Symmetry Factored Embedding And Distance

Yaron Lipman   Xiaobai Chen   Ingrid Daubechies   Thomas Funkhouser
Princeton University

Abstract

We introduce the Symmetry Factored Embedding (SFE) and the Symmetry Factored Distance (SFD) as new tools to analyze and represent symmetries in a point set. The SFE provides new coordinates in which symmetry is "factored out," and the SFD is the Euclidean distance in that space. These constructions characterize the space of symmetric correspondences between points – i.e., orbits. A key observation is that a set of points in the same orbit appears as a clique in a correspondence graph induced by pairwise similarities. As a result, the problem of finding approximate and partial symmetries in a point set reduces to the problem of measuring connectedness in the correspondence graph, a well-studied problem for which spectral methods provide a robust solution. We provide methods for computing the SFE and SFD for extrinsic global symmetries and then extend them to consider partial extrinsic and intrinsic cases. During experiments with difficult examples, we find that the proposed methods can characterize symmetries in inputs with noise, missing data, non-rigid deformations, and complex symmetries, without a priori knowledge of the symmetry group. As such, we believe that it provides a useful tool for automatic shape analysis in applications such as segmentation and stationary point detection.

1 Introduction

Symmetry plays a central role in nature. It is related to efficient and robust arrangements, and it is prevalent in both natural and man-made objects. As such, detecting, representing, and exploiting symmetries is an important research topic in many disciplines, including computer graphics and computer vision. Its numerous applications in graphics include registration, segmentation, recognition, compression, de-noising, viewpoint selection, completion, beautification, and several others.

The goal of our work is to investigate a new method for analysis and representation of symmetries in 3D objects. This problem is difficult when objects have complex, multiple symmetries (e.g., the 4 rotations and 4 reflections of the Jewelry model in Figure 1), partial symmetries (e.g., due to missing data and extra parts in the scan of the flower statue), approximate symmetries (e.g., due to noise in scanned data), and when the symmetry groups expected in the input data are not known in advance.

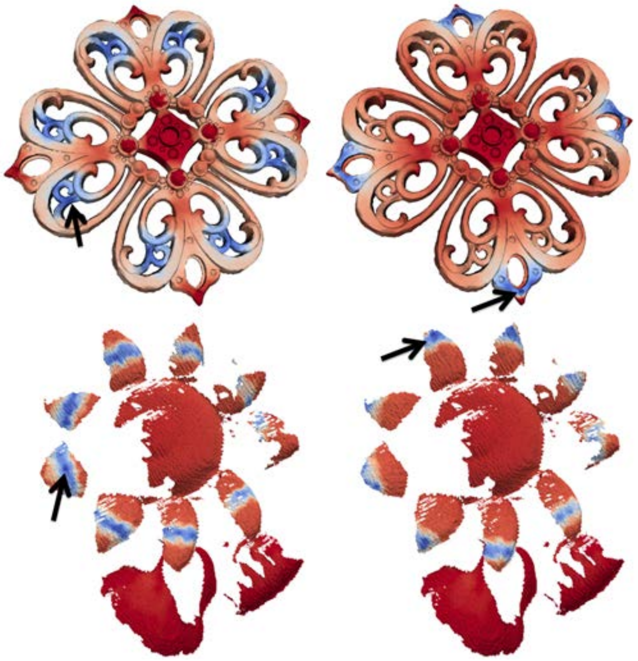


Figure 1: *Symmetry Factored Distance measured from the points marked by black arrows (blue is small distance, red is large).*

Recent approaches to address this problem have relied upon methods that analyze symmetries in terms of their generating transformations. For example, [Podolak et al. 2006] proposed a continuous measure assessing the degree to which an object is symmetric with respect to every transformation within a prescribed group (e.g., planar reflections), and [Mitra et al. 2006] described a method to detect symmetries by clustering "votes" for transformations that align boundary points with similar local shape descriptors. These methods have demonstrated their usefulness in several applications, but they have a few problems: 1) they require knowing a priori which symmetry transformations are expected (so that votes can be accumulated in an appropriate parameterization of the transformation space); 2) they require a large amount of processing and/or storage in high dimensional spaces (e.g., rigid body transformations have six dimensions); 3) they are able to represent symmetries only for transformations that can be parameterized (e.g., not arbitrary non-rigid deformations); 4) they cluster votes based on Euclidean distances in transformation space, which may not be meaningful due to the (usually) non-linear dependence on the parameterization; 5) they do not usually leverage the inter-dependence of repeated structures (e.g., rotations by 120 and 240 degrees), which appear as disjoint clusters in transformation space; and, 6) they do not explicitly produce correspondences between symmetric points, which are required for many applications (e.g., symmetrization, segmentation, de-noising, etc.).

Our approach is to find *orbits*, i.e., we search for correspondences between symmetric points rather than the transformations that align them. These correspondences are represented in a *symmetry correspondence matrix*, a (usually sparse) $n \times n$ non-negative matrix C that, roughly speaking, encodes symmetry relations between pairs of n points sampled from the input data. Of course, this approach is related to earlier methods that have computed fuzzy correspondence matrices for alignment and matching problems [Gold et al. 1998; Lipman and Funkhouser 2009]. However, a *symmetry* correspondence matrix has special properties that make it particularly well-suited for symmetry detection. In particular, the top eigenvectors of the correspondence matrix contain information that completely characterizes orbits. In the language of spectral methods on graphs, this observation can be rephrased by viewing the symmetric correspondence between points as an equivalence relation: all points symmetric with one another form an orbit, which appears as a clique in the graph induced by the symmetry correspondence matrix. As a result, the problem of finding approximate and partial symmetries in a point set reduces to the problem of measuring connectedness in the correspondence graph, a well-studied problem for which spectral methods provide a robust solution.

Analysis of the spectral properties of the symmetry correspondence matrix leads us to define a new embedding of the input shape into a higher dimensional Euclidean space (the Symmetry Factored Embedding, or SFE). Euclidean distances in this embedded space "factor out" symmetries of the original shape – accordingly we call it the Symmetry Factored Distance (SFD). More generally, the Symmetry Factored Distance between two points quantifies, in a continuous way, the extent to which they are "symmetric". This is shown in Figure 1, which visualizes the values of $\mathrm{SFD}(x, \cdot)$ (distance from one point to all others), for two models and two different choices of the source point x in each model (marked with a black arrow). Note how points in the same orbit have small distances (shown in dark blue).

The main contribution of this work is the observation that symmetry can be handled robustly and efficiently in correspondence space with the definitions of the Symmetry Factored Embedding and the Symmetry Factored Distance. To compute these representations, we provide a randomized algorithm to generate a fuzzy symmetry correspondence matrix from a 3D point set and then use spectral analysis of this matrix to compute the Symmetry Factored Embedding and Distance (Section 4). We find that these methods are able to reveal orbits of symmetric points for 3D point sets containing large amounts of noise, missing data, non-rigid deformation, and multiple partial symmetries for several types of input data, including LIDAR scans of cities and cryo-EM scans of macromolecules (Sections 5 and 6). Moreover, we demonstrate their utility in two shape analysis applications: mesh segmentation and detection of stationary points (Section 7).

2 Previous work

Understanding the symmetries of shapes is a well studied problem in many disciplines, including image processing, computational geometry, and computer vision. Recent work in computer graphics has focused on analyzing partial and approximate symmetries and on utilizing symmetries in shape analysis and mesh processing applications.

Symmetry detection in transformation space: Most recent approaches to detecting approximate and partial symmetries have focused on algorithms that cluster votes for symmetries in a parameterized "transformation space" [Imiya et al. 1999; Mitra et al. 2006; Yip 2000; Li et al. 2005]. For example, [Mitra et al. 2006] generate "votes" for transformations that align pairs of similar points and then cluster them in a space with coordinate axes labeled by the parameters of the potential symmetry transformations. These methods have several drawbacks (listed in the introduction), some of which can be overcome by utilizing discriminating local shape descriptors to ensure that votes are cast only for likely symmetric point correspondences. However, no matter how good the shape descriptors are, these methods are not effective at finding correspondences between points in complex symmetry orbits that are spread across multiple different clusters in transformation space.

Symmetry representation in transformation space: A similar vein of work has utilized symmetry measurements to define shape representations based on how symmetric an object is with respect to multiple transformations. For example, [Kazhdan et al. 2003] used Zabrodsky's symmetry distance extended to characteristic functions to define a "symmetry descriptor" that measures the symmetries of an object with respect to all planes and rotations through its center of mass. [Podolak et al. 2006] built upon this work and that of [Reisfeld et al. 1995] to define a planar reflective symmetry transform (PRST) that measures reflectional symmetries with respect to all planes through space. [Rustamov 2008] extended the PRST to consider correlations at multiple radii. These representations store a measure of symmetry for a regularly sampled set of transformations within a group, and thus they are practical only for groups with low dimensionality (e.g., rigid body transformations would require storing a six dimensional grid).

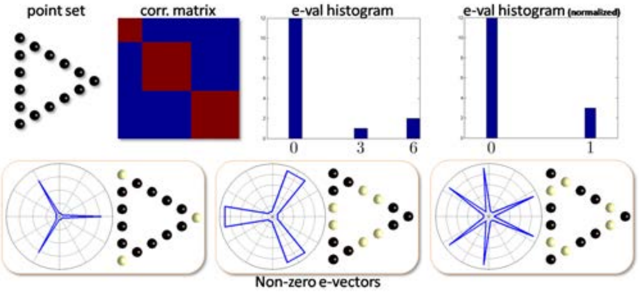
Symmetry representation on the shape: To extend symmetry analysis methods to non-rigid shapes, [Xu et al. 2009] proposes a method to accumulate votes for an "intrinsic reflectional symmetry axis" directly on the input surface. Sampled pairs of surface points vote for the curve of points geodesically equidistant to both samples to yield a continuous estimate of which points on the surface are stationary under intrinsic reflection. Grass-fire algorithms are used to extract a curve representation for the intrinsic reflectional symmetry axis and to establish correspondences between nearby symmetric points. This method was shown to be useful for segmentation and part repair, but it works only for reflection.

