ZoomOut: Spectral Upsampling for Efficient Shape Correspondence

SIMONE MELZI*, University of Verona
JING REN*, KAUST
EMANUELE RODOLÀ, Sapienza University of Rome
ABHISHEK SHARMA, LIX, École Polytechnique
PETER WONKA, KAUST
MAKS OVSJANIKOV, LIX, École Polytechnique

We present a simple and efficient method for refining maps or correspondences by iterative upsampling in the spectral domain that can be implemented in a few lines of code. Our main observation is that high quality maps can be obtained even if the input correspondences are noisy or are encoded by a small number of coefficients in a spectral basis. We show how this approach can be used in conjunction with existing initialization techniques across a range of application scenarios, including symmetry detection, map refinement across complete shapes, non-rigid partial shape matching and function transfer. In each application we demonstrate an improvement with respect to both the quality of the results and the computational speed compared to the best competing methods, with up to two orders of magnitude speed-up in some applications. We also demonstrate that our method is both robust to noisy input and is scalable with respect to shape complexity. Finally, we present a theoretical justification for our approach, shedding light on structural properties of functional maps.

CCS Concepts: • **Computing methodologies → Shape analysis**.

Additional Key Words and Phrases: Shape Matching, Spectral Methods, Functional Maps

1 INTRODUCTION

Shape matching is a task that occurs in countless applications in computer graphics, including shape interpolation [Kilian et al. 2007] and statistical shape analysis [Bogo et al. 2014], to name a few.

An elegant approach to non-rigid shape correspondence is provided by *spectral techniques*, which are broadly founded on the observation that near-isometric shape matching can be formulated as an alignment problem in certain higher-dimensional embedding

---

*Equal contribution.

Authors' addresses: Simone Melzi, University of Verona, simone.melzi@univr.it; Jing Ren, KAUST, jing.ren@kaust.edu.sa; Emanuele Rodolà, Sapienza University of Rome, rodola@di.uniroma1.it; Abhishek Sharma, LIX, École Polytechnique, kein.iitian@gmail.com; Peter Wonka, KAUST, pwonka@gmail.com; Maks Ovsjanikov, LIX, École Polytechnique, maks@lix.polytechnique.fr.
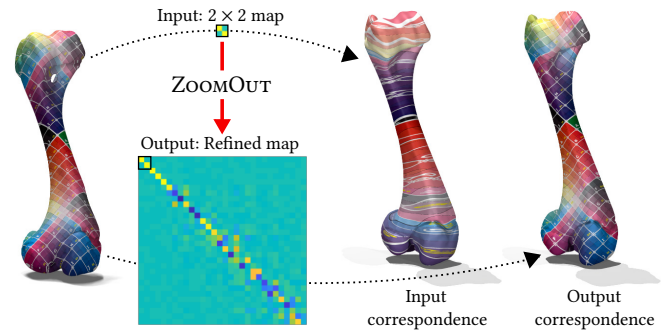
Fig. 1. Given a small functional map, here of size 2×2 which corresponds to a very noisy point-to-point correspondence (middle right) our method can efficiently recover both a high resolution functional and an accurate dense point-to-point map (right), both visualized via texture transfer from the source shape (left).

spaces [Biasotti et al. 2016; Jain and Zhang 2006; Maron et al. 2016; Ovsjanikov et al. 2012]. Despite significant recent advances and their wide practical applicability, however, spectral methods can both be computationally expensive and unstable with increased dimensionality of the spectral embedding. On the other hand, a reduced dimensionality results in very approximate maps, losing medium and high-frequency details and leading to significant artifacts in applications.

In this paper, we show that a higher resolution map can be recovered from a lower resolution one through a remarkably simple and efficient iterative spectral up-sampling technique, which consists of the following two basic steps:

(1) Convert a $k \times k$-size functional map to a pointwise map.
(2) Convert the pointwise map to a $k + 1 \times k + 1$ functional map.

Our main observation is that by iterating the two steps above, starting with an approximate initial map, encoded using a small number of spectral coefficients (as few as 2–15), we can obtain an accurate correspondence at very little computational cost.

We further show that our refinement technique can be combined with standard map initialization methods to obtain state-of-the-art results on a wide range of problems, including intrinsic symmetry detection, isometric shape matching, non-rigid partial correspondence and function transfer among others. Our method is robust to significant changes in shape sampling density, is easily scalable to meshes containing tens or even hundreds of thousands of vertices and is significantly (up to 100-500 times in certain cases) faster than existing state-of-the-art map refinement approaches, while producing comparable or even superior results. For example, Figure 1 shows a result obtained with our method, where starting from an initial 2×2 functional map, we recover a high resolution functional and an accurate pointwise correspondence.

Contributions. To summarize:
(1) We introduce a very simple map refinement method capable of improving upon the state of the art in a diverse set of shape correspondence problems; for each problem we can achieve

the same or better quality at a fraction of the cost compared to the current top performing methods.

(2) We demonstrate how higher-frequency information can be extracted from low-frequency spectral map representations.

(3) We introduce a novel variational optimization problem and develop a theoretical justification of our method, shedding light on structural properties of functional maps.

2 RELATED WORK

Shape matching is a very well-studied area of computer graphics. Below we review the methods most closely related to ours, concentrating on spectral techniques for finding correspondences between non-rigid shapes. We refer the interested readers to recent surveys including [Biasotti et al. 2016; Tam et al. 2013; Van Kaick et al. 2011] for a more in-depth treatment of the area.

Point-based Spectral Methods. Early spectral methods for shape correspondence were based on directly optimizing pointwise maps between spectral shape embeddings based on either adjacency or Laplacian matrices of graphs and triangle meshes [Jain and Zhang 2006; Jain et al. 2007; Mateus et al. 2008; Ovsjanikov et al. 2010; Scott and Longuet-Higgins 1991; Umeyama 1988]. Such approaches suffer from the requirement of a good initialization, and rely on restricting assumptions about the type of transformation relating the shapes. An initialization algorithm with optimality guarantees, although limited to few tens of points, was introduced in [Maron et al. 2016] and later extended to deal with intrinsic symmetries in [Dym and Lipman 2017]. Spectral quantities (namely, sequences of Laplacian eigenfunctions) have also been used to define pointwise descriptors, and employed within variants of the quadratic assignment problem in [Dubrovina and Kimmel 2010, 2011]. These approaches have been recently generalized by spectral generalized multidimensional scaling [Aflalo et al. 2016], which explicitly formulates minimum-distortion shape correspondence in the spectral domain.

Functional Maps. Our approach fits within the functional map framework, which was originally introduced in [Ovsjanikov et al. 2012] for solving non-rigid shape matching problems, and extended significantly in follow-up works, including [Aflalo and Kimmel 2013; Ezuz and Ben-Chen 2017; Kovnatsky et al. 2013; Rodolà et al. 2017] among others (see [Ovsjanikov et al. 2017] for an overview). These methods assume as input a set of corresponding functions, which can be derived from pointwise landmarks, dense descriptor fields, or from region correspondences. They then estimate a functional map matrix that allows to transfer real-valued functions across the two shapes, which is then converted to a pointwise map.

Although the first step reduces to the solution of a linear system of equations, this last step can be difficult and error prone [Ezuz and Ben-Chen 2017; Rodolà et al. 2015]. As a result, several strong regularizers have been proposed to promote certain desirable properties: see [Burghard et al. 2017; Huang and Ovsjanikov 2017; Litany et al. 2017; Nogneng and Ovsjanikov 2017; Rodolà et al. 2017; Wang et al. 2018b]. More recently, several other constraints on functional maps have been proposed to promote continuity of the pointwise correspondence [Poulenard et al. 2018], map curves defined on shapes [Gehre et al. 2018], extract more information from given descriptor
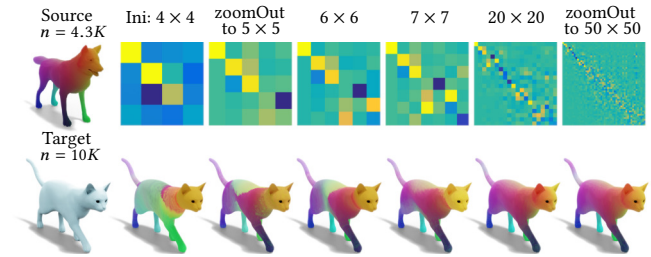
Fig. 2. ZoomOut example. Starting with a noisy functional map of size 4×4 between the two shapes we progressively upsample it using our two-step procedure and visualize the corresponding point-to-point map at each iteration via color transfer. Note that as the size of the functional map grows, the map becomes both more smooth and more semantically accurate. We denote the number of vertices by n.

constraints [Wang et al. 2018a], and for incorporating orientation information into the map inference pipeline [Ren et al. 2018].

In a concurrent work, [Shoham et al. 2019] also compute hierarchical functional maps by building explicit hierarchies in the spatial domain using subdivision surfaces. Unlike this work, our method operates purely in the spectral domain, and does not require computing additional shape hierarchies.

High-frequency Recovery. Several approaches have also observed that high-frequency information can be recovered even if the input functional map is small or noisy. This includes both optimizing an input map using vector field flow [Corman et al. 2015], recovering precise (vertex-to-point) maps [Ezuz and Ben-Chen 2017] from low frequency functional ones, and using pointwise products to extend the space of functions that can be transferred [Nogneng et al. 2018].

Iterative Map Refinement. We also note other commonly-used relaxations for matching problems based on optimal transport, e.g. [Mandad et al. 2017; Solomon et al. 2016], which are often solved through iterative refinement. Other techniques that exploit a similar formalism for solving optimal assignment include the Product Manifold Filter and its variants [Vestner et al. 2017a,b]. Map refinement has also been considered in the original functional maps approach [Ovsjanikov et al. 2012] where Iterative Closest Point in the spectral embedding has been used to improve input functional maps. Finally, in the context of shape collections [Huang et al. 2014; Wang et al. 2013; Wang and Singer 2013], cycle-consistency constraints have been used to iteratively improve input map quality. We further discuss methods most closely-related to ours in Section 4.3 below.

Although these techniques can be very effective for obtaining high-quality correspondences, methods based purely on optimization in the spatial domain can quickly become prohibitively expensive even for moderate sampling density. On the other hand, spectral techniques can provide accurate solutions for low-frequency matching, but require significant effort to recover a high-quality dense pointwise correspondence; further, such approaches are often formulated as difficult optimization problems and suffer from instabilities for large embedding dimensions.

3 BACKGROUND & NOTATION

In this section we introduce the main background notions and notation used throughout the rest of the paper.

