MapTree: Recovering Multiple Solutions in the Space of Maps

JING REN, KAUST
SIMONE MELZI, LIX, École Polytechnique
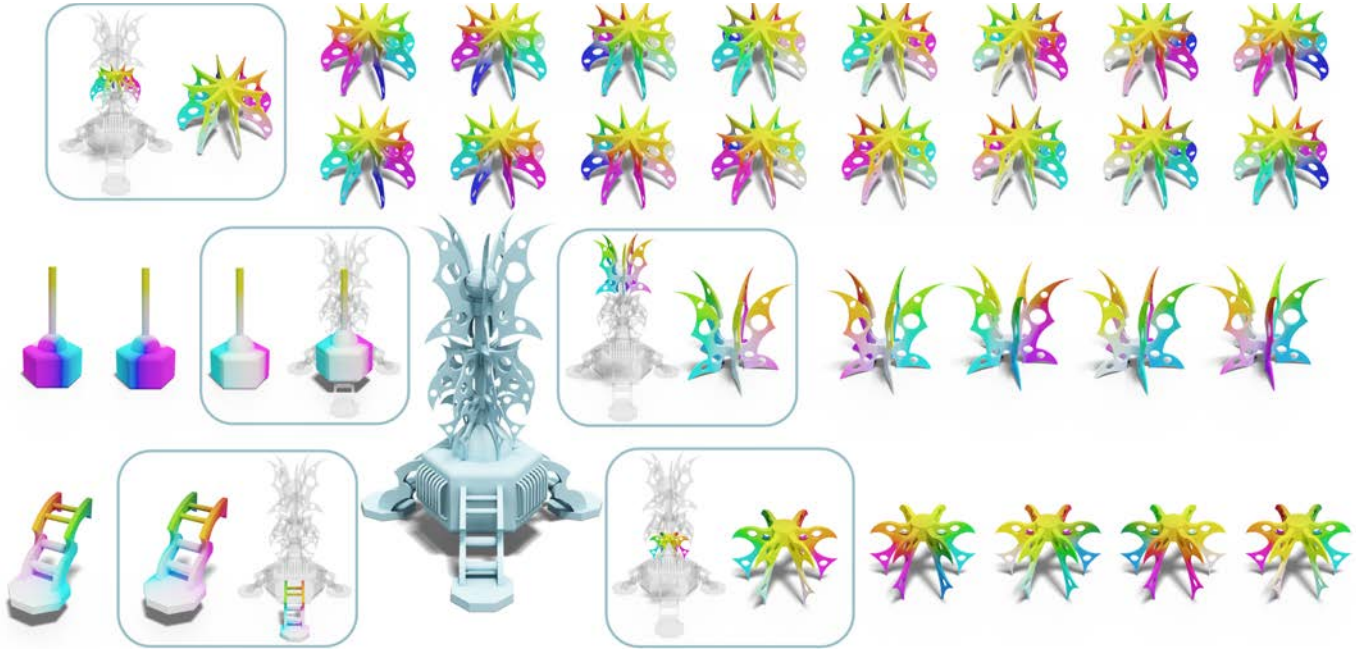MAKS OVSJANIKOV, LIX, École Polytechnique
PETER WONKA, KAUST

Fig. 1. For a given model (colored blue in the center), we first automatically isolate its disconnected components (shown in boxes). For each component, we show its location on the original shape, and a zoomed-in version where we color the vertices using their 3D coordinates. On the left/right of the component, we show a *subset* of the self-symmetric maps of this component, obtained fully automatically using our algorithm. Correspondences are shown via color transfer.

In this paper we propose an approach for computing multiple high-quality near-isometric dense correspondences between a pair of 3D shapes. Our method is fully automatic and does not rely on user-provided landmarks or descriptors. This allows us to analyze the full space of maps and extract multiple diverse and accurate solutions, rather than optimizing for a single optimal correspondence as done in most previous approaches. To achieve this, we propose a compact tree structure based on the spectral map representation for encoding and enumerating possible rough initializations, and a novel efficient approach for refining them to dense pointwise maps. This leads to a new method capable of both producing multiple high-quality correspondences across shapes and revealing the symmetry structure of a shape without a priori information. In addition, we demonstrate through extensive experiments that our method is robust and results in more accurate correspondences than state-of-the-art for shape matching and symmetry detection.

Authors' addresses: Jing Ren, KAUST, jing.ren@kaust.edu.sa; Simone Melzi, LIX, École Polytechnique, melzismn@gmail.com; Maks Ovsjanikov, LIX, École Polytechnique, maks@lix.polytechnique.fr; Peter Wonka, KAUST, pwonka@gmail.com.

1 INTRODUCTION

Computing dense correspondences across non-rigid 3D shapes is a classical problem in computer vision, computer graphics, and related fields. Dense correspondences between a pair of shapes are typically encoded by a map that assigns points on one shape to points on the other shape. There exist many optimization methods that define a map energy and try to find a desirable map by optimizing it [Bronstein et al. 2006; Ezuz et al. 2019; Huang et al. 2008; Ovsjanikov et al. 2010; Vestner et al. 2017b]. Unfortunately, the space of maps is very complex.
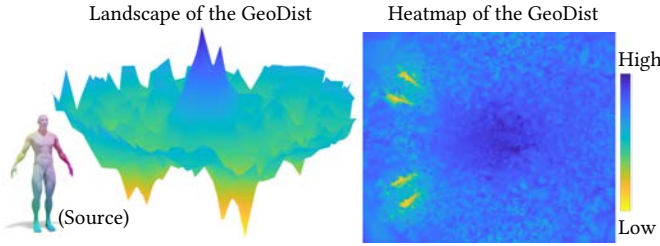
Fig. 2. We generate 10K self-maps on a human shape (shown bottom left). We then embed these maps into 2D using multi-dimensional scaling such that the geodesic distance between two point maps is preserved by the Euclidean distance in 2D. Therefore, we can visualize the geodesic distortion ("GeoDist") of each map. *Left:* the 3D landscape of the GeoDist over 10K maps. *Right:* the corresponding 2D heatmap. The maps with small (large) geodesic distortion are in color yellow (blue).

Fig. 3. *Left:* the contour of the geodesic distortion energy as shown in Fig. 2. We pick 6 initial maps (a-f) and after applying ZoomOut we obtain the refined maps (A-F), the visualization of these maps are shown on the *right*. The black lines show the intermediate steps/maps during the refinement/optimization. We can see that different initializations lead to different termination points (local minima).

First, this space is exponential in size and can contain many local minima. For example, Fig. 2 shows the commonly-used geodesic distortion (the average change in geodesic distances across all pairs of points) energy of 10K self-maps on a human shape. We can see that the energy function is non-convex with multiple local minima that correspond to different types of maps. Some local minima correspond to semantically meaningful maps that exist due to the presence of symmetries, e.g. the left-to-right symmetry of a human shape. Other local minima correspond to other smooth high-quality maps, e.g. a twisted mapping where the lower part of a human is correctly mapped, but the upper part is flipped from left-to-right. In addition, there are local minima (typically with higher energy) that correspond to undesirable maps, e.g. a mostly smooth map, but with one large error such as the head being mapped to a hand.

Second, the optimization in the space of maps is difficult and current algorithms need to make many simplifications, e.g. via relaxations [Dym et al. 2017; Fogel et al. 2013; Solomon et al. 2016], early termination, etc. As a result, algorithms may not actually converge to local minima, but terminate at points that may or may not be close to a local minimum. We call these points termination points (of a particular algorithm).

Therefore, we can identify two types of errors in existing shape matching algorithms. 1) The algorithm terminates close to the correct local minimum, but it does not fully reach it because of the simplifications in the algorithm. This typically results in many small errors at each vertex, but only very few large errors. 2) The optimization algorithm ends up in a wrong part of the space, generally, because of the initialization. This latter problem is particularly prominent as many algorithms are based on pre-defined descriptors or landmarks [Ovsjanikov et al. 2012; Ren et al. 2018; Vestner et al. 2017a]. In Fig. 3 we visualize some example maps (A-F) that are results of a map optimization algorithm, ZoomOut [Melzi et al. 2019b], starting from different initializations. In the example, four maps are recovered corresponding to symmetries: the direct map (A), the symmetric map (B), the back-to-front flipped map (C), and the doubly flipped map (D). In addition, there are two twisted maps. In map (E) the upper body is assigned the symmetric counterpart, while the the lower body arises from the direct map. In map (F) the upper body arises from the direct map and the lower body from the symmetric map. In addition, when comparing different map
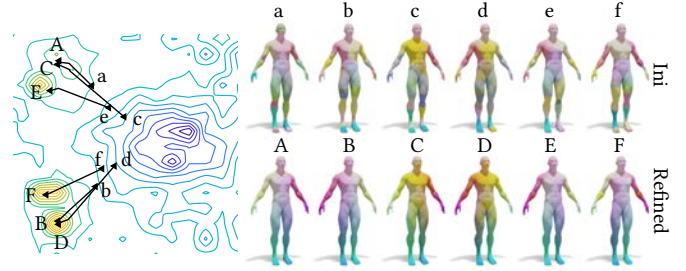
optimization algorithms, one can observe that different methods can lead to different maps, see Fig. 4.

We would like to advocate a departure from previous work, that has primarily focused on recovering a single high-quality map, and suggest that recovering multiple (or all) high quality maps in the space of maps is an important goal. Recovering multiple maps can provide information about all the intrinsic symmetries of the shapes and they also provide insights into the energy landscape of the optimization. For example, surprising low energy solutions of the geodesic distortion energy like the twisted maps can be recovered (not only accidentally) by existing methods. We therefore consider the problem of *multi-solution shape matching* and derive a new solution to this problem. We also note that multi-solution *symmetry detection* is a special case of this problem (See Fig. 1). However, our algorithm for recovering multiple maps is more than just a solution to a new problem statement. It can be used as building block to improve upon a long standing problem in traditional single-solution shape matching. For the vast majority of existing methods, the initialization is a considerable source of error and it can lead to a solution in the wrong part of the search space, e.g., a symmetric or twisted map is found when the goal is to compute the direct map. Complementary to efforts in improving the initialization or the optimization, we demonstrate that a considerable improvement in single-solution shape matching can be achieved by first computing a set of all high-quality maps and then selecting among them. As we show below, this can lead to significant improvements over the state-of-the-art even in traditional single-solution shape matching and single-solution symmetry detection.

For designing an algorithm to recover multiple high-quality maps, we can identify two criteria that an ideal solution should satisfy: (1) the recovered maps should be sufficiently different from each other. (2) each dense map should be smooth and near isometric. This includes maps that correspond to symmetries and other globally smooth maps that minimize energies that promote near isometries, e.g., based on geodesic distortion.

To tackle this problem, we propose to exploit the *spectral properties* of correspondences, encoded through the functional map representation [Ovsjanikov et al. 2012]. Key to our approach is a novel multi-resolution functional map exploration mechanism that completely avoids the need for descriptors or landmark correspondences
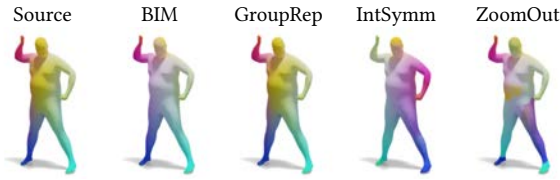
Fig. 4. Self-symmetry detection. Here we show an example where different baselines lead to different local minima. Specifically, BIM [Kim et al. 2011] obtains a back-to-front map, GroupRep [Wang and Huang 2017] leads to the direct map, IntSymm [Nagar and Raman 2018] leads to a back-to-front and left-to-right map, while ZoomOut [Melzi et al. 2019b] that is initialized by OrientRev [Ren et al. 2018] obtains a map with mixed symmetry.

and is capable of producing *multiple* meaningful dense correspondences that might exist in particular due to symmetries on the shapes. Our method is based on three observations, which we discuss below in detail: 1) pointwise maps that have large differences differ in the low frequencies of the spectral basis; 2) maps that are globally similar mainly differ in the high frequencies; 3) Large functional maps can be recovered from a small functional map [Melzi et al. 2019b]. Therefore, we explore the space of solutions through an efficient analysis of low frequency functional maps, which we then refine to obtain very accurate final correspondences. Depending on the application the recovery of multiple maps may be complemented by an algorithm for map selection, e.g., in single-solution shape matching the direct map should be selected.

To summarize, our main contributions are:
(1) A new method to solve the multi-solution shape matching problem that is able to output multiple high-quality dense correspondences.
(2) Significantly improved performance in near-isometric shape matching and symmetry detection compared to the state of the art.
(3) A compact tree structure for enumerating and organizing the space of maps between a pair of shapes, that allows exploration and map selection *a posteriori*.

2 RELATED WORK

Below, we briefly review work that is most related to ours, while focusing on intrinsic symmetry detection, spectral methods and shape matching techniques.

2.1 Intrinsic Symmetry detection

In computer vision and computer graphics, intrinsic symmetries have been widely investigated as a tool to gain insight into the structure of shapes or scenes. We refer the interested readers to recent surveys [Liu et al. 2010; Mitra et al. 2013].

