Eurographics Symposium on Geometry Processing 2012
Eitan Grinspun and Niloy Mitra
(Guest Editors)

Volume 31 (2012), Number 5

Finding Surface Correspondences Using Symmetry Axis Curves

Tianqiang Liu, Vladimir G. Kim, and Thomas Funkhouser

Princeton University

Abstract

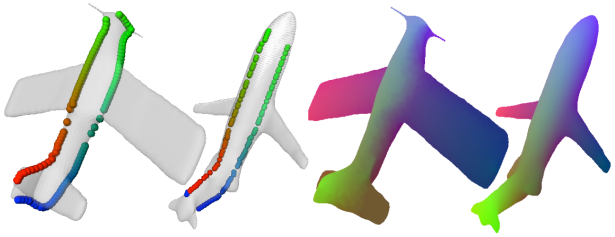
In this paper, we propose an automatic algorithm for finding a correspondence map between two 3D surfaces. The key insight is that global reflective symmetry axes are stable, recognizable, semantic features of most real-world surfaces. Thus, it is possible to find a useful map between two surfaces by first extracting symmetry axis curves, aligning the extracted curves, and then extrapolating correspondences found on the curves to both surfaces. The main advantages of this approach are efficiency and robustness: the difficult problem of finding a surface map is reduced to three significantly easier problems: symmetry detection, curve alignment, and correspondence extrapolation, each of which has a robust, polynomial-time solution (e.g., optimal alignment of 1D curves is possible with dynamic programming). We investigate of this approach on a wide range of examples, including both intrinsically symmetric surfaces and polygon soups, and find that it is superior to previous methods in cases where two surfaces have different overall shapes but similar reflective symmetry axes, a common case in computer graphics.

Categories and Subject Descriptors (according to ACM CCS):

1. Introduction

Finding semantically meaningful correspondences between points on different (non-isometric) surfaces is a fundamental problem in computer graphics with applications in morphing [Ale01], attribute transfer [KS04], and shape database analysis [ACP03]. For most of these applications, the goal is to find the map $f : S_1 \rightarrow S_2$ between surface, S_1 and S_2, that best aligns "semantically equivalent" points (e.g., when given two human bodies, the map takes a point on the right knee of one body to the equivalent point on the right knee of the other, etc.).

Although there are many good methods to produce a map with minimal distortion between two surfaces once a sparse set of feature correspondences has been found or given by a user ([Ale01]), it is still quite challenging to find a sparse set of feature correspondences completely automatically. Traditional methods detect a set of local features on the two surfaces and then perform a combinatorial search of potential correspondences between them, computing for each an estimate of the distortion induced by the deformation aligning them (e.g., [ZSCO*08]). This approach is both time-consuming and error-prone when the surfaces have significantly different local features (e.g., two *different* shapes within the same object class, as shown in Figure 1a).



a) Symmetry Axis Curves b) Surface Correspondences

Figure 1: Aligning symmetry axis curves helps find surface correspondences. Given two 3D models, we extract symmetry axis curves from both, align the curves (left), and then extrapolate the axis correspondences to produce a full surface map (right). In these images, points with the same color are predicted correspondences.

The general problem is to discover stable, semantically-equivalent features to match on significantly different surfaces.

In this paper, we investigate methods based on symmetry analysis. Our key ideas are based on three observations: 1) most objects in the real world have an extrinsic and/or intrin-

sic reflective symmetry, at least approximately; 2) finding the symmetry axis curve of a surface (the set of surface points stationary in a symmetric map) is relatively easy, especially for global symmetries; and 3) searching for corresponding features on the 1D symmetry axis curves is significantly easier than on the 2D surfaces. These observations combine to suggest an approach that leverages a symmetry axis curve detection algorithm to find surface correspondences. Essentially, the reflective symmetry axis curve serves as a global feature that can be detected and aligned robustly and efficiently, thereby simplifying the search for correspondences. Once a sparse set of correspondences are found on the symmetry axis curve, they can be extrapolated to other points on the surface with known methods.

For our investigation of this approach, we have developed a number of algorithms and experiments. First, we propose a method for detecting intrinsic reflective symmetry axis curves based on blended intrinsic maps [KLF11]. Second, we propose an axis curve alignment algorithm based on string matching that is efficient (polynomial-time) and robust to missing and extra parts. Third, we describe an algorithm based on blended intrinsic maps to extrapolate symmetry axis curve correspondences to surfaces with little conformal distortion. Fourth, we describe a method for fitting a genus-zero mesh to arbitrary surface data to facilitate intrinsic symmetry detection and surface correspondence. Finally, we present results of experiments that suggest our method out-performs the state-of-the-art on many types of real-world surfaces. While each of these contributions is a small research advance on its own, the combination leads to a significant conclusion: aligning symmetry axis curves is usually the best way to find maps between surfaces representing real-world objects in the same class.

2. Related Work

Finding correspondences between two surfaces is a classical problem that has been studied in many fields [vKZHCO10].

In some applications, maps between surfaces can be modeled with a low-dimensional transformation, and thus finding correspondences can be performed in polynomial-time. For example, rigid transformations have six degrees of freedom and can be searched efficiently with algorithms based on Hough transformations, RANSAC, Geometric Hashing, etc. Similar approaches have been applied for non-rigid transformations, including thin-plate splines [BR07] and conformal maps [LF09]. The common drawback of these approaches is that they require an analytical model for the deformation expected between two surfaces, and thus they cannot be used when surfaces in the same class have significantly different shapes.

Some methods find surface correspondences by embedding surfaces in a feature space where similar points have similar coordinates and then produce a dense map based on nearest neighbors in that space. For example, [BBK06] developed a Generalized Multidimensional Scaling (GMDS) framework where a small number of correspondences define an embedding of one surface onto another minimizing an approximation to the Gromov-Hausdorff distance. [OMMG10] showed that a single correspondence can define a Heat Kernel Map, a high dimensional embedding of a surface invariant under isometry. These methods use heuristic search procedures to find initial point correspondences, and thus may converge to a local minimum. Also, since correspondences are formed by closest points in a high-dimensional space, the resulting map is not guaranteed to be smooth, or even continuous, when surfaces are not isometric.