Discovery of repeating structures: Others have utilized the redundancy in repeating structures to detect symmetries more robustly [Bokeloh et al. 2009; Li et al. 2006; Leung and Malik 1996; Liu et al. 2007; Pauly et al. 2008; Shikhare et al. 2001]. For example, [Pauly et al. 2008] extend the voting method of [Mitra et al. 2006] by fitting parameters of a transformation generator to optimally register with clusters in transformation space. [Berner et al. 2008] and [Bokeloh et al. 2009] take a similar approach, using a subgraph matching of feature points and feature lines, respectively, to establish potential correspondences between repeated structures and a variant of iterative closest points (ICP) to simultaneously grow corresponding regions and refine matches over all detected patterns. This method is able to find repeated patterns in noisy data [Pauly et al. 2008], but requires a priori knowledge of the commutative group expected in the data and a non-linear optimization which may find a local minimum if not given a good initial guess.

Eigen-analysis methods: As symmetry detection can be viewed as a particular instance of the shape alignment problem, spectral methods previously introduced for matching, such as [Leordeanu and Hebert 2005], can be used for symmetry detection [Hays et al. 2006]. Chertuk and Keller [2010] suggest finding distinct symmetry transformations by observing top eigenvectors of the association graph's similarity matrix. The association graph matrix encodes consistency values of all couples of possible correspondence pairs and is a different matrix from the one used in this paper. Their goal is also different from ours in that they aim at finding correspondences supporting different distinct symmetry transformations, while we aim to find orbits. One problem with this approach is that correspondences from distinct symmetry transformations may be spread across different eigenvectors corresponding to similar (or same) eigenvalues resulting in inconsistencies in the extracted correspondences. For example, for a perfectly symmetric shape the dimension of the top eigenspace of the association graph's matrix equals the number of distinct symmetry transformations and it is not clear how to choose the "correct" eigenvector basis that separates the different transformations correctly. Our approach defines a stable distance using all eigenvectors and eigenvalues that is insensitive to such mixing, as well as independent of the specific choice of eigenvectors (it depends on only the eigenspaces).

3 Approach

Our goal is to detect and quantify *symmetries* in a point set $\mathcal{X} = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$. A symmetry is a group of transformations $\mathcal{G} = \{g\}$ that act on $\mathcal{X}$ and leave it intact. For example, the set of points shown below, sampled from an equilateral triangle, can be rotated by $120^\circ, 240^\circ$ around its center, and we get the same set of points after the rotation (if we forget about the labeling of the points). Likewise, the point set is preserved if we reflect these points with respect to any line connecting the center and one of the corners of the triangle. Therefore the symmetry group of this point set is the *dihedral* group. Often symmetry groups $\mathcal{G}$ are subgroups of rigid transformation in the Euclidean two or three dimensional space, but they need not be restricted to only this case; for example, self isometries of a surface are also symmetries.



A basic observation of this paper is that it is useful to think about symmetry in the *correspondence space*. We start with describing the perfect symmetric case, that is, a case similar to the triangle, in which every symmetry transformation leaves the points $\mathcal{X}$ intact ($g\mathcal{X} = \mathcal{X}$), and we address the case of imperfect/approximate symmetry afterwards.

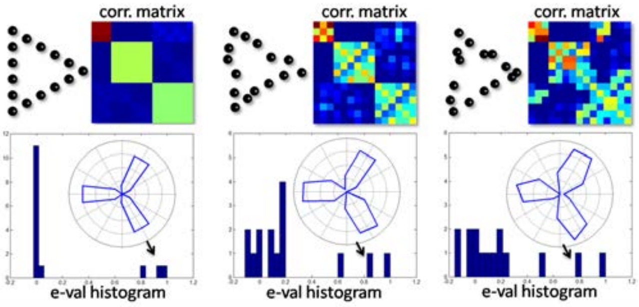
Perfect symmetry The symmetry correspondence graph is a graph whose vertices are the points in $\mathcal{X}$, with undirected edges (x_i, x_j) between points in the same *orbit*. Two points are in the same orbit if there exists a symmetry transformation which takes x_i to x_j. It is easy to check that since symmetry transformations of a shape form a *group*, the relation "x_i, x_j are in the same orbit" is an equivalence relation. This means that the symmetry correspondence graph has a very specific structure: the number of its connected components equals the number of orbits, and each such connected component is a clique. The symmetry correspondence graph can be described by an adjacency matrix $C \in \mathbb{R}^{n \times n}$, which is generally sparse (at least for discrete symmetries). If we rearrange the rows and columns of this matrix according to the orbits, then C is a block-diagonal matrix in which each block consists of only ones, and zeros appear everywhere outside the blocks. The figure above shows this matrix for the triangle point set (red is one and blue zero).

The spectral properties of the symmetry correspondence matrix C contain a lot of information about the symmetry of the shape and will play an important role in extracting data from the symmetry correspondence matrix, especially in the imperfect case. As proved in the Appendix, the number of non-zero eigenvalues equals the *number* of orbits, the magnitude of each eigenvalue is the *size* of that orbit, and there exists an eigenvector corresponding to each eigenvalue that is constant on the corresponding orbit and zero everywhere else. These eigenvectors are shown for the triangle example above: the image second from the right in the top row shows the eigenvalue histogram of the symmetry correspondence matrix, and the bottom row visualizes the three nonzero eigenvectors. Each eigenvector is shown twice: once as a polar plot, and once color coded on the point set $\mathcal{X}$. Note that the white points are exactly the orbits.

Further note that these eigenvectors can be thought of as functions defined over the point set $\mathcal{X}$, that is $f : \mathcal{X} \to \mathbb{R}$; we will therefore also refer to them as eigenfunctions. The linear space spanned by these non-zero vectors is the space of functions that are invariant to the shape's symmetry: every combination of these vectors is constant on the orbits, and points in $\mathcal{X}$ get transformed by symmetries only inside their orbit, and, vice versa, every symmetry-invariant function is constant on orbits. This means that the non-zero spectrum of the symmetry correspondence matrix encapsulates all the information about the orbits of the symmetry, regardless of its specific *type* (e.g., cyclic, dihedral, etc.). Importantly, it is separated from the "rest" of the eigenvectors by a spectral gap related to the size of the orbit. This is even more apparent after a normalization such that the rows of the symmetry correspondence matrix sum to one – then, the nonzero eigenvalues are all ones, as shown in the histogram of eigenvalues on the top right of the figure above.

Our plan is to use the top eigenvectors multiplied with their eigenvalues to define an embedding of the point set in a higher dimensional space where Euclidean distance in that space "factors out symmetry". Since in the perfect case, only the eigenvectors corresponding to nonzero eigenvalues are constant on orbits and they span the space of functions constant on orbits, this procedure will lead to an embedding where the Euclidean distance is zero between points in the same orbit and nonzero between points in different orbits. We will name this embedding the Symmetry Factored Embedding (SFE) and the corresponding Euclidean distance in that space the Symmetry Factored Distance (SFD).

Approximate symmetry "Real life" data is seldom perfectly symmetric, due to sampling, noise, warps, partial symmetry, etc. In this case, we use a *continuous* estimation of symmetry between points in the data $\mathcal{X}$ to build a *fuzzy* version of the symmetry correspondence matrix (i.e., with continuous values $[0, 1]$) and then use its spectral properties to construct the Symmetry Factored Embedding and Distance. Intuitively, our continuous estimation of symmetry between points is a relaxation of the above described binary relation "being in the same orbit". There are two justifications for using the SFE and SFD to detect approximate symmetries. First, since there exists a *spectral gap* separating the symmetry-aware eigenfunctions from the non-symmetry-related eigenfunctions, even when the symmetry correspondence matrix has missing entries or is contaminated with noise, the top eigenfunctions are stable. This is demonstrated in the image below, where we see three sets of points, sampled from a perfect equilateral triangle, with different noise levels. To the right of the point set we show in each case the fuzzy correspondence matrix as computed using our (randomized) algorithm. Note that to bring out the block structure, the rows and columns are rearranged - for visualization purposes only, since we obviously do not know this ordering, which determines the orbits, in the general case. As the shape becomes more noisy and deviates from perfect symmetry, the symmetry correspondence matrix gradually loses its block structure. Nevertheless, the top three eigenfunctions are still distinct from all the rest (see the eigenvalue histogram on the bottom row); we show the second most dominant eigenfunction in polar coordinates, and it is clear that it is stable as well under the deviation from perfect symmetry.



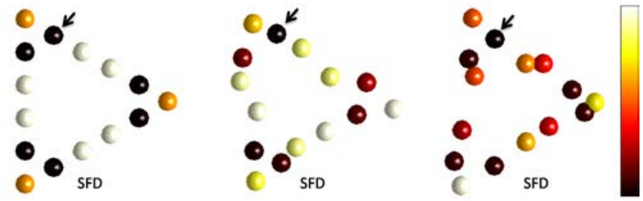
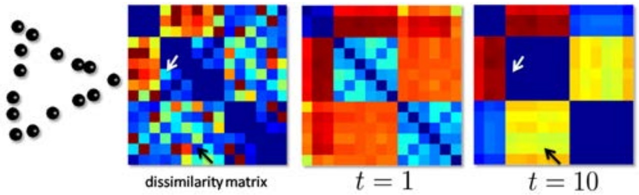


Figure 2: *The SFD on the triangle point sets.*

Second, the Symmetry Factored Embedding and Distance can be viewed as a version of the diffusion map and diffusion distance [Coifman et al. 2005; Nadler et al. 2005], with a special "symmetry-factored" kernel. Therefore, the SFD (visualized with colors for the three triangles in Figure 2) has a natural interpretation in terms of random walks on a graph. Intuitively, the SFD is measuring connectivity in the symmetry correspondence graph by integrating over *all* paths (up to a certain length) between points. In the perfectly symmetric case, the symmetry correspondence graph is a disconnected collection of cliques; as we deviate from perfect symmetry, edges are added between different cliques and removed inside the cliques. It follows that SFD, which considers all the paths up to some length t between points, is a very suitable tool to robustly detect these contaminated cliques. This is shown in the example below, which the left colored matrix shows an approximation of the symmetry dissimilarity matrix of the triangle point set (approximated by the first step of our algorithm after only few iterations). Obviously the block structure misses some strong links inside the blocks (cliques) and has some "shortcuts" between different blocks (cliques). Next to it, we show the Symmetry Factored Distance matrix for "times" (=length of path) $t = 1, 10$; we mark by white and black arrows examples where our algorithm finds missing links and rectifies shortcuts, respectively.



