, we employ *discrete approximate Killing vector fields* (AKVFs) [BCBSG10]. Killing fields are tangent vector fields that generate continuous symmetries [DC93]. But relatively few surfaces admit Killing fields since exact intrinsic symmetries are rare on arbitrary surfaces. AKVFs generalize Killing fields — these are critical points of a *Killing energy* that measures the amount of asymmetry of a given field. [BCBSG10] shows how these vector fields can be computed on a triangle mesh by solving an eigenvalue problem. The inset shows examples of AKVFs.

This paper extends [BCBSG10] by investigating the AKVFs of a *composite object* that is constructed by gluing together simple parts. We define a *localized* vector field to be one that has non-negligible norm on only one part and a *tangled* vector field to be a linear combination of localized vector fields. We show that the AKVFs of a composite surface span a subspace similar to the span of the AKVFs of its parts. Thus, we first find the AKVFs of the composite, and then we *untangle* them by finding the best linear combinations that localize onto the parts; See Figure 2 for an example of AKVFs before and after untangling. Our model of untangling is closely linked to a well-understood problem in factor analysis, and for this reason it can be accomplished efficiently using a preexisting statistical algorithm. Finally, we apply a standard clustering method to extract the intrinsic primitives and remaining asymmetric parts. The entire approach is based on developing a global understanding about the structure of the surface rather than using only localized notions of shape; thus, it is more likely to be robust to both noise and localized deformations.

One application of our method is shape segmentation, in which asymmetric connecting regions are divided among parts based on their approximate symmetries to generate a division of the entire surface. Our method compares acceptably with state-of-the-art segmentation methods in some of the object categories of the Princeton Segmentation Database [CGF09], a result that may have implications concerning human understanding of shape. More importantly, when allowing for the existence of asymmetric connecting regions, we show how to find the intrinsic primitive parts of a shape and the most prominent symmetry generators on each part, which may be useful for shape classification and other applications.

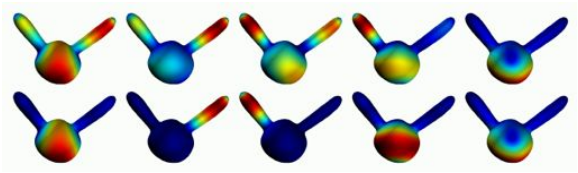


Figure 2: *AKVF norms on a simple model: (top) original and (bottom) after untangling. Note how the untangled AKVFs are localized on individual parts.*

1.1. Previous Work

Recent research in geometry processing has focused on finding and making use of intrinsic symmetries, first considered in [RBBK07]; papers such as [RBBK10] also considered *partial* symmetries on surfaces with some asymmetric parts. [MBB10] presents an approach to finding partial discrete intrinsic symmetries, identifying parts with grid-like structure without considering the continuous case.

More generally, there is a considerable body of research related to mesh segmentation, and a full survey is beyond the scope of this paper. We concentrate on the latest approaches and those methods that are most related to ours. We refer the reader to [Sha08, APP*07] for a more thorough review.

Our method is based on clustering faces of the mesh with similar local intrinsic symmetries. A closely-related approach is segmentation using slippage analysis [GG04]. Slippage motions are rigid motions of a surface that displace it in tangential directions. Surfaces that support such motions are *extrinsically* symmetric, e.g. planes, spheres and cylinders. Since this method is based on extrinsic rigid motions, it is unable to detect *intrinsic* symmetries. For example, a bent planar patch will have a single extrinsic translational symmetry, but in reality has three intrinsic symmetries (two translations and one rotation). In this way, our method can be considered an extension of [GG04] to "intrinsic slippable motions." Our method further differs from theirs in that we can use a global top-down approach to locate clusters because of the properties of AKVFs of composite objects.

An approach that explicitly partitions input shapes into sets of primitives is [AFS06]. This paper uses a bottom-up approach, explicitly fitting planes, spheres and cylinders to each segment. As in [GG04], the fitting method uses the extrinsic geometry of the surface and thus is not invariant to

near-isometric deformations; it cannot be applied to finding intrinsic primitives. A similar recent approach presented in [SSCO08] uses values of the Shape Diameter Function (SDF) as features of mesh faces and performs the clustering by segmenting the histogram of the SDF values. This approach is more general than fitting primitives, and it is able to identify bent cylinders since different points on a bent cylinder will have similar SDF values. Our method, however, is more general, identifying any near-isometric deformation of a cylinder, rather than only those that preserve radius.

Two other related methods that use local symmetries to find the clusters are [PSG*06] and [LCDF10]. The former builds feature vectors based on extrinsic symmetry planes and the latter based on distances in a space where symmetric points are mapped to the same location. Since the first method is based on extrinsic reflection symmetries, it cannot detect parts which are approximately intrinsically symmetric. The second method searches for a global symmetry between different parts of a shape, whereas we focus on parts which are themselves symmetric.

The most recent work on clustering face-based features was proposed in [KHS10], which describes a supervised learning approach. Instead of using a single criterion for segmentation, the algorithm infers human-like segmentations and labeling from training data. Although the method works remarkably well in practice, generating the training data is still a non-negligible task, and the parts it generates are motivated by semantic rather than geometric applications. Finally, two recent intrinsic segmentation methods make use of global spectral data, although the operators they consider are not related to intrinsic symmetries. In particular, [SOCG10] uses the heat kernel signature and [Reu10] uses the eigenfunctions of the Laplace-Beltrami operator to divide meshes. Those, however, exhibit a very different behavior than the eigenfunctions we are considering. For example, the Fielder vector (the first non-trivial eigenvector of the Laplace-Beltrami operator) tends to be aligned with the principal direction of an object, whereas the direction of the first eigenvector of our operator is governed by the intrinsic symmetries of the shape, as can be observed in Figure 5(c).

1.2. Contribution

Compared to existing methods, our contribution is two-fold. First, we are able to discover intrinsic primitives of a shape, extending similar work on extrinsic primitives [GG04]. Second, in contrast to previous approaches, we are able to discover these primitives using a more efficient top-down approach that analyzes the approximate intrinsic symmetries of the *whole* object. This is made possible due to our understanding of the behavior of AKVFs of composite objects.