In applications where correspondences must be found between significantly different shapes, most previous methods rely upon detection of local features (tips of protrusions, ridges and valleys, etc.) and then search for the permutation of feature correspondences that provide the surface alignment with minimal deformation. For example, [ZSCO*08] extracts features at points whose average geodesic distance to other points on the surface is locally maximal and then performs a priority-driven search of potential correspondence sets, choosing the one for which the deformation induced in the surface by aligning the corresponding feature points is minimal. This approach is extremely slow (the search space is $O(N!)$ for N feature correspondences), and thus is only practical for finding small correspondence sets and/or for aligning surfaces with distinctive local shape features that can be used to prune the search. In contrast, we search for correspondences between features on 1D curves (symmetry axis curves), which can be done efficiently using dynamic programming, and thus we can guarantee an optimal solution in polynomial time.

Other methods have performed global shape analysis to produce structures that are easier to match. For example, several researchers have extracted graph structures in which nodes represent "parts" and then reduce finding surface correspondences to a graph matching problem [HSKK01]. Unfortunately, subgraph isomorphism is still an NP-complete problem, and finding matches where M nodes in one graph correspond to N nodes in another is particularly difficult in practice, and thus these methods are most effective when the graph extraction algorithm is very robust and can generate similar topological structures for different shapes in the same object class. Our observation is that reflective symmetry axis curves provide exactly such a structure for most object classes.

Other recent work has investigated using symmetry analysis to guide mesh processing algorithms. For example, [MGP07] proposed an approach that can automatically enhance symmetries in a 3D shape; [PSG*06] introduced a method that captures reflectional symmetries, which can be leveraged for alignment, matching, segmentation, and viewpoint selection; [GPF09] leveraged symmetry detection for

remeshing, simplification, and beautification of 3D meshes. Our work can be considered an extension of this line of work to consider inter-surface correspondence. In this respect, it is related to [ZHG10], who leveraged extrinsic planar symmetry detection to guide transfer of textures between surfaces representing human faces. Their method requires human-specified landmark correspondences and is limited to transfer between similar surfaces with consistent local texture features using a single extrinsic reflective symmetry. Our method is fully automatic and works for a much broader class of shapes.

3. Key Idea

In this paper, we take advantage of global reflective symmetries to simplify the surface correspondence problem. The key idea is to extract curve(s) along the reflective symmetry axis of each surface, then search for an optimal alignment of the curves, and finally extrapolate correspondences on the curves to the rest of the surfaces (Figure 1).

The rationale for this approach is remarkably simple. First, almost every object in the real world is (at least approximately) symmetric – indeed, if you look around an office, it is difficult to find an object that does not have at least one global reflective symmetry. It seems silly not to take advantage of these symmetries when searching for surface correspondences.

Second, finding a map between symmetric points is *significantly* easier than finding a map between points on different surfaces, both because the local geometric features of semantic correspondences are more likely to be similar for symmetric maps than for inter-surface maps, and because geometric invariants (angles, distances, derivatives, etc.) must be preserved at the stationary points of a symmetry map, providing constraints that simplify map extraction.

Third, global reflective symmetry axis curves (stationary points in the symmetric map) are usually semantically stable features of objects – points on the symmetry axis curve of one object usually correspond to points on the symmetry axis curve of another. Therefore, the search for a map between points on the symmetry axis curves must consider only correspondences between 1D sets of ordered points, rather than between full 2D surfaces, a problem for which efficient algorithms based on dynamic programming are available.

Finally, computing a full inter-surface map from a given set of point correspondences is a well-studied problem for which several methods are available [Ale01], and thus we can use point correspondences found on the symmetry axis curve to establish a map for the entire surface automatically.

The main idea is that the difficult (intractable) problem of finding a map between significantly different symmetric surfaces can be reduced to a sequence of simpler problems, each of which can be solved with well-studied polynomial-time algorithms. In particular, the most difficult problem in surface mapping (finding a coarse set of semantic surface correspondences) is reduced to a 1D string matching problem (aligning symmetry axis curves), which can be solved in polynomial time with dynamic programming. This approach has broad applicability for classes of man-made objects with perfect extrinsic reflective symmetries (e.g., airplanes, chairs, etc.), as well as organic objects with approximate intrinsic reflective symmetries (e.g., people, animals, etc.), as are commonly found in 3D model repositories.

4. Methods

The input to our system is a pair of manifold, genus zero meshes, S_1 and S_2, and the output is a map, $m : S_1 \rightarrow S_2$, which gives a correspondence on S_2 for every point on S_1.

For each input S that is not manifold or genus zero, we execute a preprocessing step that constructs such a mesh M using a simple "shrink wrap" algorithm based on the variational level-set method described in [ZOMK00]. The algorithm starts with M on the bounding sphere of S and then iteratively moves vertices towards closest points on S until either they lie on S or further movement would exceed a maximum curvature threshold. M is then used for correspondence computations with other manifold, genus zero meshes. Afterwards, correspondnences found on M can be projected back onto the original input S using any interpolation scheme (we use radial-basis functions and closest points).

The core of our method finds correspondences between two manifold, genus zero meshes using the three stage processing pipeline shown in Figure 2. We first extract symmetry axis curves (2b), then align the symmetry axis curves (2c), and finally extrapolate the symmetry axis curve correspondences to a complete surface map (2d). The following subsections describe each of these steps in detail. Since each step leverages prior techniques, there are only minor research contributions within each subsection. Combining this sequence of steps into a system that automatically finds surface maps is the main contribution of our work.