Given a pair of shapes $\mathcal{M}$ and $\mathcal{N}$, typically represented as triangle meshes, we associate to them the positive semi-definite Laplacian matrices $L_{\mathcal{M}}, L_{\mathcal{N}}$, discretized via the standard cotangent weight scheme [Pinkall and Polthier 1993], so that $L_{\mathcal{M}} = A_{\mathcal{M}}^{-1} W_{\mathcal{M}}$, where $A_{\mathcal{M}}$ is the diagonal matrix of lumped area elements and $W_{\mathcal{M}}$ is the cotangent weight matrix, with the appropriate choice of sign to ensure positive semi-definiteness. We make use of the basis consisting of the first $k_{\mathcal{M}}$ eigenfunctions of the Laplacian matrix, and encode it in a matrix $\Phi_{\mathcal{M}}^{k_{\mathcal{M}}} = [\varphi_1^{\mathcal{M}}, \varphi_2^{\mathcal{M}}, \ldots, \varphi_{k_{\mathcal{M}}}^{\mathcal{M}}]$ having the eigenfunctions as its columns. We define the *spectral embedding* of $\mathcal{M}$ as the $k_{\mathcal{M}}$-dimensional point set $\{(\varphi_1^{\mathcal{M}}(x), \ldots, \varphi_{k_{\mathcal{M}}}^{\mathcal{M}}(x)) \mid x \in \mathcal{M}\}$.

Given a point-to-point map $T : \mathcal{M} \to \mathcal{N}$, we denote by Π its matrix representation, s.t. $\Pi(i, j) = 1$ if $T(i) = j$ and 0 otherwise, where i and j are vertex indices on shape $\mathcal{M}$ and $\mathcal{N}$, respectively. Note that the matrix Π is an equivalent matrix representation of any pointwise map T without extra assumptions, such as bijectivity. The corresponding *functional map* $\mathbf{C}$ is a linear transformation taking functions on $\mathcal{N}$ to functions on $\mathcal{M}$; in matrix notation, it is given by the projection of Π onto the corresponding functional basis:

$$\mathbf{C} = \Phi_{\mathcal{M}}^{+} \Pi \Phi_{\mathcal{N}}, \qquad (1)$$

where $^{+}$ denotes the Moore-Penrose pseudo-inverse. When the eigenfunctions are orthonormal with respect to the area-weighted inner product, so that $\Phi_{\mathcal{M}}^{\top} A_{\mathcal{M}} \Phi_{\mathcal{M}} = Id$, then Eq. (1) can be written as: $\mathbf{C} = \Phi_{\mathcal{M}}^{\top} A_{\mathcal{M}} \Pi \Phi_{\mathcal{N}}$. Note that $\mathbf{C}$ is a matrix of size $k_{\mathcal{M}} \times k_{\mathcal{N}}$, independent of the number of vertices on the two shapes.

A typical pipeline for computing a correspondence using the functional map representation proceeds as follows [Ovsjanikov et al. 2017]: 1) Compute a moderately-sized basis (60-200 basis functions) on each shape; 2) Optimize for a functional map $\mathbf{C}_{opt}$ by minimizing an energy, based on preservation of descriptor functions or landmark correspondences and regularization, such as commutativity with the Laplacian operators; 3) Convert $\mathbf{C}_{opt}$ to a point-to-point map. The complexity of this pipeline directly depends on the size of the chosen basis, and thus the dimensionality of the spectral embedding. Smaller bases allow more stable and efficient functional map recovery but result in approximate pointwise correspondences, while larger functional maps can be more accurate but are also more difficult to optimize for and require stronger priors.

Our main goal, therefore, is to show that accurate pointwise correspondences can be obtained even in the presence of only small, or approximate functional maps.

4 ZOOMOUT: ITERATIVE SPECTRAL UPSAMPLING

As input we assume to be given either a small functional map $\mathbf{C}_0$ or a point-to-point correspondence $T : \mathcal{M} \to \mathcal{N}$; both may be affected by noise. We will discuss the role and influence of the input map in detail in the following sections. If it is a point-to-point map, we first convert it to a functional one via Eq. (1). For simplicity, we first state our method and then provide its theoretical derivation from a variational optimization problem in Section 4.4.

Given an input $k_{\mathcal{M}} \times k_{\mathcal{N}}$ functional map $\mathbf{C}_0$ our goal is to extend it to a new map $\mathbf{C}_1$ of size $(k_{\mathcal{M}}+1) \times (k_{\mathcal{N}}+1)$ without any additional information. We do so by a simple two-step procedure:
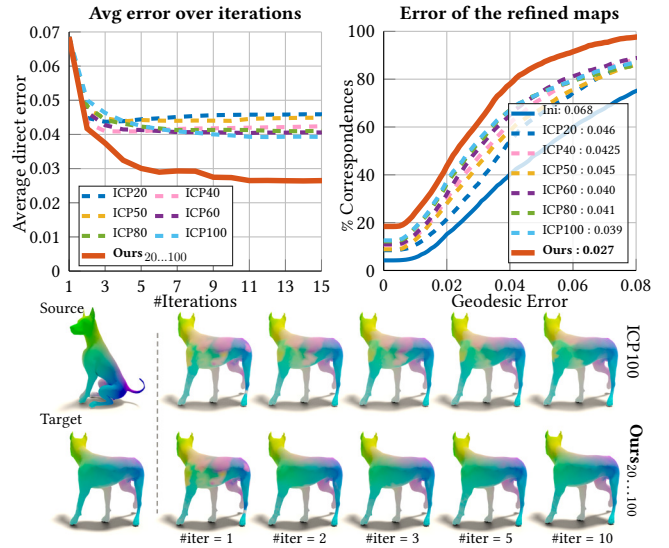

Fig. 3. Comparison of map quality during ICP iterations in different (fixed) dimensions vs. ZoomOut from 20 to 100 with step 5. Top row: average error of pointwise maps during refinement and the error summary of the refined maps after 15 iterations. Note that regardless of dimension, ICP gets trapped in a local minimum. Bottom row: visualization of the refined maps at iteration 1, 2, 3, 5, and 10 of ICP with dimension 100 vs. our method.

(1) Compute a point-to-point map T via Eq. (2), and encode it as a matrix Π.
(2) Set $\mathbf{C}_1 = (\Phi_{\mathcal{M}}^{k_{\mathcal{M}}+1})^{\top} A_{\mathcal{M}} \Pi \Phi_{\mathcal{N}}^{k_{\mathcal{N}}+1}$.

We then iterate this procedure to obtain progressively larger functional maps $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n$ until some sufficiently large n. As we demonstrate below, this remarkably simple procedure, which can be implemented in only a few lines of code (see Appendix B), can result in very accurate functional and pointwise maps even given very small and possibly noisy input. To compute a pointwise map from a given $\mathbf{C}$ in step (1), we solve the following problem:

$$T(p) = \underset{q}{\arg\min} \|\mathbf{C}(\Phi_{\mathcal{N}}(q))^{\top} - (\Phi_{\mathcal{M}}(p))^{\top}\|_2, \ \forall \ p \in \mathcal{M} \qquad (2)$$

where $\Phi_{\mathcal{M}}(p)$ denotes the pth row of the matrix of eigenvectors $\Phi_{\mathcal{M}}$. This procedure gives a point-to-point map $T : \mathcal{M} \to \mathcal{N}$, and can be implemented via a nearest-neighbor query in $k_{\mathcal{M}}$-dimensional space. It is also nearly identical, up to change in direction, to the pointwise map recovery described in the original functional maps article [Ovsjanikov et al. 2012, Section 6.2] but differs from other recovery steps, introduced, e.g., in [Ezuz and Ben-Chen 2017] as we discuss below.

Figure 2 shows an example of ZoomOut on a pair of animal shapes from the TOSCA dataset [Bronstein et al. 2008]. Starting with a 4×4 functional map, we show both the functional and point-to-point (visualized via color transfer) maps throughout our upsampling iterations. Note how the pointwise map becomes both more smooth and accurate as the functional map grows.

We use the term "upsampling" in the description of our method to highlight the fact that at every iteration ZoomOut introduces additional frequencies and thus intuitively adds samples *in the spectral domain* for representing a map.
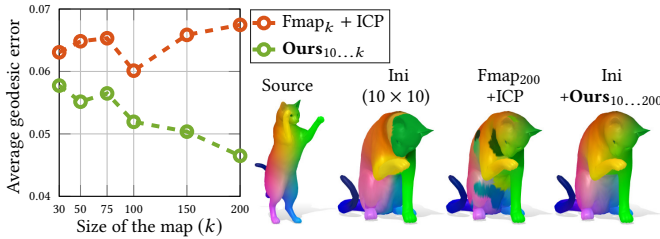
Fig. 4. Impact of the input functional map size. Given a pair of shapes, we use a fixed set of descriptors and the approach of [Nogneng and Ovsjanikov 2017] to compute a functional map of size $k \times k$ and refine it with ICP. Alternatively, we compute a functional map of size 10×10 using the same approach and upsample it to $k \times k$ using our method. Differently from the ICP baseline, our method leads to improvement as k grows. On the right we show a qualitative illustration for $k = 200$.

4.1 Map Initialization

We initialize our pipeline by optimizing for a $k_{\mathcal{M}} \times k_{\mathcal{N}}$ functional map $\mathbf{C}_0$ using an existing approach; we tested recent techniques, including [Ren et al. 2018; Rodolà et al. 2017] among others, across different settings described in detail in Section 5.

The key parameter for the initialization is the size of the functional map, and in most settings, we set $k_{\mathcal{M}} = k_{\mathcal{N}} = k$ for some small k. This value ranges between 4 and 20 in all of our experiments, and allows us to obtain high quality maps by upsampling $\mathbf{C}_0$ to sizes up to 200×200 depending on the scenario. We have observed that the key requirement for the input map $\mathbf{C}_0$ is that although it can be noisy and approximate, it should generally disambiguate between the possible symmetries exhibited by the shape. Thus, for example, if 4 basis functions are sufficient to distinguish left and right on the animal models shown in Figure 2, then with a functional map of this size our method can produce an accurate final correspondence. Perhaps the most difficult case we have encountered is in disambiguating front and back in human shapes which requires approximately 15 basis functions. This is still significantly smaller than typical values in existing functional map estimation pipelines, which are based on at least 60 to 100 basis functions to compute accurate maps.

4.2 Acceleration Strategies

We propose three ways to accelerate ZoomOut.

4.2.1 Larger step size. The basic method increases the size of the functional map by one row and one column at each iteration. This choice is supported by our theoretical analysis below, which suggests that increasing by one at each iteration helps to promote isometric maps, when they are present. In practice our method also achieves good accuracy with larger increments ranging between 2 and 5 (see supplementary materials for an illustration). We also note that in some settings (e.g., in the context of partial correspondence or in challenging non-isometric pairs), it is more reasonable to have rectangular functional maps with more rows than columns. There, we increase the number of rows with higher increments than that of columns. We point out these explicitly in Section 5.