While early symmetry detection work primarily focused on *extrinsic* or embedding-dependent symmetries [Mitra et al. 2006], a significant number of techniques have also been proposed for *intrinsic* symmetry detection, including [Lipman et al. 2010; Ovsjanikov et al. 2008; Raviv et al. 2010; Xu et al. 2012, 2009] among others. Importantly, most early intrinsic symmetry detection methods either only detect similar parts, and thus do not produce dense correspondences [Lipman et al. 2010; Xu et al. 2012, 2009], suffer from robustness issues [Ovsjanikov et al. 2008] or result in difficult non-convex optimization problems [Raviv et al. 2010], making them

sensitive to initialization and strongly relying on descriptors and subsampling.

More closely related to ours are recent methods based on advances in shape correspondence, especially the functional maps framework, including [Liu et al. 2015; Nagar and Raman 2018; Ren et al. 2018; Wang and Huang 2017]. These methods also aim to produce dense correspondences but in most cases require additional information such as detection of extremities [Liu et al. 2015] or *a priori* knowledge of symmetry orbits of a specified landmark of the shape [Wang and Huang 2017]. The method of Nagar et al. [Nagar and Raman 2018] that we compare to below, is related to our approach in its exploration of near-diagonal functional map structure, but relies on very sensitive eigenfunction sign detection.

2.2 Shape Matching

Our framework is also related to the problem of shape matching, and thus to methods that look for dense correspondences between non-rigid 3D shapes. For an in-depth review of this area we refer the readers to [Biasotti et al. 2016; Tam et al. 2013]. Several approaches to shape matching directly solve for correspondences between points on the two surfaces by minimizing an explicit energy, e.g., [Bronstein et al. 2006; Huang et al. 2008; Ovsjanikov et al. 2010]. The main limitation of these methods is that they often lead to complex combinatorial problems. An alternative is to first map the shapes to a canonical domain (e.g. a sphere), and then solve for the correspondence between these parametric representations [Aigerman and Lipman 2016; Aigerman et al. 2015; Lipman and Funkhouser 2009], or blend across multiple such maps [Kim et al. 2011].

One successful strategy for shape matching is to relax the search space from permutation matrices to a space more amenable to continuous optimization, e.g., doubly stochastic matrices [Dym et al. 2017; Fogel et al. 2013; Solomon et al. 2016], but also other successful relaxations exist [Azencot et al. 2019; Kezurer et al. 2015; Leordeanu and Hebert 2005; Maron et al. 2016; Solomon et al. 2012]. This, however, requires a possibly error-prone projection step to compute an integer solution, and makes it difficult to compute *multiple* correspondences, which might exist due to symmetries.

Our goal is to automatically and simultaneously produce multiple high-quality maps present due to different symmetries. Though there are no existing methods to solve our problem directly, there is some pioneering work for shape matching where multiple maps are *involved*. For example, multiple maps are generated and then blended into a single map in [Kim et al. 2011]. [Sahillioğlu 2018] proposed a genetic method for shape matching, where a "population" of maps are maintained for "evolution". However, these approaches do not aim to produce multiple different high-quality dense correspondences and thus they cannot be easily adapted to our setting. [Sung and Kim 2013] output multiple maps that potentially contain the symmetric map for a bilateral shape but is hard to generalize to shapes with multiple symmetric axes. [Sahillioğlu and Yemez 2013] considered the symmetry ambiguity when computing a single map to avoid symmetry flips. All of the above mentioned methods involve multiple maps, however, none of the them aim to recover *all* possible symmetry-aware maps. The ones that produce multiple maps provide either a given class of maps (like the direct and

the symmetric for bilateral shapes) or a set of maps without any property (like intermediate maps).

2.3 Shape Matching with Functional Maps

Our approach is based on the functional map representation [Ovsjanikov et al. 2012]. The vast majority of methods that use this framework for shape matching start with a set of descriptor functions, derived from point signatures or from landmarks, and use them jointly with global map quality criteria to compute a correspondence [Aflalo and Kimmel 2013; Kovnatsky et al. 2013; Nogneng and Ovsjanikov 2017] (we refer to [Ovsjanikov et al. 2017] for an overview). While computing *a functional* map reduces to solving a least squares system, the conversion from a functional map to a point-wise map is not trivial and can lead to errors and noise [Ezuz and Ben-Chen 2017; Rodolà et al. 2015]. To improve accuracy, several desirable map attributes have been promoted via regularizers for the functional map estimation first using geometric insights [Burghard et al. 2017; Eynard et al. 2016; Gehre et al. 2018; Litany et al. 2017b; Nogneng and Ovsjanikov 2017; Ren et al. 2018; Rodolà et al. 2017; Shoham et al. 2019; Wang et al. 2018a,b], and more recently using learning-based techniques [Halimi et al. 2019; Litany et al. 2017a; Roufosse et al. 2019]. Nevertheless, despite significant progress, the reliance on descriptors and decoupling of continuous optimization and pointwise map conversion remains common to all existing methods.

Our method is also related to spectral embedding methods. Some previous methods [Jain et al. 2007; Mateus et al. 2008; Ovsjanikov et al. 2008; Rustamov 2007] try to solve the shape matching (or symmetry detection) problem by directly aligning each dimension of the spectral embedding to obtain correspondences. In the functional map language this means forcing the functional maps to be diagonal [Nagar and Raman 2018]. Our method aims to explore block-diagonal functional maps as initializations that we then refine to dense correspondences without imposing diagonality. As we demonstrate below, this is crucial to obtain high quality correspondences in diverse and challenging cases.

2.4 Map Refinement

A common strategy for improving estimated correspondences consists in iterative map refinement as a post-processing step, e.g. [Mandad et al. 2017; Solomon et al. 2016; Vestner et al. 2017a,b]. The simplest refinement in the functional maps framework is the Iterative Closest Point algorithm in the spectral domain [Ovsjanikov et al. 2012]. Recently, other more advanced refinement methods for both functional and pointwise maps have been proposed in [Ezuz et al. 2019; Ren et al. 2018], that, respectively, try to minimize the bi-directional geodesic Dirichlet energy, and promote the bijectivity, smoothness and coverage of the correspondences. When shape *collections* are considered, a common strategy is to use cycle consistency constraints [Huang et al. 2014; Wang et al. 2013; Wang and Singer 2013]. Most closely related to ours, is a recent ZoomOut method proposed in [Melzi et al. 2019b], and based on iterative conversion between functional and pointwise maps.

The graph isomorphism problem. We also mention briefly that computing all high quality (e.g., nearly isometric) maps between

Table 1. Our method outputs 8 maps for this pair of human shapes, where the first four maps contain the complete left-to-right, or back-to-front symmetries, and the last four contain a combination across the upper and lower body. We report map quality measures, including orthogonality ("Ortho"), Laplacian Commutativity ("lapComm"), orientation preservation ("Orient"), edge length distortion ("EdgeDist"), and as-rigid-as-possible ("ARAP") distortion. Note that according to most metrics all maps are comparable.

Measure	map1	map2	map3	map4	map5	map6	map7	map8
Ortho	19.9	20.3	19.9	19.9	18.6	17.3	19.3	20.5
lapComm	12.0	11.3	16.1	16.7	21.2	21.6	23.0	24.3
Orient	0.72	1.24	1.28	0.85	0.79	1.23	0.91	1.18
EdgeDist	1.46	1.97	2.56	3.73	4.54	4.30	5.73	4.57
ARAP	1.28	1.39	1.47	1.45	2.99	2.90	3.36	3.36

shapes is related to counting *all isomorphisms* between graphs. This problem has been studied from a theoretical perspective, [Kobler et al. 2012; Mathon 1979], and although complexity equivalence results exist, no polynomial time algorithm is known.

3 MOTIVATION, BACKGROUND & NOTATION

Our goal is to find multiple high-quality maps on a shape pair or a single shape that encode different symmetries. In order to motivate our work, we would like to briefly review three simple solutions to the problem.

The first idea is to exhaustively enumerate all maps in the point-wise map space (with size $O(n^n)$, n is the number of vertices) and select some maps based on some criterion (e.g., geodesic distortion energy). This is infeasible when the shapes have more than about 10 vertices due to the size of the space.

The second idea would be to use random sampling to only evaluate a subset of the maps. However, the probability of obtaining a high-quality map from random sampling is almost zero given the large size of the search space.

A third idea would be to take an existing optimization algorithm and start it with multiple different initializations. Common approaches for initializing maps include using pointwise descriptors, e.g., [Eisenberger et al. 2019; Vestner et al. 2017b] or initial segment or landmark correspondences [Ezuz et al. 2019; Kleiman and Ovsjanikov 2018; Ren et al. 2018]. For these methods it is actually very challenging to modify the initialization in a meaningful way so that the optimization methods would explore different parts of the map space. It would be necessary to run these methods with different random initializations. However, there is also no guarantee that different parts of the space will be explored and that a random initialization will lead to a meaningful map. While this approach is more promising than randomly sampling maps, it would still require a large number of random initializations. We will compare to this approach as a baseline to demonstrate that only few distinct maps can be recovered compared to our method (e.g. Table 3)

A second challenge also becomes evident. Simply proposing a set of maps is not sufficient for an application. We need to have a map

selection algorithm that either selects the direct map, a particular symmetric map, or all meaningful symmetric maps from a set of candidate maps.

We propose a novel algorithm, called *MapTree*, that tackles these two challenges. First of all, instead of dealing with the unorganized point-wise map search space, we propose to solve the same problem in the *spectral* domain based on the following observations. (1) each point-wise map can be represented as a functional map in a reduced basis [Ovsjanikov et al. 2012]. (2) the functional-map representation utilizes the eigenfunctions associated with specific frequencies (ordered eigenvalues). Therefore, the functional-map space can be well organized w.r.t. the frequencies which makes it much easier for map exploration (3) a single point-wise map can be converted to functional maps with different dimensions, where the low dimension encodes the global information of the point-wise map (such as the symmetry orientation), while the high dimension encodes the local details of the map. Therefore, to disambiguate two maps with good-quality, we can convert them to functional maps with different dimension for comparison.

Based on these observations, we propose a tree structure to explore the functional map space, where at low dimensions we can enumerate all possible functional maps in good-quality. The idea of exploring low-dimensional functional maps has been used for symmetry detection in e.g., [Nagar and Raman 2018] and to some extent in [Wang and Huang 2017]. However, those methods are limited by specific choice of the dimensionality (typically very low such as 6-10) in order to maintain efficiency. Moreover, our method is not limited to symmetry detection and can naturally be applied to compute multiple maps across shapes.

Here we clarify the notations that are used in our method. Given two shapes S_1 and S_2 represented as triangle meshes, we associate to each the Laplace-Beltrami operator using the cotangent weight discretization [Meyer et al. 2003]. The first k eigenfunctions of this operator on shape S_i form a basis denoted by $\Phi_{S_i}^{(k)} = \left[\varphi_1^{S_i}, \varphi_2^{S_i}, \cdots, \varphi_k^{S_i}\right]$, having the eigenfunctions stored as columns in a matrix.

Given a pointwise map $T_{12} : S_1 \rightarrow S_2$, that maps each vertex in shape S_1 to a vertex in shape S_2, we represent it with a binary matrix Π_{12}, such that the $\Pi_{12}(i, j) = 1$ if $T_{12}(i) = j$ and 0 otherwise. Throughout our paper, we use Π to denote pointwise correspondences implicitly restricting the matrix to be binary with exactly one value 1 at each row.

As introduced in [Ovsjanikov et al. 2012] the functional map representation of T_{12} is a linear transformation mapping *functions* on S_2 to *functions* on S_1 (note the change in direction compared to T_{12}). When a functional map corresponds to a pointwise map it can be written as a matrix:

$$C_{21} = \Phi_{S_1}^{\dagger} \Pi_{12} \Phi_{S_2}, \tag{1}$$

where † denotes the Moore-Penrose pseudo-inverse.

4 MAP SPACE EXPLORATION

Our overall objective is to recover a set of high quality maps between a given shape pair. As mentioned above, the recovered maps should be sufficiently different and, ideally all be approximately smooth and near isometric.
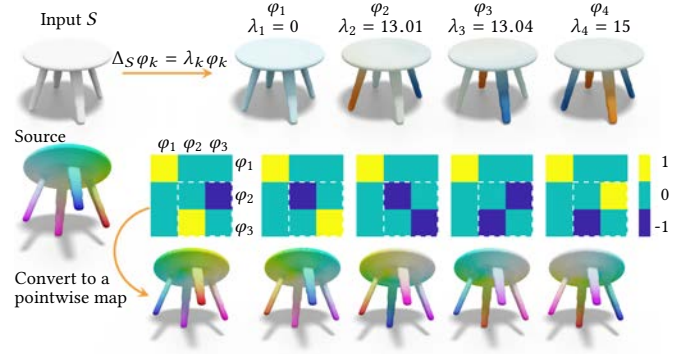

Fig. 5. Illustration of repeated eigenvalues. *Top*: A table with four rotational and four reflectional symmetries, and the first four eigenvalues/functions of its LB operator. Note that the second (φ_2) and the third eigenfunctions (φ_3) correspond to similar eigenvalues. *Middle and Bottom*: we plot different functional maps and associated pointwise correspondences via color transfer. Note that that different permutations between φ_2 and φ_3 lead to different self-symmetric maps.

To tackle this problem, our main idea is to exploit the spectral map representation, in order to efficiently enumerate accurate low-dimensional functional maps, which we refine to near-isometric pointwise correspondences. We organize the functional maps into a tree structure, where each level corresponds to the map size, and a node corresponds to all functional maps that share a particular leading principal submatrix. Our general approach then consists in progressively expanding this tree to obtain high quality diverse solutions. In this section, we first discuss how the spectral techniques can help us explore the map space. We then describe our approach for recovering the map tree hierarchy starting with a pair of shapes.