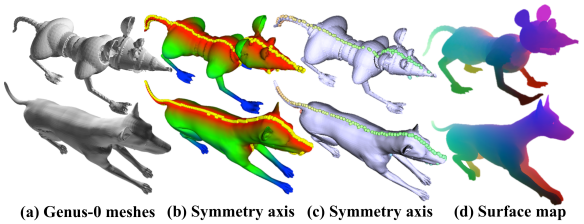


Figure 2: System processing pipeline.

4.1. Symmetry Axis Extraction

The first step is to extract a set of candidate symmetry axis curves from each surface. Given a densely sampled genus-zero triangle mesh, M, we aim to find a set of curves on M

that are stationary under a reflective symmetry r, $r : M \to M$, where r is a nearly-isometric map that takes the entire surface to itself. For intrinsic (extrinsic) symmetries, these are the curves on M whose points are all nearly equal geodesic (Euclidean) distance from both p and $r(p)$ for all points p on M. In our case, since the input surface is genus-zero after preprocessing, each candidate symmetry axis forms a closed 1D curve C on M, which we output as a sequence of mesh vertices. For each candidate C, we also produce a quality score, $Q_{Axis}(C)$, estimating how good the curve is for surface correspondence.

Several methods are possible to extract a symmetry axis curve on a mesh [vKZHCO10]. We have adopted an approach that first generates a symmetry map from the surface onto itself, then computes a function measuring the distance from a surface point p to its correspondence in that map, and finally extracts a symmetry axis curve C as the 0-level set of that distance function. We found this approach to be more robust that other methods (e.g., PIRS voting [XZT*09]), first because our algorithm for generating symmetry maps explicitly tries to maintain conformality and thus is more robust to deviations from isometry, and second because it is easier to extract a symmetry axis curve from a distance function than from functions produced by voting, which may have very subtle ridges on thin extremities (e.g., the tail of a cat) unless extremely large numbers of votes are cast. Moreover, the symmetry map produced in this step can be used to evaluate the distortion of the map, which can be used to estimate the quality of the symmetry axis curve, $Q_{Axis}(C)$, and it can be used to facilitate correspondence extrapolation later in the processing pipeline.

The main challenge then is to generate candidate symmetry maps. For extrinsic reflectional symmetries (planar reflections that map the entire surface onto itself), this problem is trivial, and several good algorithms are available (e.g., [OO96, PSG*06]). However, for intrinsic symmetries (isometric maps from the surface onto itself), the problem is more difficult, especially when the best symmetry map is not perfectly isometric. So, we take a two-phase approach, where we first check whether the input surface has approximate extrinsic reflective symmetries, and then check for approximate intrinsic reflective symmetries only if an extrinsic one cannot be found.

To check for extrinsic reflective symmetries, we perform a rigid transformation of the mesh so that its centroid is at the origin and its principal axes are aligned with Cartesian axis vectors. Then, we consider reflections across planes defined by $x = 0, y = 0, z = 0$ as candidates for an extrinsic, global reflective symmetry transformation [OO96]. For each plane, we reflect points sampled uniformly from the surface across the plane and compute the average distance δ to the closest point on the mesh. If δ is less than a threshold ($\varepsilon = 0.03\sqrt{Area(M)}$), we accept the plane reflection as an extrinsic symmetry and create a candidate map r that takes each vertex v to the vertex on the mesh closest to its reflection. The axis curve C is then extracted as the 0-level set of $d(p, r(p))$ using topological thinning, and the axis quality is determined by $Q_{Axis}(C) = Length(C)$.

To find intrinsic reflective symmetry maps, we consider conformal maps that correspond the mesh to itself. Like [KLF11], the basis of our approach is blended intrinsic maps (BIM) – i.e., each candidate symmetry map is a weighted blend of conformal maps constructed from triplets of feature points. However, using ideas in [KLCF10], we can specialize the algorithm to work more efficiently for reflectional symmetry detection than it does in the general case by applying filters to the selection of feature point triplets that ensure consistency of symmetric spatial relationships and surface properties. Specifically, we first detect feature points at extrema of the average geodesic distance (AGD) function [ZSCO*08]. Then, we construct conformal maps for triplets of feature point correspondences, considering each feature point, a, paired with two others, b and c, and forming the conformal map defined by $(a \to a, b \to c, c \to b)$, which represents the hypothesis that a is on the symmetry axis and $b \leftrightarrow c$ are a symmetric correspondence. Then, consistent sets of the low-distortion maps are identified by eigenanalysis of a conformal map similarity matrix, and symmetry maps are produced by blending conformal maps with weights provided by the top eigenvectors. Rather than producing only a single best map, as was done in [KLF11], our method generates maps associated with all of the eigenvectors associated with the top eigenvalues, ranked by estimates of area distortion, $c(p_i)$, averaged over points p_i sampled on the surface. For each map r, we extract a candidate symmetry axis curve, C, as the 0-level set of $d(p, r(p))$ using topological thinning, and estimate its quality with

$$Q_{Axis}(C) = Length(C) \times \frac{avg(c(p_i)) - c_t}{1 - c_t} \qquad (1)$$

where $c_t = 0.5$, and discard it if $Q_{Axis}(C)$ is negative (the symmetry map has large distortion). The final result then is a set of symmetry axis curves with associated symmetry maps and quality estimates.

4.2. Symmetry Axis Alignment

Our next step finds the optimal alignment (map) between symmetry axis curves.

In this section, we first define the axis alignment problem and then describe an efficient algorithm to solve it. Since symmetry axes are 1D curves, we are able to find an optimal solution in polynomial-time.

Problem formulation. Given two sets of symmetry axis curves, $C_1 = \{C_1^1, C_1^2, \ldots, C_1^{k_1}\}$ and $C_2 = \{C_2^1, C_2^2, \ldots, C_2^{k_2}\}$, extracted from meshes M_1 and M_2, our goal is to find a pair of axis curves (C_1^*, C_2^*), and an optimal alignment c^* that maximize a quality measure $Q(C_1^i, C_2^j, c)$, where $(1 \le i \le k_1)$

(a) Symmetry map (b) Symmetry axis

Figure 3: Extraction of intrinsic symmetry axis curves. a) The left image shows a reflective symmetry map, where corresponding points are shown in the same color. b) The right image shows distances between symmetric correspondences, where red is small and blue is large. The symmetry axis curve (yellow) is the 0-level set of this distance function.