4.2.2 Approximate nearest neighbors. We can also use approximate nearest neighbor instead of exact nearest neighbors during upsampling. This is particularly useful in higher dimensions where such
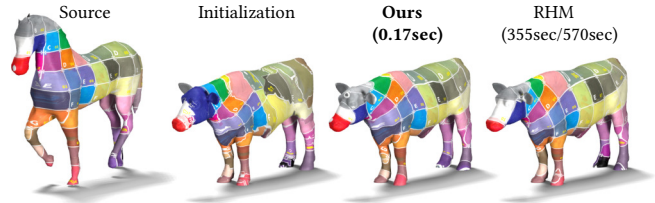
Fig. 5. Comparison with RHM [Ezuz et al. 2019]. Both methods are initialized with a 17×10 functional map provided by the authors of [Ezuz et al. 2019]. The reported runtimes (excluding pre-computation) are for a CPU implementation of our method with acceleration, and a (GPU/CPU) implementation of RHM. The runtime of pre-computation for our method is 7s (and 70s for RHM). Our solution has comparable quality and is more than 2 orders of magnitude faster.

queries can become expensive. In practice, we have observed that using the FLANN library [Muja and Lowe 2014] can lead to a 30x time improvement with negligible impact on final quality ($\sim$0.001% decrease of average accuracy).

4.2.3 Sub-sampling. In the presence of perfect information, a functional map $\mathbf{C}$ of size $k \times k$ is fully determined by k point correspondences. Thus, it is possible to sample a small number (typically a few hundred) points on each shape, perform our refinement using the spectral embedding of only those points, and then convert the final functional map to a *dense* pointwise correspondence only once. In practice we simply use Euclidean farthest point sampling starting from a random seed point.

4.3 Relation to Other Techniques

While closely related to multiple existing techniques, our method is fundamentally different in several ways that we highlight below.

Iterative Closest Point. ICP refinement of functional maps [Ovsjanikov et al. 2012] differs in that our method progressively *increases* the dimension of the spectral embedding during refinement. This crucial difference allows us to process smaller initial functional maps, which are easier to compute, and avoids getting trapped in local minima at higher dimensions, significantly improving the final accuracy. Figure 3 shows the accuracy of our method compared to ICP with different dimensions. All methods in this figure refine the same initial pointwise map at #iter = 1, which is computed using [Ren et al. 2018] with the orientation-preserving term. Moreover, differently from ICP our approach does not force the singular values of functional maps to be 1, and inverts the direction of the pointwise and functional maps in a way that is consistent with the directions of a map and its pull-back. As we show in Section 4.4, rather than promoting area-preserving pointwise maps as done in ICP, our method implicitly optimizes an energy that promotes full isometries. In Figure 4 we further illustrate that our method produces significantly more accurate maps in higher dimensions. We initialize the maps with the approach of [Nogneng and Ovsjanikov 2017] using the WKS descriptors and 2 landmarks.

BCICP. [Ren et al. 2018] is a recent powerful technique for improving noisy correspondences, based on refining maps in both the spectral and spatial domains, while incorporating bijectivity, smoothness and coverage through a series of sophisticated update
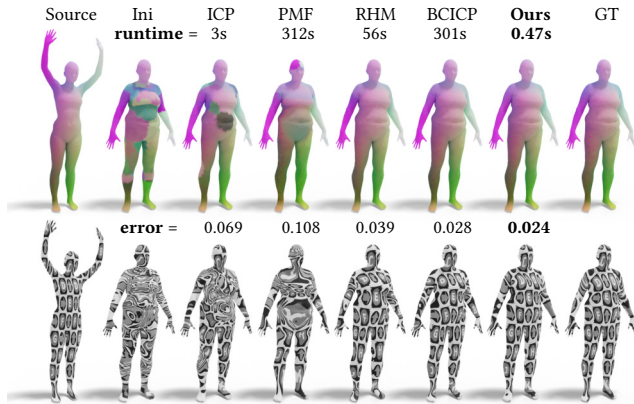
Fig. 6. Refinement example. Given the initialization computed from WKS descriptors, we compare our method with existing refinement techniques, by visualizing the maps via color transfer (first row) and texture transfer (second row). We also report the average error and the runtime for each method. Note that our method is 120x faster than RMH and 640x faster than BCICP, while resulting in lower error.

steps. While often accurate, this method requires the computation of geodesic distances, is inefficient, and suffers from poor scalability. As an extension of the original ICP, this method also uses spectral embeddings of *fixed* size. As we show in our tests, our very simple approach can achieve similar and even superior accuracy at a fraction of the time cost.

Reversible Harmonic Maps (RHM). [Ezuz et al. 2019] is another recent approach for map refinement, based on minimizing the bi-directional geodesic Dirichlet energy. In a similar spirit to ours, this technique is based on splitting the alignment in a higher-dimensional embedding space from the computation of pointwise maps. However, it requires the computation of all pairs of geodesic distances, and results in least squares problems with size proportional to the number of points on the shapes. Furthermore, similarly to ICP and BCICP, the embedding dimension is fixed throughout the approach. As a result, our approach is significantly more efficient (see Figure 5), scalable, and, as we show below, more accurate in many cases.

Spatial refinement methods. Spatial refinement methods such as PMF [Vestner et al. 2017a,b] operate via an alternating diffusion process based on solving a sequence of linear assignment problems; this approach demonstrates high accuracy in challenging cases, but is severely limited by mesh resolution. Other approaches formulate shape correspondence by seeking for optimal transport plans iteratively via Sinkhorn projections, but they either scale poorly [Solomon et al. 2016] or can have issues with non-isotropic meshes [Mandad et al. 2017]. Interestingly, although fundamentally different, a link exists between ZoomOut and PMF that we describe in the supplementary materials.

In Figure 6 we show qualitative comparisons with the methods above on pairs of remeshed shapes from the FAUST [Bogo et al. 2014] dataset. We provide a more complete evaluation with state-of-the-art refinement methods in Section 5.
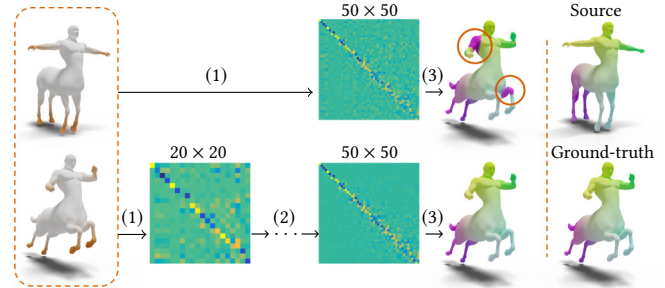

Fig. 7. We use an existing functional map pipeline (1) to compute either a 50×50 (top row) or 20×20 (bottom row) functional map using the same input descriptors. We then upsample (2) the smaller map to also have size 50x50 using our technique, and convert both to pointwise maps (3). Our approach leads to better results as can be seen, e.g., on the arms and legs.

4.4 Derivation and Analysis

In this section we provide a theoretical justification for our method by first formulating a variational optimization problem and then arguing that ZoomOut provides an efficient way of solving it.

4.4.1 Optimization Problem. We consider the following problem:

$$\min_{\mathbf{C} \in \mathcal{P}} E(\mathbf{C}), \text{ where } E(\mathbf{C}) = \sum_k \frac{1}{k} \left\| \mathbf{C}_k^T \mathbf{C}_k - I_k \right\|_F^2 . \tag{3}$$

Here $\mathcal{P}$ is the set of functional maps arising from pointwise correspondences, $\mathbf{C}_k$ is the principal $k \times k$ submatrix of $\mathbf{C}$ (i.e., the submatrix of $\mathbf{C}$ consisting of the first k rows and columns), and I_k is an identity matrix of size k. In other words, Eq. (3) aims to compute a pointwise map associated with a functional map in which every principal submatrix is orthonormal.

The energy in Eq. (3) is different from the commonly used penalty promoting orthonormal functional maps [Kovnatsky et al. 2013, 2016; Ovsjanikov et al. 2012] in two ways: first we explicitly constrain $\mathbf{C}$ to arise from a point-to-point map, and second we enforce orthonormality of every principal submatrix rather than the full functional map of a given fixed size. Indeed, an orthonormal functional map corresponds to only a *locally area-preserving* point-to-point correspondence [Rustamov et al. 2013]. Instead, the energy in Eq. (3) is much stronger and promotes complete isometries as guaranteed by the following theorem (proved in Appendix A):

THEOREM 4.1. *Given a pair of shapes whose Laplacian matrices have the same eigenvalues, none of which are repeating, a functional map $\mathbf{C} \in \mathcal{P}$ satisfies $E(\mathbf{C}) = 0$ if and only if the corresponding pointwise map is an isometry.*

To derive ZoomOut as a method to solve the optimization problem in Eq. (3) we first consider a single term inside the sum, and write the problem explicitly in terms of the binary matrix Π representing the pointwise map:

$$\min_{\Pi} \left\| \mathbf{C}_k^T \mathbf{C}_k - I_k \right\|_F^2 = \min_{\Pi} \left\| \mathbf{C}_k \mathbf{C}_k^T - I_k \right\|_F^2 , \tag{4}$$

$$\text{where } \mathbf{C}_k = (\Phi_{\mathcal{M}}^k)^+ \Pi \Phi_{\mathcal{N}}^k . \tag{5}$$

This problem is challenging due to the constraints on Π. To address this, we use *half-quadratic splitting,* by decoupling Π and $\mathbf{C}_k$. This

leads to the following two separate sub-problems:

$$\min_{\Pi} \|(\Phi^k_{\mathcal{M}})^+ \Pi \Phi^k_{\mathcal{N}} C^T_k - I_k\|^2_F, \tag{6}$$

$$\min_{C_k} \|C_k - (\Phi^k_{\mathcal{M}})^+ \Pi \Phi^k_{\mathcal{N}}\|^2_F. \tag{7}$$

Now we remark that Eq. (6) does not fully constrain Π since it only penalizes the image of Π within the vector space of $\Phi^k_{\mathcal{M}}$. Instead, inspired by a related construction in [Ezuz and Ben-Chen 2017] we add a regularizer $\mathcal{R}(\Pi) = \|(I - \Phi^k_{\mathcal{M}}(\Phi^k_{\mathcal{M}})^+)\Pi \Phi^k_{\mathcal{N}} C^T_K\|^2_{A_{\mathcal{M}}}$, where we use the weighted matrix norm $\|X\|^2_{A_{\mathcal{M}}} = tr(X^T A_{\mathcal{M}} X)$ and $A_{\mathcal{M}}$ is the area matrix of shape $\mathcal{M}$. This regularizer penalizes the image of $\Pi \Phi^k_{\mathcal{N}} C^T_k$ that lies outside of the span of $\Phi^k_{\mathcal{M}}$, which intuitively means that no spurious high frequencies should be created. Finally, it can be shown (see proof in the appendix) that solving Eq. (6) with the additional term $\mathcal{R}(\Pi)$ is equivalent to solving:

$$\min_{\Pi} \|\Pi \Phi^k_{\mathcal{N}} C^T_k - \Phi^k_{\mathcal{M}}\|^2_F. \tag{8}$$

The problem in Eq. (8) has a closed-form solution, which reduces to the nearest-neighbor search described in Eq. (2) above. Moreover, the problem in Eq. (7) is solved simply via $C_k = (\Phi^k_{\mathcal{M}})^+ \Pi \Phi^k_{\mathcal{N}}$ since the minimization is unconstrained.