4.1 How to efficiently explore the space of sufficiently different but high-quality maps?

As noted above, Fig. 2 illustrates that the commonly used geodesic distortion energy is strongly non-convex with multiple strong local minima. Fig. 3 further shows that two maps with similar geodesic distortion error can encode completely different symmetries. Finally, in Table 1 we show another shape pair, and report multiple commonly-used map quality criteria. Remark that these criteria cannot disambiguate maps with different symmetry effectively.

To better structure the space of different maps, we propose to use the multi-scale functional map representation. Namely, for each pointwise map T_{12} from S_1 to S_2, we can convert it into a functional map C_{21} by Eq. (1) of a particular size. Our general observation is that small (low-dimensional) functional maps provide a convenient way of structuring the map space for three reasons: 1) they allow to encode sufficiently different pointwise maps efficiently 2) as demonstrated below they can be explored effectively due to their small size, 3) they can be *refined* to obtain accurate pointwise correspondences.

The discriminating ability of low-dimensional functional maps comes from two complementary results. First, given two pointwise maps that are sufficiently *close* to each other, their functional map representation, *especially in the low frequencies* will be similar. Indeed, we have the following theorem (proved in Appendix A):

ALGORITHM 1: Functional Map Expansion Rules

Input : Two shapes S_i (i = 1,2) with corresponding eigenfunctions $\{\varphi_k^{S_i}\}$ and eigenvalues $\{\lambda_k^{S_i}\}$; an initial functional map $C_{\text{ini}} \in \mathbb{R}^{k_2 \times k_1}$; A refinement method $\mathcal{R}$

Output: A set of functional maps C expanded from C_{ini}

(1) group similar eigenvalues for both shapes: find the index p_i for shape S_i such that $\lambda_{p_i}^{S_i} - \lambda_{k_i}^{S_i} \leq \epsilon < \lambda_{p_i+1}^{S_i} - \lambda_{k_i}^{S_i}$, $i = 1, 2$

(2) enumerate the mappings between $\{\varphi_{k_1}^{S_1}, \cdots, \varphi_{p_1}^{S_1}\}$ and $\{\varphi_{k_2}^{S_2}, \cdots, \varphi_{p_2}^{S_2}\}$; store the assignment as a $p_2 \times p_1$ matrix C_{perm} (with entry 0, ±1)

(3) construct $C = \begin{pmatrix} C_{\text{ini}} & \mathbb{0} \\ \mathbb{0} & C_{\text{perm}} \end{pmatrix}$, where $\mathbb{0}$ is a zero-matrix

(4) refine each C using the given refinement method $\mathcal{R}$. Convert the refined maps to functional maps of size equal to that of C, and output the result.

ALGORITHM 2: Map Tree Exploration

Input : A pair of shapes S_i (i = 1,2) with corresponding eigenfunctions $\{\varphi_k^{S_i}\}$ and eigenvalues $\{\lambda_k^{S_i}\}$; some stopping criterion and some tree pruning rules

Output: A set of maps and the corresponding map tree

Initialization : Initialize the map tree with a node $C_{\text{ini}} = \frac{\sum \varphi_1^{S_2}}{\sum \varphi_1^{S_1}}$ with the status "un-explored"

while *Stopping Criterion Not Met* **do**

    for *each **un-explored leaf** C^* of the map tree* **do**

        (1) Apply Algo. 1 and obtain a set of functional maps C

        (2) Update the map tree by attaching each $C \in C$ to the node C^* with the default status "un-explored"

        (3) Prune the tree w.r.t. the given pruning rule and label the pruned leaves as "explored".

    end

end

THEOREM 4.1. *Given two pointwise maps $T_{12}^1, T_{12}^2 : S_1 \to S_2$ that are δ-close, their functional map matrix representation C_{21}^1, C_{21}^2 will satisfy the following bound $\sum_j \left(C_{21}^1(j, i) - C_{21}^2(j, i) \right)^2 < (\delta c \lambda_i)^2 A_1 \; \forall i$.*

Here two maps T^1, T^2 are δ-close if for any vertex v the geodesic distance between $T_{12}^1(v)$ and $T_{12}^2(v)$ is smaller than δ, c is some constant (which depends on S_2), A_1 is the surface area of shape S_1, and λ_i is the i-th eigenvalue of shape S_2. This bound suggests that for two maps that are close to each other, i.e., δ is small, we need larger i (higher frequencies) to disambiguate two maps.

Moreover, while Theorem 4.1 only establishes the bound in one direction, showing that nearby pointwise maps will result in similar low frequency functional maps, a complementary result can also be found. Specifically if two pointwise maps are *sufficiently different* their low frequency functional map representations must also be different. Intuitively, this is because for two different pointwise maps it must be possible to find indicator functions of regions that are mapped differently by them. If those regions are large enough so that their projection onto the low frequency eigenfunctions are significant, this implies that the functional maps must be different in the sense of the Frobenius norm. We formalize this intuition in Appendix A.2.

In summary, there are several advantages of the spectral representation via functional map: (1) the Frobenius-norm between two functional maps is a cheap but relatively accurate approximation of the distance between two pointwise maps. (2) for two maps that are far from each other (e.g., with different symmetry), we only need relatively low frequencies to convert functional maps that can disambiguate the two pointwise maps (see Fig. 5 for an example). (3) for two maps that are close to each other, we then need higher frequencies for disambiguation. Therefore, we can organize pointwise maps in the spectral domain along the frequencies.

To exploit these properties we propose a pipeline that effectively explores low-frequency functional maps and then refines them to higher-frequency functional maps and accurate pointwise correspondences. For the latter stage, we extend the approach proposed in [Melzi et al. 2019b] that allows to obtain high quality maps from small low frequency functional maps.

4.2 Hierarchical Map Recovery

As suggested above, we propose to organize the functional maps in a tree structure (which we call the Map Tree) from low frequency to high frequency. The nodes at each level k of the tree represent different functional maps of size k-by-k, and two nodes across adjacent levels k and $k + 1$ are connected if and only if the functional map matrices have the same leading principal k-by-k submatrix. Such a map tree structure provides a convenient way of map organization using the ordered nature of the Laplacian spectrum. It also helps to explore the map space efficiently.

Our main idea is to build the map tree in a coarse to fine fashion. Namely, we start with the root node and progressively test whether the descendant nodes can arise from a high quality map. Our basic algorithm, which we call **Algorithm A**, is summarized as follows:

(1) Initialize the set $\mathcal{S} = \{C_0\}$ to contain the single functional map of size 1×1 equal to the ratio of the values of the constant basis functions.

(2) For each $C_i \in \mathcal{S}$:
  (a) Let C_i^+ and C_i^- be two functional maps obtained by adding an extra row and column to C_i, with zero values, except for the bottom right diagonal, set to 1 and −1 respectively.
  (b) Apply a map refinement algorithm $\mathcal{R}$ given C_i^+ as initialization. If the resulting pointwise map T is of sufficiently high quality, convert T to a functional map of the same size as C_i^+ and add it to $\mathcal{S}$. Do the same for C_i^-.
  (c) Repeat the previous two steps until all functional maps in $\mathcal{S}$ have size greater than some threshold κ.

Intuitively, this algorithm progressively tries different diagonal functional map initializations and applies a refinement algorithm $\mathcal{R}$ to them. In order to analyze this map exploration approach, we need an assumption on the refinement algorithm. Specifically, we will call algorithm $\mathcal{R}$ *complete* if, given an initial functional map C of size $k \times k$, and assuming there exists at least one isometric map, whose functional map representation has C as the leading principal $k \times k$ sub-matrix, then algorithm $\mathcal{R}$ will recover such an isometry, and an empty set otherwise.

The following theorem shows that under certain assumptions, **Algorithm A** is guaranteed to recover the map tree up to level κ.

THEOREM 4.2. *Suppose the Laplacians of* S_1, S_2 *have the same eigenvalues, none of which are repeating. Let* $\mathcal{M}$ *denote the tree constructed from all isometries between* $S_2 \to S_1$. *Further suppose that the refinement algorithm* $\mathcal{R}$ *is complete. Then the subtree of* $\mathcal{M}$ *until level* κ *will coincide with the tree built from the output of Algorithm A.*

PROOF. See Appendix. □

We remark that although there are 2^k possible functional maps of size $k \times k$ that have $+1$ or -1 on the diagonal the complexity of **Algorithm A** depends on the number of recovered isometries. For example if there exists only one isometric correspondence, this algorithm will only test $2k$ initializations.

Practical Implementation. We propose the Map Tree Exploration algorithm (Algo. 2) with Functional Map Expansion Rules (Algo. 1) to implement the idea of Algorithm A with several practical modifications.

First, we take special care in the presence of eigenvalues that are close to being repeating, which can arise particularly for symmetric shapes (see Fig. 5 for an example). In this case, as has been observed in, e.g. [Ovsjanikov et al. 2008; Wang and Huang 2017] the corresponding eigenfunctions will form an isometry-invariant *subspace*. Therefore, we must allow not only ± 1 on the diagonal, but also consider all orthogonal *combinations* among the corresponding eigenfunctions. To achieve this, we need two building blocks: (1) detecting repeating eigenvalues, and (2) enumerating combinations among the grouped eigenfunctions. Importantly, since our method tries a set of initializations for refinement, we only need a *conservative* estimate for declaring eigenvalues to be repeating, since poor initalizations will be discarded. Therefore, we detect repeating eigenvalues by simple thresholding. For enumerating possible initializations we consider all permutations (allowing sign flips) among grouped eigenfunctions. We have observed this to perform remarkably well, due, in particular, to the robustness of our refinement method, although theoretically the invariant subspace can be continuous. Algorithm 1 describes the details of the functional map expansion rules used in practice. We stress that our approach is geared towards a *discrete* set of maps, and we do not consider continuous spaces, such as isometries between two spheres. Also note that the detected repeating eigenvalues of two shapes might not have the same size, therefore, the expanded functional map after applying Algorithm 1 can be rectangular.

Algorithm 2 describes our complete map tree construction method, which uses the expansion Algorithm 1 as a subroutine. Note that Algorithm 2 also relies on pruning rules for deciding when a map is of sufficiently good quality. In practice we use the following:

(1) Discard a refined map if its orthogonality error $E_{\text{ortho}}(C) > \epsilon_1$ or its Laplacian Commutativity error $E_{\text{lapComm}}(C) > \epsilon_2$.

(2) Prune the leaf C that converges to the same functional map as another leaf.

Since the orthogonality ($E_{\text{ortho}}(C) = \|CC^T - I\|_F^2$) and the Laplacian Commutativity ($E_{\text{lapComm}}(C) = \|C\Delta_1 - \Delta_2 C\|_F^2$) are strong indicators for detecting bad functional maps, We use them to remove the leaves/functional maps that are unlikely to be good. We
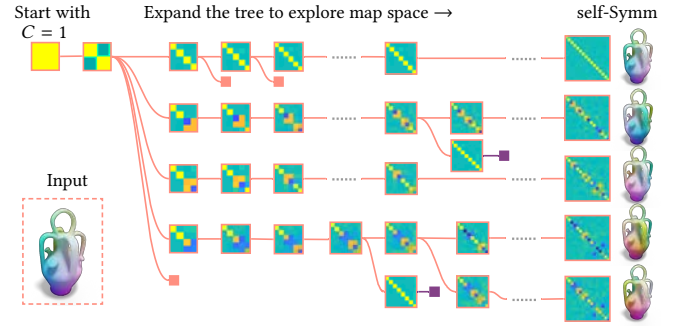

Fig. 6. Illustration of Algorithm 2 for exploring the self-symmetry space of a vase. The red/purple squares represent nodes pruned due to poor quality and duplicates of refined maps respectively. The last column shows the final computed functional and corresponding point-wise maps that capture different self-symmetries.

introduce rule (2) to avoid initializations that converge to the same functional map after refinement. The rationale behind it is that, as we show in the next section, our new refinement method has strong convergence power, capable of producing high quality maps even from poor initializations. Thus, we use rule (2) to remove initializations that lead to the same maps to avoid redundancy. Fig. 6 shows how a map tree is constructed using our algorithm.

the

4.3 Bijective ZoomOut

In Algorithm 1 (step 4), we need a refinement method $\mathcal{R}$ to refine the expanded functional map from the enumeration. For efficiency, we can use some spectral-based methods such as ICP [Ovsjanikov et al. 2012] or ZoomOut [Melzi et al. 2019b]. ZoomOut is the current state-of-the-art refinement for near-isometric shape matching, which progressively increases the frequency during the refinement. Specifically, it proposes the following energy to minimize, where each principal $k \times k$ submatrix is constrained to be orthonormal:

$$E_{\text{ZM}}(C_{21}) = \sum_k \frac{1}{k}\left\|C_{21}^{(k)}(C_{21}^{(k)})^T - I_k\right\|^2 \tag{2}$$

$$\exists\Pi_{12}, \text{s.t. } C_{21} = \Phi_{S_1}^{\dagger}\Pi_{12}\Phi_{S_2}$$

To remove the direction bias and improve upon ZoomOut, we propose to add the bijectivity constraint on the two functional maps mapping in different directions. Specifically, we propose to minimize the following energy:

$$E(C_{12}, C_{21}) = E_{\text{ZM}}(C_{12}) + E_{\text{ZM}}(C_{21}) +$$

$$\sum_k \frac{1}{k}\left\|C_{12}^{(k)}C_{21}^{(k)} - I_k\right\|^2 + \sum_k \frac{1}{k}\left\|C_{21}^{(k)}C_{12}^{(k)} - I_k\right\|^2 \tag{3}$$

$$\exists\Pi_{12}, \Pi_{21}, \text{s.t. } C_{12} = \Phi_{S_2}^{\dagger}\Pi_{21}\Phi_{S_1}, C_{21} = \Phi_{S_1}^{\dagger}\Pi_{12}\Phi_{S_2}$$

We use a similar half-quadratic splitting technique as [Melzi et al. 2019b] to decouple the pointwise maps Π and the corresponding functional maps C. We provide the pseudo-code and implementation details in the appendix.