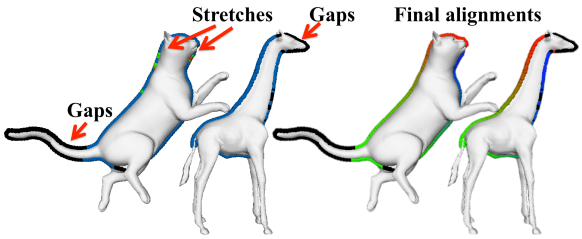


Figure 4: Symmetry axis alignment. A sequence alignment may have gaps and stretches, highlighted with red arrows on the left. The optimal alignment is shown with correspondences shown in different colors on the right.

and $(1 \leq j \leq k_2)$:

$$(C_1^*, C_2^*, c^*) = \arg \max_{C_1^i, C_2^j, c} \{Q(C_1^i, C_2^j, c)\}, \quad (2)$$

We define the quality of an alignment between any two curves C_1^i and C_2^j as the product of three terms:

$$Q(C_1^i, C_2^j, c) = Q_{Axis}(C_1^i) \cdot Q_{Axis}(C_2^j) \cdot Q_{Align}(C_1^i, C_2^j, c) \quad (3)$$

The first two terms, $Q_{Axis}(C_1^i)$ and $Q_{Axis}(C_1^j)$, describe the quality of each axis independently, as defined in Equation 1. The last term, $Q_{Align}(C_1^i, C_2^j, c)$, describes the quality of a pairwise alignment c between two axis curves – its definition and computation are the focus of this section.

Definition of Q_{Align}. $Q_{Align}(C_1^i, C_2^j, c)$ provides a measure estimating how well a correspondence c aligns semantic features of two axis curves C_1^i and C_2^j. For the classes of objects considered in this paper (e.g., Figure 4), it should be robust to non-isometric distortions (e.g., cat neck vs. giraffe neck), extra or missing parts (e.g., cat tail vs. giraffe), and/or differences due to errors in symmetry axis curve extraction. Moreover, it should be defined so that its optimizer can be found efficiently.

With these goals in mind, for each pair of symmetry axis curves, $A \in \{C_1^i\}$ and $B \in \{C_2^i\}$, we discretize both curves into an ordered set of N ($N = 200$) uniformly-spaced sample points ($A = \{a_1, a_2, \ldots a_N\}$ and $B = \{b_1, b_2, \ldots b_N\}$) and then define the alignment quality to be the inverse of the string edit distance of alignment c:

$$Q_{Align}(A, B, c) = 1/(D(A, B, c) + \varepsilon)$$
$$D(A, B, c) = \sum_{0 \leq k < K} \gamma(\mathbf{a}_{i_k}, \mathbf{b}_{j_k}) \quad (4)$$

where $\gamma(\cdot, \cdot)$ denotes the cost of a string edit operation. If we use λ to denote an empty string, correspondences and gaps can both be represented by alignment pairs, giving three possible edit operations: (1) substitutions ($a_i \rightarrow b_j$), (2) insertions, ($a_i \rightarrow \lambda$), and (3) deletions, ($\lambda \rightarrow b_j$). A valid alignment, c, is a sequence of alignment pairs, $(\mathbf{a}_{i_0}, \mathbf{b}_{j_0}), (\mathbf{a}_{i_1}, \mathbf{b}_{j_1}), \ldots, (\mathbf{a}_{i_{K-1}}, \mathbf{b}_{j_{K-1}})$, where $\mathbf{a}_i \in \{A, \lambda\}$ and $\mathbf{b}_i \in \{B, \lambda\}$, where point order in the strings is preserved in the sequence of alignment pairs, and where points paired with λ cannot also be paired with other points.

We define $\lambda(\cdot, \cdot)$ to balance efficient computations and discriminative alignments. For each correspondence assignment with λ, we assign a constant gap penalty: $\gamma(a_i, \lambda) = \gamma(\lambda, b_j) = 0.5$. For assignments between each pair of points, a_i and b_j, we define $\gamma(\mathbf{a}_i, \mathbf{b}_j)$ to be the $L2$ distance betweeen shape descriptors computed at a_i and b_j. In our implementation, the shape descriptor for a point p on a symmetry axis curve C of mesh M has 52 features in total. 32 of them represent the histogram of geodesic distances from p to all other points on M (normalized so that each dimension has zero mean and unit standard deviation) and 20 of them represent the local curvature profile of M at p. The curvature profile contains 10 features the curvature of C at p at different scales ranging from 0.05 to 0.15 and 10 features representing the curvature profile of M in the direction orthogonal to C at p at the same scales. Although these shape descriptors (the curvature profiles) are not invariant to isometries or other simple deformations, we find that they are more robust that many alternatives (e.g., HKS, Gaussian curvature, etc.) to intra-class variations found in our data sets.

Computation of Q_{align}. To find the alignment c with the minimal value of Q_{align}, we repeatedly employ an algorithm based on dynamic programming. Since the algorithm is so fast, we execute it for every pair of extracted curves using both possible alignment directions and every possible starting point, and then return the best result.

