

Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization

Robin Magnet
LIX, École Polytechnique, IP Paris
rmagnet@lix.polytechnique.fr

Jing Ren
ETH Zurich
jing.ren@inf.ethz.ch

Olga Sorkine-Hornung
ETH Zurich
sorkine@inf.ethz.ch

Maks Ovsjanikov
LIX, École Polytechnique, IP Paris
maks@lix.polytechnique.fr

Abstract

We introduce pointwise map smoothness via the Dirichlet energy into the functional map pipeline, and propose an algorithm for optimizing it efficiently, which leads to high-quality results in challenging settings. Specifically, we first formulate the Dirichlet energy of the pulled-back shape coordinates, as a way to evaluate smoothness of a pointwise map across discrete surfaces. We then extend the recently proposed discrete solver and show how a strategy based on auxiliary variable reformulation allows us to optimize pointwise map smoothness alongside desirable functional map properties such as bijectivity. This leads to an efficient map refinement strategy that simultaneously improves functional and point-to-point correspondences, obtaining smooth maps even on non-isometric shape pairs. Moreover, we demonstrate that several previously proposed methods for computing smooth maps can be reformulated as variants of our approach, which allows us to compare different formulations in a consistent framework. Finally, we compare these methods both on existing benchmarks and on a new rich dataset that we introduce, which contains non-rigid, non-isometric shape pairs with inter-category and cross-category correspondences. Our work leads to a general framework for optimizing and analyzing map smoothness both conceptually and in challenging practical settings.

1. Introduction

Shape correspondence is a fundamental task in Geometry Processing, acting as a building block for many downstream applications [46, 43, 9]. One of the key challenges in designing a successful general-purpose shape matching method is the choice of the objective function that should promote high quality correspondences and, at the same time, be easy enough to optimize in order to be applicable on complex, densely sampled geometric objects.

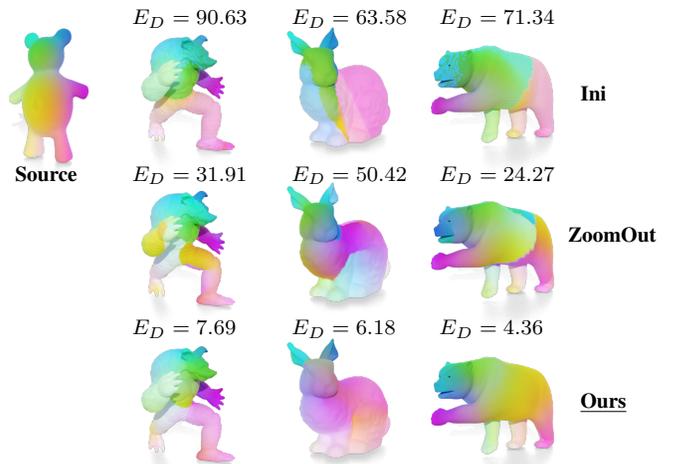


Figure 1. Our method can deal with noisy inputs and produce high-quality and smooth pointwise maps for non-isometric shape pairs. As a comparison, ZoomOut [28], the current state-of-the-art refinement method, cannot explicitly control the map smoothness and can have large discontinuous patches in the obtained maps. We report the smoothness metric E_D for each map.

A widely acknowledged desirable objective in non-rigid shape matching is *smoothness*, which intuitively promotes local consistency or continuity of computed correspondences, while being less restrictive than, e.g., isometries or conformal maps. Several works have incorporated smoothness into the map computation pipelines either via auxiliary energy terms [16], or by structuring the search space privileging continuous, often low frequency, correspondences or deformation fields, e.g., [12, 13]. Despite the utility of smoothness as a supervising signal in map computation, existing approaches can either be difficult to scale to dense meshes or are incorporated in an ad-hoc manner. Moreover, there is no coherent framework for comparing different existing strategies for promoting map smoothness using a single consistent computational and conceptual formalism.