5 RESULTS

In this section, we show extensive quantitative and qualitative results on three applications: (1) multi-solution shape matching,

Table 2. We compare our MapTree method to other simple baselines on a pair of human shapes from the FAUST dataset. We sample 500 random point-wise maps and apply different refinement methods. We then report the quality of the refinement methods and the total runtime.

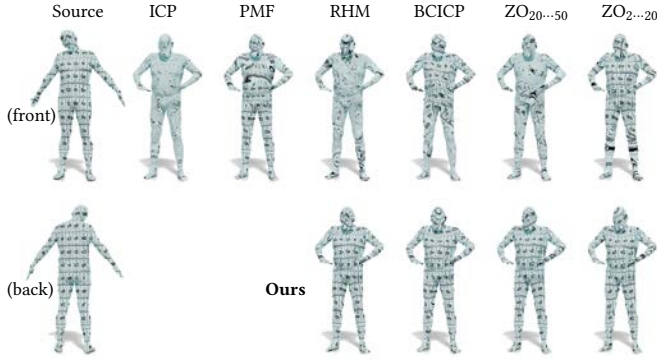| Methods	# diffMaps	Acc	GeoDist	Dirichlet	Conformal	runtime (h)
Random Ini	500	0.623	2.088	1092	6.958	-
ICP$_{50}$	500	0.463	1.880	75.78	21.83	1.946
PMF$_{50}$	127	0.050	**0.286**	44.07	4.049	82.21
BCICP$_{50}$	500	0.154	1.727	30.49	13.83	63.62
RHM$_{50}$	500	0.285	1.726	53.76	6.956	5.841
ZoomOut$_{20\cdots50}$	500	0.372	1.767	48.09	8.697	0.221
ZoomOut$_{2\cdots20}$	454	0.031	0.775	13.68	2.944	0.080
MapTree (Ours)	**10**	**0.019**	0.375	**11.26**	**1.815**	**0.002**

Fig. 7. We show the shape pair that is tested in Table 2. *Top*: we show the best refined map of each method. *Bottom*: we show the four obtained maps using our method, including the direct map, symmetric map, back-to-front map, and doubly flipped map via texture transfer.

where our method is applied to produce multiple maps/symmetries; (2) standard single-solution non-rigid shape matching; (3) self-symmetry detection. In addition, we provide information on the parameters of the algorithm and its runtime.

To compare different methods, we measure the accuracy and the smoothness of the computed maps:

- **Accuracy**: The *accuracy* of a given map is obtained by measuring the *geodesic* distance between the computed correspondence and the given ground-truth correspondence, where the geodesic distance is normalized by the surface area to make this measurement comparable across different shape pairs.
- **Smoothness**: We use three different metrics to measure the smoothness of a given map, namely, the geodesic distortion, Dirichlet energy, and conformal distortion:
  - **Geodesic Distortion**: Assume that matrix $G_k(i, j)$ stores the normalized geodesic distances between all pairs of vertices i and j for shape S_k. Then for a given map $T_{12} : S_1 \rightarrow S_2$, we measure its *geodesic distortion* on every pair of (i, j) as:

$$\sum_i \sum_j \left(G_1(i, j) - G_2\big(T_{12}(i), T_{12}(j)\big) \right)^2$$

  - **Dirichlet Energy**: We measure the Dirichlet energy of the *mapped* vertex positions on the target shape.
  - **Conformal Distortion**: we use the same metric as proposed in [Ezuz et al. 2019], which is modified from [Hormann and Greiner 2000]: for each face f in S_1, we compute $\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} - 2$,

Table 3. Quantitative comparison to (1) RandIni + ZoomOut, (2) IntSymm, and (3) OrientRev + ZoomOut. We report the number of detected high-quality maps (i.e., corresponding to some intrinsic symmetries) and the runtime on several shapes with different resolution.

Shape	Figure Number	#Vtx	# High-quality maps				Runtime (sec)			
			(1)	(2)	(3)	Ours	(1)	(2)	(3)	Ours
Table	Fig. 5	14K	1	1	1	**7**	2359	8.63	13.0	39.0
Vase	Fig. 6	15K	2	1	1	**4**	1683	4.4	11.6	25.2
Cup	Fig. 10	30K	**3**	1	1	3	9474	19.8	34.6	30.0
Glasses	Fig. 11	2K	1	1	1	**3**	44.61	0.82	1.71	3.24
Knot	Fig. 12	5K	3	0	1	**44**	350.6	-	3.23	56.2
Legs	Fig. 19	12K	1	1	1	**3**	2016	2.15	11.2	3.69
Baby	Fig. 26	2K	1	1	1	**3**	37.22	1.24	1.56	6.50
Human pair	Fig. 7	5K	2	-	-	**4**	286.2	-	-	7.21
Tiger pair	Fig. 8	5K	1	-	-	**2**	311	-	-	6.27
Gorilla pair	Fig. 9	5K	2	-	-	**4**	173.4	-	-	9.96
Cat pair	Fig. 17	5K	1	-	-	**2**	156.7	-	-	10.8
Centaur pair	Fig. 18	5K	**2**	-	-	2	140.6	-	-	11.5

where σ_i are the singular values of the linear transformation that maps f from S_1 to S_2. We then average this value over all the faces in S_1.

5.1 Multi-Solution Shape Matching

Our algorithm MapTree has no published direct competitor for the problem of multi-solution shape matching. We therefore try to create a baseline method that consists of a modification of existing techniques. The most suitable idea seems to be to start with many different random point-to-point maps as initialization and use an existing optimization technique. Specifically, we randomly generated 500 pointwise maps between a human shape from the FAUST-remeshed dataset, and apply different refinement methods including recent state-of-the-art methods RHM [Ezuz et al. 2019] and ZoomOut [Melzi et al. 2019b]. A suitable competitor must be able to at least find the symmetric or the direct map in such an example. Therefore, when measuring the map accuracy, we consider both the direct and symmetric direction and pick the one that gives a smaller error. Accuracy and different smoothness metrics are reported in Table 2.

As a conclusion, we can observe that only ZoomOut works in combination with random initialization. Other methods have an excessive computation time (e.g. over 50h for PMF and BCICP) and with the exceptions of PMF they cannot converge to competitive high-quality maps for any of the 500 random initializations. See Fig. 7 for a visual comparison of the best map found by each method. We thus select random initialization + ZoomOut as baseline method.

For our quantitative evaluation, we would like to test how many distinct high-quality near isometric maps can be recovered by different methods. To simplify this test, we only count near isometric maps associated with a symmetry and do not consider other low energy maps such as twisted maps as solution. Since we only have ground truth data for some of the maps, we use visual inspection to decide if the maps are smooth and correspond to a symmetry and the geodesic distortion metric to confirm that they are low energy. We compiled a small test dataset consisting of 7 interesting shapes for the recovery of maps from the shape to itself and 5 shape pairs for the recovery of maps between two shapes. In this test, we compare to three competitors: 1) our baseline consisting of 500
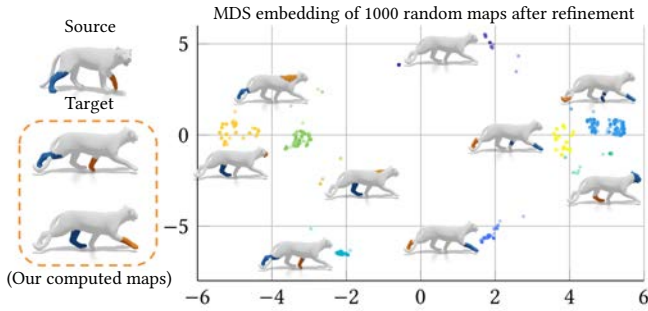
Fig. 8. *Left:* For this pair of tigers, we apply our method and obtain the direct and the symmetric map visualized via segment transfer. *Right:* We randomly generate 1000 point-wise maps and apply ZoomOut for refinement. We then embed the refined maps in 2D using MDS. We apply a simple K-means to cluster the 2D points and color each point/map accordingly. For each cluster, we show one representative map that is closest to the direct/symmetric map. See Fig. 16 in Appendix for the corresponding map visualization via color transfer.
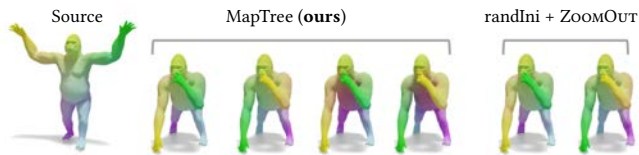

Fig. 9. For this pair of gorillas, our method obtains four semantically meaningful maps. As a comparison, applying ZoomOut to 500 random point-wise maps obtains two maps.

random initializations + ZoomOut 2) IntSymm [Nagar and Raman 2018] and 3) OrientRev [Ren et al. 2018] + ZoomOut. The last two competitors are listed for completeness only, as they are the current state of the art methods for single-solution symmetry detection. These methods can only recover a maximum of one map by design. Our results are shown in Table 3. We can observe that our method is the only algorithm that can reliably recover multiple high-quality near isometric maps. The result of random initialization + ZoomOut recovers fewer maps except for the cup and the centaur pair.

We provide an example visualization to show why random initialization + ZoomOut has difficulty recovering multiple distinct maps using multi-dimensional scaling (MDS) to embed the refined maps in 2D (see Fig. 8). We cluster the embedded points/maps and show the best map of each cluster. To better visualize the map, we show how each map transfers the front-left leg and the back-right leg. The visualization of the complete map via color transfer can be found in Fig. 16 in the appendix. We can notice that most of the maps are of low-quality, e.g., they map a leg to the back or to the head. Some maps are also not smooth and map one leg to two legs. Only one cluster contains high quality maps, but none of the maps recover the symmetry. This shows that the size of the map search space is too large so that it is almost impossible to sample the random initial maps that can lead to different high-quality meaningful maps. In contrast, our method recovers the two meaningful (direct and symmetric) maps as shown in the insets of Fig. 8 and Fig. 16. We show another example where our tree structure can help to find more meaningful maps on a pair of gorillas in Fig. 9.
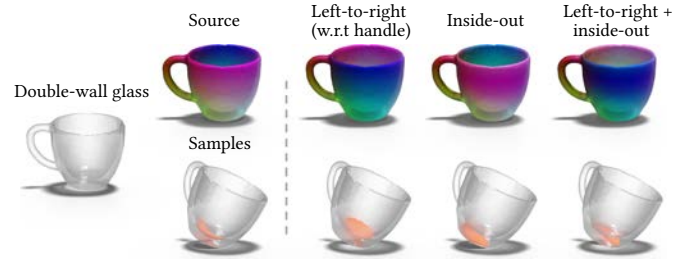

Fig. 10. Symmetry of a double-wall glass cup computed with our method. *Top*: three self-symmetric maps visualized via color transfer. *Bottom*: to better visualize the inside-out symmetries, a small set of samples on the bottom of the *inner wall* of the cup are highlighted in orange. We then visualize how these samples are mapped according to the symmetries shown in the top row. Note that for the two inside-out symmetric maps, the mapped samples are on the bottom of the outer wall.

Fig. 11. Symmetries detected with our method on a "glasses" shape visualized via color-transfer. Observe the detected left-to-right, upside-down, and the simultaneous double symmetry flip.

Fig. 12. For this challenging shape with smooth symmetry group, the IntSymm method fail to output a symmetric map and OrientRev method produce a noisy map with mixed symmetry. As a comparison, our method still outputs several high-quality self-symmetric maps.

Finally, we showcase three models to visualize interesting maps that can be recovered by our method. For example, on the cup model (Fig. 10) we can recover an interesting inside-out map and on the glasses (Fig. 11) there are left-to-right and upside-down symmetries. Fig. 12 shows a challenging case of a knot shape with a smooth/continuous symmetry group. In this case, IntSymm even fails to output a single map (instead, it outputs an identity map) and "OrientRev" produces a rather noisy map, while our method is still able to output several accurate self-symmetric maps.

5.2 Non-rigid shape matching

We apply our map tree exploration Algorithm 2 to the SHREC'19 Challenge dataset [Melzi et al. 2019a], which involves 430 shape pairs from 44 shapes with different mesh resolution, triangulation, and partiality. We consider this to be one of the most challenging datasets available for shape matching and we use it to demonstrate that our MapTree algorithm can significantly outperform the state-of-the-art methods in traditional single solution shape matching. We compare to the best and most closely-related shape matching methods and list the average error, geodesic distortion, and runtime of each method in Table 4.

In single-solution shape matching the algorithms are expected to output a single map with the direct orientation (i.e, map the

Table 4. Quantitative results on the SHREC'19 Challenge. For the map-tree based methods, the geodesic distortion (shown in brackets) is averaged over *all produced maps*.