4 Methods

Our implementation of the proposed methods follows the computational pipeline shown in Figure 3. Given a point set $\mathcal{X}$ (top-left corner), we first construct a fuzzy correspondence matrix by a randomized voting procedure (bottom left, Section 4.1). Then, we compute the Symmetry Factored Embedding, $\Pi(\mathcal{X})$, that directly provides the Symmetry Factored Distance (Section 4.2), which can be visualized as a (block-structured) correspondence matrix (top row) or by a distance field from selected points (three are shown in the bottom right). Each of these steps is discussed in detail in the following subsections. For the sake of clarity, we focus the discussion on methods for finding global symmetries that are subgroups of rigid motions, and later we will generalize to partial and non-rigid symmetries (Section 5).

4.1 Symmetry Correspondence Matrix

The first step in our algorithm is constructing a fuzzy symmetry correspondence matrix $C \in \mathbb{R}^{n \times n}$ for the point set $\mathcal{X} = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$ (in this paper $d = 2, 3$). The entries $C_{ij} \in [0, 1]$ quantify continuously how much the points x_i, x_j belong to the same orbit.

We start by describing our computation for the *dissimilarity* measure of symmetry between pairs of points undergoing global, rigid transformations. We define the matrix $S \in \mathbb{R}^{n \times n}$ by

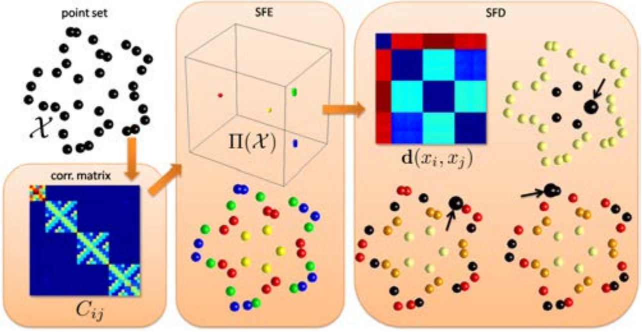


Figure 3: *Algorithm pipeline. Given a point set $\mathcal{X}$: 1) approximate the symmetry correspondence matrix C_{ij}, 2) using spectral properties of C, compute the Symmetry Factored Embedding $\Pi(\mathcal{X})$, and 3) extract Euclidean distances in the embedded space which are the Symmetry Factored Distance $\boldsymbol{d}(x_i, x_j)$.*

$$S_{ij} = S(x_i, x_j) = \inf_{g \in \mathcal{T} : g x_i = x_j} D(\mathcal{X}, g\mathcal{X}), \tag{1}$$

where $\mathcal{T}$ denotes the rigid transformations, $D(\mathcal{X}, \mathcal{Y})$ is a deviation measure between point sets $\mathcal{X}, \mathcal{Y}$, for example we used Root Mean-Squared Deviation (RMSD):

$$D(\mathcal{X}, \mathcal{Y}) = \left(\frac{\sum_{i=1}^{n} d(x_i, \mathcal{Y})^2 + \sum_{j=1}^{n} d(y_j, \mathcal{X})^2}{2n} \right)^{1/2}, \tag{2}$$

where $d(x_i, \mathcal{Y}) = \min_j \|x_i - y_j\|$ is the Euclidean distance from point x_i to the set $\mathcal{Y}$.

Intuitively S_{ij} measures how well can $\mathcal{X}$ be preserved by a rigid transformation that takes x_i to x_j. It is not hard to see that if $S_{ij} = 0$ then x_i, x_j are in the same orbit of a perfectly symmetric shape. Therefore this definition can be seen as generalization of the "in orbit" equivalence relation introduced before; it provides a continuous value rather than one or zero.

The dissimilarity matrix S can be approximated in a number of ways, of which we provide one possible method based on randomized voting at the end of this subsection. Once we obtain S we convert it to the (unnormalized) symmetry correspondence matrix via

$$\widetilde{C}_{ij} = e^{-\left(\frac{S_{ij}}{\sigma\ diam}\right)^2}, \tag{3}$$

where $diam = \max_{ij} \|x_i - x_j\|$ is the point set's diameter, and $\sigma > 0$ is a localization parameter which set the confidence in the higher values of the dissimilarity symmetry measure. Usually we use $\sigma = 0.1\% - 1\%$ such that the resulting matrix $\widetilde{C}$ is sparsely filled. The last step is turning $\widetilde{C}$ to the symmetry correspondence matrix C by making it row stochastic (meaning each row sum up to one). Intuitively, C_{ij} encodes the *probability* of each point x_j being in the orbit of point x_i. Therefore, C can be seen as a random walk matrix where jumping around in the same orbit is very likely, while passing between orbits, especially ones which do not have adjacent points is not likely.

We observe that, when translating a point set by some vector v the RMSD satisfies $D(\mathcal{X}, \mathcal{X} + v) \leq \|v\|$, therefore since the definition of S_{ij} is based on the infimum over all rigid transformation (including translation) we obtain the upper bound

$$S_{ij} \leq D(\mathcal{X}, \mathcal{X} + (x_j - x_i)) \leq \|x_j - x_i\|. \tag{4}$$

This bound provides a clear connection with diffusion maps [Nadler et al. 2005]: the dissimilarity used there (sometimes called affinity) is $\|x_i - x_j\|$, and the corresponding similarity measure satisfies $e^{-\|x_i - x_j\|^2/2\epsilon} \geq \widetilde{C}_{ij}$ for $\sigma = \sqrt{2\epsilon}/diam$. This shows that our random walk matrix is exactly like diffusion, except we have *shortcuts* for symmetry; points in the same orbit are linked with short (ideally zero length) edges.

Construction of S: Calculating S directly from the definition (1) requires searching over the orthogonal transformation group for every pair of points. Instead, we use a randomized algorithm in the spirit of previous voting techniques [Mitra et al. 2006]. First, we initialize our matrix $S_{ij} = \infty$. Next, we repeatedly pick two pairs of points x_1, x_2 and y_1, y_2 randomly of approximately the same distance $|\|x_1 - x_2\| - \|y_1 - y_2\|| < \alpha \cdot diam$ (we use $alpha \approx 0.02$), and which are not too close to each other $\|x_1 - y_1\| > \epsilon \cdot diam$ or $\|x_2 - y_2\| > \epsilon \cdot diam$ (we use $\epsilon \approx 0.025$), since we always know that the identity is perfect symmetry (the diagonal of S is always zero). Next, we augment each pair with the approximated normal at the first point in each pair $\bar{n}(x_1), \bar{n}(y_1)$. We only consider pairs which have approximately the same angle w.r.t their normal $\langle n(x_1), x_2 - x_1 / \|x_2 - x_1\| \rangle \approx \langle n(y_1), y_2 - y_1 / \|y_2 - y_1\| \rangle$ (we allowed deviation of upto 5 degrees). We then orthogonalize each pair of points (as a vector) w.r.t the normal and complete it to an orthogonal frame where the direction of the third added vector is randomized (to account for reflections): $v_1 = n(x_1)$, $\tilde{v}_2 = (x_2 - x_1) - \langle x_2 - x_1, n(x_1) \rangle n(x_1)$, $v_2 = \frac{\tilde{v}_2}{\|\tilde{v}_2\|}$, $v_3 = \pm v_1 \times v_2$. Next, the rigid transformation g defined by these two frames is applied to the entire point set ($g\mathcal{X}$), and the deviation from the original point set $D(\mathcal{X}, g\mathcal{X})$ is measured. If g aligns at-least some fraction of the points in $\mathcal{X}$ (usually $0.2 - 0.4$), measured by counting the number of mutually closest points, then the value $D(\mathcal{X}, g\mathcal{X})$ is cast into S using a $\min$ operation at the entries S_{ij} corresponding to these mutually closest pairs. Once the voting process ends, we translate the minimal value of S to zero, and take minimum of each entry S_{ij} with $\|x_i - x_j\|$, that is $S_{ij} \leftarrow \min \{S_{ij}, \|x_i - x_j\|\}$, justified by the bound in eq. (4).

4.2 Symmetry Factored Embedding and Distance

Once we have the symmetry correspondence matrix we can calculate the *Symmetry Factored Embedding*. The general technique of looking at the first eigenvectors of a similarity matrix and using them as "good coordinates" for dimensionality reduction and clustering has been widely exploited in machine learning literature and related fields [Shi and Malik 1997; Belkin and Niyogi 2001]. The new observation in this paper is that these techniques are well suited for detecting and quantifying *orbits of symmetry* in correspondence space.

The SFE is defined by using eigenfunctions of the random walk matrix C. We use the normalization suggested by [Nadler et al. 2005] (described in the Appendix for completeness). Denote by ψ_k and λ_k, $k = 1..n$, the eigenvectors and eigenvalues (resp.) of C, that is $C\psi_k = \lambda_k \psi_k$. Note that all eigenvalues of C are bounded between -1 and 1.