For each possible pair of curves, alignment direction, and starting point, the discrete optimization problem in Equation 4 can be solved optimally with Dynamic Time Warping, a method traditionally used for speech analysis [RS80] and recently used for shape matching [MP05]. The basic idea is the optimal solution of the current sequences can be computed by the optimal solutions of their prefixes, and it can be computed recursively as below (ignoring boundary con-

ditions for brevity):

$$D(A_{1:i},B_{1:j}) = \min \begin{cases} D(A_{1:i-1},B_{1:j-1}) + \gamma(a_i,b_j) \\ D(A_{1:i-1},B_{1:j}) + \min(\gamma(a_i,b_j),\gamma(a_i,\lambda)) \\ D(A_{1:i},B_{1:j-1}) + \min(\gamma(a_i,b_j),\gamma(\lambda,b_j)) \end{cases} \quad (5)$$

This approach is applied for every pair of symmetry axis curves extracted from both meshes, $C_1 = \{C_1^1, C_1^2, \ldots, C_1^{k_1}\}$ and $C_2 = \{C_2^1, C_2^2, \ldots, C_2^{k_2}\}$, and then the best alignment is output as the final result.

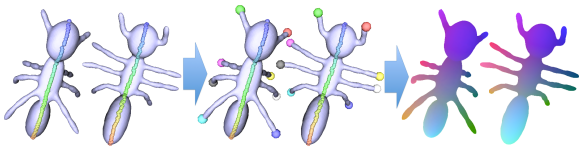
The overall complexity of this algorithm is $O(K^2N^3)$, where K is the number of axes extracted per mesh and N is the number of discrete points on each symmetry axis curve (usually 200). Each dynamic program takes $O(N^2)$ for each of $O(K^2)$ pairs of curves, $O(N)$ starting points, and 2 directions. In practice, it runs in ~ 5 seconds per pair of meshes. It would be possible to improve the speed even further by a factor of $O(N/\log N)$ using a method by [MP05], but we did not implement it since the speed of this step is fast already.

Finding vertex correspondences from the alignment. The output of Dynamic Time Warping is a correspondence between point samples, which may contain one-to-many correspondences. To convert this one-to-many correspondences into a one-to-one vertex correspondence, we replace the curve segment defined by the multiple corresponding points with its middle sample point. We then project each sample point onto its closest vertex, and remove correspondences where vertices are duplicated. The output is then a duplicate-free set of correspondences between vertices on two symmetry axis curves.

4.3. Correspondence Extrapolation

The next step is to extrapolate the correspondences on the symmetry axis curves to the rest of the genus-zero triangle mesh. Given a set of vertex correspondences, $c = \{a_i, b_i\}$, where $a_i \in A$ is a point on symmetry axis C_1 of mesh M_1 and $b_i \in B$ is on on C_2 of M_2, the goal is to generate a map, $m : M_1 \rightarrow M_2$, for all vertices in M_1.

This is a classic inter-surface mapping problem, for which a number of solution methods have been proposed [Ale01]. One approach is to embed the surfaces into a high-dimensional space based on geodesic distances to points with known correspondences and then to establish correspondences between closest points in the embedded space (e.g., GMDS [BBK06]). Another approach is to map both surfaces to a canonical domain (e.g., a sphere or a coarse mesh), where given correspondence points align and distortion is minimal according to some metric (e.g., angle deviations), and then interpolate the map in that domain [KS04, PSS01, SAPH04]. We employ a two-stage hybrid algorithm that leverages both of these ideas.



(a) Symmetry axis alignment **(b) Feature point correspondences** **(c) Surface map**

Figure 5: Correspondence Extrapolation. Correspondences on the symmetry axis curves are first extrapolated to a stable set of feature points, and then interpolated to the rest of the surface.

Our unique problem is that the set of correspondences provided by the symmetry axis curve alignment, c_C, usually includes points only within central regions of the surfaces, and thus correspondences may be extrapolated to other regions, sometimes over considerable distances, which is unstable and leads to map distortions. To address this problem, we proceed in two phases (Figure 5). In the first phase, we employ a linear assignment based only on axis correspondences c_C to find new correspondences c_F between highly-distinctive feature points far from the symmetry axis curves (e.g., tips of protrusions). In the second phase, we compute complete inter-surface map by approximately interpolating the correspondences given by both c_C and c_F. In this way, the correspondence set is enriched with well-spaced, stable correspondences before it is interpolated over the entire surface. Details of this two-stage algorithm follow.

In the first phase, our goal is to *extrapolate* the set of correspondences c_C given on the symmetry axis curves. Since we cannot robustly compute a complete surface map directly, we focus only on stable feature points far from the axis for which we can robustly extrapolate correspondences. In our implementation, we extract a set of feature points, F_1 and F_2, at persistent maxima of geodesic distances from the symmetry axis curve, using the same persistence definition as in [DLL*10]. We then search for correspondences between these points using a method based on the ideas of Generalized Multidimensional Scaling (GMDS) [BBK06]. The feature points, F_1 and F_2, are embedded into a high dimensional space where the i-th dimension represents a normalized distance to the point in the i-th correspondence of the symmetry axis curve. Then, we conduct a linear assignment based on the distance in the embedded space, and build a new correspondence set c_F that includes pairs of feature points (f_1, f_2). This procedure tends to find correspondences that are stable, well spread-out on the surface, and semantically correct.

In the second phase, we fill in a full surface map based on the correspondences discovered between axis and feature points. Since the given correspondences now span the entire surface, this is a straight-forward inter-surface mapping problem. In our implementation, we use a variant of blended

intrinsic maps [KLF11], where conformal maps defined by triplets of corresponding points are blended to form a smooth map over the entire surface. With this approach the main design decision is to select which conformal maps to compute and blend. If there are $|c_C|$ correspondences between points on the symmetry axis curves and $|c_F|$ correspondences between feature points, then we could blend $\binom{|c_C|+|c_F|}{3}$ conformal maps, many of which would have redundant information. Our strategy is to blend conformal maps only constructed from triplets that contain two correspondences from the axis curves and one from the feature points. Specifically, we select $M = 50$ evenly spaced points on C_1, add another point that is separated by 10% of the axis length, and form a triplet from these two points combined with each of the feature correspondences (for a total of $M \times |c_F|$ triplets, or approximately 500 in practice). These conformal maps are blended using the "confidence weights" based on area distortion as described in [KLF11] to form the final map.