Method \ Measurement	Error ($\times 10^{-3}$)	Geodesic Distortion	Runtime (sec)
BIM [Kim et al. 2011]	81.2	2.314	169
Genetic [Sahillioğlu 2018]	339.4	5.458	25.11
OrientOp (Ini) [Ren et al. 2018]	141.2	1.953	5.81
Ini + ICP [Ovsjanikov et al. 2012]	130.2	1.596	3.82
Ini + PMF [Vestner et al. 2017b]	143.5	1.862	429
Ini + BCICP [Ren et al. 2018]	85.0	1.147	209
Ini + RHM [Ezuz et al. 2019]	125.2	1.403	25.2
Ini + ZoomOut [Melzi et al. 2019b]	118.5	1.367	0.38
MapTree (Ours) - GT selection	**28.42**	**0.93** (1.16)	35.5
MapTree (Ours) - Auto selection	**39.99**	**0.94** (1.16)	35.5

BIM OrientOp ICP PMF BCICP RHM ZoomOut
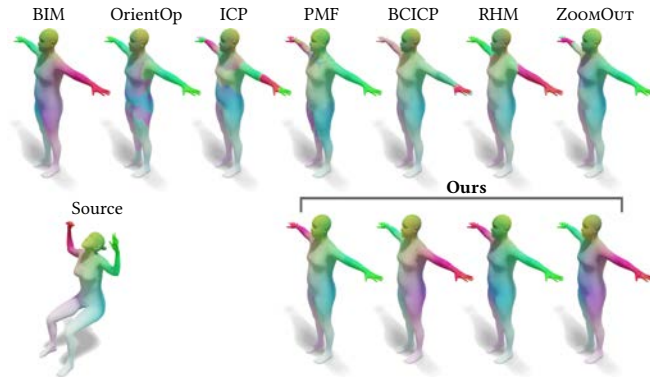
Ours

Source

Fig. 13. Qualitative comparison of maps obtained with different methods. When the initialization (OrientOp) is of poor quality, the refinement methods including ICP, PMF, BCICP, RHM, and ZoomOut fail to recover a good map. Meanwhile, BIM outputs a back-to-front symmetric map which is not preferable for symmetry detection. In contrast, our method can output a set of smooth maps with different orientations for different applications.

left of a shape to the left, assuming there is a left-to-right symmetry). Competing methods are: BIM [Kim et al. 2011], and several functional map based methods, where we use the best known automatic initialization [Ren et al. 2018], and refine it using different methods, such as ICP [Ovsjanikov et al. 2012], PMF [Vestner et al. 2017b], BCICP [Ren et al. 2018], RHM [Ezuz et al. 2019], and ZoomOut [Melzi et al. 2019b]. Fig. 13 shows an example of the maps obtained from different methods. In Table 4, we also compare to the genetic algorithm [Sahillioğlu 2018], where sparse correspondences are computed and interpolated. As shown in the table, the average direct error (0.339) and the average error that allows symmetry flips (0.289) are significantly higher than ours (0.047). In Fig. 22 we visualize the computed maps by the genetic algorithm on three example shape pairs. Note that the genetic algorithm can produce accurate sparse correspondences but fails to produce high-quality and smooth dense maps.

We propose two versions of our algorithm: semi-automatic and automatic. In the semi-automatic method we assume that the user selects the best map among the available maps. We simulate this user selection by selecting the left-to-left direct map closest to the ground truth and denote this method as *GT selection*. The average error of our method is 28 while the best baseline gives 81.

Table 5. Self-symmetry detection on SHREC'19. We compare the accuracy and different smoothness measurement (including geodesic distortion, Dirichlet energy, and conformal distortion) of the self-symmetric maps from different baselines.

Methods \ Measurement	Accuracy ($\times 10^{-3}$)	GeoDist ($\times 10^2$)	Dirichlet Energy	Conformal Distortion	Runtime (sec)
BIM	83.69	**1.418**	**3.278**	1.970	164
GroupRep	311.1	5.254	13.41	7.787	3.95
IntSymm	**62.50**	1.945	12.17	7.123	2.05
OrientRev (Ini)	137.2	4.682	22.07	12.69	0.52
Ini + ICP	108.9	3.604	10.49	6.235	8.33
Ini + PMF	119.4	2.444	15.98	9.605	425
Ini + RHM	118.7	4.166	7.352	4.369	28.7
Ini + BCICP	96.72	2.466	5.633	3.741	157
Ini + ZoomOut	80.30	2.858	6.601	3.838	6.58
MapTree - GT	**39.62**	1.512	3.763	**0.949**	65.2
MapTree - Auto	**47.48**	1.507	3.833	**0.929**	65.2

The main idea of our automatic selection algorithm is to use cycle consistency. We propose a simple approach to achieve this in the spirit of [Huang et al. 2014], but using the fact that our method produces multiple maps for each shape pair. Specifically, for every pair S_1, S_2 in the benchmark, our algorithm produces 10 maps $\{C_{12}^i\}_{i=1}^{10}$. We perform an automatic map selection in two steps. We first initialize the selection for each shape pair with the map that minimizes the orientation-preservation error introduced in [Ren et al. 2018]. We then iteratively update this selection by considering triplets of shapes, and by minimizing the following cycle consistency energy:

$$
\begin{aligned}
E_{\text{CyCons}}(C_{12}^i) = \sum_{j \in J} &\|C_{1j}^\star C_{j2}^\star - C_{12}^i\|_F + \|C_{12}^i C_{2j}^\star - C_{1j}^\star\|_F \\
&+ \|C_{j1}^\star C_{12}^i - C_{2j}^\star\|_F
\end{aligned}
\tag{4}
$$

where J is the set of all shape pairs in the benchmark that form a triplet with S_1 and S_2. Thus, we update the map selection for the pair S_1, S_2 by selecting C_{12} among the 10 possible maps, the one that minimizes this energy, while keeping the selection for the other shape pairs (identified by the superscript $\star$) in Equation (4) fixed from the previous iteration. This very simple strategy converges remarkably fast, within 2-3 iterations over all map pairs.

The results of the cycle consistency-based automatic selection are also shown in Table 4 as "MapTree (ours) - Auto selection." This combination makes our method a fully automatic method for shape matching, which achieves the state-of-the-art accuracy with a 50% improvement over the best existing baseline. It is also interesting to note that the average geodesic distortion of all the produced maps across the complete dataset is 1.16. This is remarkable, because it shows that most of our recovered maps have smaller geodesic distortion than the direct map recovered by competing methods. It is also an indication that geodesic distortion alone is not sufficient to distinguish between the direct map and other high quality maps.

We perform a second large scale experiment on more than 200 pairs of SCAPE human shapes, where we also achieved a significant improvement over the best baseline by 23%. The details of the test on SCAPE and some additional visualizations including the geodesic distortion and texture transfer can be found in Appendix B.

Fig. 14. Partial intrinsic symmetries of the Neptune statue. *Left*: the input model is not strictly symmetric due to the topology change. *Middle*: we run our algorithm directly and obtain the self-map visualized via color transfer. *Right*: we only highlight the shape regions that have intrinsic symmetry.

5.3 Self-symmetry detection.

Our map tree algorithm can also be used to improve upon the state-of-the-art in single-solution self-symmetry detection. Table 5 reports a quantitative evaluation on self-symmetry detection on the SHREC'19 dataset that consists of 44 shapes with different poses. Our method produces eight self-maps per shape. We then select the preferable self-symmetric map among the 8 maps in two ways: (1) "GT selection": we use the given ground-truth self-symmetric map to select the map with the smallest distance to the ground-truth. (2) "Auto selection": we simply select the map that is far from the identity map with a small Laplacian-commutativity error. Note that this is a fully automatic method. Comparing to the best baseline IntSymm [Nagar and Raman 2018] on this dataset, our method achieves 24% improvement on accuracy with better geodesic distortion. Also note that, the maps produced by our method achieve the smallest conformal distortion compared to all the baselines, and achieve comparable Dirichlet energy and geodesic distortion to BIM, while outperforming all other methods.

Partial intrinsic symmetries. We also applied our self-symmetry exploration algorithm to shapes with *partial* intrinsic symmetries without modification. Fig. 14 shows such an example, where the input Neptune model is not strictly symmetric. We use our method and obtain the self-map as visualized in the middle of Fig. 14. We can see that most of the body region contains a symmetry, while the remaining part does not due to the topology change of having the hand attached to the spear and the feet attached to the statue's base. We can then extract the region with symmetry from the map, as shown in the right of Fig. 14, where only the region that contains symmetry is colored w.r.t. the symmetric vertex via color transfer, and the remaining region is colored white. See Fig. 24, 25 and Fig. 26 in Appendix for more examples. We observe that our method produces high quality results in these challenging cases, due to the robustness and strong convergence power of our map refinement even if, e.g., the bijectivity constraint is not strictly satisfied.

5.4 Robustness

Our method is robust w.r.t. mesh triangulation and decimation. Specifically, the SHREC'19 dataset consists of 44 shapes with different mesh resolution and triangulation. Table 4 and 5 shows that
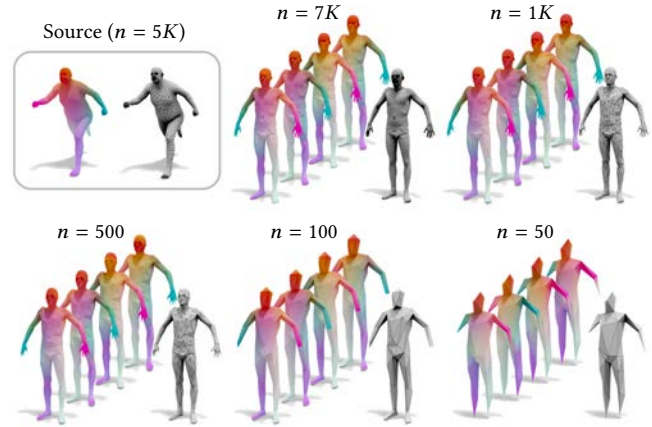

Fig. 15. Robustness w.r.t. decimation. For a human shape with different triangulation (source from SHREC'19, target from FAUST), we downsample the target shape to the number of n vertices. We use the *same* set of parameters and apply our MapTree algorithm. Note that even at the resolution of $n = 100$, our method still obtains four reasonable maps that are consistent with other resolutions.

our method is robust w.r.t. mesh triangulation. We further test the robustness of our method w.r.t. decimation (see Fig. 15). For a given shape pair, we downsample the target shape to different number of vertices. Our method still produces accurate and consistent maps.

5.5 Parameters

For our MapTree construction algorithm, we set ϵ in Algorithm 1 to 1, and the ϵ_1 and ϵ_2 in the pruning rules to 0.5 for most of our tests. For some cases we also set ϵ_1, ϵ_2 to larger values to allow more leaves to be explored with a prior knowledge that the investigated shape (pair) contains more complicated symmetries. We stop the tree construction process when the leaf functional map has size larger than 10 (for non-human shapes) or 20 (for human shapes).

5.6 Runtime & Implementation

We implemented the proposed methods in MATLAB, and conducted the experiments on a workstation with a 3.10GHz processor and 64 GB memory. The (average) runtime for different tests in seconds is reported (see Table 3, 4, 5). We used the sub-sampling strategy to speed-up our MapTree algorithm similar to the one used in [Melzi et al. 2019b]. Specifically, we only sample a small number of points (e.g., 200-500) on the shapes to compute a functional map in our MapTree algorithm. In the vast majority of cases our complete pipeline takes 60 seconds end-to-end. We will release our complete implementation for full reproducibility of all of the results.

6 CONCLUSION, LIMITATIONS & FUTURE WORK

In this paper, we proposed a novel approach to recover multiple high quality correspondences between a shape or a shape pair. To achieve this, we propose a compact tree structure to efficiently organize and explore map space without requiring any landmarks or initialization. We presented a large variety of experiments demonstrating that our method can be used to explore the self-symmetry space or even the partial intrinsic symmetry of a single shape, and

to explore the map space between a shape pair, which is not possible for the previous methods. We also show the quantitative results on challenging datasets showing that our method can be used as an automatic shape matching technique that does not require any descriptors or landmarks. The results show that our method significantly outperforms current state-of-the-art. Our algorithm is easy to implement with a small set of parameters.

Our method still has some limitations. First, we only target discrete sets of maps and symmetries, and do not consider continuous spaces such as the symmetry group of a sphere. Second, while outputting a set of high-quality maps can be beneficial, for the specific application of shape matching, it is not trivial to correctly select the preferred map, even from a small set. Third, we only consider manifold meshes in our setting. However, using a Laplace-Beltrami discretization such as [Belkin et al. 2009] or a recent work [Sharp and Crane 2020], our method could be directly generalized to non-manifold meshes or even point clouds. For partial matching, [Rodolà et al. 2017] can be helpful to design partial functional map expansion rules to potentially extend our work.

We also plan to investigate potential applications that can exploit our map tree structure. For example, Fig. 19 in the Appendix shows that the multiple detected maps can help the shape modeling process. Meanwhile, Fig. 5 shows that the global symmetry information is encoded in the relatively low frequencies. This suggests that our map tree structure encodes level-of-symmetry which might be useful for shape modeling and, e.g., making the triangulation of a given shape symmetric w.r.t. multiple symmetry axes.