The Symmetry Factored Embedding $\Pi^t : \mathcal{X} \to \mathbb{R}^n$ is defined

$$\Pi^t(x_i) = \left(\lambda_1^t \psi_1(x_i), \lambda_2^t \psi_2(x_i), ..., \lambda_n^t \psi_n(x_i)\right), \tag{5}$$

where t is a "time" parameter (we mostly used $t = 20 - 80$). This setting gives more importance to eigenvectors with higher magnitude eigenvalues (hence more symmetry-aware), and ignore the ones with small eigenvalues (less symmetry-aware). Further note that although we pick eigenvectors in eigenspaces possibly with dimensionality larger than one, the SFD is nevertheless invariant to this choice; in other words our construction is dependent only upon the eigenspaces rather a specific (and arbitrary) choice of eigenvectors (see TheoremA.3 in Appendix).

The Symmetry Factored Distance is then defined as the Euclidean distance in the embedded space, that is $\mathbf{d}^t(x_i, x_j)^2 =$

$$\left\|\Pi^t(x_i) - \Pi^t(x_j)\right\|^2 = \sum_{k=1}^n \lambda_k^{2t} \left|\psi_k(x_i) - \psi_k(x_j)\right|^2. \tag{6}$$

4.3 Symmetry Invariant Function space

A consequence of the previous discussion is that we can characterize the space of functions $f : \mathcal{X} \to \mathbb{R}$ defined over the point set $\mathcal{X}$ that are invariant under the (approximated) symmetry. In the perfect symmetric case symmetry invariance means that

$$f(g(x_i)) = f(x_i),$$

for all $i = 1, ..., n$ and every $g \in \mathcal{G}$. In particular, we can take any function $u : \mathcal{X} \to \mathbb{R}$ defined over the point set and "symmetrize" it. That is, making it adhere to the internal symmetry of the point set.

The main idea is that this symmetrization can be done in the framework introduced above. In particular, we show that multiplying powers of the symmetry correspondence matrix by a vector (function) $h \in \mathbb{R}^{n \times 1}$, that is $h_s = C^t h$, will lead to its symmetrization. Basically, the reason is that by multiplying h with the powers of the symmetry correspondence matrix we actually average its values over the paths of the symmetry correspondence graph, dominated by orbits.

More precisely, we will use the following decomposition of C (see Appendix). Denote by $\Psi = (\psi_1, ..., \psi_n)$ the matrix with the eigenfunctions of C as columns, and denote by $\Phi = (\Psi^{-1})^T$ its transpose inverse. We denote the columns of Φ by $\phi_1, ..., \phi_n$. Next, since C is diagonalized by Ψ we have $C = \Psi \Lambda \Psi^{-1} = \Psi \Lambda \Phi^T$, where $\Lambda_{ii} = \lambda_i$ is a diagonal matrix with the eigenvalues at the diagonal. Then in particular

$$C^t = \sum_{k=1}^n \lambda_k^t \psi_k \phi_k^T. \tag{7}$$

When multiplying a vector h with this matrix we get

$$C^t h = \sum_{k=1}^n \lambda_k^t \psi_k \left\langle \phi_k, h \right\rangle. \tag{8}$$

Plugging $t = 0$, we get $h = \sum_{k=1}^n \psi_k \langle \phi_k, h \rangle$ which is basically writing the decomposition of the vector h in the basis of eigenfunctions Ψ. Since $|\lambda_i| \leq 1$ where the symmetric eigenfunctions have eigenvalues close to positive one, and non-symmetric eigenfunctions have eigenvalues distributed around zero (see Section 3), eq. (8) "projects" the function h on the space of symmetry invariant functions. Symmetrization of functions is a useful geometric processing tool for which explore applications in Section 7.

5 Generalizations

The symmetry analysis method described in the previous sections has two independent components: 1) the creation of the symmetry correspondence matrix, C, and 2) the analysis of this matrix based on its spectral properties to quantify orbits. The second component is very general and fits the abstract definition of symmetry as group acting on points sets – i.e., without relying upon the properties of any specific transformation group. The first, however, was described in the previous section specifically to deal with subgroups of rigid motions acting globally on the point set $\mathcal{X}$. It is therefore natural to consider ways to replace the first component with methods that measure different types of symmetry. In this section, we consider two methods, one for analyzing partial extrinsic, and another for global intrinsic symmetries.

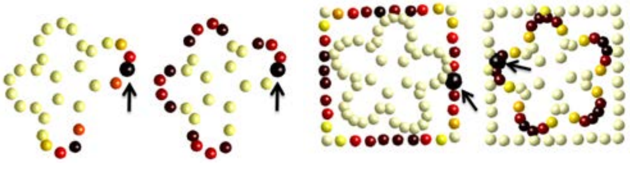
Partial Symmetry There are three main types of partial rigid symmetry: 1) the point set $\mathcal{X}$ is a *subset* of some point set $\mathcal{Y}$ with global extrinsic symmetry, 2) the point set $\mathcal{X}$ *contains* a subset $\mathcal{Y} \subset \mathcal{X}$ which has global extrinsic symmetry, and 3) the point set $\mathcal{X}$ can be subdivided into disjoint sets $\mathcal{X} = \sqcup_\ell \mathcal{Y}_\ell$ such that each of which is satisfying (1) or (2). Obviously, the third case contains the first two.

We treat these cases with a simple modification to $D(\cdot,\cdot)$, the deviation measure between point sets. Basically, we need to design $D(\cdot,\cdot)$ to produce low distance if some sufficiently large part of the point set $\mathcal{X}$ has been matched. That is, we set the distance to zero if some fraction $p \in (0,1]$ of the points in $\mathcal{X}$ has been matched. p can be seen as the prescribed amount of shape that defines symmetry, that is, a rigid transformation g which leaves a p fraction of the points $\mathcal{X}$ close to $\mathcal{X}$ is considered a "symmetry". To achieve this we modify the RMSD as follows: $D_p(\mathcal{X},\mathcal{Y}) =$

$$\left(\frac{\sum_{i \in I_p(x_i,\mathcal{Y})} d(x_i,\mathcal{Y})^2 + \sum_{j \in J_p(y_j,\mathcal{X})} d(y_j,\mathcal{X})^2}{2pn} \right)^{1/2}, \quad (9)$$

where $I_p(x_i,\mathcal{Y}) = \left\{ i : d(x_i,\mathcal{Y}) \le p^{th} \left\{ d(x_i,\mathcal{Y}) \right\}_{i=1}^n \right\}$, where $p^{th} \left\{ d(x_i,\mathcal{Y}) \right\}_{i=1}^n$ denotes the $p-$th percentile of the set $\left\{ d(x_i,\mathcal{Y}) \right\}_{i=1}^n$, and $J_p(y_j,\mathcal{X})$ is defined similarly.

The image below (a) demonstrates the tradeoff between different values of p, in partial $2D$ flower point set. The SFD (measured from the arrow marked points) based on using $D_p(\cdot,\cdot)$ with $p = 1$ is shown on the left, results in discovering the *global* reflective symmetry. On the other hand taking $p = 0.4$ (right) reveals that this shape can be seen as a complete $5-$fold flower with a missing part. In (b), using $p = 0.4$ again, we show that a point set which has subsets with *different* symmetry groups (type 3 above) is getting separated using the SFD (measured from the marked points). This suggests that the SFD could be used to perform complex segmentation of shapes/images that otherwise would not be possible.



(a) (b)

Global Intrinsic Symmetry When working in correspondence space, it is also possible to think about the process of finding orbits without a specific notion of a transformation group. In this section, we demonstrate this idea for the global intrinsic ("non-rigid") symmetry case [Ovsjanikov et al. 2008; Raviv et al. 2007; Bronstein et al. 2009]. In particular, we show that it is possible to take existing intrinsic symmetry descriptors and improve their performance using the previously described "diffusion of symmetry". Given a manifold mesh $M = (\mathcal{X}, E, F)$, with $\mathcal{X}$ as its vertices, E edges set, and F faces, we use the symmetry-invariants as implied by Ovsjanikov et al. [2008], namely the pointwise absolute values of the Laplacian eigenvectors (assuming non-repeating eigenvalues). That is, if we denote by $u_1, ..., u_K$ the first K eigenvectors of the cotangent Laplacian on meshes [Pinkall and Polthier 1993], the global intrinsic symmetry dissimilarity measure is then

$$S_{ij}^2 = \sum_{k=1}^{K} \left(|u_k(x_i)| - |u_k(x_j)| \right)^2. \quad (10)$$

We can now use this dissimilarity matrix as explained in Section 4. Figure 4 shows the first (i.e., most dominant) two eigenvectors of the Laplacian (left) and the symmetry correspondence matrix C (right). Note that the latter integrates the information from the Laplacian's eigenvectors and better respect the approximate intrinsic reflective symmetry of the model (see for example the arrow marked area).

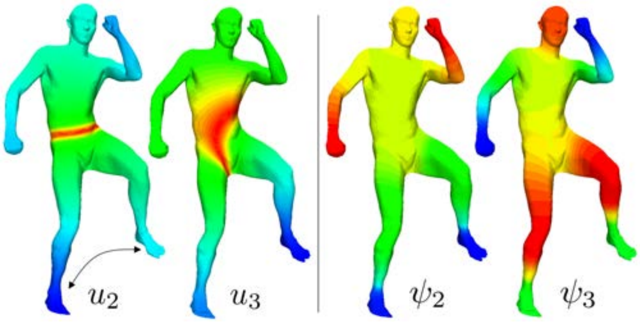


Figure 4: *Comparison between the first two eigenvectors of the Laplacian (left) and the first two eigenvectors of the symmetry correspondence matrix C (right). The color map is normalized to better show values near zero (in dark blue).*

6 Results

To investigate the properties of the Symmetry Factored Embedding/Distance, we ran a series of experiments with different types of inputs and made comparisons to alternative approaches. The experiments were done on a 2.2GHz Opteron 275 processor. All meshes/point clouds were first subsampled to around $1k-2k$ points using an iterative algorithm which iteratively picks the farthest (in Euclidean sense) point in $\mathcal{X}$ from the sample set and adds it to the set. Once the SFD is computed on the sample set we transfer it back to the original point set using Shepard's interpolation scheme.