5. Results

We have executed a series of experiments designed to test the performance of our method, to understand its speed, range of applicability, and performance relative to the state-of-the-art.

Benchmark Results: Our first experiment tests how well our algorithm performs for finding maps between nearly isometric, genus-zero, watertight meshes. For this experiment, we follow the testing methodology prescribed by the Surface Correspondence Benchmark described in [KLF11]. The benchmark provides pairs of surface meshes representing humans and animals taken from the SCAPE, TOSCA, and SHREC Watertight 2007 data sets, and provides code to evaluate and compare automatic surface mapping methods with plots that show geodesic distance errors between predicted correspondences and ground truth correspondences provided by people.

Figure 6 shows our results. In each plot, the horizontal axis represents geodesic error and the vertical axis represents the fraction of ground truth correspondences within predictions closer than that error threshold (higher curves are better). For the sake of brevity, we show results of our method (solid lines) only in comparison to Blended Intrinsic Maps (BIM) [KLF11] (dotted lines), which significantly out-performs Mobius Voting [LF09], GMDS [BBK06], and Heat Kernel Maps [OMMG10], and all other methods tested on this data set. From the results, we see that our method (solid lines) performs comparable to BIM (dotted lines) for the NonRigidWorld (pink) and Human (green) data sets, and better for the Animal (black) and SCAPE data sets (blue). These results are encouraging because BIM is already very good on some of these data sets (e.g., NonRigidWorld and SCAPE). The improved result for animals best shows the advantage of our approach – often the symmetry axis for four-legged animals is robust to extract, align, and extrapolate, even in cases where the surface shapes are very different (e.g., Figure 4).

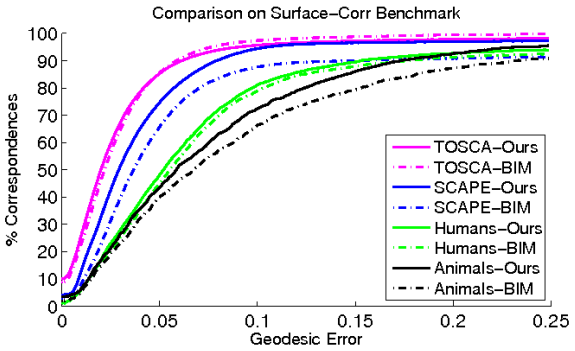


Figure 6: Results for the Surface Correspondence Benchmark.

Impact of each step on the benchmark results: To investigate how much each step of computational pipeline contributes to the final surface correspondence errors in these benchmark tests, we ran a series of experiments in which the pipeline was started from "perfect" data provided by a human at each successive stage of the pipeline. Specifically, we compare final correspondences when a) the pipeline runs in full starting with the original surface inputs, b) when the symmetry axis extraction step is skipped, and the symmetry axis alignment step starts with human-extracted symmetry axes for all models, c) when the first two steps are skipped, and the correspondence extrapolation step starts from human-specified correspondences on every pair of symmetry axes, and d) when all but the very last algorithm are skipped, and the system only extrapolates correspondences from human-specified correspondences on the symmetry axes and extremal features. Results are shown for the Animals data set of the Surface Correspondence Benchmark in Figure 7. In consideration of the fact that the magenta curve is shifted upward by $\sim 20\%$ since 4 out of 21 ground-truth correspondences are given, the proximities of curves to one another indicate that the main source of error comes from the final step of the pipeline – correspondence extrapolation – i.e., the results do not change much even if perfect axis alignments and extremal feature points are provided by a human to the last stage of the pipeline. We attribute this to the failure of blended intrinsic maps to align semantic features. Perhaps other methods that consider local shape features instead of conformality and/or area preservation will produce better results for this benchmark.

Global symmetry detection benchmark results: To evaluate the quality of our symmetry detection algorithm in isolation, we compared the symmetry maps produced by the first step of our pipeline to previous algorithms for that task using the benchmark described in [KLCF10]. Specifically, we predict a full map for each input model from every point

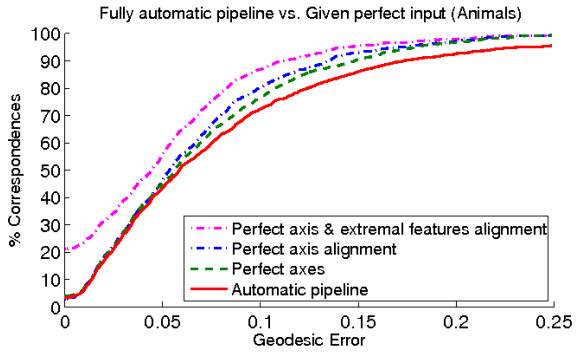


Figure 7: Given ground truth in different stages

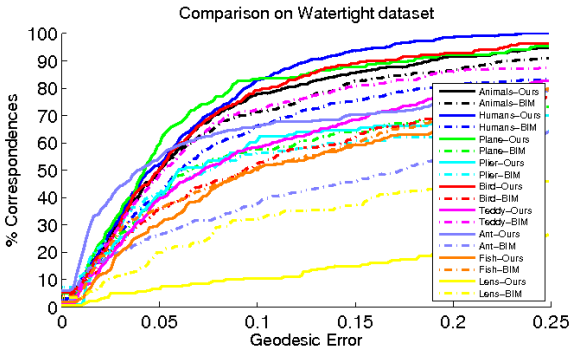


Figure 8: Results for SHREC Watertight 2007 pairs.

to its symmetry correspondence and then apply the metrics of [KLCF10] to evaluate how well the predicted maps align with human-specified ground truth. According to the "correspondence rate" metric, we extract "acceptable" correspondences for 91% of points on average over all three datasets (vs 84% for [Kim et al 2010]). According to the "mesh rate" statistic, we extract "acceptable" maps for 83% of meshes (versus 77%). Our method for extrinsic symmetry axis extraction never fails according to these metrics. As such, we believe that this algorithm could be useful by itself for symmetry map prediction in applications beyond the one proposed here.