Finally, in this derivation we assumed a specific value of k. In practice we start with a particular value k_0 and progressively increase it. This is motivated by the fact that if a principal $k \times k$ submatrix is orthonormal, it provides a very strong initialization for the larger problem on a $(k + 1) \times (k + 1)$ matrix since only a single new constraint on the additional row and column must be enforced. This leads to our method ZOOMOUT:

(1) Given $k = k_0$ and an initial C_0 of size $k_0 \times k_0$.
(2) Compute $\arg\min_{\Pi} \|\Pi \Phi^k_{\mathcal{N}} C^T_k - \Phi^k_{\mathcal{M}}\|^2_F$.
(3) Set $k = k + 1$ and compute $C_k = (\Phi^k_{\mathcal{M}})^+ \Pi \Phi^k_{\mathcal{N}}$.
(4) Repeat the previous two steps until $k = k_{\max}$.

4.4.2 Empirical Accuracy. We demonstrate that this simple procedure is remarkably efficient in minimizing the energy in Eq. (3). For this in Figure 8 we plot the value of the energy during the iterations of ZOOMOUT from $k = 20$ to $k = 120$ with step 5 on 100 pairs of shapes from the FAUST dataset, and compare it to the ICP refinement using $k = 120$. We also evaluate a method in which we perform the same iterative spectral upsampling as in ZOOMOUT but use the point-

Fig. 8. Value of $E(C)$ across iterations

wise map recovery from [Ezuz and Ben-Chen 2017] instead of Eq. (2). For all methods, at every iteration we convert the computed pointwise map to a functional map C of fixed size 120×120 and report $E(C)$. Our approach results in maps with energy very close to the ground truth, while Deblur with upsampling performs poorly, highlighting the importance of the adapted pointwise recovery method. In the supplementary materials we further detail the differences between the two methods and their relation to PMF.
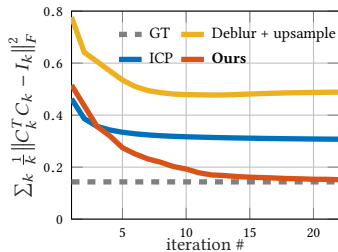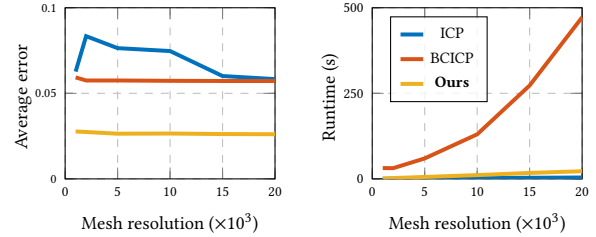
Fig. 9. Scalability and accuracy test on 6 pairs of scanned bones. The source shape has 5K vertices. We compare to ICP and BCICP on the same target shape with different resolution (ranging from 1K to 20K vertices).

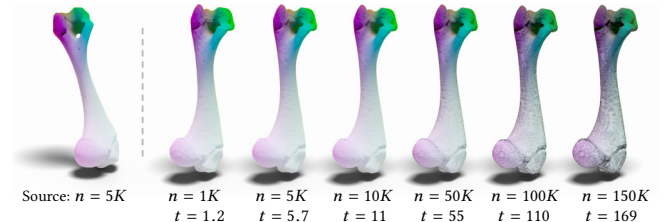| Source: $n = 5K$ | $n = 1K$ $t = 1.2$ | $n = 5K$ $t = 5.7$ | $n = 10K$ $t = 11$ | $n = 50K$ $t = 55$ | $n = 100K$ $t = 110$ | $n = 150K$ $t = 169$ |

Fig. 10. Scalability. The vertices of the bone shapes are colored black to show the resolution (zoom in for better view), while RGB colors encode the computed map, via pull-back from the source. The corresponding runtime for our upsampling, from 5×5 to 50×50 *without any acceleration*, is reported below each shape (in seconds).

Finally, in Figure 7 we also show the result of an existing functional map estimation pipeline with orientation preservation [Ren et al. 2018] for a map of size 50×50 with careful parameter tuning for optimality, which nevertheless leads to noise in the final point-to-point map. Initializing the map to size 20×20 using exactly the same descriptors and up-sampling it to a larger size with our approach leads to a significant improvement.

5 RESULTS

We conducted an extensive evaluation of our method, both in terms of its empirical properties (Section 5.1) and in relation to existing methods, as we showcase across several applications (Section 5.2).

5.1 Performance of ZOOMOUT

We start by showing an evaluation of scalability, as well as of the stability and smoothness of our method.

5.1.1 Scalability. In Figure 9 we assess the scalability of our method using shapes of humerus bones of wild boars acquired using a 3D sensor. Each bone was scanned independently, and the ground truth was provided by domain experts as 24 consistent landmarks [Gunz and Mitteroecker 2013] on each shape. We show the average runtime and accuracy over 6 maps w.r.t. different target mesh resolution; the input descriptors (one landmark point and WKS descriptors [Aubry et al. 2011]) for the initialization are fixed. While BCICP, the current state-of-the-art method, quickly becomes prohibitively expensive at high resolution, both ICP and ZOOMOUT without acceleration have approximately linear complexity. On the other hand, the accuracy for BCICP and our method are stable w.r.t. different resolutions, while ICP is less accurate and more unstable. Figure 10 also shows an example with meshes having ~150K vertices.
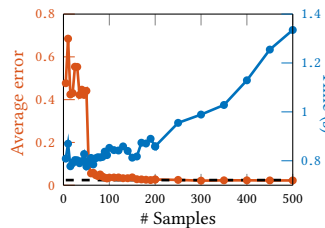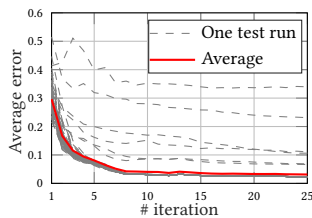
Fig. 11. Acceleration by sampling
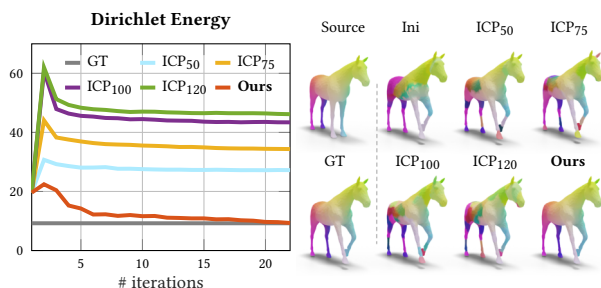

Fig. 12. Stability of zoomOut

Fig. 13. Average Dirichlet energy of pointwise maps on 20 TOSCA pairs, starting with a computed 20 × 20 functional map, refined either using ICP in different dimensions or ZoomOut until 120 × 120. Our method converges to a smoother map, with Dirichlet energy closer to the ground truth.
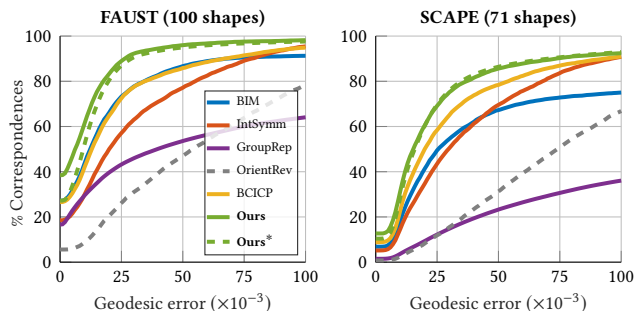

Fig. 14. Error summary of symmetry detection. We compare with the recent state-of-the-art methods IntSymm [Nagar and Raman 2018] and GroupRep [Wang and Huang 2017], as well as to the baseline Blended Intrinsic Maps (BIM) [Kim et al. 2011] and BCICP.

Figure 11 shows the results of our sub-sampling strategy for acceleration on one pair of bones, where the source has 20K vertices, and the target has 5K vertices. The corresponding runtime (blue curve) and average error (red curve) w.r.t. different sampling size for the source shape are reported. We can see that around 100 samples on a 20K mesh are enough to produce a refined map with similar quality to that of our method without sampling (whose average is shown as dashed black).

5.1.2 Stability. We also evaluate the stability of our method w.r.t. noise in the initial functional map. Here we test on a single shape pair from FAUST initialized using the approach of [Ren et al. 2018] while fixing the size of the computed functional map to 4. Given this 4 × 4 initial functional map, we add white noise to it and use our method to refine the map. Figure 12 shows the average error over iterations for 100 independent random tests. This plot shows that our method is robust to noise in the input, even if the input maps can have errors up to approximately 40% of the shape radius. At the same time, our algorithm can efficiently filter out the noise within a small number of iterations. Note that in 94 cases out of 100 the refined maps converged to a nearly identical final result, while in the remaining 6, the refinement led to maps that are mixed with symmetric ambiguity since there is too much noise introduced into their initialization.

5.1.3 Smoothness. The maps refined with our method are typically very smooth, although this constraint is not enforced explicitly. Figure 13 shows a quantitative measurement of the smoothness compared to ICP with different dimensions on 20 pairs of shapes from the TOSCA dataset [Bronstein et al. 2008], starting with a 20 × 20 functional map computed via [Nogneng and Ovsjanikov

2017]. Map smoothness is measured as the mean Dirichlet energy of the normalized coordinates of the target shape mapped on the source through the given point-to-point map. Our method clearly provides smoother maps, and approaches the ground truth after a few iterations.

5.2 Practical Applications

We applied our method across a range of application scenarios, including symmetry detection, map refinement among complete shapes, partial matching and function transfer. In each application we demonstrate a quantitative improvement as well as a significant speedup compared to the best competing method. Note that in all experiments, we use the same initialization for all competing methods to guarantee a fair comparison.

Unless otherwise stated, ICP uses the same dimension as the output dimension of ZoomOut. "Ours" refers to applying ZoomOut on the complete meshes, while "Ours*" refers to ZoomOut with sub-sampling for acceleration. In both cases, we always output dense correspondences between complete meshes. To measure the accuracy, we only accept *direct* ground-truth maps (except for the symmetry detection application, where the symmetric ground-truth maps are considered). For texture transfer, we first convert the point-wise map to a functional map with size 300×300, then we use this functional map to transfer the uv-coordinates from source to target.