Complex symmetry groups Our first experiment focuses on the ability of the Symmetry Factored Distance (SFD) to characterize orbits in complex symmetry groups comprising of rigid transformations. Figure 5 shows two representative examples. In this image (and several others in the paper), the SFD is measured from one source point (black arrow) to all others, where dark blue represents zero distance, dark red represents the largest distance, and other distances are normalized to exaggerate distances around zero. On the left, the virus has icosahedral symmetry (a symmetry group of size 120), while the mechanical part on the right has a type of dihedral symmetry (symmetry group of size 12). Note how the SFD correctly reveals the orbits for the source points even in these complex symmetry groups.

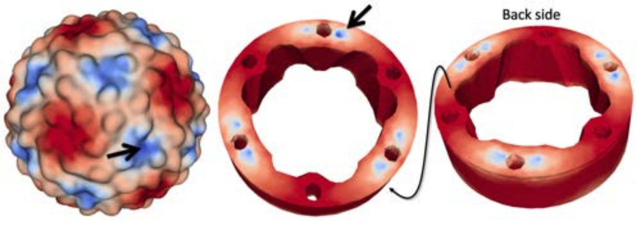


Figure 5: *The SFD visualized on two models with global extrinsic symmetry.*

Robustness to noise Our second experiment investigates the robustness of our method to noise. We computed the SFD from a point on a model perturbed by increasing amounts of noise and visualized the deviation as the amount of noise increases. Figure 7 shows an example result for a cube. Each row shows the SFD from

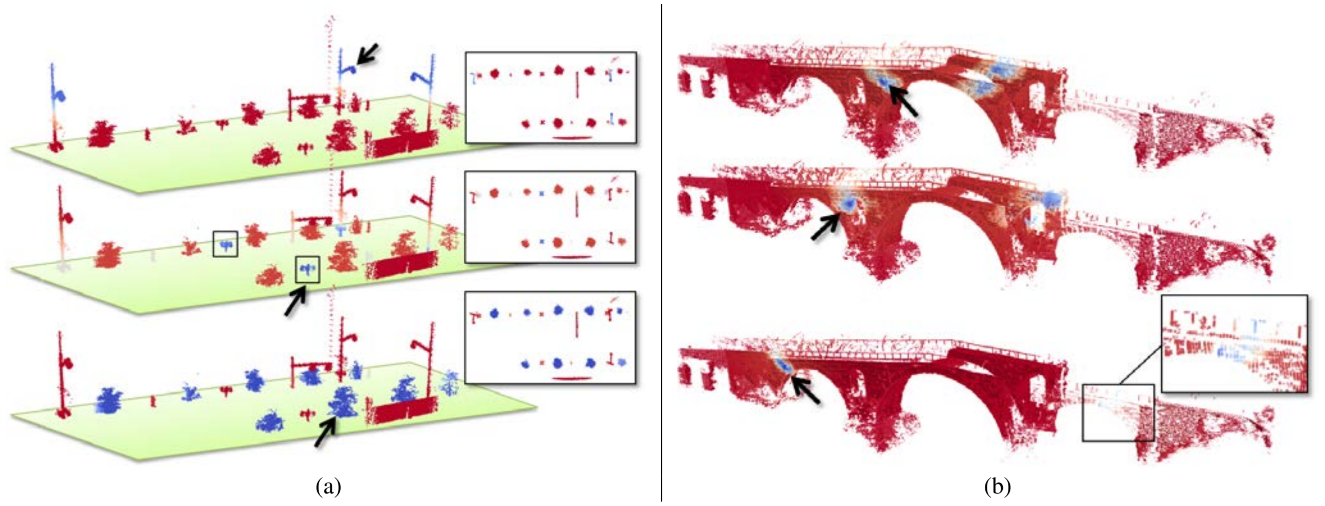


Figure 6: *The SFD visualized on two models with partial extrinsic symmetry.*

a fixed source point (marked on the leftmost cube) through increasing levels of noise. Note that although this example seems simple, the symmetry group of the cube is octahedral, and as such, it comprises 48 distinct transformations. Yet, our method can find orbits on the cube even with large amounts of noise.

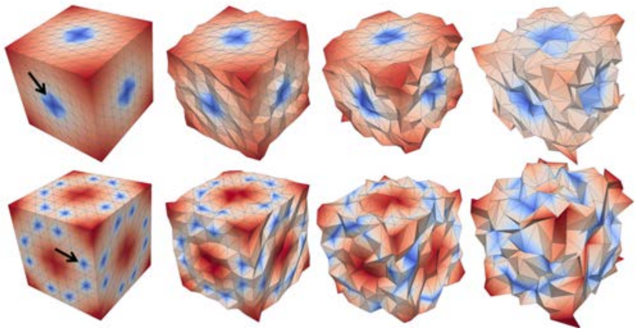


Figure 7: *The SFD visualized on a series of cubes with increasing noise level.*

Partial extrinsic symmetry Our third experiment investigates computation of the SFE and SFD for point sets with partial extrinsic symmetry. The SFD is shown for several examples with missing data and extra parts in Figures 1, 8, and 6. Figure 1 (bottom row) shows a noisy scan of a flower statue in which the SFD characterizes the 9-fold symmetry of the petals despite missing data and extraneous parts. Figure 8 shows three more examples: a) a vase model with a bottom part that is (approximately) 4-fold symmetric and a top part that is only reflective symmetric, b) a 6-fold symmetric mechanical part with a missing piece (note how the SFD still recovers the existing part of the orbit correctly), and c) an aerial LIDAR scan of cars in a row where two cars are sampled very sparsely and the trees and ground are extraneous parts. In the latter case (c), note that observing only local properties of the marked cars (on the right) is probably not enough to recognize them as cars or to associate them with the rest of cars in the row due to poor sampling – symmetry is an important cue here.

Figure 6 shows more difficult examples. The left part of the figure (a) shows objects extracted from a LIDAR scan of a street scene (the ground plane is shown for visualization only, and the inset views show the scene from the top). It contains both partial translational and reflectional symmetries between semantically similar objects, but no segmentation nor tranformation group is provided to the system (besides an up-vector for the ground plane). The SFDs from a point on a traffic light, a fire hydrant, and a bush are shown. Note that difficult elements to recognize like small and poorly-sampled water hydrants (marked with black squares in the middle image) are correctly clustered together due mainly to the strong symmetry detected in the scene. Figure 6 (b) shows a LIDAR scan of a bridge with D_2 symmetry (reflection and rotation by 180° though its middle), non-uniform sampling, noise and missing parts. In spite of these problems the SFD captures the approximate notion of symmetric orbits very well – note the repeating "blue" elements with SFD close to the three different selected source points (marked with a black arrow), found even when the point cloud is very poorly sampled (as shown in the bottom right inset). These examples provide evidence for the robust nature of the Symmetry Factored Embedding in the presence of noisy and partial data.

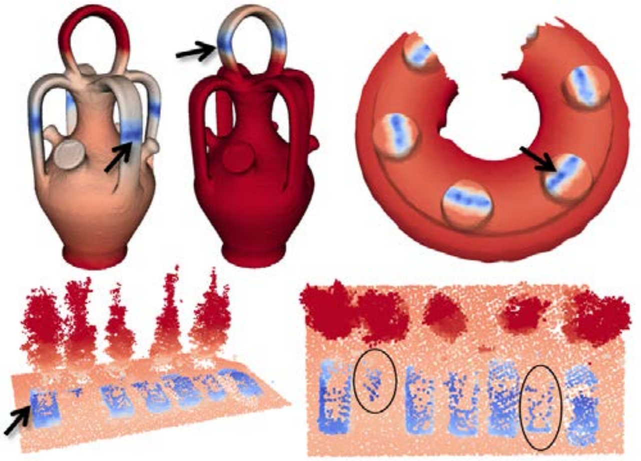


Figure 8: *The SFD visualized on a group of models with partial extrinsic symmetry.*

Global intrinsic symmetry Next, we show the SFD computed for models with non-rigid deformation and compare them to the dissimilarities computed using the Laplacian eigenvectors. In each image pair of Figure 9, we show S_{ij} as computed from eq. (10) on the left, and we show the SFD on the right. Note that the SFD better clusters the symmetric points with respect to the approximate intrinsic bilateral reflection symmetry.

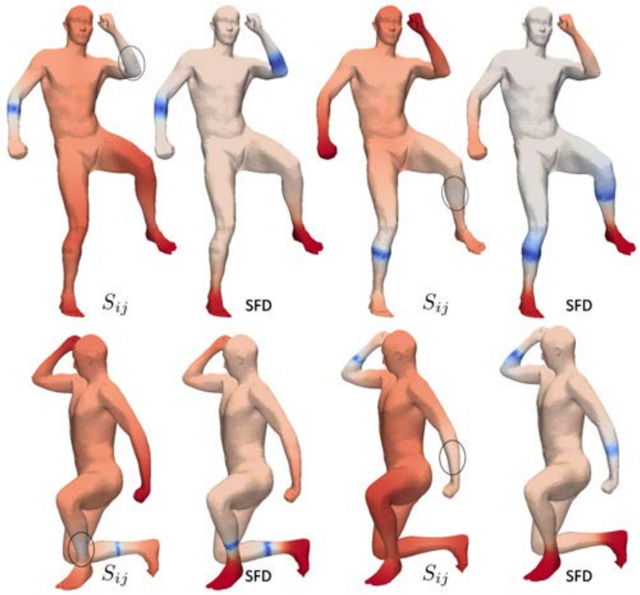
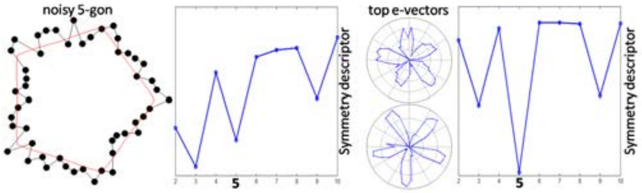
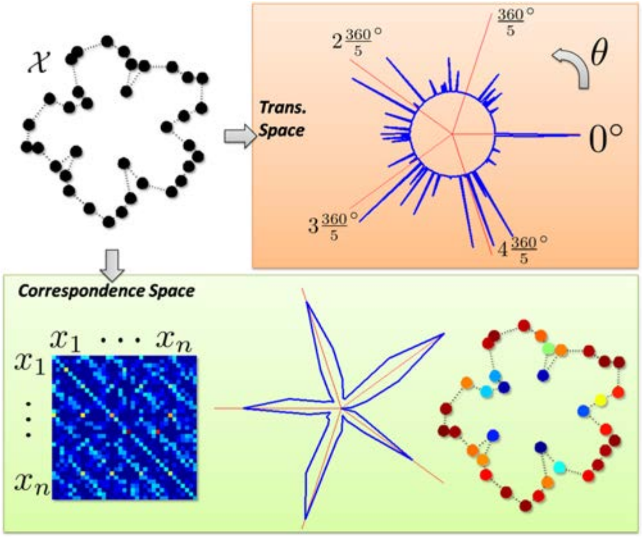