The inputs to our method are a triangle mesh, the required number of parts k, and the number of AKVFs N. We compute the first N AKVFs of the mesh as in [BCBSG10] and apply an optimization procedure to find a set of N localized vector fields. Using these fields, we assign a feature vector of N values to each face. Finally, we use k-means clustering on these features to retrieve a set of k cluster centers. By extracting isolines of functions based on the distances to these cluster centers, we output a partitioning of the shape into approximately symmetric parts and an asymmetric remainder.

The rest of the paper is organized as follows. In the next section, we remind the reader how AKVFs are found according to [BCBSG10] and investigate the behavior of AKVFs on a composite object. In Sections 3 and 4 we describe the main building blocks of our algorithm: untangling the AKVFs and clustering them. In Section 5 we describe the full algorithm and show some experimental results and possible applications. We conclude with a discussion and future research directions in Section 6.

2. AKVFs of Composite Surfaces

2.1. Discrete AKVFs

Any tangent vector field on a surface generates a deformation of the surface: simply let all points "flow" along the vector field for a fixed amount of time t. Killing vector fields have the additional property that the deformations they generate are *isometric* to the initial surface, or, equivalently, that if all points on the surface flow for a fixed amount of time, pairwise geodesic distances on the surface will be preserved.

Only a limited set of surfaces admit Killing vector fields, including developable surfaces, spheres and surfaces of revolution. Using a variational formulation, however, one can generalize the notion of Killing field to define *approximate* Killing fields. To this end, we formulate an energy functional called the *Killing energy* based on the L^2 norm of the deviation from the over-determined condition satisfied by exact Killing fields. More precisely, a vector field X on a surface Σ is Killing if and only if its covariant derivative tensor is anti-symmetric. The covariant derivative is a generalization to surfaces of the Jacobian matrix of a planar vector field, hence it can be thought of as a 2×2 matrix $M_q(X)$, given at each point q on the surface. Hence, the vector field X is Killing if and only if $\forall q \in \Sigma$, $P_q(X) = 0$, where $P_q(X) = M_q(X) + M_q(X)^T$. See [BCBSG10] for more details on Killing vector fields, and the derivation of P. The Killing energy of X is $\mathcal{K}_\Sigma(X) := \int_\Sigma \|P(X)\|^2 / \int_\Sigma \|X\|^2$, and AKVFs are the critical points of $\mathcal{K}_\Sigma$. Alternatively, they are solutions of the eigenvalue problem $P^*P(X) = \lambda X$, where P^* is the L^2-adjoint of P, and the lowest eigenvalue of P^*P is the Killing energy of Σ.

In the discrete setting, we are given a triangle mesh $M = (V, F, E)$, where V, F and E are the sets of vertices, faces and edges, respectively. To calculate AKVFs we need a method to discretize vector fields and an accompanying discretization of the operator P^*P. A common approach to the discretization of differential operators on meshes is the use of Discrete Exterior Calculus (DEC) [DKT06], which repre-

sents vector fields using discrete *one-forms* given by a vectors of size $|E|$. The operator P^*P can be discretized using a matrix $R \in \mathbb{R}^{|E|\times|E|}$. For completeness we give the exact expressions for computing R in the additional material. The eigenvectors of R are discrete one-forms which can be interpolated using Whitney forms (see [DKT06]) to generate tangential vector fields on the faces of the mesh.

If a surface has an AKVF that is close to an exact Killing field, the smallest eigenvalue of R will be close to zero, and its matching eigenvector will generate the associated approximate symmetry. If we have a more complex shape, such as the one in Figure 2, it is not clear *a priori* what the eigenvectors associated to the smallest eigenvalues of R represent. We show that if the shape is a *composite* consisting of several parts possessing approximate intrinsic symmetries and the transition regions between the parts are small enough, then the AKVFs behave more or less as if these parts were independent of each other.

This independence is a manifestation of a general property of elliptic equations admitting a variational formulation, such as those defining Killing and Laplace-Beltrami operator eigenvalue problems. To analyze this phenomenon we first consider an idealized setting, where the transition regions between the parts are infinitesimal; we will discuss real-world scenarios later on. We also consider only a surface that decomposes into two component parts, since the results below generalize in a straightforward manner to multiple parts connected in more complex ways.

2.2. Composite Surfaces with ε-Transition Regions

Let Σ be a compact surface embedded in $\mathbb{R}^3$ and suppose that we can decompose it into two components Ω_1 and Ω_2, whose interiors are disjoint, as well as a "small" transition surface $\mathcal{N}_\varepsilon$, such that $\Sigma = \Omega_1 \cup \mathcal{N}_\varepsilon \cup \Omega_2$, and $\partial\mathcal{N}_\varepsilon = \partial\Omega_1 \cup \partial\Omega_2$. Suppose further that each component Ω_i equals a "large" surface Σ_i from which a small ball of radius $0 < \varepsilon \ll 1$, centered at a point on Σ_i, has been removed; Finally, suppose that the transition region $\mathcal{N}_\varepsilon$ can be contained in a ball of radius $\mathcal{O}(\varepsilon)$ and has surface area $\mathcal{O}(\varepsilon^2)$. An example for such a surface is the dumbbell (see Figure 4(b)), where: Σ_1 and Σ_2 are two spheres, Ω_1 and Ω_2 are two spheres with small holes, and $\mathcal{N}_\varepsilon$ is the "neck" region.