ACKNOWLEDGMENTS

REFERENCES

Yonathan Aflalo and Ron Kimmel. 2013. Spectral multidimensional scaling. *PNAS* 110, 45 (2013), 18052–18057.

Noam Aigerman and Yaron Lipman. 2016. Hyperbolic Orbifold Tutte Embeddings. *ACM Transactions on Graphics* 35, 6 (Nov. 2016), 217:1–217:14.

Noam Aigerman, Roi Poranne, and Yaron Lipman. 2015. Seamless Surface Mappings. *ACM Transactions on Graphics (TOG)* 34, 4, Article Article 72 (July 2015), 13 pages.

Marc Arnaudon, Anton Thalmaier, and Feng-Yu Wang. 2017. Gradient Estimates on Dirichlet Eigenfunctions. *arXiv preprint arXiv:1710.10832* (2017).

Omri Azencot, Anastasia Dubrovina, and Leonidas Guibas. 2019. Consistent Shape Matching via Coupled Optimization. *Computer Graphics Forum* 38, 5 (2019), 13–25.

Mikhail Belkin, Jian Sun, and Yusu Wang. 2009. Constructing Laplace Operator from Point Clouds in Rd. In *Proc. Symposium on Discrete Algorithms (SODA)*. 1031–1040.

Silvia Biasotti, Andrea Cerri, Alex Bronstein, and Michael Bronstein. 2016. Recent trends, applications, and perspectives in 3D shape similarity assessment. *Computer Graphics Forum* 35, 6 (2016), 87–119.

Alex Bronstein, Michael Bronstein, and Ron Kimmel. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences* 103, 5 (2006), 1168–1172.

Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. 2017. Embedding shapes with Green's functions for global shape matching. *Computers & Graphics* 68 (2017), 1–10.

Harold Donnelly. 2006. Eigenfunctions of the Laplacian on compact Riemannian manifolds. *Asian Journal of Mathematics* 10, 1 (2006), 115–126.

Nadav Dym, Haggai Maron, and Yaron Lipman. 2017. DS++: a flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 184.

Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. 2019. Divergence-Free Shape Correspondence by Deformation. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 1–12.

D. Eynard, E. Rodolà, K. Glashoff, and M. M. Bronstein. 2016. Coupled Functional Maps. In *2016 Fourth International Conference on 3D Vision (3DV)*. 399–407.

Danielle Ezuz and Mirela Ben-Chen. 2017. Deblurring and Denoising of Maps between Shapes. *Computer Graphics Forum* 36, 5 (2017), 165–174.

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. 2019. Reversible Harmonic Maps Between Discrete Surfaces. *ACM Trans. Graph.* 38, 2 (2019), 15:1–15:12.

Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d'Aspremont. 2013. Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems*. 1016–1024.

Anne Gehre, Michael Bronstein, Leif Kobbelt, and Justin Solomon. 2018. Interactive curve constrained functional maps. *Computer Graphics Forum* 37, 5 (2018), 1–12.

Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. 2019. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4370–4379.

Kai Hormann and Günther Greiner. 2000. *MIPS: An efficient global parametrization method*. Technical Report. ERLANGEN-NUERNBERG UNIV (GERMANY) COMPUTER GRAPHICS GROUP.

Qixing Huang, Fan Wang, and Leonidas Guibas. 2014. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 36.

Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J Guibas. 2008. Non-rigid registration under isometric deformations. *Computer Graphics Forum* 27, 5 (2008), 1449–1457.

Varun Jain, Hao Zhang, and Oliver van Kaick. 2007. Non-rigid spectral correspondence of triangle meshes. *International Journal of Shape Modeling* 13, 01 (2007), 101–124.

Itay Kezurer, Shahar Z Kovalsky, Ronen Basri, and Yaron Lipman. 2015. Tight relaxation of quadratic matching. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 115–128.

Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 79.

Yanir Kleiman and Maks Ovsjanikov. 2018. Robust Structure-Based Shape Correspondence. In *Computer Graphics Forum*. Wiley Online Library.

Johannes Kobler, Uwe Schöning, and Jacobo Torán. 2012. *The graph isomorphism problem: its structural complexity*. Springer Science & Business Media.

Artiom Kovnatsky, Michael Bronstein, Alex Bronstein, Klaus Glashoff, and Ron Kimmel. 2013. Coupled quasi-harmonic bases. *Computer Graphics Forum* 32, 2pt4 (2013), 439–448.

Marius Leordeanu and Martial Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. IEEE, 1482–1489.

Yaron Lipman, Xiaobai Chen, Ingrid Daubechies, and Thomas Funkhouser. 2010. Symmetry factored embedding and distance. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 103.

Yaron Lipman and Thomas Funkhouser. 2009. MÖBius Voting for Surface Correspondence. *ACM Trans. Graph.* 28, 3, Article 72 (July 2009), 12 pages.

Or Litany, Tal Remez, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. 2017a. Deep Functional Maps: Structured Prediction for Dense Shape Correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 5659–5667.

Or Litany, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. 2017b. Fully spectral partial shape matching. *Computer Graphics Forum* 36, 2 (2017), 247–258.

Xiuping Liu, Shuhua Li, Risheng Liu, Jun Wang, Hui Wang, and Junjie Cao. 2015. Properly Constrained Orthonormal Functional Maps for Intrinsic Symmetries. *Computer and Graphics* 46, C (Feb. 2015), 198–208.

Yanxi Liu, Hagit Hel-Or, Craig Kaplan, and Luc Van Gool. 2010. Computational Symmetry in Computer Vision and Computer Graphics. *Foundations and Trends in Computer Graphics and Vision* 5 (01 2010), 1–199.

Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. 2017. Variance-Minimizing Transport Plans for Inter-surface Mapping. *ACM Transactions on Graphics* 36 (2017), 14.

Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. 2016. Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 73.

Diana Mateus, Radu Horaud, David Knossow, Fabio Cuzzolin, and Edmond Boyer. 2008. Articulated Shape Matching Using Laplacian Eigenfunctions and Unsupervised Point Registration. In *Proc. CVPR*. 1–8.

R Mathon. 1979. A note on the graph isomorphism counting problem. *Inf. Process. Lett.* 8, 3 (1979), 131–132.

Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2019a. SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association.

Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. 2019b. ZoomOut: Spectral Upsampling for Efficient Shape Correspondence. *ACM Transactions on Graphics (TOG)* 38, 6, Article 155 (Nov. 2019), 14 pages.

https://doi.org/10.1145/3355089.3356524

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. 2003. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*. Springer, New York, NY, 35–57.

Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. 2006. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 560–568.

Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. 2013. Symmetry in 3D Geometry: Extraction and Applications. *Computer Graphics Forum* 32, 6 (2013), 1–23.

Rajendra Nagar and Shanmuganathan Raman. 2018. Fast and Accurate Intrinsic Symmetry Detection. In *The European Conference on Computer Vision (ECCV)*.

Dorian Nogneng and Maks Ovsjanikov. 2017. Informative Descriptor Preservation via Commutativity for Shape Matching. *Computer Graphics Forum* 36, 2 (2017), 259–267.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30:1–30:11.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. 2017. Computing and Processing Correspondences with Functional Maps. In *ACM SIGGRAPH 2017 Courses*. Article 5, 5:1–5:62 pages.

Maks Ovsjanikov, Quentin Merigot, Facundo Memoli, and Leonidas Guibas. 2010. One Point Isometric Matching with the Heat Kernel. *CGF* 29, 5 (2010), 1555–1564. https://doi.org/10.1111/j.1467-8659.2010.01764.x

Maks Ovsjanikov, Jian Sun, and Leo Guibas. 2008. Global intrinsic symmetries of shapes. *Comp. Graph. Forum* 27, 5 (2008), 1341–1348.

Dan Raviv, Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. 2010. Full and Partial Symmetries of Non-rigid Shapes. *International Journal of Computer Vision* 89 (July 2010), 18–39.

Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2018. Continuous and Orientation-preserving Correspondences via Functional Maps. *ACM Transactions on Graphics (TOG)* 37, 6 (2018).

Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. 2017. Partial functional correspondence. *Computer Graphics Forum* 36, 1 (2017), 222–236.

Emanuele Rodolà, Michael Moeller, and Daniel Cremers. 2015. Point-wise Map Recovery and Refinement from Functional Correspondence. In *Proc. Vision, Modeling and Visualization (VMV)*.

Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. 2019. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*. 1617–1627.

Raif M Rustamov. 2007. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*. Eurographics Association, 225–233.

Yusuf Sahillioğlu. 2018. A genetic isometric shape correspondence algorithm with adaptive sampling. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 1–14.

Yusuf Sahillioğlu and Yücel Yemez. 2013. Coarse-to-fine isometric shape correspondence by tracking symmetric flips. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 177–189.

Nicholas Sharp and Keenan Crane. 2020. A Laplacian for Nonmanifold Triangle Meshes. In *Proc. SGP*.

Meged Shoham, Amir Vaxman, and Mirela Ben-Chen. 2019. Hierarchical Functional Maps between Subdivision Surfaces. *Computer Graphics Forum* (2019).

Justin Solomon, Andy Nguyen, Adrian Butscher, Mirela Ben-Chen, and Leonidas Guibas. 2012. Soft Maps Between Surfaces. *Computer Graphics Forum* 31, 5 (2012), 1617–1626.

Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 72.

Min-Hyuk Sung and Junho Kim. 2013. Finding the M-best consistent correspondences between 3D symmetric objects. *Computers & graphics* 37, 1-2 (2013), 81–92.

Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. 2013. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE TVCG* 19, 7 (2013), 1199–1217.

Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alex Bronstein, Michael Bronstein, and Ron Kimmel. 2017a. Efficient deformable shape correspondence via kernel matching. In *3D Vision (3DV), 2017 International Conference on*. IEEE, 517–526.

Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. 2017b. Product Manifold Filter: Non-rigid Shape Correspondence via Kernel Density Estimation in the Product Space. In *Proc. CVPR*. 6681–6690.

Fan Wang, Qixing Huang, and Leonidas J. Guibas. 2013. Image Co-segmentation via Consistent Functional Maps. In *Proc. ICCV*. 849–856.

Hui Wang and Hui Huang. 2017. Group representation of global intrinsic symmetries. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 51–61.

Larry Wang, Anne Gehre, Michael Bronstein, and Justin Solomon. 2018a. Kernel Functional Maps. *Computer Graphics Forum* 37, 5 (2018), 27–36.

Lanhui Wang and Amit Singer. 2013. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA* 2, 2 (2013), 145–193.

Tuanfeng Y. Wang, Tianjia Shao, Kai Fu, and Niloy J. Mitra. 2019. Learning an intrinsic garment space for interactive authoring of garment animation. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 220:1–220:12. https://doi.org/10.1145/3355089.3356512

Y Wang, B Liu, K Zhou, and Y Tong. 2018b. Vector Field Map Representation for Near Conformal Surface Correspondence. *Computer Graphics Forum* 37, 6 (2018), 72–83.

Kai Xu, Hao Zhang, Wei Jiang, Ramsay Dyer, Zhiquan Cheng, Ligang Liu, and Baoquan Chen. 2012. Multi-scale partial intrinsic symmetry detection. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 181.

Kai Xu, Hao Zhang, Andrea Tagliasacchi, Ligang Liu, Guo Li, Min Meng, and Yueshan Xiong. 2009. Partial intrinsic reflectional symmetry of 3D shapes. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 138.

A THEORETICAL ANALYSIS

A.1 Proof of Theorem 4.1

Suppose we are given two compact surfaces M, N and two smooth maps $T_1, T_2 : N \rightarrow M$. We consider the pullback operators $T_{F1}, T_{F2} : \mathcal{L}_2(M) \rightarrow \mathcal{L}_2(N)$, defined as $T_{F1}(f) = g_1$, where $g_1(x) = f(T_1(x))$, and similarly $T_{F1}(f) = g_2$, where $g_2(x) = f(T_2(x))$.

We will call T_1 and T_2 δ-close if the following is satisfied: for each $x \in N$, $d^M(T_1(x), T_2(x)) < \delta$ for some $\delta \geq 0$. Here d^M denotes the geodesic distance on M.

Now let φ_i be an eigenfunction of the Laplace-Beltrami operator corresponding to eigenvalue $\lambda_i \geq 0$. Our goal is to relate $T_{F1}(\varphi_i)$ to $T_{F2}(\varphi_i)$ using the bound above.

Observe that if $g = T_{F1}(\varphi_i) - T_{F2}(\varphi_i)$ then

$$g(x) = \varphi_i(T_1(x)) - \varphi_i(T_2(x)) \tag{5}$$

Using the results from [Arnaudon et al. 2017; Donnelly 2006] we recover the following bound: whenever M is a smooth compact manifold, there exists a universal constant C (depending on the manifold), such that for any eigenfunction φ_i, its gradient satisfies the following:

$$\|\nabla \varphi_i\|_\infty \leq C\lambda_i \tag{6}$$

Combining this bound with the assumption that T_1 and T_2 are δ-close, and using the gradient theorem for the line integral on the surface we get that:

$$\left|\varphi_i(T_1(x)) - \varphi_i(T_2(x))\right| \leq \delta C\lambda_i \; \forall \; x \in N. \tag{7}$$

This implies that:

$$\|T_{F1}(\varphi_i) - T_{F2}(\varphi_i)\|_2^2 = \int_{x \in N} \left(\varphi_i(T_1(x)) - \varphi_i(T_2(x))\right)^2 d\mu(x) \tag{8}$$

$$\leq (\delta C\lambda_i)^2 \text{Area}(N) \tag{9}$$

Now recall that the ith column of the functional map representation of a pointwise map T encodes the coefficients of $TF(\varphi_i^M)$ expressed in φ_i^N. Therefore, by Plancherel's theorem we obtain Theorem 4.1:

THEOREM. *Given two pointwise maps T_1, T_2 that are δ-close, their functional map matrix representation C_1, C_2 will satisfy the following bound $\sum_j (C_1(j, i) - C_2(j, i))^2 < (\delta C\lambda_i)^2 Area(N)$ for any i.*

Note that this bound implies that the higher frequencies become more and more "unstable". Or, alternatively, in order to disambiguate

pointwise maps that are very close, one must consider higher and higher frequencies.