Figure 9: *Comparison between using the symmetry invariants S_{ij} and the SFD to find intrinsic correspondences (orbits) in human models.*

Comparison to methods based on correlation The next experiment evaluates the robustness of our method to noise in comparison to the algorithm presented by Kazhdan et al. [2004]. Kazhdan's algorithm takes as input a function over the sphere and provides a symmetry distance for every k, indicating "how symmetric" the function is with respect to k-fold rotations. The computation of this distance is based on autocorrelation and thus it should be robust to noise. To make a comparison with this method, we take the 2D sphere (i.e., circle) and generate a very noisy 5-gon parameterized over it. The figure below shows the noisy point set (in red is the ground-truth 5-gon), along with a graph (second from the left) presenting Kazhdan's symmetry descriptor for $k = 2, 3, ...10$. As seen, using their algorithm for detecting the symmetry will yield 3-fold as the most likely symmetry in the data. However, using our algorithm, that is looking at the top eigenvectors of the symmetry correspondence matrix (the top two are shown in polar plots), the 5-fold symmetry is more apparent. If we use Kazhdan's method to search for k-fold symmetries in the top non-constant eigenvector produced by our method, the 5-fold symmetry is easily detected, as seen in the global minimum in the rightmost graph.



Comparison to methods based on voting It is also instructive to compare our method to alternative approaches that detect symmetries in transformation space, in this case a simple method that accumulates votes for rotational symmetries. The figure below shows the result of performing a few iterations of the voting algorithm described in Section 4.1, casting the *same* "votes" both in the transformation space (in this case 2D rotations parameterized over the circle), and correspondence space. The density map in transformation space should contain five clusters as the point-set has approximated 5-fold symmetry (we disregarded reflections in this example). However, it is clear that at least one rotation is missed (by $360/5°$) by the votes cast. Furthermore, the other transformations are seen only vaguely (the red lines show the ground truth). On the other hand, the top non-constant eigenvector (bottom middle and right) of the symmetry correspondence matrix (bottom left) still robustly detects the 5-fold symmetry. Finding orbits is generally more robust than finding transformations since orbit clustering exploits consistency relations between found transformations to recover missed transformations.



Comparison to direct analysis of the dissimilarity matrix Another possible alternative would be to directly use the dissimilarity measures S_{ij} produced by our voting algorithm to estimate the "symmetry factored distance" between pairs of points. In the perfect case, like the one presented in Section 3, this approach may suffice. However, in cases of imperfect symmetries, the symmetry diffusion provided by our method is important to robustly integrate the "observations" into a consistent set of symmetry orbits. This is demonstrated in Figure 10 for two examples: the flower model and a row of cars acquired with a LIDAR scanner. We show a side by side comparison of taking the dissimilarity measure S_{ij} (left) with the SFD (right) from the same source point (black arrows). Note how the SFD better models the close symmetry relationship between all the cars, even the ones that are poorly sampled by the LIDAR scanner.

Timing and complexity In this subsection we provide theoretical analysis of our algorithm and report running times. We specifically analyze the complexity of the voting algorithm used to approximate the matrix S for global extrinsic symmetries, which dominates the overall computational complexity of that implementation. Similar analysis can be performed for the case of partial extrinsic symmetries. Afterwards, we discuss the complexity of the spectral analysis.

The voting algorithm as described in Section 4.1 takes $O(n \log n)$ for each iteration (vote), assuming a spatial data structure (kd-tree) is used for closest point query. As to the number of votes required, since consistency of orbits is taken care of by the spectral analysis, it is enough to reveal a *generating set* of transformations (that is, a set of transformation which can build any transformation in the symmetry group). The probability of getting the randomized two pairs of points revealing some generating transformation is proportional to $k/n\sqrt{n}$ on average, where k is the averaged orbit size generated by the subgroup defined by that generating transformation: the probability of getting the first point in the second pair right is

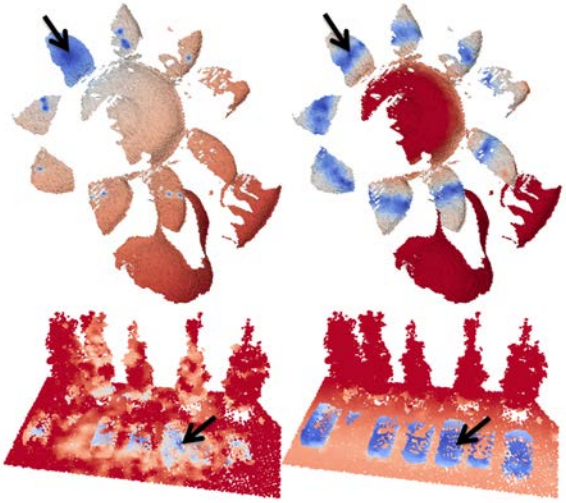


Figure 10: *Comparing the dissimilarity measure S_{ij} (left) to the SFD (right).*

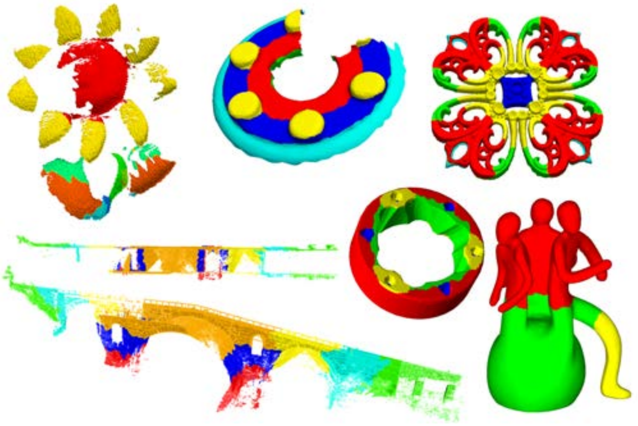


Figure 11: *Symmetry-aware segmentation. We use the SFE with the basic k-means algorithm.*

k/n on average, since there are k (on average) repeating appearances of the first point in the first pair, and the probability of getting the second point right assuming the first is right is proportional to $1/\sqrt{n}$ on average (due to the requirement that the distance between the pairs is approximately equal). The overall complexity of approximating S_{ij} is then $O(n^2\sqrt{n}\log n)$, where the constant can be bounded *independently* of the symmetry group type, or even better, depends on the reciprocal of the minimal average orbit size of the generating set's subgroups. So the higher the symmetry - the better.

As to practical running times, we mostly ran our algorithm $cn\sqrt{n}$ iterations based on the analysis above, where $c = 1 - 3$. The table below lists observed running times for some of the examples in the paper.

Model	pnts	time	Model	pnts	time
Cube	386	29s	Julius	1000	231s
Bridge	1000	258s	Flower	1000	290s
Chair	1000	358s	Dragon	1000	850s
Street	1000	441s	Horse	1000	920s
Jewelery	1200	893s	Mech.1	1200	750s
Cars	1200	675s	Mech.2	1500	852s
Cryo EM	2194	1774s	Virus	2000	1052s

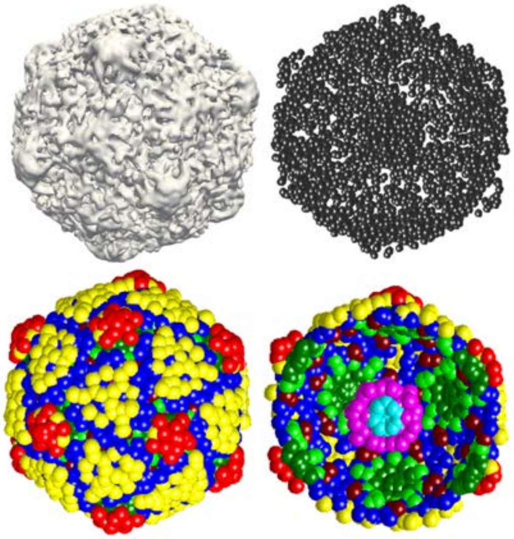


Figure 12: *Detecting symmetry in Polio-virus through clustering Cryo-EM data in Symmetry Factored Embedding space.*

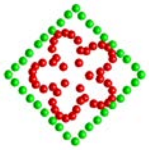
For the spectral analysis of the matrix C it is enough to take the first few eigenvectors due to the spectral gap. Since the matrix C is usually sparse (orbit sizes are finite and much smaller than n) the eigen-analysis of the matrix C can be done in time $O(n^{1.5})$ in number of samples. Except for the non-rigid case, we have not exploited this property, since calculating the first 100 eigenvectors in 1000×1000 matrix takes around 5 seconds in Matlab, which is negligible in comparison to the time complexity of approximating S.

7 Applications

In this section, we explore two shape analysis applications of the methods presented in Section 4: symmetry-aware segmentation of point sets and approximating the locus of stationary points.