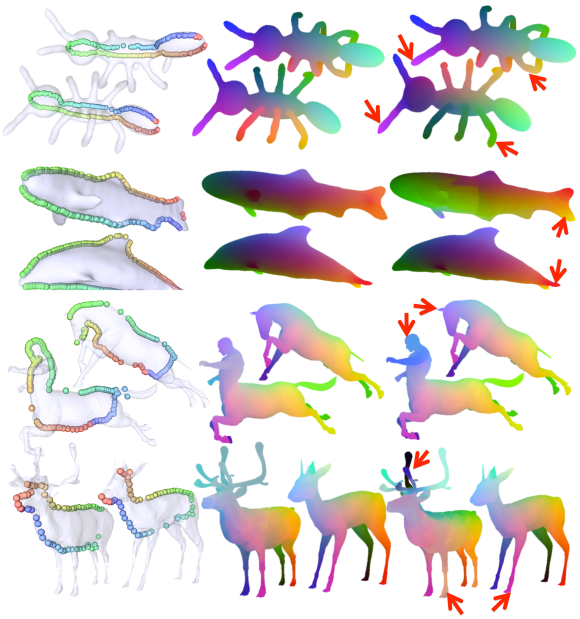
SHREC Watertight 2007 results. To test our method on a wider range of inputs, we also compared results for symmetric object classes in the SHREC Watertight 2007 Data Set [GBP07]. Again, we show our results in solid lines and BIM in dotted lines (Figure 8). Here there is a much bigger difference in the results: our method provides better results for object classes where multiple nearly-isometric mappings are possible, and yet only one of them aligns the symmetry axis. This is the case for Airplanes (BIM may map wings to fuselage) and Ants (BIM maps front to back and/or top to bottom), as shown in Figure 9. On the other hand, our method performs worse in cases when the symmetry axes are extracted poorly due to non-isometries in the symmetry map (Teddy, Figure 10(a)), when extracted symmetry axis curves are very short (Glasses, Figure 10(b)), and when the surfaces have few feature points to guide surface map extrapolation (e.g., Fish). Overall, our method has better results than BIM for six object classes and worse results for three.

More difficult examples: While the results of tests on standard data sets are informative for comparison, they do not test the full range of applicability of our method. Figures 1 and 9 show several more interesting examples. In general, we find that our approach is most useful when the input surfaces are both symmetric and share a similar symmetry axis curve, but vary significantly in overall shape. In these cases, other algorithms that attempt to find a nearly isometric map between the two surfaces will have great difficulty. For example, Figure 9 shows cases where BIM fails (red arrows in right column), due to instability at thin junctions between parts (ants), non-isometric regions (fish tails), and/or extra parts (centaur-horse and deer-buck). In contrast, finding a symmetric map for each model in these cases can be done robustly, since the two sides of the same surface are almost perfectly isometric. Moreover, since the symmetry axis curves align semantic features (e.g., nose-to-nose, tail-to-tail, etc.), and the symmetry axis curves span the surface sufficiently for extrapolation, our algorithm has no trouble producing a correct map in these cases (third column). We believe that most pairs of 3D surfaces to be aligned for applications in 3D graphics fit these criteria.

Our implementation also can handle polygon soups and other surface inputs that present problems for other surface correspondence algorithms. In Figure 9, the deer in the bottom row have 9 and 13 connected components, and the mouse in Figure 2 has 209. Previous methods that solve for correspondences based on a smooth surface deformation model (e.g, BIM) cannot handle this type of input. By first shrink-wrapping the model in a genus-zero triangle mesh, we are able to employ algorithms that find correspondences based on symmetry axes that can then be projected back on the original surface data.

Failure Cases: Our method is not applicable for all object types. Of course, it fails for objects that do not have a reflective symmetry, and thus it would not be useful for alignment of partially occluded scan data, for example, unless partial reflective symmetry axes could be identified robustly. Also, it fails when axis curves are not extracted properly and when symmetry axis curves are too short and few feature points are available for extrapolation (Figure 10). These problems could probably be addressed by using other symmetry detection algorithms and/or extracting other features to guide extrapolation (e.g., ridges and valleys), but those are topics for future investigation.

Timing: The computational complexity of our algorithm is $O(F^6 S \log M + N^3 + NFS \log M)$, where $M \sim 10{,}000$ is the number of vertices in each genus-zero mesh, $S = 128$ is the

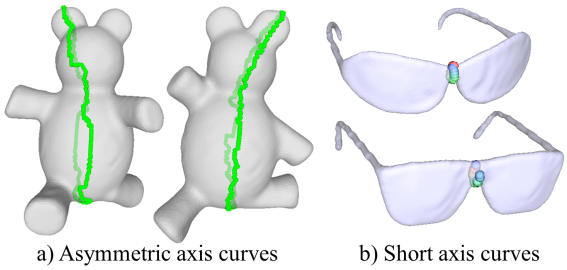


(a) Axis Alignments (b) Our Results (c) BIM Results

Figure 9: Results for difficult cases. Each of the four rows shows one example. a) A pair of input meshes is shown on the left. b-c) Our extracted and aligned symmetry axes and extrapolated surface maps are shown color coded by correspondence. d) Surface maps computed with Blended Intrinsic Maps (BIM) are shown for comparison on the right. Errors produced by BIM are indicated by red arrows.

number of points sampled on each mesh, $F \sim 5-10$ is the number of feature points on each mesh, and $N = 200$ is the number of sample points on each symmetry axis curve. The slowest steps are the ones that require computing blended intrinsic maps, as they require a $\log M$ search to find the closest vertex for S sample points for each of ~ 200 maps. Ironically, the problem of finding coarse point correspondences, which traditionally is the hardest challenge in surface mapping (intractable in the general case), can be solved optimally with the least computation ($O(N^3)$) of any step in our system.