We consider the eigenvalue problem for a linear, elliptic partial differential system of equations for a one-form ω of the form $P^*P(\omega) = \lambda\omega$ where P is a first-order partial differential operator. We would like to compare the spectral data of P^*P on Σ with the spectral data of P^*P on Σ_1 and Σ_2. Let $\lambda_1 \le \lambda_2 \le \cdots$ be the eigenvalues of P^*P on Σ, counted with multiplicity; and let $\omega_1, \omega_2, \ldots$ be the corresponding orthonormalized eigenvectors. Next, merge all the eigenvalues of P^*P restricted to Σ_1 and Σ_2 into one set $\{\mu_n : n \in \mathbb{N}\}$ sorted in increasing order (again, accounting for multiplicities), and denote by u_n the corresponding eigenvectors. Our spectral comparison will estimate the discrepancy between the eigenvalues λ_n and μ_n for each n up to a threshold. To additionally compare eigenvectors, we further introduce the *proxy eigenvectors* denoted $\tilde{u}_1, \tilde{u}_2, \ldots$, where the $\tilde{u}_n$ equals the appropriate AKVF eigenvector from one of the Σ_i multiplied by a smooth cut-off function $\chi^{(i)}$ that vanishes on the neck region and the other Σ_i.

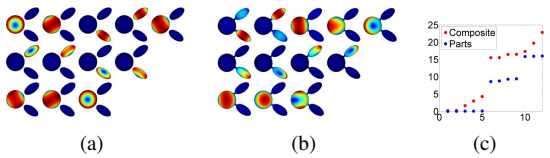
Proposition 1. *There exist constants $\varepsilon_0, C > 0$ depending only on the eigenvalues of Σ_1, Σ_2 and a number $M(\varepsilon)$ with $\lim_{\varepsilon\to 0} M(\varepsilon) = \infty$ so that the spectral data of P^*P satisfies:*

1. *If $\varepsilon < \varepsilon_0$ then for all n s.t. $\lambda_n < M(\varepsilon)$ we have*
$$|\lambda_n - \mu_n| \le C/|\log(\varepsilon)|.$$
2. *Let $P_{n,\delta}$ be the L^2-orthogonal projector onto the subspace $\mathcal{W}_{n,\delta} := span\{\tilde{u}_k : k \text{ s.t. } |\lambda_n - \mu_k| \le \delta\}$. If $\varepsilon < \varepsilon_0$ then for all $\delta > 0$ and n s.t. $\lambda_n < M(\varepsilon)$ we have*
$$\omega_n = P_{n,\delta}(\omega_n) + \eta$$
where $\eta \perp \mathcal{W}_{n,\delta}$ and satisfies $\|\eta\|_{L^2}^2 \le C/\left(\delta^2|\log(\varepsilon)|\right)$.

The first part of the Proposition states that up to a threshold $M(\varepsilon)$ the eigenvalues of the composite object are close to the eigenvalues of the parts if the neck region is small enough. The reason for the threshold is that the transition regions themselves begin to contribute to the spectrum of P^*P when the eigenvalue under consideration is sufficiently large. The spectral data of P^*P above the threshold would contain eigenvalues coming from the transition region and corresponding eigenfields supported (up to small L^2 error) on the transition region. We can say that the threshold increases to infinity with ε because we have assumed that the transition region decreases in size with ε in a uniform way.

The second statement is more complicated to account for multiplicity. This is because it can happen that a finite number $\lambda_{n_1}, \ldots, \lambda_{n_k}$ of the eigenvalues of P^*P are very close together; in this case, any linear combination of the corresponding eigenvectors $\omega_{n_1}, \ldots, \omega_{n_k}$ would be approximately an eigenvector that should be compared to a linear combination of $\tilde{u}_k$ for appropriate k. Thus the Proposition asserts that the *subspace* spanned by $\omega_{n_1}, \ldots, \omega_{n_k}$ is approximately equal to the *subspace* spanned by these $\tilde{u}_k$. Results of this kind are fairly well-known in the mathematical literature.

For completeness, we give a full proof of this result and the appropriate references in the supplemental material.



(a) (b) (c)

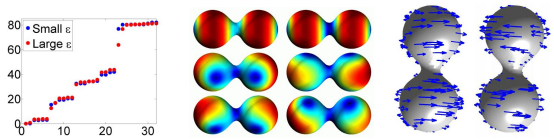
Figure 3: *A demonstration of Proposition 1: (a) the AKVFs of the parts, (b) the AKVFs of the composite, and (c) the spectrum of the composite (red) and the parts (blue).*

Figure 3 demonstrates the Proposition on a surface consisting of three parts (a sphere and two ellipsoids) connected by small transition regions. Figure 3(a) shows the norm of the first 12 AKVFs of the three parts (u_n for $n = 1, \ldots, 12$), grouped by similar eigenvalues. Figure 3(b) shows the first 12 AKVFs of the composite (ω_n), grouped the same way. Finally, in Figure 3(c) we see the sorted eigenvalues of the parts (μ_n), and the composite (λ_n). Within each group, the exact ordering is immaterial, since the eigenvalues are similar. It is evident that within each group the eigenvectors of the parts match the eigenvectors of the composite, up to effects cause by the "neck" regions. In general, the eigenvalues of the composite are bigger than the eigenvalues of the parts, since the "neck" hampers the exact symmetry of the parts.

2.3. Toy Examples

To demonstrate Proposition 1, we consider a dumbbell surface Σ as the one in Figure 4. Assume Σ is cylindrically symmetric about the axis through the sphere centers, that the spheres have radius 1, and that the neck has radius $\varepsilon \ll 1$.

Figure 4(a) compares the first 32 eigenvalues of Σ for two values of ε. The first six eigenvalues of the components are zero since each sphere has three exact KVFs, and the rest are clustered with varying multiplicity. The spectrum of Σ is closer to those of the components when ε is smaller. Specifically, the first six eigenvalues of Σ are close to zero and separated from the others.

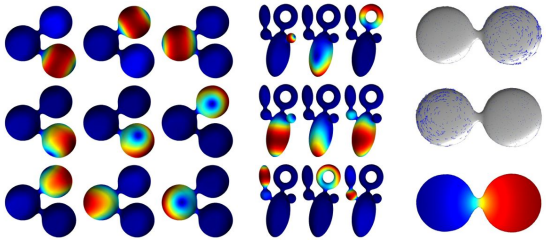


(a) (b) (c)

Figure 4: *(a) The first 32 eigenvalues for two neck sizes; (b) norms of the first six AKVFs; and (c) the first two AKVFs, the sum and difference of which generates localized fields.*

Figure 4(b) shows the norms of the first six AKVFs of Σ. The AKVFs of the composite are not localized on the two spheres, and have similar support on both. A closer look in Figure 4(c), however, shows that the first AKVF rotates both spheres in the same direction, whereas the second AKVF rotates the spheres in opposite directions. Taking the sum and difference of the first and second eigenvectors reveals two new "almost" eigenvectors, which are localized on the two spheres. In general, some unknown linear combination would generate "almost" eigenvectors localized on the parts.