5.2.1 Symmetry Detection. We first apply our approach for computing pose-invariant symmetries. This problem has received a lot of attention in the past and here we compare to the most recent and widely used techniques. In this application we are only given a single shape and our goal is to compute a high-quality intrinsic symmetry, such as the left-right symmetry present in humans. This problem is slightly different from the pairwise matching scenario, since the identity solution must be ruled out. We do so by leveraging a recent approach for encoding map orientation in functional map computations [Ren et al. 2018]. Namely, we compute an initial 10 × 10 functional map by solving an optimization problem with exactly the same parameters as in [Ren et al. 2018] and WKS descriptors as input, but instead of orientation-preserving, we promote orientation-*reversing* maps. This gives us an initial functional map which we then upsample to size 100×100. Figure 14 shows the error

Table 1. **Symmetry Detection**. Given approximate symmetric maps (OrientRev [Ren et al. 2018]), we refine them using our method or BCICP, and compare the results to several state-of-the-art methods, including BIM, IntSymm, and GroupRep. Here we report the average error and runtime over 100 FAUST shapes and 71 SCAPE shapes. We also include the results of our method with sub-sampling for acceleration (called Ours*).

| Measurement		Average Error ($\times 10^{-3}$)		Average Runtime (s)	
Method \ Dataset		FAUST	SCAPE	FAUST	SCAPE
BIM [Kim et al. 2011]		65.4	133	34.6	41.7
GroupRep [Wang and Huang 2017]		224	347	8.48	16.7
IntSymm [Nagar and Raman 2018]		33.9	60.3	1.35	1.81
OrientRev (Ini) [Ren et al. 2018]		68.0	110	0.59	1.07
Ini + BCICP [Ren et al. 2018]		29.2	49.7	195.1	525.6
Ini + Ours		**16.1**	**46.2**	22.6	62.7
Ini + Ours*		18.5	46.6	**1.78**	**3.66**
Improv. w.r.t	Ini + Ours	**44.9%**	**7.0%**	8×	8×
state-of-the-art	Ini + Ours*	36.6%	6.2%	**110×**	**140×**
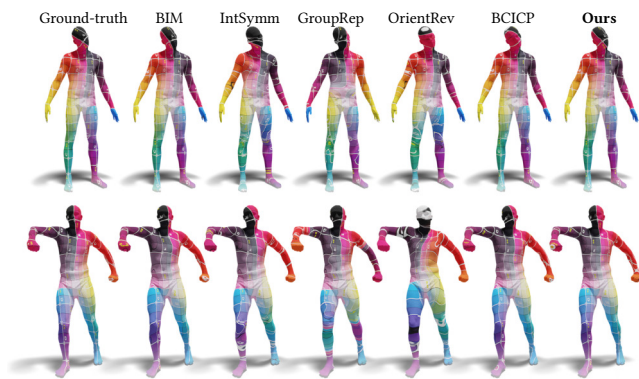
Fig. 15. Symmetry detection. We show two examples with FAUST (first row) and SCAPE (second row) and visualize the symmetric maps from different methods via texture transfer. Note that our method with acceleration is over 100× faster than BCICP, while achieving comparable or better quality.

curves on the SCAPE [Anguelov et al. 2005] and FAUST benchmarks (for which we have the ground truth symmetry map), while Table 1 reports the average error and runtime. Note that the shapes in both datasets are not meshed in a symmetric way, so a successful method must be able to handle, often significant, changes in mesh structure.

Our approach achieves a significant quality improvement compared to all state-of-the-art methods, and is also significantly faster. With acceleration, we achieve a speedup of more than 100x on a workstation with a 3.10GHz CPU and 64GB RAM. Figure 15 further shows a qualitative comparison. Finally, we remark that for human shapes the first four Laplacian eigenfunctions follow the same structure disambiguating top-bottom and left-right. Therefore we can use a fixed 4×4 diagonal functional map with entries $1, 1, -1, -1$ as an initial guess for human symmetry detection. Results with this initialization are shown in the supplementary materials.

5.2.2 Refinement for shape matching. We applied our technique to refine maps between pairs of shapes and compared our method with recent state-of-the-art refinement techniques, including RHM [Ezuz et al. 2019], PMF [Vestner et al. 2017b], BCICP [Ren et al. 2018], Deblur [Ezuz and Ben-Chen 2017], as well as the standard refinement ICP [Ovsjanikov et al. 2012].
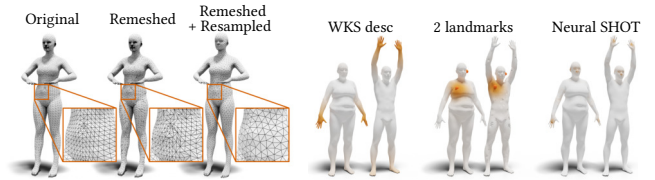
Fig. 16. Different triangulation

Fig. 17. Different descriptors

Table 2. **Quantitative evaluation of refinement for shape matching**. The Original and Remeshed datasets include 300 shape pairs. The Resampled dataset includes 190 FAUST pairs and 153 SCAPE pairs.

		Average Error ($\times 10^{-3}$)			Average Runtime (s)		
Method \ Dataset		Original	Remeshed	Resampled	Original	Remeshed	Resampled
Ini		67.3	44.0	46.5	-	-	-
ICP		54.0	36.3	29.3	10.2	10.1	5.32
Deblur		61.9	38.6	44.4	10.9	11.7	10.4
RHM		41.9	33.3	32	41.4	42.5	47.4
PMF		26.4	25.9	86.4	736.5	780.2	311.5
BCICP		21.6	19.5	26	183.7	117.8	364.2
Ours		**15.8**	**13.3**	**21.7**	9.60	9.64	6.49
Ours*		17.5	14.5	24.6	**1.14**	**1.15**	**0.68**
Improv.	Ours	**26.9%**	**31.8%**	**16.5%**	19×	12×	56×
	Ours*	19.0%	25.6%	5.4%	**160×**	**100×**	**535×**

For each dataset (FAUST and SCAPE), we consider three different versions. (1) Original: where all the meshes have the same triangulation. (2) Remeshed: we randomly flipped 12.5% of the edges (using gptoolbox [Jacobson et al. 2018]) keeping the vertex positions unchanged to maintain a perfect ground-truth. (3) Remeshed + Resampled (called "Resampled" in Table 2): we use the datasets provided in [Ren et al. 2018], where each shape is remeshed and resampled independently, having different number of vertices (around 5k) and often significantly different triangulation. As such, these are more challenging than the original datasets on which near-perfect results have been reported in the past. Figure 16 shows a FAUST shape in the three versions.

To demonstrate that our algorithm works with different initializations, we use three different types of descriptors to compute the initial functional maps (with size 20×20) for the three datasets: (1) WKS; (2) descriptors derived from two landmarks (see the two spheres highlighted in the middle of Figure 17); (3) Learned SHOT descriptors [Roufosse et al. 2018]: the descriptors computed by a non-linear transformation of SHOT, using an unsupervised deep learning method trained on a mixed subset of the remeshed and resampled SCAPE and FAUST dataset. For the experiments with WKS descriptors, we also use the orientation-preserving operators [Ren et al. 2018] to disambiguate the symmetry of the WKS descriptors.

Table 2 reports the average error and runtime, while the corresponding summary curves are in the supplementary materials. Figure 6 shows a qualitative example. Our method without acceleration achieves 26.9%, 31.8%, and 16.5% improvement in accuracy over the state-of-the-art while being 10 to 50 times faster. With acceleration, our method is more than 100-500× faster than the top existing method while still producing accuracy improvement. Our method is also much simpler than BCICP (see Appendix B for an overview of the source code of BCICP and our method). Interestingly, we also note that the method in [Roufosse et al. 2018] overfits severely when trained directly on functional maps of size 120 and

Table 3. **SHREC19 summary**. We compare with the refinement techniques RHM, PMF, BCICP and the baseline ICP on 430 shape pairs. We report an accuracy improvement over BCICP (the top performing method on this benchmark), and a significant gap in runtime performance over all methods.

Method		#samples	Measurement	
			Avg. Error ($\times 10^{-3}$)	Avg. Runtime (s)
Initialization		-	60.4	-
ICP		-	47.0	87.3
Deblur		-	55.4	102.1
RHM		-	42.6	2313
PMF		500	56.2	72.9
		1000	51.8	118.1
		5000	83.2	349.3
BCICP		500	40.7	90.0
		1000	33.6	163.7
		5000	30.1	437.9
Ours*		**500**	**28.8**	**1.5**
Improv.	Ours*	500	4%	**290×**

results in an average error of 97.5. In contrast, training on smaller functional maps and using our upsampling leads to average error of 21.7. Please see the supplementary for an illustration. We provide evaluation of other quantitative measurements such as bijectivity, coverage, and edge distortion in Appendix C. We also provide additional qualitative examples and comparison to the Deblur method on non-isometric shapes in Appendix D.

5.2.3 Matching different high-resolution meshes. SHREC19 [Melzi et al. 2019] is a recent benchmark composed of 430 human pairs with different connectivity and mesh resolution, gathered using 44 different shapes from 11 datasets. Each shape is aligned to the SMPL model [Loper et al. 2015] using the registration pipeline of [Marin et al. 2018], thus providing a dense ground truth for quantitative evaluation. This benchmark is challenging due to high shape variance and due to the presence of high-resolution meshes (5K to 200K vertices, see supplementary materials for examples). In Table 3 we report full comparisons in terms of average error and runtime.

Since BCICP and PMF require a full geodesic distance matrix as input, we apply them on simplified shapes (we used MATLAB's `reducepatch` for the remeshing). The refined maps are then propagated back to the original meshes via nearest neighbors; please see the supplementary materials for more details.

We initialize ZOOMOUT with the 20×20 functional map provided as baseline in [Melzi et al. 2019], and upsample this map to size 120×120 with a step of size 5. Our method achieves the best results while being over 290× faster. We also highlight that although we have a similar accuracy as BCICP, we better preserve the local details as shown in Figure 18, since we avoid the mesh simplification and map transfer steps.

In the supplementary materials, we further compare to methods that are applicable on full-resolution meshes directly. The experiment is conducted on a subset of SHREC19 and our method achieves a significant improvement in accuracy.

5.2.4 Point cloud surfaces. Several standard methods for meshes typically fail when applied to point clouds. We tested our approach on point clouds generated from the FAUST and TOSCA datasets, by sampling points within mesh triangles uniformly at random. We estimate the Laplace operator on point clouds as proposed in [Belkin
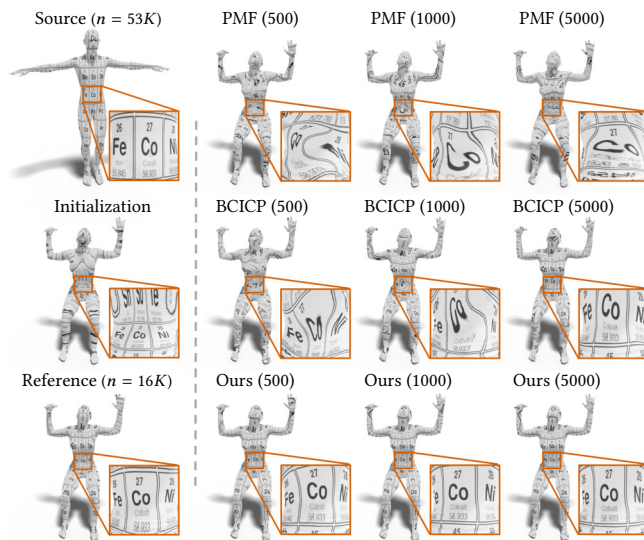

Fig. 18. Different sampling density. Here we show an example from the SHREC19 benchmark on a pair of shape with 53K and 16K vertices respectively. We compare with PMF and BCICP under different sampling density (500, 1000, and 5000 samples). The computed maps are visualized via texture transfer. Our method achieves the best global accuracy while preserving the local details at the same time. Further, our method is much less dependent on the sampling density than BCICP or PMF.