Symmetry-aware segmentation Symmetry is an important cue for segmentation of shapes into functional parts. It has been used previously by [Podolak et al. 2006] to decompose meshes. However, their method focused specifically on planar reflective symmetries and would not extend well to handle arbitrary symmetries.

In this section, we investigate the idea of using the SFD to facilitate symmetric segmentation of meshes. The main observation is that the distances between points in symmetric orbits are reduced in the Symmetry Factored Embedding space (zero in the case of perfect symmetry), and so clustering points in that space should produce decompositions with symmetric points in the same cluster. To experiment with this idea, we execute a standard k-means clustering algorithm on points in the embedded space $\Pi(\mathcal{X}) = \{\Pi(x_i)\}_{i=1}^{n}$ and then decompose the input based on the resulting clusters (we smooth the index function to get slightly smoother boundaries of clusters).



The inset figure shows a simple example for a 2D point set with $k = 2$ (two clusters). Note the different symmetry groups of each cluster. More complex 3D examples are shown in Figure 11. There are cases with partial symmetry (bridge, flower, mechanical part in top row) and complex symmetry (jewelry, mechanical part in middle row). Figure 12 shows the re-

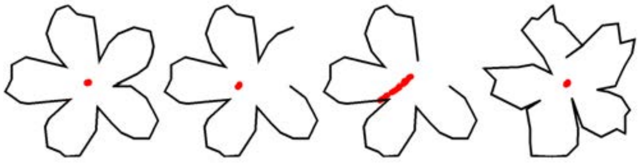


Figure 13: *Finding stationary points in 2D star shape. We use the functional symmetrization procedure on the coordinate functions to robustly locate the set of stationary points (in red). Note that the procedure automatically reveals changes in global symmetry (5-fold to line reflection), and is stable to noise.*

sult of segmentation of virus density map based on the Symmetry Factored Embedding: given a Cryo-EM volumetric density map of the polio-virus (isosurface extraction shown on the top-left), we identify local maximas (top-right), and cluster them based on a k-means algorithm in the SFE space to reveal the icosahedral symmetry (bottom-left). The image on the bottom-right shows a cut through the middle of the virus point set to reveal its internal symmetry. These examples all depend on characterization of complex, partial, and/or approximate symmetries that would be hard to detect with other methods.

Stationary points of extrinsic symmetry Stationary points of extrinsic symmetry are points $q \in \mathbb{R}^d$ left fixed by the symmetry transformations of $\mathcal{X}$, that is $g(q) = q$ for all $g \in \mathcal{G}$. They provide important information about the structure of a shape and can be used for recognition, alignment, and symmetrization of 3D meshes.

We find that detection of stationary points is straight-forward in our framework. The basic observation is that centroid of an orbit of finite symmetry group is a stationary point of symmetry [Zabrodsky et al. 1993]. Let $x \in \mathcal{X}$ be an arbitrary point, then the centroid q of the orbit of x, that is the set $\{gx \mid g \in \mathcal{G}\}$, is $q = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} gx$ (see Lemma A.4 in Appendix). Then, for all $\widehat{g} \in G$ we have

$$\widehat{g}q = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}(\widehat{g}g)x = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} gx = q,$$

where the first equality is due to linearity of the operator $\widehat{g}$ and the second equality is due to the fact that $\widehat{g}\mathcal{G} = \mathcal{G}$.

This suggests that finding stationary points can be done by averaging the coordinates of orbits. This can be done robustly in the framework introduced in Section 4.3, utilizing the symmetrization of functions defined over the point set $\mathcal{X}$. In particular, if we assume $\mathcal{X}$ is written as a matrix $\mathbb{R}^{n\times d}$ where each row i contains the coordinates of x_i w.r.t. some global coordinate system, then multiplying $\mathcal{X}$ with C^t, that is $C^t\mathcal{X}$, averages $\mathcal{X}$ over the orbits. This can be seen, for example, in the perfect symmetric case where the non-zero eigenvectors ψ_k and the corresponding ϕ_k are constant on orbits (ϕ_k are left eigenvector and in the perfect case $\phi_k = \psi_k$). Therefore, eq. (8) shows that $C^t\mathcal{X}$ equals the piecewise constant vector which is constant on every orbit and equals the orbit's centroid.

Figure 13 shows the symmetrized coordinate functions $C^t\mathcal{X}$ as red points, for four 2D contours. Note, the robustness of their positions in the center of the flower, despite noise and deviation from perfect 5-fold symmetry in the input (leftmost and rightmost). In the second example, a piece of the contour has been removed, but the stationary point stays approximately the same. When removing a larger part of the input in the third image, a reflective symmetry becomes the dominant symmetry of the shape and the distribution of the red points becomes a line.

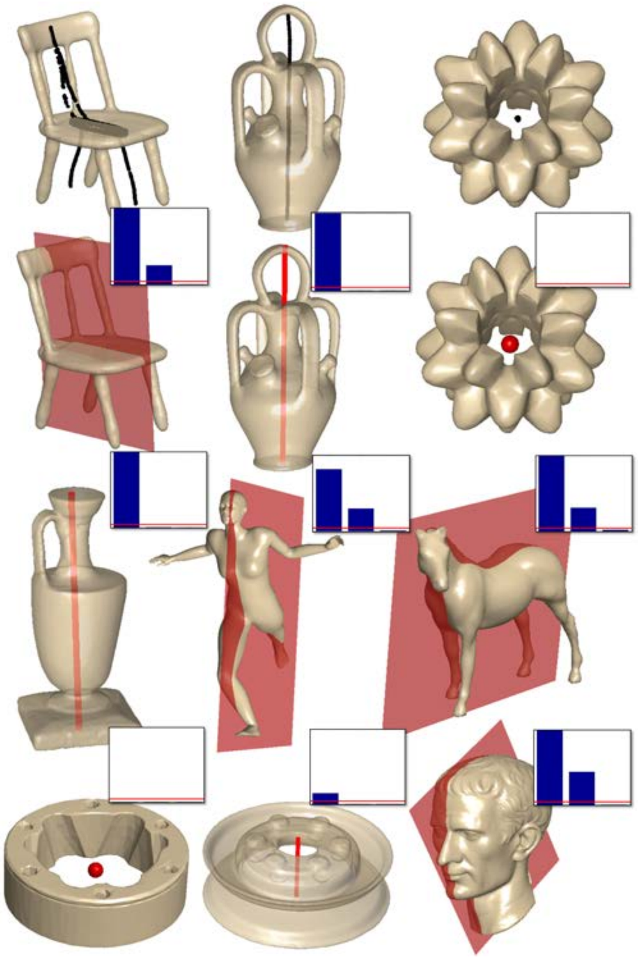


Figure 14: *Finding approximate stationary locus of symmetry for a collection of 3D models.*

Figure 14 shows coordinate symmetrization for stationary points approximation in 3D. The top row shows the stationary points in black. In the other rows, we demonstrate how the stationary points can be used to automatically identify the stationary set *type* of an object (point, line, plane, or none). Our procedure is quite simple: we first scale the models' diameter to one, then, once the coordinate functions of the point set has been symmetrized, that is $\mathcal{X}_s = C^t\mathcal{X}$, we perform a Principle Component Analysis (PCA) on the set $\mathcal{X}_s$, and analyze the sizes of the three eigenvalues with respect to a threshold $\epsilon = 0.001$ – if there is only one above threshold, then the stationary set is a line, if two - a plane, and if none - a point. If there are three eigenvalues above the threshold, we can say that the point set is "asymmetric." These cases are shown in different examples of Figure 14 – eigenvalues are shown as bar charts next to each figure, and $\epsilon = 0.001$ is visualized as red line in those plots. Note, for example, that the mechanical part (bottom-left) has point (not line) symmetry, a fact which is rather delicate to discover with other methods.

Lastly, Figure 15 shows a combination of the symmetry-aware clustering application described above and stationary point approximation described in this section: we first segment the shape by clustering orbits, and then we find the stationary points for each part and fit a primitive (point, line, or plane) with the appropriate dimensionality. This analysis provides a concise description of the shape parts based on symmetry.

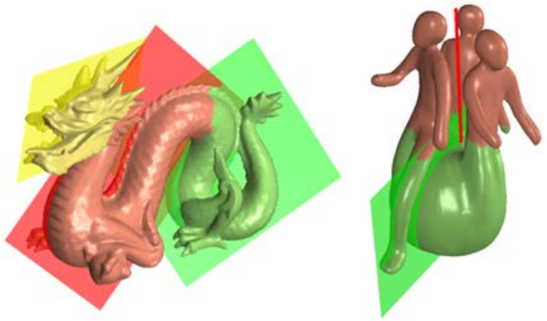


Figure 15: *Incorporating symmetry-aware segmentation and stationary points extraction for decomposing models to symmetric part.*

8 Conclusions, limitations, and future work

In this paper, we introduce the Symmetry Factored Embedding (SFE) and the Symmetry Factored Distance (SFD). The main observation is that analyzing symmetries in correspondence space with spectral analysis provides a robust way to detect and quantify symmetry in point sets. During experiments, we find that the SFE and SFD can characterize complex symmetry groups, provide robustness to noise, and be applied on data sets with partial symmetries and non-rigid deformations. Moreover, they can be used to segment symmetric objects and identify stationary points.

This work is just a first step, and it has several limitations. For example, our current measure of deviation for partial extrinsic symmetry does not directly enforce that continuous parts of the object align with one another, but rather only that a prescribed fraction aligns. This formulation can cause the algorithm to neglect small, but important, parts of the shape (like two small bumps on a sphere). This problem could be alleviated by considering more sophisticated dissimilarity measures when constructing S. Another issue relates to the setting of parameters. Our system has three: 1) σ separates different orbits – it intuitively defines how much we "trust" the values of the dissimilarity measure (like a smooth truncating function), 2) t prescribes the time of diffusion, or equivalently deciding which eigenvectors should be used according to the magnitude of their eigenvalues, and 3) p indicates the fraction of the object that should be aligned by a partial symmetry. In our current implementation, we set them manually. An interesting topic of future work would be to derive theoretical or practical ways to either avoid them or set them automatically.