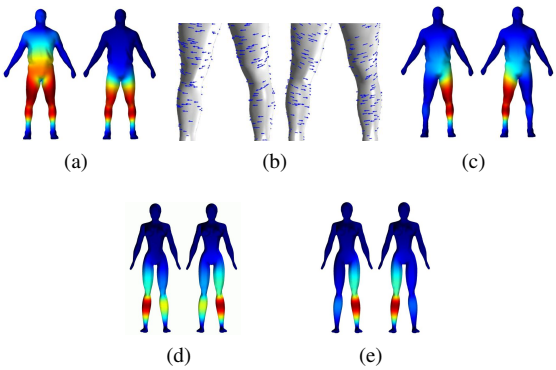
Figure 5(a,b) shows more examples of AKVFs of surfaces created by gluing together primitives with necks of varying size using the MeshMixer software [SS10]. Note that localization persists even if the parts have higher genus, e.g. a torus. Furthermore, although we primarily use extrinsic primitives in these examples, our formulation is purely intrinsic, and thus holds for almost-isometric deformations of extrinsic primitives, such as the octopus models in Figure 9.



(a) (b) (c)

Figure 5: *(a,b) Examples of AKVFs of composite surfaces. (c) Different behavior of AKVFs (top, center) versus the Fiedler vector (bottom), on a shape where the rotational symmetries of the parts are not aligned with the principal direction of the shape.*

Figure 5(c) shows the different behaviors of the Laplace-Beltrami eigenvectors and the Killing eigenvectors. The model is composed of two flat ellipsoids, each having rotational symmetry, connected at the equator by a thin "neck". The first two AKVFs of the composite, shown in Figure 5(c, top and center) indeed correspond to the first eigenvectors of the parts, those that generate the extrinsic rotational symmetry. The Fiedler vector - the first eigenvector of the Laplace-Beltrami operator - shown in Figure 5(c, bottom), on the other hand, follows the principal direction of the shape.



(a) (b) (c)

(d) (e)

Figure 6: *(a) Norms of the first two AKVFs; (b) the AKVFs rotate in the same and opposite directions, respectively; (c) the norm of the sum and difference of the AKVFs from (b); and (d,e) the norm of the first two AKVFs and a linear combination of them for another model.*

2.4. Real-World Composite Models

The spectral comparison theorem of the previous section is in essence a *perturbative* result that holds when ε is small.

There are generalizations of Proposition 1 that handle the case when the transition region has fixed length but infinitesimal width [JM92] (where the neck contributes in lower eigenvalues), and that derive asymptotic expansions of the spectral data in powers of ε [Gad05] (providing intuition for what happens when ε increases). But an exact analysis of the behaviour of the spectral data is difficult for large ε.

However, real-world composite models are exactly those for which we cannot expect the transition regions to be infinitesimal. Yet experimental evidence suggests that the relation between the eigenvectors of the composite and its parts is mostly preserved. Figure 6 shows of AKVFs of two models from the Princeton Segmentation Benchmark [CGF09] and linear combinations localized on the parts.

Note how the phenomena exhibited on the spheres repeats for the man model in Figure 6(b): the first AKVF rotates both legs in the same direction, whereas the second rotates each leg in the opposite direction. Figure 6(c) shows the sum and difference of the first two AKVFs, which are localized on each leg. Figure 6(d,e) show the first two AKVFs, and a linear combination of them for another model.

3. Untangling AKVFs

We have seen that the AKVFs of a composite surface span a linear space close to the one spanned by the AKVFs of its parts. Thus, if we have extracted the AKVFs of a composite, there should exist linear combinations that generate a set of vector fields localized on the parts; we call the technique for finding these fields "untangling." In this section, we describe how to untangle a given set of input vector fields $\omega_1, \ldots, \omega_N$ to produce an alternative set $\tilde{u}_1, \ldots, \tilde{u}_N$, where each $\tilde{u}_i$ is a linear combination of the inputs optimized to have maximally separate support that represents a different part of Σ.

3.1. Tangling Energy

The overlap between two vector fields ω_1 and ω_2 on Σ can be measured using a "tangling energy" designed to identify pairs of fields whose supports are similar:

$$E_{Tang}(\omega_1, \omega_2) = \int_\Sigma \|\omega_1\|^2 \|\omega_2\|^2$$

Note that uniformly scaling a field ω does not affect its support or Killing eigenvalue, so before untangling we normalize ω_i such that $\int_\Sigma \|\omega_i\|^2 = 1$. Then, the integrand above is large where both ω_1 and ω_2 have large magnitudes, showing where the fields are tangled locally.

Usually, more than two fields need to be untangled, hence we extend the energy function E_{tang} using pairwise products:

$$E_{Tang}(\omega_1, \ldots, \omega_N) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \int_\Sigma \|\omega_i\|^2 \|\omega_j\|^2 \qquad (1)$$

On a discrete mesh $M = (V, F, E)$, the vector fields are represented using discrete one-forms. To be consistent with this DEC formulation, we express the norm of ω_i on each edge as a vector of length $|E|$ as well. For the edge $e \in E$, we take $(\|\omega\|)_e = \sqrt{(\star_1)_e}|(\omega)_e|$, where $\star_1$ is the discrete diagonal Hodge star for one forms, given by a diagonal matrix containing the standard cotangent weights [Hir03]. This definition is consistent with the usual DEC definition of the L^2 inner product, which sums these norms on the whole mesh: $\langle \omega, \omega \rangle = \omega^\top \star_1 \omega$.