A.2 Lower bound of functional map representation

Suppose we are given two shapes $\mathcal{M}, \mathcal{N}$, and two pointwise maps between them $T_1, T_2 : \mathcal{N} \rightarrow \mathcal{M}$. This induces two functional maps $T_{F1}, T_{F2} : L^2(\mathcal{M}) \rightarrow L^2(\mathcal{N})$ given via pull-backs. Specifically for any function $f \in L^2(\mathcal{M})$, the corresponding function $g \in L^2(\mathcal{N})$ is given by $T_{F1}(f) = g$, where $g(y) = f(T_1(y))$ for any $y \in \mathcal{N}$ (and similarly for T_{F2} defined via pull-back with respect to T_2).

We define the functional map matrices $\mathbf{C}_1, \mathbf{C}_2$ by expressing T_{F1} and T_{F2} in the Laplace-Beltrami basis. Specifically, $\mathbf{C}_1(i,j) = <T_{F1}(\varphi_j^{\mathcal{M}}), \varphi_i^{\mathcal{N}} >_{\mathcal{N}}$, where $\varphi_j^{\mathcal{M}}$ is the ith eigenfunction of the Laplace-Beltrami operator on $\mathcal{M}$, and $< \cdot, \cdot >_{\mathcal{N}}$ is the standard L_2 inner product for functions on $\mathcal{N}$.

Our goal is to bound the difference between T_1, T_2 and the difference between the matrices $\mathbf{C}_1, \mathbf{C}_2$. Specifically we would like to show that if T_1 and T_2 are *sufficiently different* then the submatrices of $\mathbf{C}_1, \mathbf{C}_2$ will be different as well.

For this we consider any region $\mathcal{R} \subseteq \mathcal{M}$ and denote by $\mathcal{P}_1, \mathcal{P}_2$ its pre-images via T_1 and T_2. Namely, $\mathcal{P}_1 = \{y \in \mathcal{N}, \text{ s. t. } T_1(y) \in \mathcal{M}\}$. Note that if $\iota_{\mathcal{R}}$ is the indicator function of $\mathcal{R}$ and $\iota_{\mathcal{P}_1}, \iota_{\mathcal{P}_2}$, are indicator functions of $\mathcal{P}_1, \mathcal{P}_2$, then we have $T_{F1}(\iota_{\mathcal{R}}) = \iota_{\mathcal{P}_1}$ and $T_{F2}(\iota_R) = \iota_{\mathcal{P}_2}$.

Now observe that for any region $\mathcal{R}$ we have:

$$\|T_{F1}(\iota_{\mathcal{R}}) - T_{F2}(\iota_{\mathcal{R}})\|_2^2 = \int_{\mathcal{N}} (\iota_{\mathcal{P}_1}(y) - \iota_{\mathcal{P}_1}(y))^2 dy$$
$$= \text{Area}(\mathcal{P}_1 \ominus \mathcal{P}_2),$$

where $\ominus$ denotes the symmetric difference of the two sets $\mathcal{P}_1 \ominus \mathcal{P}_2 = (\mathcal{P}_1 \setminus \mathcal{P}_2) \cup (\mathcal{P}_2 \setminus \mathcal{P}_1)$.

Suppose that the given maps T_1, T_2 are *sufficiently different* in the following sense: there exists a region $\mathcal{R}$ s.t. its two pre-images $\mathcal{P}_1, \mathcal{P}_2$ via T_1 and T_2 are such that $\text{Area}(\mathcal{P}_1 \ominus \mathcal{P}_2) > A$ where A some constant. Moreover, suppose that the projection of $\iota_{\mathcal{R}}$ onto the first $k_{\mathcal{M}}$ eigenfunctions of the LB operator on $\mathcal{M}$ is accurate up to ε error: $\|\iota_{\mathcal{R}} - \sum_{i=1}^{k_{\mathcal{M}}} a_i \varphi_i^{\mathcal{M}}\|_2^2 < \varepsilon \text{Area}(\mathcal{R})$, where $a_i = < \varphi_i^{\mathcal{M}}, \iota_{\mathcal{R}} >_{\mathcal{M}}$. Similarly, we assume that the projection of $\iota_{\mathcal{P}_1 \ominus \mathcal{P}_2}$ onto the first $k_{\mathcal{N}}$ eigenfunctions of the LB operator on $\mathcal{N}$ is accurate up to δ error: $\|\iota_{\mathcal{P}_1 \ominus \mathcal{P}_2} - \sum_{i=1}^{k_{\mathcal{N}}} b_i \varphi_i^{\mathcal{N}}\|_2^2 < \delta \text{Area}(\mathcal{P}_1 \ominus \mathcal{P}_2)$, where $b_i = < \varphi_i^{\mathcal{N}}, \iota_{\mathcal{P}_1 \ominus \mathcal{P}_2} >_{\mathcal{N}}$. Finally, for simplicity we will also assume that T_1, T_2 are both locally area-preserving, although a more refined variant of this bound can also be established under less restrictive assumptions (e.g., only assuming bijectivity and smoothness).

Now let $\tilde{\mathbf{C}}_1$ and $\tilde{\mathbf{C}}_2$ be the $k_{\mathcal{N}} \times k_{\mathcal{M}}$ leading principal submatrices of $\mathbf{C}_1, \mathbf{C}_2$ (i.e., sub-matrices obtained by taking the first $k_{\mathcal{N}}$ rows and first $k_{\mathcal{M}}$ columns of $\mathbf{C}_1, \mathbf{C}_2$). Our goal is to bound the Frobenius norm: $\|\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2\|_F^2$ as a function of the constants A, ε, δ introduced above.

For this remark that:

$$\|T_{F1}(\iota_{\mathcal{R}} - \sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}}) - T_{F2}(\iota_{\mathcal{R}} - \sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}})\|_2$$

$$= \|T_{F1}(\iota_{\mathcal{R}}) - T_{F2}(\iota_{\mathcal{R}}) - T_{F1}(\sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}}) - T_{F2}(\sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}})\|_2$$

$$\geq \|\iota_{\mathcal{P}_1 \ominus \mathcal{P}_2}\|_2 - \|T_{F1}(\sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}})\|_2 - \|T_{F2}(\sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}})\|_2.$$

$$= \sqrt{\text{Area}(\mathcal{P}_1 \ominus \mathcal{P}_2)} - \|T_{F1}(\sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}})\|_2 - \|T_{F2}(\sum_{i=k_{\mathcal{M}}+1}^{\infty} a_i \varphi_i^{\mathcal{M}})\|_2.$$

$$(10)$$

Now recall that by assumption of local area preservation we have $\|T_{F1}(f)\|_2 = \|f\|_2$ for any function f. Thus, if we let $\mathbf{a}$ be the first $k_{\mathcal{M}}$ coefficients of $\iota_{\mathcal{R}}$ in the LB basis of $\mathcal{M}$ as introduced above, then:

$$\frac{\|(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\mathbf{a}\|}{\|\mathbf{a}\|} \geq \frac{\sqrt{\text{Area}(\mathcal{P}_1 \ominus \mathcal{P}_2)}(1 - \sqrt{\delta})}{\sqrt{\text{Area}(\mathcal{R})}} - 2\sqrt{\varepsilon}. \quad (11)$$

Finally, the Frobenius norm is bounded by the L2 norm we have:

$$\|(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\|_F \geq \|(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)\|_2 = \max_{\mathbf{x}} \frac{\|(\tilde{\mathbf{C}}_1 - \tilde{\mathbf{C}}_2)(\mathbf{x})\|_2}{\|\mathbf{x}\|_2} \quad (12)$$

$$\geq \frac{\sqrt{\text{Area}(\mathcal{P}_1 \ominus \mathcal{P}_2)}(1 - \sqrt{\delta})}{\sqrt{\text{Area}(\mathcal{R})}} - 2\sqrt{\varepsilon}. \quad (13)$$

A.3 Proof of Theorem 4.2

For convenience, we restate the theorem:

THEOREM. *Suppose the Laplacians of S_1, S_2 have the same eigenvalues, none of which are repeating. Let $\mathcal{M}$ denote the tree constructed from all isometries between $S_2 \rightarrow S_1$, and suppose that the refinement algorithm $\mathcal{R}$ is complete. Then the subtree of $\mathcal{M}$ until level κ will coincide with the tree built from the output of Algorithm A.*

PROOF. Under the assumptions of the theorem, any pointwise isometry T_{12} will induce a *diagonal* functional map with values that are ±1. This is simply because an isometry preserves geodesics (edge lengths in the discrete setting) and thus the local area elements, which implies that the corresponding functional map must be both orthonormal and commute with the Laplace-Beltrami operators. Since none of the eigenvalues are repeating the corresponding functional map must be diagonal.

To prove Theorem 4.2 suppose there exists a node n in $\mathcal{M}$ which is not present in the tree built from the output of Algorithm A (we denote that tree by $\mathcal{M}_A$). Let d_n represent the diagonal of the functional map corresponding to n. Note that d_n is a vector of size κ. Now for every node m in $\mathcal{M}_A$ let d_m encode the diagonal of its functional map, and let $p(d_m)$ be the first entry of d_m that does not equal the corresponding entry of d_n. In other words $p(d_m) = \min_i$ s.t. $d_m(i) \neq d_n(i)$. Let $m*$ be the node maximizing $p(d_m)$, i.e. $m* = \arg\max_m p(d_m)$, and let $p_{\max} = p(d_{m*})$. Now assume w.l.o.g. that $d_n(p_{\max}) = 1$ whereas $d_{m*}(p_{\max}) = -1$. However, by construction of Algorithm A, to create the node $m*$ it must have considered

---

ALGORITHM 3: Bijective ZoomOut

Input : A pair of shapes S_i (i = 1,2) with corresponding
eigenfunctions $\{\varphi_k^{S_i}\}$; initial maps Π_{12} and Π_{21}

Output: Refined maps Π_{12} and Π_{21}

for $k = k_{ini} : k_{step} : k_{fianl}$ **do**

$\quad \Phi_1 = \Phi_k^{S_1}, \quad \Phi_2 = \Phi_k^{S_2}$

$\quad C_{12} = \Phi_2^\dagger \Pi_{21}\Phi_1, \quad C_{21} = \Phi_1^\dagger \Pi_{12}\Phi_2$

$\quad \Pi_{12} = \text{NNsearch}(\Phi_2 C_{21}^T, \Phi_1), \quad \Pi_{21} = \text{NNsearch}(\Phi_1 C_{12}^T, \Phi_2)$

$\quad C_{12} = \begin{pmatrix} \Phi_2 \\ \Pi_{12}\Phi_2 \end{pmatrix}^\dagger \begin{pmatrix} \Pi_{21}\Phi_1 \\ \Phi_1 \end{pmatrix}, \quad C_{21} = \begin{pmatrix} \Phi_1 \\ \Pi_{21}\Phi_1 \end{pmatrix}^\dagger \begin{pmatrix} \Pi_{12}\Phi_2 \\ \Phi_2 \end{pmatrix}$

$\quad \Pi_{12} = \text{NNsearch}\left(\begin{pmatrix} \Phi_2 C_{21}^T & \Phi_2 C_{12} \end{pmatrix}, \begin{pmatrix} \Phi_1 & \Phi_1 \end{pmatrix} \right)$

$\quad \Pi_{21} = \text{NNsearch}\left(\begin{pmatrix} \Phi_1 C_{12}^T & \Phi_1 C_{21} \end{pmatrix}, \begin{pmatrix} \Phi_2 & \Phi_2 \end{pmatrix} \right)$
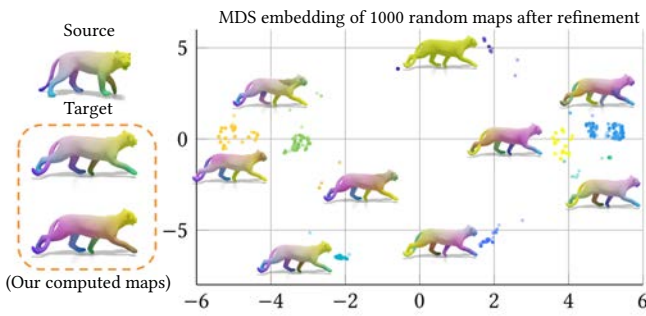
end

Fig. 16. (Left) For this pair of tigers, we apply our method and obtain the direct and the symmetric map visualized via color transfer. (Right) We randomly generate 1000 point-wise maps and apply ZoomOut for refinement. We then embed the refined maps in 2D with MDS. We apply a simple K-means to cluster the 2D points and color each point/map accordingly. For each cluster, we show one representative map that is closest to the direct/symmetric map.

an initialization consisting of a functional map of size whose diagonal equals to vector v_1, of size p_{max} where $v_1(i) = d_{m*}(i), i = 1..p_{max}$, and, again by construction, must have tested the initialization consisting of a functional map whose diagonal equals to v_2, where $v_2(i) = d_{m*}(i), i = 1..p_{max} - 1$ and $v_2(p_{max}) = 1$. Note $\mathcal{M}_A$ does not contain nodes m, s.t. $d_m(i) = v_2(i), 1..p_{max}$. However, we supposed that n exists in $\mathcal{M}$, and therefore there must exist at least one isometry whose functional maps has as principal submatrix a diagonal matrix with values equal to v_2. But this contradicts the assumption that refinement method $\mathcal{R}$ is *complete*. Therefore, any node in $\mathcal{M}$ must exist in $\mathcal{M}_A$. Conversely, since every node in $\mathcal{M}_A$ corresponds to some isometry it must also exist in $\mathcal{M}$, which proves the theorem. □

B ADDITIONAL RESULTS

In Algorithm 3, we give more details about the bijective ZoomOut that is used in our method. Fig. 16 shows the corresponding map visualization as discussed in Fig. 8. Fig. 17 and Fig. 18 show two examples of multi-solution shape matching, where we compare our method to a simple baseline: applying ZoomOut to 500 random
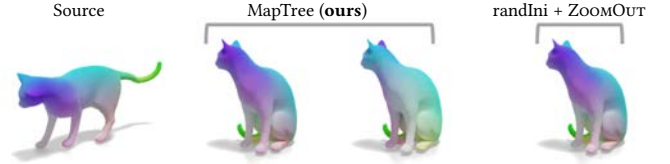

Fig. 17. For the cat pair, our method obtains both the direct map and the symmetric map simultaneously, while the baseline, applying ZoomOut to 500 random initial maps only contains a direct map.