In terms of future work, there are several ways the ideas in this paper could be extended. First, on the theoretical level, it seems that new shape measures could be defined in terms of the matrix C and its spectral properties, like quantifying *how* symmetric a shape is without prescribing a transformation group. Second, in terms of algorithms, it seems natural to allow attributes associated with input points of $\mathcal{X}$ to be considered when measuring the deviation between points. Lastly, we believe this framework can be used in order to solve for correspondences between *different* shapes with symmetry.

References

BELKIN, M., AND NIYOGI, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, MIT Press, 585–591.

BERNER, A., BOKELOH, M., WAND, M., SCHILLING, A., AND SEIDEL, H.-P. 2008. A graph-based approach to symmetry detection. In *IEEE/EG Symposium on Volume and Point-Based Graphics*.

BOKELOH, M., BERNER, A., WAND, M., SEIDEL, H.-P., AND SCHILLING, A. 2009. Symmetry detection using line features. *Computer Graphics Forum (Eurographics) 28*, 2, 697–706.

BRONSTEIN, A., BRONSTEIN, M., BRUCKSTEIN, A., AND KIMMEL, R. 2009. Partial similarity of objects, or how to compare a centaur to a horse. *Int. J. Comp. Vis. 84*, 2, 163–183.

CHERTOK, M., AND KELLER, Y. 2010. Spectral symmetry analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear*.

COIFMAN, R. R., LAFON, S., LEE, A. B., MAGGIONI, M., NADLER, B., WARNER, F., AND ZUCKER, S. W. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences 102*, 21 (May), 7426–7431.

GOLD, S., RANGARAJAN, A., PING LU, C., AND MJOLSNESS, E. 1998. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition 31*, 957–964.

HAYS, J., LEORDEANU, M., EFROS, A. A., AND LIU, Y. 2006. Discovering texture regularity as a higher-order correspondence problem. In *9th European Conference on Computer Vision*.

IMIYA, A., UENO, T., AND FERMIN, I. 1999. Symmetry detection by random sampling and voting process. In *CIAP99*, 400–405.

KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. A reflective symmetry descriptor for 3D models. *Algorithmica 38*, 1 (Oct.).

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3D shape matching. In *Symposium on Geometry Processing*.

LEORDEANU, M., AND HEBERT, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, 1482–1489.

LEUNG, T., AND MALIK, J. 1996. Detecting, localizing and grouping repeated scene elements from an image. In *European Conference on Computer Vision*, 546–555.

LI, W., ZHANG, A., AND KLEEMAN, L. 2005. Fast global reflectional symmetry detection for robotic grasping and visual tracking. In *ACRA05*.

LI, M., LANGBEIN, F., AND MARTIN, R. 2006. Constructing regularity feature trees for solid models. *Geom. Modeling Processing*, 267–286.

LIPMAN, Y., AND FUNKHOUSER, T. 2009. Mobius voting for surface correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (Aug.).

LIU, S., MARTIN, R., LANGBEIN, F., AND ROSIN, P. 2007. Segmenting periodic reliefs on triangle meshes. In *Math. of Surfaces XII*, Springer, 290–306.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics*, vol. 25, 560–568.

Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. 2005. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *ArXiv Mathematics e-prints*.

Ovsjanikov, M., Sun, J., and Guibas, L. 2008. Global intrinsic symmetries of shapes. *Computer Graphics Forum (Symposium on Geometry Processing) 27*, 5, 1341–1348.

Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., and Guibas, L. 2008. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics 27*, 3, #43, 1–11.

Pinkall, U., and Polthier, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 15–36.

Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T. 2006. A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3 (July).

Raviv, D., Bronstein, A., Bronstein, M., and Kimmel, R. 2007. Symmetries of non-rigid shapes. In *Int. Conf. on Comp. Vis.*

Reisfeld, D., Wolfson, H., and Yeshurun, Y. 1995. Context-free attentional operators: The generalized symmetry transform. *IJCV 14*, 2, 119–130.

Rustamov, R. 2008. Augmented planar reflective symmetry transform. *The Visual Computer 24*, 6, 423–433.

Shi, J., and Malik, J. 1997. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 888–905.

Shikhare, D., Bhakar, S., and Mudur, S. 2001. Compression of large 3d engineering models using automatic discovery of repeating geometric features. 233–240.

Xu, K., Zhang, H., Tagliasacchi, A., Liu, L., Li, G., Meng, M., and Xiong, Y. 2009. Partial intrinsic reflectional symmetry of 3d shapes. *ACM Transactions on Graphics (SIGGRAPH ASIA) 28*, 5.

Yip, R. 2000. A hough transform technique for the detection of reflectional symmetry and skew-symmetry. *PRL 21*, 2, 117–130.

Zabrodsky, H., Peleg, S., and Avnir, D. 1993. Continuous symmetry measures, ii: Symmetry groups and the tetrahedron. *Journal of the American Chemical Society 115*, 8278–8289.

A Appendix

We prove several properties related to the constructions described in the paper. We start with the eigen-analysis of the symmetry correspondence matrix:

Theorem A.1. *The eigen-structure of the (not row normalized) symmetry correspondence matrix C in the case of perfect symmetry has the following properties: 1) every orbit correspond to a single non-zero eigenvalue λ_k, $k = 1..K$. 2) The magnitude of each eigenvalue λ_k equals the size of its corresponding orbit. 3) To each eigenvalue λ_k there corresponds an eigenvector ψ_k which is constant on the corresponding orbit and zero elsewhere. 3) the rest of the eigenvalues $\lambda_{K+1}, ..., \lambda_n$ are all zero. Lastly, the row-normalized correspondence matrix has the same eigenvectors, however the non-zero eigenvalues are all one.*

Proof. The proof is simple and relay on the observation that in the perfect symmetry case the symmetry correspondence matrix can be rearranged to be of perfect block structure. This rearrangement is done by conjugating the matrix with a permutation matrix P which puts the indices of orbital points together: $\widehat{C} = P^T C P$. Therefore, we can assume the matrix C is already in block structure. Every block of the matrix corresponds to an invariant subspace, so we can analyze each block independently. Every block (all ones matrix) has a single constant eigenvector of eigenvalue equal the size of the block (orbit), and all the rest of the eigenvalues are zero. Taking into account that every block corresponds to one orbit, the result follows. The row-normalized case can be treated similarly. □

In this paper we use the normalization suggested by [Nadler et al. 2005] to the eigenvectors of the random-walk matrix C: the matrix C is similar to the symmetric matrix $C_s = D^{1/2} C D^{-1/2}$, where $D_{ii} = \sum_j \widetilde{C}_{ij}$ is the diagonal matrix with the sum of rows of $\widetilde{C}$ in the diagonal. Therefore it has orthogonal decomposition $C_s = U \Lambda U^T$ and it has the same eigenvalues as C, therefore,

$$C = D^{-1/2} C_s D^{1/2} = \left(D^{-1/2} U\right) \Lambda \left(D^{-1/2} U\right)^{-1}.$$

From this last equation we see that $\Psi = D^{-1/2} U$ consists of columns (right) eigenvectors of C. Note that these vectors are orthonormal w.r.t the inner product defined by D, that is $\Psi^T D \Psi = I_d$. The reason Nadler et al. [Nadler et al. 2005] performs this normalization is to make the Euclidean distance in the diffusion map space equal the diffusion distance. In our context this implies that the SFD can be interpreted as diffusion distance (the proof repeats Nadler's arguments):

Theorem A.2. *The symmetry factored distance can be seen as diffusion distance in the symmetry correspondence graph.*

Next we prove that the symmetry factored distance is invariant to specific choice of eigenvectors. In other words, that it only depends on the eigenspaces of C.

Theorem A.3. *The symmetry factored distance depends only on the eigenspaces.*

Proof. Let $\Theta = \Psi M$ be a different choice of eigenvectors of C with the same eigenvalues that is $C\Theta = \Theta\Lambda$. Then M has to be block (of sizes matching the dimensions of eigenspaces) orthogonal matrix since $I_d = \Theta^T D \Theta = M^T \Psi^T D \Psi M = M^T M$. The SFD $\mathbf{d}^t(x_i, x_j)$ is the Euclidean norm of the difference between the i^{th} and the j^{th} rows of the matrix $\Psi \Lambda^t$. On the other hand when using the basis Θ we compare the rows of the matrix $\Psi M \Lambda^t$. Since every eigenspace is left invariant by M, and the eigenvalue is fixed on every eigenspace the part of the SFD contributed from every eigenspace is the same using the two bases. □

Lastly we prove the formula for the centroid of an orbit used in Section 7:

Lemma A.4. *For a discrete symmetry transformation group $\mathcal{G}$, the centroid of an orbit of a point x is given by the formula $q = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} gx$.*

Proof. Denote by $\mathcal{G}_x$ the *stabilizer* of the point x, that is the subgroup $\mathcal{G}_x = \{h \in \mathcal{G} \mid hx = x\}$. Write $\mathcal{G}$ as union of its left cosets w.r.t $\mathcal{G}_x$, that is $\mathcal{G} = \cup_{i=1}^{L} g_i \mathcal{G}_x$, where $L = |\mathcal{G}| \,/\, |\mathcal{G}_x|$, $\{g_i x\}_{i=1}^{L}$ is the orbit of x and $g_i x \neq g_{i'} x$. Then

$$\sum_{g \in \mathcal{G}} gx = \sum_{i=1}^{L} \sum_{h \in \mathcal{G}_x} g_i(hx) = \sum_{i=1}^{L} g_i x \, |\mathcal{G}_x| \,,$$

and therefore

$$\frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} gx = \frac{|\mathcal{G}_x|}{|\mathcal{G}|} \sum_{i=1}^{L} g_i x = \frac{1}{L} \sum_{i=1}^{L} g_i x.$$

□