To define discrete tangling energy, we work in matrix notation. We define a matrix $\Phi \in \mathbb{R}^{|E| \times N}$ whose i-th column is ω_i. We then take $H = \sqrt{\star_1}$ and $|\Phi|$ to be the matrix that contains the absolute values of the elements of Φ. It is easy to see that the i-th column of the matrix $H|\Phi|$ is the vector $\|\omega_i\|$. Since $(H|\Phi|)^2_{ei} = (H_{ee}|\Phi_{ei}|)^2 = (H_{ee}\Phi_{ei})^2$, we have that the discrete tangling energy is given by:

$$E_{Tang}(\Phi) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{e=1}^{|E|} (H\Phi)^2_{ei} (H\Phi)^2_{ej} \qquad (2)$$

3.2. Untangling Vector Fields

The objective of the AKVF untangling algorithm is to find a set of coefficients defining a linear combination of the eigenvectors ω_i that minimizes the discrete tangling energy E_{Tang}. This linear combination can be expressed as $\Phi\mathcal{A}$, where $\mathcal{A} \in \mathbb{R}^{N \times N}$. To avoid degenerate situations and regularize the system, we limit ourselves to rotation matrices $\mathcal{A} \in SO(N)$. Formally, we solve the optimization problem:

$$\underset{\mathcal{A} \in SO(N)}{\text{minimize}} \quad E_{Tang}(\Phi\mathcal{A}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{e=1}^{|E|} (H\Phi\mathcal{A})^2_{ei} (H\Phi\mathcal{A})^2_{ej} \qquad (3)$$

This optimization problem is very similar to the problem of rotation in factor analysis [Har76], and the untangling energy is related to the so-called *orthomax criterion*. The goal in factor analysis is to extract a small set of "factors" that explain a larger set of given observations, e.g. by using Principal Component Analysis. These factors are required to be sparse, such that any observation can be described as a linear combination of a small number of factors. Thus, the usual procedure is to first extract the factors and then to find the optimal linear combination that will lead to sparsity. Although our situation is not directly related to factor analysis, factor sparsification is exactly what we are attempting to accomplish in "untangling" the AKVFs. As it turns out, the discrete tangling energy given in Equation 2 is identical to the statistical "parsimony criterion" described in [Har76].

Rotation has been well-studied in factor analysis, and efficient methods have been developed for finding a local minimum of the optimization problem in Equation 3. Specifically, an efficient algorithm using SVD was proposed in [Hor65] and used later for sparse modeling of medical images in [SSL06]. The algorithm is based on the fact

that the solution $\mathcal{A}$ is equal to $\arg\max_{\mathcal{A}}(\mathrm{tr}(\mathcal{A}^\top Q))$, where $Q = \Phi^\top H^\top (H\Phi\mathcal{A})^{\hat{}3}$ (the notation $X^{\hat{}3}$ means the element-wise exponentiation by three of the elements in the matrix X). Thus, one can keep Q fixed, solve for the matrix $\mathcal{A}$ which maximizes $\mathrm{tr}(\mathcal{A}^\top Q)$, and then update the estimate for Q using $\mathcal{A}$. Although the algorithm appears in several other sources, we provide it here for completeness in Algorithm 1; note that each iteration is fairly efficient, since it involves only matrix multiplication and finding the SVD of a relatively small $N \times N$ matrix.

Input: $\Phi_{in} \in \mathbb{R}_{|E|\times N}$, H, tol, $maxiter$
Output: Φ_{out}
begin
  $\mathcal{A} \leftarrow Id$; $\Phi_{out} \leftarrow H\Phi_{in}$; $d \leftarrow 0$
  for $i \leftarrow 1$ **to** $maxiter$ **do**
    $d_{old} \leftarrow d$
    $[L,D,M] = svd((H\Phi_{in})^\top \Phi_{out}{}^{\hat{}3})$
    $\mathcal{A} \leftarrow LM^\top$; $d \leftarrow trace(D)$; $\Phi_{out} \leftarrow H\Phi_{in}\mathcal{A}$
    if $|d - d_{old}|/d < tol$ **then**
      break
    end
  end
  $\Phi_{out} \leftarrow \Phi_{in}\mathcal{A}$
end

Algorithm 1: Procedure for untangling AKVFs.

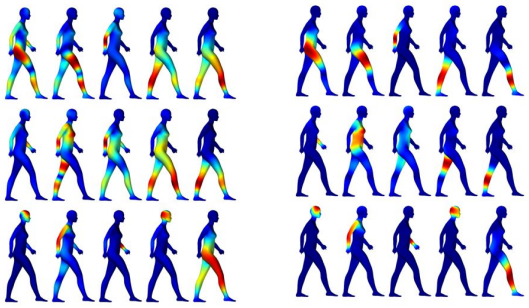


Figure 7: *AKVFs before (left) and after (right) untangling.*

For our experiments, we used $tol = 10^{-5}$ and $maxiter = 500$. Figure 7 shows examples of the norms of a few AKVFs before and after untangling; the untangled AKVFs are localized on the parts of the composite object, as we required. In the following section we will show how to use these localized vector fields for part discovery. Note that it can be the case that two EKVFs have similar support and cannot be untangled; for instance, this situation may occur when a part exhibits both rotational and translational symmetry. In this case, factor analysis simply produces multiple fields localized on the same part; no undesired cancellation occurs since this would imply the existence of a smaller intrinsic symmetry on the part that is not present. Multiple fields localized on the same part may help classify the *type* of symmetry on the part but should not affect part discovery or segmentation, so the particular linear combination of fields with similar support produced by factor analysis is unimportant.

4. Part Discovery with Untangled AKVFs

In Section 2 we showed the relation between the AKVFs of a composite and those of its parts. Using this, we demonstrated in Section 3 how to extract localized vector fields from the set of AKVFs of the composite. Assuming our surface is composed of a few parts, these extracted vector fields should be close to the AKVFs of the parts. If we now formulate a clustering problem using these new vector fields as features, we should be able to discover the intrinsic approximately symmetric parts. Indeed, given a surface Σ and a set of N localized vector fields $\tilde{u}_i$, we are looking for k components Ω_i and a connecting region $\mathcal{N}$ such that $\Sigma = \mathcal{N} \cup (\cup_i \Omega_i)$ and each $\tilde{u}_i$ has non-negligible norm only on one of the Ω_j's.