Fig. 18. For the example of centaurs, the baseline and our method both obtain the direct map and the symmetric map.
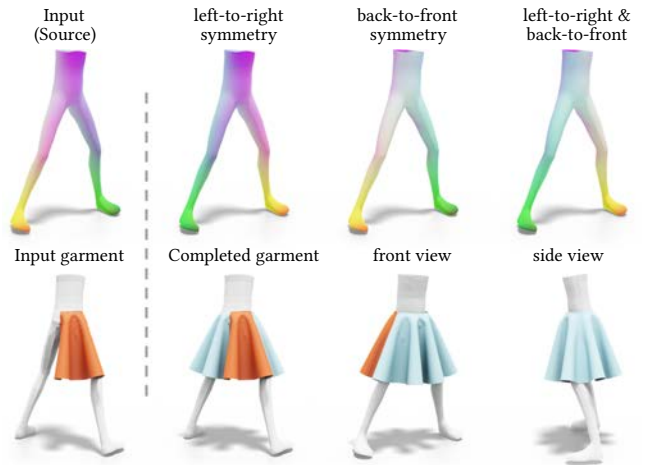

Fig. 19. *Top*: given an input shape, we compute three different self-symmetric maps visualized via color-transfer. *Bottom*: given a small piece of garment, we used the computed symmetries to complete the garment. We then show the completed garment in three different views where the input garment is highlighted as red and the automatically generated garment is colored blue. The mesh and garment are from the work [Wang et al. 2019].

pointwise maps. In the following, we show more qualitative and quantitative results of our method.

B.1 Self-symmetry detection

Fig. 19 shows a use case of how the detected self-symmetries can help the modeling process. To model a symmetric model, in general the artist need to manually setup the symmetry axes during the sculpting. However, it is not trivial to setup such symmetry axes when we want to have the model being symmetric w.r.t. another shape. For example, as shown in Fig. 19, we would like to model a dress that is symmetric w.r.t. the given legs shape. In this case, we can use our method to detect the self symmetries of the legs shape and use the obtained symmetries to complete the input garment and make it symmetric from back to front and from left to right w.r.t. the legs shape.

Table 6. Self-symmetry detection on FAUST dataset with 100 human shapes.

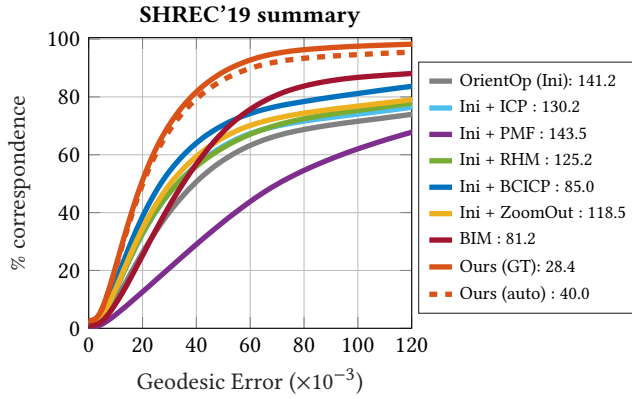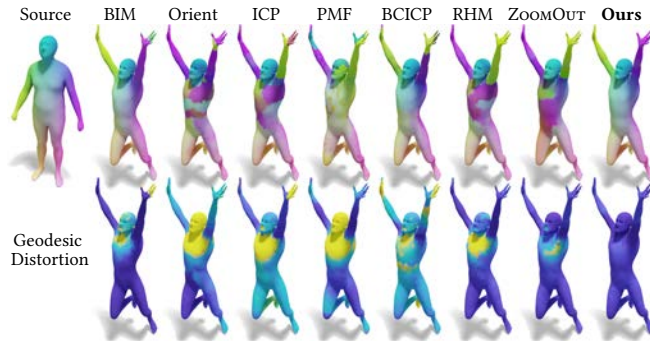| Method \Measurement	Accuracy ($\times 10^{-3}$)	Geodesic distortion ($\times 10^2$)	Runtime (s)
BIM	65.4	2.35	34.6
GroupRep	224	10.2	8.48
IntSymm	33.9	2.61	1.35
OrientRev (Ini)	68.0	5.52	0.59
Ini + ICP	38.3	3.94	13.3
Ini + BCICP	29.2	2.33	195
Ini + ZoomOut	16.1	1.86	22.6
Ours - GT selection	**15.3**	**1.81** (2.09)	61.0

Fig. 20. Shape matching on SHREC'19 Challenge. The summary curves of different methods average over 430 shape pairs.

Fig. 21. An example pair from SHREC'19. *Top*: we visualize the computed maps from different methods. *Bottom*: we visualize the geodesic error of each map, where blue indicates a lower error, and yellow indicates a larger error.

B.2 Correspondences between a shape pair

In Table 4 we report the accuracy and geodesic distortion of the maps obtained by different methods on 430 shape pairs from the SHREC'19 dataset. Here we show the cumulative curve of the errors in Fig. 20. Fig. 21 shows an example shape pair from the SHREC'19 dataset, for which we visualize both the computed maps and the geodesic distortion. Fig. 22 visualizes three maps generated by the genetic method [Sahillioğlu 2018] on shape pairs from the SHREC'19 dataset.

We also perform a similar test to 210 shape pairs from the SCAPE dataset (see Table 7). Similarly, we performed a simple ablation study and show that the map tree with our refinement method is

Fig. 22. Example maps generated by the genetic algorithm [Sahillioğlu 2018]. For each example pair, we show the source shape on the left, and visualize the map on the target shape on the right via color transfer.
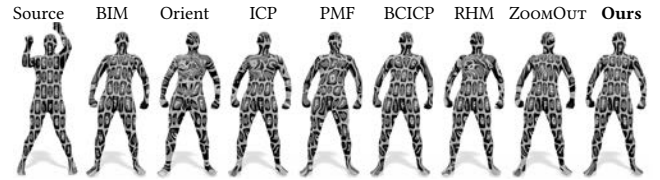

Fig. 23. Here we show an example pair from SCAPE dataset and visualize the obtained maps via texture transfer.

Table 7. We test 210 shape pairs from SCAPE dataset and report the average accuracy and runtime for different baseline methods.

Method / Measurement	Avg. Error ($\times 10^{-3}$)	Avg. Runtime (s)
BIM [Kim et al. 2011]	107.6	105.5
OrientOp (Ini) [Ren et al. 2018]	106.6	13.89
Ini + ICP [Ovsjanikov et al. 2012]	71.39	13.90
Ini + PMF [Vestner et al. 2017b]	119.7	385.8
Ini + BCICP [Ren et al. 2018]	**55.92**	287.1
Ini + RHM [Ezuz et al. 2019]	67.82	26.6
Ini + ZoomOut [Melzi et al. 2019b]	61.90	1.35
MapTree (Ours) - GT selection	**29.3**	60.2
MapTree (Ours) - Auto selection	**42.9**	60.2

more accurate and efficient than using the OrientOp initialization plus the ZoomOut refinement. We also use the cycle consistency to automatically select the maps with the preferred direct orientation from our output 5 maps, and the corresponding accuracy is 23% better than the best baseline. Fig. 23 shows a qualitative example of the SCAPE shape pair.

B.3 Partial symmetry detection

Fig. 24 and Fig. 25 show other two examples where the input shape is not symmetric intrinsically. In these cases, our method can still be applied directly to obtain some semantic partial symmetries. Our method can also be applied to explore the multi-scale structure of an input model with multiple objects. Specifically, we first detect the connected components of the input model, and apply our map exploration method to each component and each pair of components. Fig. 1 shows an example of a shape, in which our method detected different number of symmetry axes across different connected components. Fig. 26 shows another complex shape, and the results of our method, used to explore how a single connected component can be mapped to itself and to the other components.

Input: three dancing girls A,B,C   Symm Axis ①   ②   ③
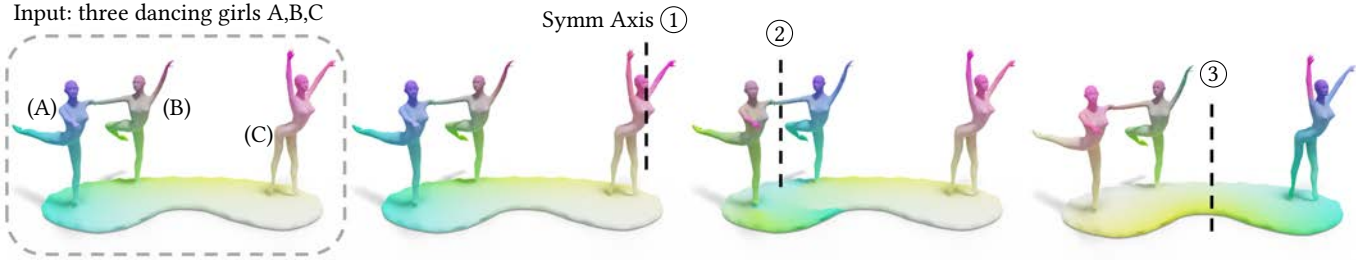

Fig. 24. In this example three dancing girls are connected by a base, which makes the overall shape not symmetric. We apply our self-symmetry exploration algorithm directly to obtain self-maps, visualized via color transfer. Note that the partial symmetric axis (1) detected the symmetry of the upper body of the girl (C); the symmetry axis (2) detected the partial symmetry between the girl A and B; the symmetry axis (3) detected a more global partial symmetry, where the girl (A) and (C) are mapped to each other, and (B) is mapped to itself from left-to-right.

Input (three different views)   A subset of detected partial intrinsic symmetries

Fig. 25. For non-symmetric shapes, some previous methods [Lipman et al. 2010; Xu et al. 2012, 2009] can detect the *partial* intrinsic symmetries which are given as *clusters* of points without any exact pointwise matches (please see Fig.14 in [Xu et al. 2012]). Our method can be applied to non-symmetric shapes as well and obtain the partial symmetric point-wise maps directly.

Input   Disconnected components   (A) Self-symm of one component   (B) maps between two components   (C) maps between two components
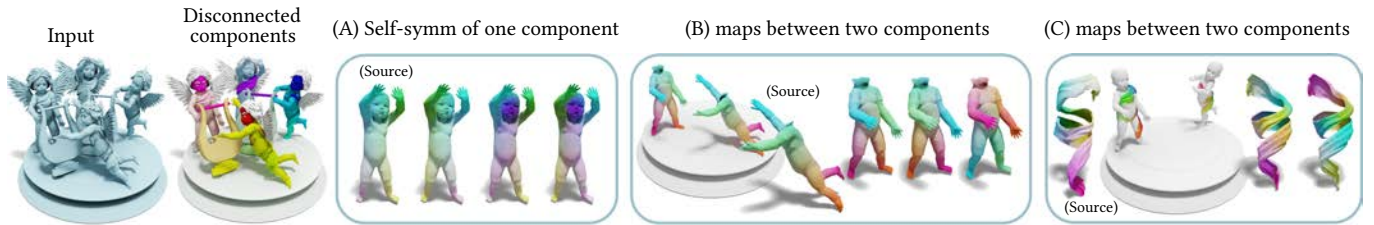

Fig. 26. Multi-scale symmetry detection. For an input model with multiple objects, we can first decompose it into individual connected components. (A) for a single component, we explore its self-symmetry space. Here we show *three* obtained self-symmetric maps of the baby angel, including the left-to-right, back-to-front, and the double flip self-symmetry. (B) we also explore different maps between two components. In this example, we find the direct map, the symmetric map, and the arm-to-leg correspondences between the body of two angels in different pose. (C) shows another example of finding the correspondences between two pieces of cloth.