In practice, on a SunFire X4100 computer with an AMD Opteron 275 Dual-Core 2.2GHz processor, preprocessing of polygon soup models takes approximately one minute, extracting the symetry axis curves from an extrinsically symmetric mesh with 12,500 vertices takes around 5 seconds or extracting intrinsic symmetry axis curves takes 1 minute, aligning symmetry axis curves takes 5 second, extrapolating the correspondence to intrinsic surfaces takes 3 minutes, and transferring the map back to polygon soups takes around 2.5 minutes (if desired). The input preprocessing and symmetry axis extraction steps must be done only once per mesh, and so their compute times may be amortized across many executions if a surface will be mapped to more than one other surface in its lifetime. The pairwise compute time is dominated by the correspondence extrapolation, which could be significantly improved with other methods (we used Blended intrinsic maps to favor ease of implementation and quality of results). In any case, these compute times are appropriate for automatic batch processing of models, as the system is intended to be used.



a) Asymmetric axis curves b) Short axis curves

Figure 10: Failure cases.

6. Conclusion

This paper describes a method for automatically finding a map between surfaces based on extraction and alignment of symmetry axis curves. Besides this main idea, the primary research contribution is the design of a system that includes algorithms for genus-0 mesh fitting, intrinsic reflective symmetry detection, symmetry axis curve alignment, and correspondence extrapolation. Results of experiments with this system demonstrate that it can find surface maps at least as well as the state-of-the-art for nearly isometric surface pairs and better that other methods for many difficult cases, including ones where the input is a polygon soup.

Our system is an early investigation of how to use global symmetry detection for automatic discovery of surface maps. As such, it has several limitations, which suggest topics for future work. First, it's current algorithms for symmetry map detection and correspondence extrapolation rely upon robust detection of feature points on and off the symmetry axis. This is just an implementation detail of the method we've chosen, but it affects the final results of our system in some cases. Second, our work considers only global reflective symmetries: other global symmetry groups (e.g., rotation) might be just as valuable. In broader terms, it would be interesting to consider other large-scale shape features that are stable, easy to detect, and fast to align. Some work has been done on extracting and matching internal symmetry axes and other part-based structures (e.g., [HSKK01]), but topological representations are generally unstable with respect to shape variations. Finding other stable global shapes features is an important topic in shape analysis. At this time, we conjecture that the global reflective symmetry axis curve is a sweet spot in this regard, but it is

possible that better features will be discovered for alignment in future work.

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of all body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (proc. SIGGRAPH)* (2003). 1

[Ale01] ALEXA M.: Recent advances in mesh morphing. *Computer Graphics Forum 21*, 2 (2001), 173–198. 1, 3, 6

[BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences (PNAS)* (2006). 2, 6, 7

[BR07] BROWN B. J., RUSINKIEWICZ S.: Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2007). 2

[DLL*10] DEY T., LI K., LUO C., RANJAN P., SAFA I., WANG Y.: Persistent heat signature for pose-oblivious matching of incomplete models. In *Computer Graphics Forum* (2010), vol. 29, pp. 1545–1554. 6

[GBP07] GIORGI D., BIASOTTI S., PARABOSCHI L.: Shape retrieval contest 2007: Watertight models track. *SHREC competition* (2007). 8

[GPF09] GOLOVINSKIY A., PODOLAK J., FUNKHOUSER T.: Symmetry-aware mesh processing. *Mathematics of Surfaces LNCS 5654* (September 2009). 2

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 203–212. 2, 9

[KLCF10] KIM V. G., LIPMAN Y., CHEN X., FUNKHOUSER T.: Mobius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum (Proc. of Symposium on Geometry Processing)* (2010). 4, 7, 8

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *Transactions on Graphics (Proc. of SIGGRAPH 2011)* (2011). 2, 4, 7

[KS04] KRAEVOY V., SHEFFER A.: Cross-parameterization and compatible remeshing of 3d models. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)* (2004). 1, 6

[LF09] LIPMAN Y., FUNKHOUSER T.: Mobius voting for surface correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (Aug. 2009). 2, 7

[MGP07] MITRA N. J., GUIBAS L., PAULY M.: Symmetrization. *ACM Transactions on Graphics (SIGGRAPH) 26*, 3 (2007), #63, 1–8. 2

[MP05] MARZAL A., PALAZÓN V.: Dynamic time warping of cyclic strings for shape matching. *Pattern Recognition and Image Analysis* (2005), 644–652. 5, 6

[OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. In *Computer Graphics Forum* (2010), vol. 29, pp. 1555–1564. 2, 7

[OO96] O'MARA D., OWENS R.: Measuring bilateral symmetry in digital images. *IEEE-TENCON - Digital Signal Processing Applications* (1996). 4

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3d shapes. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2006). 2, 4

[PSS01] PRAUN E., SWELDENS W., SCHRÖDER P.: Consistent mesh parameterizations. *Proc. of SIGGRAPH 2001* (2001). 6

[RS80] RABINER L., SCHMIDT C.: Application of dynamic time warping to connected digit recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on 28*, 4 (1980), 377–388. 5

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2004). 6

[vKZHCO10] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Eurographics State-of-the-Art report* (2010). 2, 4

[XZT*09] XU K., ZHANG H., TAGLIASACCHI A., LIU L., LI G., MENG M., XIONG Y.: Partial intrinsic reflectional symmetry of 3d shapes. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 138. 4

[ZHG10] ZENG W., HUA J., GU X.: Symmetric conformal mapping for surface matching and registration. *International Journal of CAD/CAM 9*, 1 (2010). 3

[ZOMK00] ZHAO H., OSHER S., MERRIMAN B., KANG M.: Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding 80*, 3 (2000), 295–314. 3

[ZSCO*08] ZHANG H., SHEFFER A., COHEN-OR D., ZHOU Q., VAN KAICK O., TAGLIASACCHI A.: Deformation-driven shape correspondence. In *Computer Graphics Forum* (2008), vol. 27, pp. 1431–1439. 1, 2, 4