In the discrete setting, we are given a triangle mesh $M = (V,F,E)$. To extract the components from the $\tilde{u}_i$, we first map each face to a feature vector given by a function $\varphi : F \to \mathbb{R}^N$. Since we want to find the parts on which the $\tilde{u}_i$ are localized, we use the norm of the interpolated $\tilde{u}_i$ at the center of each face for the features. The mapping φ is thus defined as

$$\varphi(f) = (\|\tilde{u}_1(f_c)\|, \|\tilde{u}_2(f_c)\|, ..., \|\tilde{u}_N(f_c)\|),$$

where $f \in F$ and f_c is the barycenter of f.

Since we know the $\tilde{u}_i$ are localized on the Ω_j, we expect the points in high-dimensional space to be grouped such that in each group only a small number of coordinates is nonzero. As a simple example, consider a model consisting of three parts, and take $N = 3$. In this case, each face will be mapped to a point in $\mathbb{R}^3$ that lies on one of the coordinate axes. This separation can be seen easily in Figure 8(a), which shows the feature space $\varphi(F)$ of the "three spheres" object from Figure 5(a). In this case, the AKVFs were localized to begin with and hence are equal to the $\tilde{u}_i$.

(a) (b)

Figure 8: *For 3 AKVFs the feature space, $\mathbb{R}^3$, can be easily visualized. (a) The segmentation of the model from Figure 5(a). (b) its matching segmented 3D feature space.*

To generate a coarse partition of Σ we cluster the faces in feature space using the distance $d(f_i,f_j) = 1 - \cos(\angle(\varphi(f_i),0,\varphi(f_j)))$. The distance is chosen to be scale-invariant because scaling a feature vector does not affect which fields are the most prominent at a given point on the surface. If the points lie on the same coordinate axis, this distance will be close to zero, mapping such points to the

same cluster. Note that this approach is identical to projecting the points on the sphere and using squared Euclidean distances. In the general case, each part might have more than one AKVF, since the original part might have had multiple symmetries, and then the features of each part lie on a different low-dimensional hyperplane in $\mathbb{R}^N$. In that case as well, using the angle-based distance will result in the correct clusters. We use a standard k-means classifier with this metric to compute a set of cluster centers $C_j \in \mathbb{R}^N, j = 1,\ldots,k$ and assign each face to its closest center to get a coarse partitioning of the faces S_i. Figure 8 shows the resulting partitioning for this simple model on the original surface and in feature space. Since k-means returns a local minimum depending on its starting point, we run the algorithm a few times, choosing the result in which $\sum_{i=1}^{k}\sum_{f\in S_i} d(\varphi(f),C_i)$ is minimal.

Optionally, we might want to generate smoother boundary curves not limited to the edges. To create a smooth boundary curve between two neighboring face clusters S_i and S_j, we compute the zero-isoline of the function $g_{ij} : M \to \mathbb{R}$ given by $g_{ij}(p) = d(C_i,\varphi(p)) - d(C_j,\varphi(p))$. This is done by defining φ on the vertices and interpolating it to the interior of the faces. φ on the vertices is defined as the area weighted average of φ on the neighboring faces.

Alternatively, to find Ω_i with stronger intrinsic symmetries and a nonempty asymmetric connecting region $\mathcal{N}$, we consider all pairs of adjacent face clusters, S_i and S_j, and extract the ε and $-\varepsilon$ isolines of the function g_{ij} for $\varepsilon > 0$ controlling the size of $\mathcal{N}$. Since this function is meaningless on other clusters, we only consider parts of these isolines curves which are either in S_i or in S_j. Faces between these curves are marked as transition regions.

Figures 9 and 10 shows part decompositions and segmentations extracted using this technique, given an untangled set of $\tilde{u}_i$. The number of parts k was given as a parameter, and we used $N = 2k$ AKVFs, as discussed in the next section. For some models we show a segmentation by taking $\varepsilon = 0$, and for others we show the symmetric parts and remove the connecting region $\mathcal{N}$.

5. Experimental Results

5.1. Implementation Details

Given the building blocks detailed in the previous two sections, description of the entire symmetric part discovery technique is fairly straightforward:

1. Compute AKVFs $\omega_1,\ldots,\omega_N$.
2. Untangle $\omega_1,\ldots,\omega_N$ using Algorithm 1.
3. Cluster faces using k-means as described in Section 4.

With code from [BCBSG10] for computing AKVFs, the part discovery algorithm is straightforward to implement. Using Matlab's implementation of k-means and SVD, code for producing an unrefined mesh segmentation fits into a relatively short script. To generate the smooth isolines, we refined the original mesh to contain vertices that coincide with the 0 and ε isolines. As a post-processing step, if $\varepsilon > 0$, we remove boundary regions that border on a single symmetric part and add them to that part. Additionally, we remove parts smaller than some fixed size. The process takes 20-30 seconds to run on average-sized meshes from the Princeton Segmentation Benchmark using a laptop wih a 3.06GHz processor; a refined implementation likely could run faster.

5.2. Parameters and Limitations

The algorithm has relatively few parameters, and none exerts undue influence on the output. The parameters needed to run the algorithm are as follows:

- k, the number of desired segments. This parameter obviously affects the resolution of features likely to be discovered; a potential topic for future research is the development of techniques for estimating this value.
- N, the number of AKVFs to compute. This parameter determines the number and size of the possible features to identify; low values will leave out important AKVFs describing symmetric surface components, while high values of N generate high-frequency AKVFs of limited usefulness. Since each part is supposed to contribute a maximum number of three AKVFs (if it is a deformation of a sphere or a plane) and usually two AKVFs (if it is a deformation of a general surface of revolution), we set $N = 2k$. We added upper and lower bounds of 16 and 32, to make sure we have a sufficiently large range of AKVFs.
- For all our experiments we used 100 k-means iterations, and $\varepsilon = 10^{-5}$ for Algorithm 1, which in all cases was more than sufficient to reach convergence.