Table 4. **Quantitative evaluation on point cloud surfaces.** Our method is both more accurate and faster than ICP on average.

Measurement \ Method	Ini	ICP	ICP$_{20}$	ICP$_{120}$	**Ours***	Improv. Ours
Average Error ($\times 10^{-3}$)	51.0	49.7	31.4	36.9	**22.3**	29.0%
Average Runtime (s)	-	29.6	8.3	305.2	**4.0**	2×

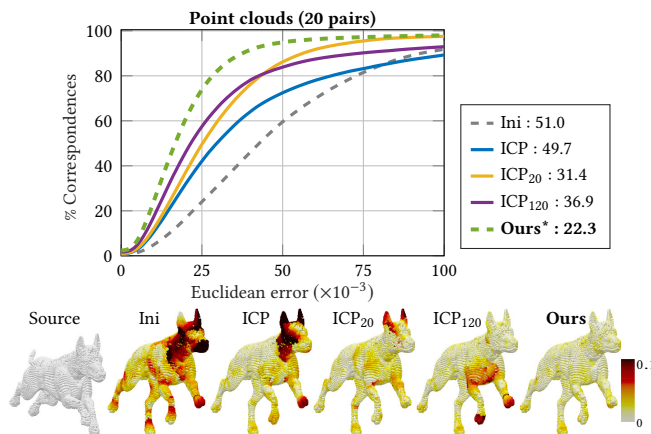Fig. 19. Results on non-rigid point cloud surfaces. We tested on 10 FAUST pairs and 10 TOSCA pairs. Below, we visualize geodesic error directly on the point clouds, defined as the Euclidean distance between the estimated matches and the ground truth. The heatmap grows from white (zero error) to dark red ($\geq$ 10% deviation from ground truth).

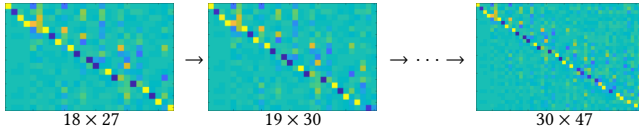et al. 2009]. The initial 20×20 functional map is estimated with

Fig. 20. Partial matching involves functional maps **C** with slanted diagonal. To account for this particular structure, we iteratively increase the two dimensions of **C** by different amounts, see Equations (9)-(10). This allows correct upsampling, as shown in this example.
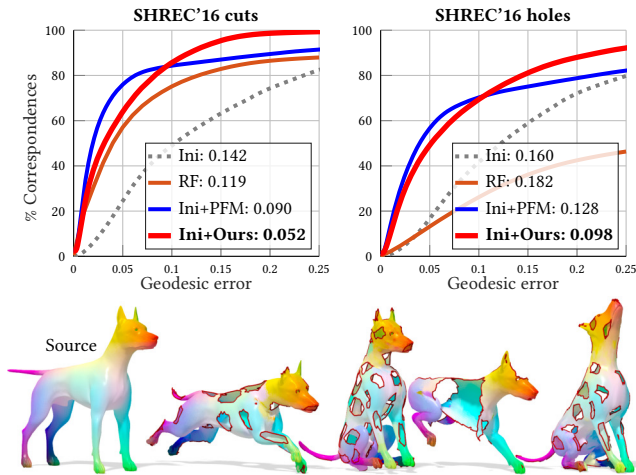

Fig. 21. *Top*: Comparisons on the SHREC'16 Partiality benchmark with the state of the art method Partial Functional Maps (PFM) [Rodolà et al. 2017] and with the Random Forests (RF) baseline [Rodolà et al. 2014]. Average runtimes are 6sec for our method and 70sec for PFM, both initialized with a 4 × 4 ground truth **C**. *Bottom*: Qualitative results on the dog class.

the approach of [Nogneng and Ovsjanikov 2017], using WKS and 2 landmarks (Ini). We then upsample from 20 to 120 with steps of size 5, and compare with ICP, ICP_{20} and ICP_{120}. Quantitative and qualitative results are shown in Table 4 and Figure 19.

5.2.5 Partial Matching. A particularly challenging setting of shape correspondence occurs whenever one of the two shapes has missing geometry. In [Rodolà et al. 2017] it was shown that, in case of partial isometries, the functional map matrix **C** has a "slanted diagonal" with slope proportional to the area ratio $\frac{A(\mathcal{N})}{A(\mathcal{M})}$ (here, $\mathcal{M}$ is a partial shape and $\mathcal{N}$ is a complete shape). Our spectral upsampling method can still be applied in this setting. To do so, we *weakly* enforce the expectation of a slanted diagonal by allowing rectangular **C**. Namely, we define the update rules for the step size as follows:

$$k_{\mathcal{M}} \mapsto k_{\mathcal{M}} + 1 \tag{9}$$

$$k_{\mathcal{N}} \mapsto k_{\mathcal{N}} + 1 + \lceil \frac{k_{\mathcal{N}}}{100}(100 - r) \rceil \tag{10}$$

where r is an estimate for rank(**C**) obtained via the formula $r = \max_{i=1}^{k_{\mathcal{M}}}\{i \mid \lambda_i^{\mathcal{M}} < \max_{j=1}^{k_{\mathcal{N}}} \lambda_j^{\mathcal{N}}\}$ after setting $k_{\mathcal{M}} = k_{\mathcal{N}} = 100$ (see [Rodolà et al. 2017, Eq. 9] for details). In the classical case where both $\mathcal{M}$ and $\mathcal{N}$ are full and nearly isometric, the estimate boils down to $r = \min\{k_{\mathcal{M}}, k_{\mathcal{N}}\} = 100$ and Eq. (10) reduces to $k_{\mathcal{N}} \mapsto k_{\mathcal{N}} + 1$; see Figure 20 for an illustration.
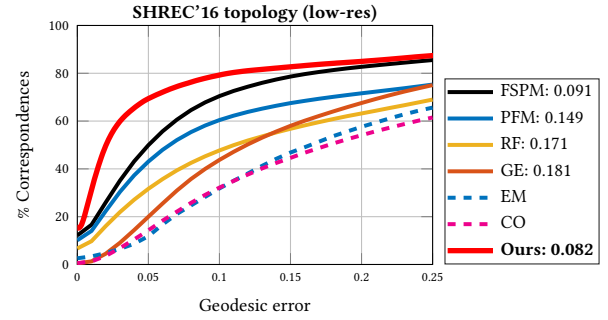
Fig. 22. Comparisons on the SHREC'16 Topology benchmark. Competing methods include PFM, RF, Green's Embedding (GE) [Burghard et al. 2017], Expectation Maximization (EM) [Sahillioğlu and Yemez 2012], Convex Optimization (CO) [Chen and Koltun 2015], and Fully Spectral Partial Matching (FSPM) [Litany et al. 2017]. Dashed curves indicate sparse methods.

Fig. 23. *Top*: the regions with topology noise are highlighted in orange; *Bottom*: maps computed using our method visualized via color transfer.

For these tests we adopt the SHREC'16 Partial Correspondence benchmark [Cosmo et al. 2016a], consisting of 8 shape classes (humans and animals) undergoing partiality transformations of two kinds: regular 'cuts' and irregular 'holes'. All shapes are additionally resampled independently to ∼10K vertices. Evaluation is performed over 200 shape pairs in total, where each partial shape is matched to a full template of the corresponding class. Quantitative and qualitative results are reported in Figure 21.

5.2.6 Topological Noise. We further explored the case of topological changes in the areas of self-contact (e.g., touching hands generating a geodesic shortcut). For this task, we compare with the state of the art on the SHREC'16 Topology benchmark [Lähner et al. 2016] (low-res challenge), consisting of 25 shape pairs (∼ 12K vertices) undergoing nearly isometric deformations with severe topological artifacts. We initialize our method with a 30 × 30 matrix **C** estimated via standard least squares with SHOT descriptors [Tombari et al. 2010]. Since self-contact often leads to partiality, we use the rectangular update rules (9)-(10). Results are reported in Figure 22. Figure 23 shows some example maps computed using our method.

5.2.7 Different Basis. In [Melzi et al. 2018] it was proposed to address the partial setting by considering a Hamiltonian $H_{\mathcal{M}} = L_{\mathcal{M}} + V_{\mathcal{M}}$ in place of the standard manifold Laplacian, where $V_{\mathcal{M}} = \text{diag}(1 - v)$ is a localization potential concentrated on the support of a given (soft) indicator function $v : \mathcal{M} \to [0, 1]$; eigenfunctions of $H_{\mathcal{M}}$ are supported on v. We performed experiments showing that spectral upsampling can still be applied *as-is* to improve the quality of maps, when these are represented in this alternative basis. In these tests we initialized as in [Melzi et al. 2018], and evaluated on

Table 5. Results in the transfer of different classes of functions, average on 20 pairs from FAUST dataset. Initial map size is 40×30 (Ini), final size of ours is 210×200. The methods marked with † are initialized with the initial functional maps refined by ICP. See text for details.

function	Ini	ICP	p2p†	ICP$_{200}$	Prod†	Ours	Ours†
HeatKernel	0.80	0.18	0.15	0.17	0.19	**0.10**	**0.10**
HeatKernel$_{200}$	0.95	0.84	0.52	0.34	0.65	**0.29**	**0.29**
HKS	0.66	0.55	0.21	0.21	0.28	0.14	**0.13**
WKS	0.51	0.15	0.06	0.11	0.13	**0.04**	**0.04**
XYZ	0.67	0.13	0.09	0.12	0.15	**0.05**	**0.05**
Indicator	0.77	0.30	0.18	0.20	0.26	**0.17**	**0.17**
SHOT	0.87	0.82	0.87	0.74	0.78	**0.73**	**0.73**
AWFT	0.45	0.26	0.18	0.19	0.24	**0.14**	**0.14**
Delta	0.98	0.93	0.67	0.43	0.82	**0.38**	**0.38**

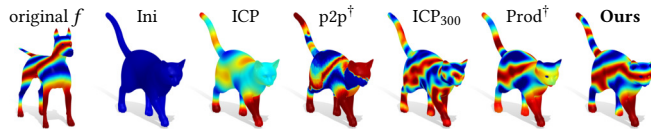original f Ini   ICP   p2p†   ICP$_{300}$ Prod†   **Ours**

Fig. 24. Function transfer example on a non-isometric pair from TOSCA. We show the original function on the source shape (leftmost) and the transfer results for the different methods. The functional map is upsampled from size 40×30 to 310×300. We mark the methods initialized with ICP with †.

the entire dataset of [Cosmo et al. 2016b], consisting of 150 cluttered scenes and 3 query models (animals). The results are reported in the supplementary materials.