The main limitation of this approach is its dependence on mesh quality. Robust curvature estimates and other precautions must be used to compute useful AKVFs on noisy surfaces. Remeshing models improved AKVF computation in some examples. A less prominent limitation of the intrinsic primitive technique is that it can be difficult to find parts that are on different scales, since smaller features are likely to be associated with larger eigenvalues that might be ignored.

Outputs of part discovery occasionally might not align with segmentations generated by humans given the goal of finding intuitively meaningful rather than symmetric parts. As with any other segmentation or part discovery technique, the use of the method presented here must be paired with its application. While the AKVF method reliably discovers parts with intrinsic symmetries it does not always identify surface regions with asymmetric but semantically recognizable features; this behavior is expected and potentially desirable depending on the use of the algorithm.

5.3. Applications

The discovery of basic parts of a mesh is a fundamental task in geometry processing, and there exist several applications for a fully-intrinsic approach to such a problem. We present two short applications of our technique.

5.3.1. Segmentation

When $\varepsilon = 0$, the computed symmetric parts are guaranteed to be adjacent. Thus, they segment the input surface Σ into k parts with similar AKVF structures, which reflect the basic symmetries of the surface patches. Figure 9 shows such segmentations of some sample surfaces. These not only reveal the symmetric parts of the surfaces but in many cases also provide an intuitive divisions of the surfaces. The construction of the intrinsic primitive segmentation thus suggests that intrinsic symmetries may play some role in human perception of the segments in a given model. The figure also compares between our segmentation and two state-of-the-art segmentation methods: Shape Diameter Function [SSCO08], and Randomized Cuts [GF08]. The comparison shows that in many cases our segmentation agrees with those methods, and in a few cases even improves on them. It is worth noting that we did not post-process our cut lines: they are just the iso-lines of a distance function as defined in Section 4.

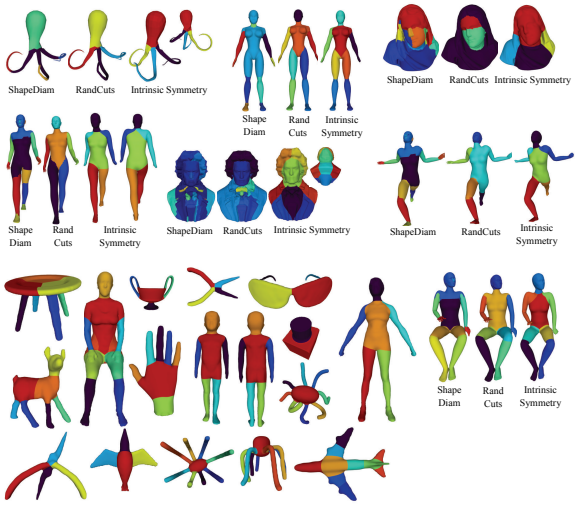


Figure 9: *Segmentations of various shapes (bottom), with comparisons to [SSCO08] and [GF08] algorithms (top).*

The results in Figure 9 reveal pros and cons of the proposed approach. For instance, the leg segments on the octopi demonstrate invariance to isometric deformation. The legs of the human model on the upper right are segmented differently above and below the knees because they cut off the translational symmetries of the upper and lower parts; smaller k values keep the full legs in individual segments because the weaker translational symmetries are ignored in favor of the rotational symmetries of the entire legs. Similar dependences on k can be seen in the fawn and table models (lower left), where extra segments appear because k is fixed *a priori*; these issues might be resolved by replacing k-means with a method that chooses the number of clusters.

For completeness, Figure 11 compares the "Rand index" of intrinsic primitive segmentation to that of other published approaches in various object categories from the Princeton Segmentation Benchmark [CGF09]. This value measures the dissimilarity between the produced segmentations and the benchmark's ground truth dataset using the fraction of pairs of faces that both agree or disagree on the segment to which they belong; see [CGF09] for details. The new approach compares fairly well with the other approaches, although it does not provide the best possible segmentations according to the benchmark. As explained in Section 5.2, the Benchmark's comparison between intrinsic primitive segmentations and those produced by humans is not a perfect evaluation since the objective of intrinsic primitives is to search for a particular type of part. Even so, the reasonable performance indicated in Figure 11 shows that the segmentations are meaningful beyond the mathematics of AKVFs.

5.3.2. Part Discovery

Rather than forcing every patch on Σ to belong to a "symmetric" part, taking $\varepsilon > 0$ allows for a nontrivial asymmetric connecting region $\mathcal{N}$, producing parts with clearer symmetries. This adjustment shows the true value of the intrinsic primitive algorithm as a technique for part discovery rather than segmentation. Figure 10 shows part decompositions of some sample models. Additionally, each part is shown with flow lines indicating some of its principal symmetries. Finding the symmetries of a given part Ω_i is as simple as finding those untangled vector fields $\tilde{u}_j$ with the largest values of $\int_{\Omega_i} ||\tilde{u}_j||^2 / \int_{\Sigma} ||\tilde{u}_j||^2$ and projecting them onto Ω_i.

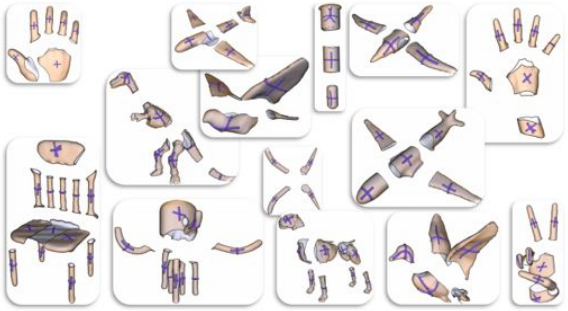


Figure 10: *Various models broken into symmetric parts, with flows of prominent AKVFs marked to show the intrinsic symmetries detected.*