5.2.8 Transfer of functions. Functional maps can be used to transfer functions without necessarily converting to pointwise correspondences. This application, however, can be hindered by the fact that small functional maps can only transfer low-frequency information. A recent approach [Nogneng et al. 2018] has tried to lift this restriction by noting that higher frequency functions can be transferred using "extended" bases consisting of pointwise products of basis functions. Our approach is similar in spirit since it also allows to extend the expressive power of a given functional map by increasing its size and thus enabling transfer of higher-frequency information.

We evaluated our method by directly comparing with the state of the art [Nogneng et al. 2018]. For 9 different classes of functions we compute the error as the norm of the difference between the transferred function and the ground truth g (obtained by transferring using the ground truth pointwise map), normalized by the norm of g. The functions considered are: Heat Kernel computed with 30 and with 200 eigenfunctions, descriptors HKS [Sun et al. 2009], WKS [Aubry et al. 2011], SHOT [Tombari et al. 2010], AWFT [Melzi et al. 2016], the coordinates of the 3D embedding, binary indicator of region, and the heat kernel with a very small time parameter approximating a delta function defined around a point. The results are reported in Table 5. We use the same parameters adopted in [Nogneng et al. 2018], and average over 20 random FAUST pairs. We refine the initial map (Ini) of size 40×30, computed using the approach of [Nogneng and Ovsjanikov 2017], to 210×200 with a step size of 1. We also compare to ICP: ICP refinement applied to Ini; p2p: function transfer using the point-to-point map obtained by ICP; ICP$_{200}$: ICP applied to a functional map of dimension 210×200 estimated through the same pipeline adopted for Ini; Prod: the method

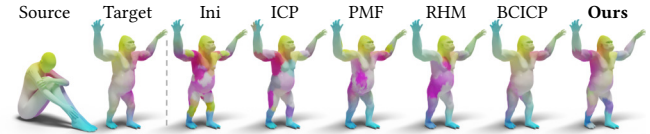Source   Target   Ini   ICP   PMF   RHM   BCICP   **Ours**

Fig. 25. Failure case. Here we show a challenging case where the initial map has left-to-right, back-to-front, and arm-to-leg ambiguity. When refining such a low-quality initial map, our method sometimes fails to produce a good refined map. However, our refinement still outperforms the regular ICP method with respect to the quality of the computed correspondences.

proposed in [Nogneng et al. 2018]. We outperform all the competitors for all the classes.

We also compare the results obtained by our method initializing the functional map after applying ICP, and the two are almost the same everywhere. A transfer example of a high-frequency function between a dog and a cat shapes from TOSCA is visualized in Figure 24. Our refinement achieves the best results with respect to all the competitors even in this non-isometric pair. In the supplementary materials we report other qualitative comparisons.

6 CONCLUSION, LIMITATIONS & FUTURE WORK

We introduced a simple but efficient map refinement method based on iterative spectral upsampling. We presented a large variety of quantitative and qualitative results demonstrating that our method can produce similar or better quality on a wide range of shape matching problems while typically improving the speed of the matching by an order of magnitude or more. We find it remarkable that our method has such strong performance, even though it is conceptually simple and only requires a few lines of code to implement. In many cases, our method outperforms very complex frameworks that consist of multiple non-trivial algorithmic components.

Our method still comes with multiple limitations. First, while being robust to noise, its success still depends on a reasonable initialization. Starting with a bad initialization, such as random functional maps, our method would produce poor results. Second, the method still relies on some parameters that have to be tuned for each application. Specifically, we need to identify the number of basis functions in the initialization and the final number of basis functions. Additionally, the step size during upsampling has to be chosen for optimal speed, but using a step size of one is always a safe choice. Finally, our method is very robust to deviations from perfect isometries, but still will fail for significantly non-isometric shape pairs. See examples in Figure 25 and in Appendix D. In future work, we would like to investigate how to automatically compute the minimal size of the input functional map and plan to extend our work to other settings such as general graphs and images.

REFERENCES

Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. 2016. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision* 118, 3 (2016), 380–392.

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

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. *ACM Transactions on Graphics* 24, 3 (July 2005), 408–416.

Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 1626–1633.

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.

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*. IEEE, Columbus, Ohio, 3794–3801.

Alex Bronstein, Michael Bronstein, and Ron Kimmel. 2008. *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY.

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

Qifeng Chen and Vladlen Koltun. 2015. Robust Nonrigid Registration by Convex Optimization. In *International Conference on Computer Vision (ICCV)*. IEEE, 2039–2047.

Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. 2015. Continuous matching via vector field flow. *Computer Graphics Forum* 34, 5 (2015), 129–139.

Luca Cosmo, Emanuele Rodolà, Michael Bronstein, Andrea Torsello, Daniel Cremers, and Yusuf Sahillioğlu. 2016a. Partial Matching of Deformable Shapes. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval (3DOR '16)*. Eurographics Association, 61–67. https://doi.org/10.2312/3dor.20161089

Luca Cosmo, Emanuele Rodolà, Jonathan Masci, Andrea Torsello, and Michael Bronstein. 2016b. Matching deformable objects in clutter. In *Proc. 3D Vision (3DV)*. 1–10.

Anastasia Dubrovina and Ron Kimmel. 2010. Matching shapes by eigendecomposition of the Laplace-Beltrami operator. In *Proc. 3DPVT*, Vol. 2.

Anastasia Dubrovina and Ron Kimmel. 2011. Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures. *Advances in Adaptive Data Analysis* 3, 01n02 (2011), 203–228.

Nadav Dym and Yaron Lipman. 2017. Exact recovery with symmetries for Procrustes matching. *SIAM Journal on Optimization* 27, 3 (2017), 1513–1530.

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.

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

Philipp Gunz and Philipp Mitteroecker. 2013. Semilandmarks: a method for quantifying curves and surfaces. *Hystrix, the Italian Journal of Mammalogy* 24, 1 (2013), 103–109.

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.

Ruqi Huang and Maks Ovsjanikov. 2017. Adjoint Map Representation for Shape Analysis and Matching. *Computer Graphics Forum* 36, 5 (2017), 151–163.

Alec Jacobson et al. 2018. gptoolbox: Geometry Processing Toolbox. http://github.com/alecjacobson/gptoolbox.

Varun Jain and Hao Zhang. 2006. Robust 3D shape correspondence in the spectral domain. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*. IEEE, 19–19.

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.

Martin Kilian, Niloy J Mitra, and Helmut Pottmann. 2007. Geometric modeling in shape space. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 64.

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

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

Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein. 2016. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision*. Springer, 680–696.

Zorah Lähner, Emanuele Rodolà, Michael Bronstein, Daniel Cremers, Oliver Burghard, Luca Cosmo, Alexander Dieckmann, Reinhard Klein, and Yusuf Sahillioğlu. 2016.

Matching of Deformable Shapes with Topological Noise. In *Proc. 3DOR*. 55–60.

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

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-person Linear Model. *TOG* 34, 6 (2015), 248:1–248:16.

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.

Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. 2018. FARM: Functional Automatic Registration Method for 3D Human Bodies.

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.

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

Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. 2016. Shape Analysis with Anisotropic Windowed Fourier Transform. In *International Conference on 3D Vision (3DV)*.

Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. 2018. Localized Manifold Harmonics for Spectral Shape Analysis. *Computer Graphics Forum* 37, 6 (2018), 20–34.

Marius Muja and David G. Lowe. 2014. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36 (2014).

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

Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael Bronstein, and Maks Ovsjanikov. 2018. Improved Functional Mappings via Product Preservation. *Computer Graphics Forum* 37, 2 (2018), 179–190.

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

Ulrich Pinkall and Konrad Polthier. 1993. Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics* 2, 1 (1993), 15–36.

Adrien Poulenard, Primoz Skraba, and Maks Ovsjanikov. 2018. Topological Function Optimization for Continuous Shape Matching. *Computer Graphics Forum* 37, 5 (2018), 13–25.

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)*.

Emanuele Rodolà, Samuel Rota Bulò, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. 2014. Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4177–4184.

Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. 2018. Unsupervised Deep Learning for Structured Shape Matching. *arXiv preprint arXiv:1812.03794* (2018).

Raif M Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. 2013. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 72.

Yusuf Sahillioğlu and Yücel Yemez. 2012. Minimum-distortion isometric shape correspondence using EM algorithm. *IEEE transactions on pattern analysis and machine intelligence* 34, 11 (2012), 2203–2215.

Guy L Scott and Hugh Christopher Longuet-Higgins. 1991. An algorithm for associating the features of two images. *Proc. R. Soc. Lond. B* 244, 1309 (1991), 21–26.

Meged Shoham, Amir Vaxman, and Mirela Ben-Chen. 2019. Hierarchical Functional Maps between Subdivision Surfaces. *Computer Graphics Forum* (2019). https://doi.org/10.1111/cgf.13789

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.

Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum* 28, 5 (2009), 1383–1392.

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.

Federico Tombari, Samuele Salti, and Luigi Di Stefano. 2010. Unique signatures of histograms for local surface description. In *Proc. ECCV*. Springer, 356–369.

Shinji Umeyama. 1988. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence* 10, 5 (1988), 695–703.

Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. 2011. A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707.

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.

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.

A THEORETICAL ANALYSIS

Proof of Theorem 4.1. We will prove this theorem with the help of the following well-known lemma, for which we give the proof in the Supplementary Material for completeness:

Lemma A.1. *Let us be given a pair of shapes $\mathcal{M}, \mathcal{N}$ each having non-repeating Laplacian eigenvalues, which are the same. A point-to-point map $T : \mathcal{M} \rightarrow \mathcal{N}$ is an isometry if and only if the corresponding functional map $\mathbf{C}$ in the complete Laplacian basis is both diagonal and orthonormal.*

Proof. To prove Theorem 4.1 first suppose that the map T is an isometry, and thus, thanks to Lemma A.1, the functional map $\mathbf{C} = \Phi_{\mathcal{M}}^{+}\Pi\Phi_{\mathcal{N}}$ is diagonal and orthonormal. From this, it immediately follows that every principal submatrix of $\mathbf{C}$ must also be orthonormal implying $E(\mathbf{C}) = 0$.