6. Conclusions and Discussion

We have presented an algorithm for dividing a given surface Σ into nearly-symmetric regions Ω_i and an optional asymmetric connecting region $\mathcal{N}$. The algorithm is not only effective for finding parts with self-symmetry but also mathematically justified and provably intrinsic, as shown by examining the structure of AKVFs on composite surfaces. The part discovery method has relatively few parameters and is straightforward to implement given pre-existing numerical and geo-

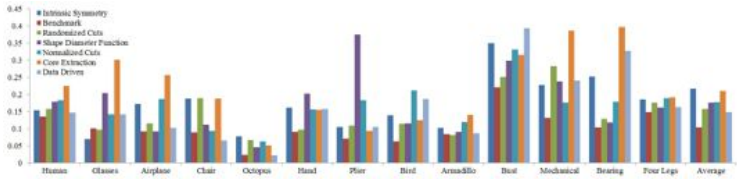


Figure 11: *Comparison of Rand indices computed by the benchmark in [CGF09].*

metric algorithm building blocks. A number of potential applications could make use of the decomposition of a shape into symmetric parts. Symmetric parts are straightforward to parametize, and the symmetries easily could help guide shape editing. Matching and correspondence techniques also can make use of the additional data about the symmetries provided by the AKVFs with support on a given segment.

Beyond its potential applications, this exploratory new technique also leaves a number of potential avenues for future research. Since intrinsic primitives are paired with their significant AKVFs, it should be possible to find group-theoretic descriptions of each part's self symmetries for use in understanding global structure. The presence or lack of symmetries could be used to compare surfaces and synthesize new geometry, and near-symmetries could be refined to denoise or improve surface models. The structure of the connecting region $\mathcal{N}$ also may provide some clues to the connectivity of various features on Σ.

Intrinsic primitives provide a unique perspective on the geometry of a given model and clearly demonstrate the value of AKVFs. Their application could play a key role in the larger research goal of understanding the structure and repetition present in discrete surfaces.

Acknowledgements

The authors would like to acknowledge the following grants: The NSF GRF program, the NDSEG program of the DoD, the Hertz Foundation Fellowship, the Weizmann Institute WiS award, NSF grants FODAVA 0808515, CCF 0808515, an AEA program grant from KAUST, a Google research grant, and a seed grant from the Stanford CS Department.

References

[AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput. 22*, 3 (2006), 181–193. 2

[APP*07] AGATHOS A., PRATIKAKIS I., PERANTONIS S., SAPIDIS N., AZARIADIS P.: 3d mesh segmentation methodologies for CAD applications. *CAD and Applications 4*, 6 (2007). 2

[BCBSG10] BEN-CHEN M., BUTSCHER A., SOLOMON J., GUIBAS L.: On discrete Killing vector fields and patterns on surfaces. In *Comp. Graph. Forum (Proc. of SGP)* (2010). 2, 3, 8

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3d mesh segmentation. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers* (2009), pp. 1–12. 2, 6, 9, 10

[DC93] DO CARMO M.: *Riemannian Geometry*. Birkhäuser, Boston, 1993. 2

[DKT06] DESBRUN M., KANSO E., TONG Y.: Discrete differential forms for computational modeling. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (2006), pp. 39–54. 3, 4

[Gad05] GADYLSHIN R.: On the eigenvalues of a "dumbbell with a thin handle". *Izv. Ross. Akad. Nauk Ser. Mat. 69*, 2 (2005). 6

[GF08] GOLOVINSKIY A., FUNKHOUSER T.: Randomized cuts for 3d mesh analysis. *ACM Tr. Graph. 27*, 5 (2008), 1–12. 9

[GG04] GELFAND N., GUIBAS L.: Shape segmentation using local slippage analysis. In *Proc. of SGP 2004* (2004). 2, 3

[Har76] HARMAN H.: *Modern factor analysis*, 3d ed., rev. ed. University of Chicago Press, Chicago :, 1976. 6

[Hir03] HIRANI A.: *Discrete exterior calculus*. PhD thesis, California Institute of Technology, 2003. 6

[Hor65] HORST P.: *Factor Analysis of Data Matrices*. Holt, Rinehart and Winston, New York, 1965. 6

[JM92] JIMBO S., MORITA Y.: Remarks on the behavior of certain eigenvalues on a singularly perturbed domain with several thin channels. *Comm. Partial Diff. Eq. 17*, 3-4 (1992). 6

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers* (2010), pp. 1–12. 3

[LCDF10] LIPMAN Y., CHEN X., DAUBECHIES I., FUNKHOUSER T.: Symmetry factored embedding and distance. In *ACM SIGGRAPH 2010 papers* (2010). 3

[MBB10] MITRA N. J., BRONSTEIN A., BRONSTEIN M.: Intrinsic regularity detection in 3d geometry. In *ECCV* (2010), pp. 398–410. 2

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3d shapes. *ACM Trans. on Graphics (Proc. Siggraph) 25* (2006), 549–559. 3

[RBBK07] RAVIV D., BRONSTEIN A., BRONSTEIN M., KIMMEL R.: Symmetries of non-rigid shapes. In *Proc. Workshop on Non-rigid Registration and Tracking through Learning* (2007). 2

[RBBK10] RAVIV D., BRONSTEIN A., BRONSTEIN M., KIMMEL R.: Full and partial symmetries of non-rigid shapes. *Intl. Journal of Computer Vision 89*, 1 (2010), 18–39. 2

[Reu10] REUTER M.: Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *Int. J. Comput. Vision 89*, 2-3 (2010), 287–308. 3

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556. 2

[SOCG10] SKRABA P., OVSJANIKOV M., CHAZAL F., GUIBAS L.: Persistence-based segmentation of deformable shapes. In *Proc. NORDIA 2010, Proc. CVPR 2010* (2010). 3

[SS10] SCHMIDT R., SINGH K.: MeshMixer: an interface for rapid mesh composition. In *SIGGRAPH 2010 Talks* (2010). 5

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput. 24*, 4 (2008), 249–259. 3, 9

[SSL06] STEGMANN M., SJÖSTR K., LARSEN R.: Sparse modeling of landmark and texture variability using the orthomax criterion. In *Intl. Symp. on Medical Imaging 2006* (2006). 6