To prove the converse, suppose that $\mathbf{C} \in \mathcal{P}$. Then $E(\mathbf{C}) = 0$ implies that every principal submatrix of $\mathbf{C}$ is orthonormal. By induction on k this implies that $\mathbf{C}$ must also be diagonal. Finally since $\mathbf{C} \in \mathcal{P}$, again using Lemma A.1 we obtain that the corresponding pointwise map must be an isometry. □

A.1 Map Recovery

Our goal is to prove that Eq. (6) with the regularizer $\mathcal{R}(\Pi) = \|(I - \Phi_{\mathcal{M}}^{k}(\Phi_{\mathcal{M}}^{k})^{+})\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{K}^{T}\|_{A_{\mathcal{M}}}^{2}$ is equivalent to solving $\min_{\Pi} \|\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} -$

$\Phi_{\mathcal{M}}^{k}\|_{F}^{2}$: In other words:

$$\arg\min_{\Pi} \|(\Phi_{\mathcal{M}}^{k})^{+}\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - I_{k}\|_{F}^{2} + \|(I - \Phi_{\mathcal{M}}^{k}(\Phi_{\mathcal{M}}^{k})^{+})\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{K}^{T}\|_{A_{\mathcal{M}}}^{2}$$

$$= \arg\min_{\Pi} \|\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - \Phi_{\mathcal{M}}^{k}\|_{F}^{2}. \tag{11}$$

For this we use the following result: for any matrix X and basis B that is orthonormal with respect to a symmetric positive definite matrix A, i.e. $B^{T}AB = Id$, and thus $B^{+} = B^{T}A$, if we let $\|X\|_{A}^{2} = tr(X^{T}AX)$ then: $\|X\|_{A}^{2} = \|B^{+}X\|_{F}^{2} + \|(I - BB^{+})X\|_{A}^{2}$. To see this, observe that $\|B^{+}X\|_{F}^{2} = tr(X^{T}ABB^{T}AX)$ while $\|(I - BB^{+})X\|_{A}^{2} = tr\left(X^{T}(I - ABB^{T})A(I - BB^{T}A)X\right) = tr\left(X^{T}(A - ABB^{T}A)X\right)$ since $B^{T}AB$. We now use this result with $X = \Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - \Phi_{\mathcal{M}}^{k}$, $B = \Phi_{\mathcal{M}}^{k}$ and $A = A_{\mathcal{M}}$. This gives:

$$\|\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - \Phi_{\mathcal{M}}^{k}\|_{A_{\mathcal{M}}}^{2} = \|(\Phi_{\mathcal{M}}^{k})^{+}\left(\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - \Phi_{\mathcal{M}}^{k}\right)\|_{F}^{2}$$
$$+ \|\left(I - \Phi_{\mathcal{M}}^{k}(\Phi_{\mathcal{M}}^{k})^{+}\right)\left(\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - \Phi_{\mathcal{M}}^{k}\right)\|_{A_{\mathcal{M}}}^{2}$$
$$= \|(\Phi_{\mathcal{M}}^{k})^{+}\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - I_{k}\|_{F}^{2} + \|\left(I - \Phi_{\mathcal{M}}^{k}(\Phi_{\mathcal{M}}^{k})^{+}\right)\Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T}\|_{A_{\mathcal{M}}}^{2}.$$

It remains to show that $\arg\min_{\Pi} \|X\|_{A_{\mathcal{M}}}^{2} = \arg\min_{\Pi} \|X\|_{F}^{2}$ with $X = \Pi\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T} - \Phi_{\mathcal{M}}^{k}$. For this note simply that since Π represents a pointwise map, both problems reduce to finding the row of $\Phi_{\mathcal{N}}^{k}\mathbf{C}_{k}^{T}$ that is closest to each of the rows of $\Phi_{\mathcal{M}}^{k}$.

Note that in supplementary material we derive both an alternative approach to ZoomOut and, as mentioned in Section 4.3. provide a link between our approach and PMF.

B IMPLEMENTATION

This Appendix lists standard Matlab code for our method and BCICP, which is the most competitive method to ours while being orders of magnitude slower. Note that a fully-working version of ZoomOut **can be implemented in just 5 lines of code**, while BCICP relies on the computation of all pairs of geodesics distances on both shapes, and even after pre-computation, is more than 250 lines of code relying on numerous parameters and spread across a main procedure and 4 utility functions.

B.1 Source Code - ZoomOut

```
1  function [C,P]=ZoomOut(M,N,C,k_final)
2
3  for k=size(C,1):k_final-1
4    x = knnsearch(N.Phi(:,1:k)*C',M.Phi(:,1:k));
5    P = sparse(1:M.n,x,1,M.n,N.n);
6    C = M.Phi(:,1:k+1)'*M.A*P*N.Phi(:,1:k+1);
7  end
```

B.2 Source Code - BCICP [Ren et al. 2018]

```
1  Precompute:
2  (1) complete pairwise geodesic distance matrix of each
        shape;
3  (2) vertex one-ring neighbor;
4  (3) edge list of each mesh
5
```

```
6   function [T12,T21]=BCICP(S1,S2,T12,T21,K)
7   B1 = S1.Phi(:,1:50);
8   B2 = S2.Phi(:,1:50);
9   for k=1:K
10      [T21,T12]=refine_pMap(T21,T12,S1,S2);
11      C12 = B2\B1(T21,:);
12      C21 = B1\B2(T12,:);
13      T12 = knnsearch(B2*C21',B1);
14      T21 = knnsearch(B1*C12',B2);
15      [T21,T12]=refine_pMap(T21,T12,S1,S2);
16      C1 = B1\B1(T21(T12),:);
17      C1 = mat_projection(C1);
18      C2 = B2\B2(T21(T12),:);
19      C2 = mat_projection(C2);
20      T21 = knnsearch(B2(T12,:)*C2',B2);
21      T12 = knnsearch(B1(T21,:)*C1',B1);
22  end
23  end
24
25  function [T21,T12]=refine_pMap(T21,T12,S1,S2)
26  for k=1:4
27      T12 = improve_coverage(T12,S1,S2);
28      T21 = improve_coverage(T21,S1,S2);
29      T12 = improve_smoothness(T12,S1,S2);
30      T21 = improve_smoothness(T21,S1,S2);
31      T12 = fix_outliers(T12,S1,S2);
32      T21 = fix_outliers(T21,S1,S2);
33  end
34  end
35
36  function [T12]=improve_coverage(T12,S1,S2)
37  % around 120 lines of code
38  function [T12]=improve_smoothness(T12,S1,S2)
39  % around 50 lines of code
40  function [T12]=fix_outliers(T12,S1,S2)
41  % around 50 lines of code
42  ...
```

C ADDITIONAL MEASUREMENTS

Given 10 random shape pairs from the FAUST original dataset, Table 6 shows the performance summary of different refinement methods w.r.t. the following measurements as used in [Ren et al. 2018]. Specifically, we evaluate the maps T_{12} and T_{21} between a pair of shapes S_1 and S_2:

- **Accuracy**. We measure the geodesic distance between T_{12} (and T_{21}) and the given ground-truth correspondences.
- **Un-Coverage**. The percentage of vertices/areas that are NOT covered by the map T_{12} (or T_{21}) on shape S_2 (or S_1)
- **Bijectivity**. The composite map $T_{21} \circ T_{12}$ (or $T_{12} \circ T_{21}$) gives a map from the shape to itself. Thus, we measure the geodesic distance between this composite map and the identity.
- **Edge distortion**. We measure how each edge in S_1 (or S_2) is distorted by the map T_{12} (or T_{21}) as follows:

$$e_{v_i \sim v_j} = \left(\frac{d_{S_2}(T_{12}(v_i), T_{12}(v_j))}{d_{S_1}(v_i, v_j)} - 1 \right)^2$$

We then average the distortion error over all the edges as a measure for the map smoothness.

Note that the PMF method optimizes for a permutation matrix directly, that is why the computed maps covered all the vertices and give almost zero bijectivity error in Table 6 (the bijectivity error is not strictly zero because the maps T_{12} and T_{21} are computed

Table 6. **Additional measurements.** Besides the map accuray, we also measure the coverage, bijectivity, and edge distortion as a smoothness measure on 10 random shape pairs from the original FAUST dataset.

Measurement\ Method	Ini	Refinement methods					
		ICP	PMF (gauss)	RHM	BCICP	**ours**	ours*
Accuracy ($\times 10^{-3}$)	98.4	85.8	36.3	63.9	49.9	**33.3**	36.8
Un-Coverage (%)	72.3	42.4	**0**	44.5	15.9	23.6	30.7
Bijectivity ($\times 10^{-3}$)	104	89.6	**1.90**	24.6	5.48	15.6	14.8
Edge distortion	10.9	26.4	37.3	3.69	5.49	**1.16**	4.37

independently). The method BCICP includes heuristics to explicitly improve the coverage, bijectivity, and smoothness. Even though our method is not designed to optimize these measurements, it still achieves reasonable performance. Note that our method gives the smallest edge distortion, which suggests that our method not only gives the most accurate map but also the smoothest map w.r.t. all the competing methods.
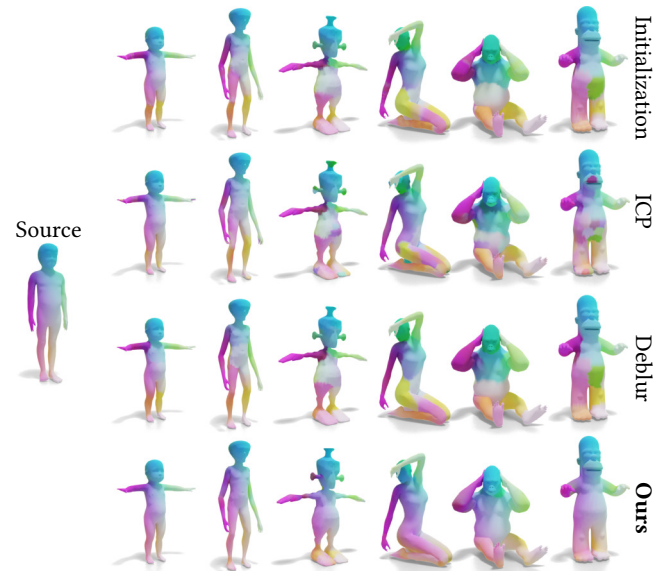
D COMPARISON TO DEBLUR

Fig. 26. Comparison to [Ezuz and Ben-Chen 2017] on non-isometric shape pairs.

The work of [Ezuz and Ben-Chen 2017] also provides an approach for recovering a point-wise map from a functional map, based on a different energy. Specifically, our energy defined in Eq. (3) and the resulting point-wise map conversion step defined in Eq. (6) are different from the deblurring energy defined in Eq. (4) in [Ezuz and Ben-Chen 2017]. Figure 26 shows a qualitative comparison between our method and this method, which we call "Deblur." We use 5 landmarks to compute the initial maps (first row), and we then apply ICP, Deblur, and ours to refine the initial maps. Note that in these examples, we rescale the target shapes to the same surface area as the source shape. We can see that even when the shape pair is far from isometry, our method can still produce reasonable maps, even though our theory relies on the isometry assumption.