In this paper we focus on the functional map framework, which was originally proposed as a tool for near-isometric



Figure 2. **DEFORMTHINGS4D-MATCHING Dataset.** We construct a new dataset for non-isometric shape matching based on the DEFORMTHINGS4D [25]. We show some example humanoid shapes and visualize the cross/inter-category correspondences via color transfer. Note that the shapes in the same category are remeshed independently (zoom in to see the mesh wireframes).

shape matching [32] and has since then been significantly extended to different tasks [42, 21] and correspondence models [40, 24, 10], among many others. The key advantages of this framework are its efficiency and flexibility. The efficiency of functional maps-based approaches stems from representing maps as small matrices using a reduced basis, which leads to small-scale optimization problems. At the same time, this framework is flexible and can incorporate a wide range of desirable constraints using simple linear algebraic formulations, e.g., [14, 31, 33].

Although originally functional map-based methods focused on constraints in the functional (spectral) domain, recent works have started to highlight and exploit links that exist between pointwise and functional map representations, while leveraging the strengths of both [39, 16, 38]. Specifically, a recent discrete optimization scheme was proposed in [38], demonstrating that many desirable map properties can be optimized directly in the pointwise map representation. Unfortunately, while the class of energies considered in [38] covers many existing functional map objectives, such as bijectivity or commutativity with the Laplacian, it does not address desirable pointwise map properties such as map smoothness. This can lead to local inconsistencies, such as discontinuous mapped patches, thus severely limiting the utility of the computed maps in practice.

In this paper we introduce a novel method that allows to explicitly promote pointwise map smoothness within the functional map framework. Our method is based on, first, formulating smoothness as the optimization of the Dirichlet energy of the pointwise map, and second, an iterative method for solving this energy optimization by extending the method introduced in [38]. This allows our approach to be used alongside other desirable objectives, while explicitly promoting smooth and locally consistent maps. We therefore both extend the scope of discrete map optimization to new energies not covered in [38] and use this insight to develop an efficient non-rigid shape matching approach that directly promotes pointwise map smoothness.

In addition to introducing a novel method for promoting smooth maps within the functional maps framework, we also investigate multiple previous approaches for computing smooth maps in different settings [16, 13, 45, 3] and show how they can be interpreted as variants of each other

and thus compared within a unified formalism. This allows us to design a *family* of different approaches, parametrized by the choice of the smoothness energy and its associated optimization strategy. We propose a coherent formalism within which various energies can be compared and demonstrate their relative utility in different settings. Finally, we observe that most public datasets focus on near-isometric pairs, making it non-trivial to evaluate accuracy and smoothness in more realistic scenarios, which can involve diverse and non-isometric shapes. To fill this gap we introduce a new challenging dataset based on DEFORMTHINGS4D [25], but with additional cross-category ground truth maps (Fig. 2). We use this dataset alongside existing benchmarks in a comprehensive comparison of various approaches computing smooth correspondences. To summarize, our key **contributions** include:

1. We show how pointwise map smoothness can be formulated and optimized within the functional map framework, by extending the discrete solver proposed in [38].
2. Based on this construction, we introduce a simple and effective map refinement method that is both computationally efficient and leads to high-quality results in non-isometric settings (Fig. 1).
3. We show how several previously proposed methods are intimately related both to our approach and within themselves, and propose a coherent framework, allowing us to directly compare ways to promote smoothness within a consistent formalism and computational strategy.
4. We construct a new dataset for non-rigid shape matching tasks with inter-category correspondences for animal shapes, and inter-/cross-category correspondences for humanoid shapes that are independently remeshed.

2. Related Work

In this section, we briefly review the previous works of shape matching, commonly used map evaluation metrics, and various map solvers, that are most related to this work. We refer to recent surveys [9, 43, 4] for more thorough discussions of shape matching.

Shape Matching Our work focuses on the problem of shape matching, that looks for dense correspondences between two non-rigid 3D shapes. One solution to shape

matching is to solve for correspondences directly by minimizing an explicit and carefully designed energy [6, 20, 34], which can lead to complex combinatorial problems with high computational complexity. An alternative solution is to find correspondences between parametric representations, where the input shapes are mapped into a canonical domain [26, 2, 1]. Our work is based on the functional map representation [32, 33], which computes correspondences between functions defined on the shapes. Different regularizers have been proposed to promote the accuracy of functional maps [31, 30, 39, 19, 49, 50]. Computing a functional map is usually reduced to solving a least-square system, which has a relatively low computation cost, but recovering a point-wise correspondence from the computed functional map is error-prone [41, 15, 38]. To further improve the accuracy of the recovered point-wise correspondences, different refinement methods have been proposed as a post-processing step [44, 27, 48, 47]. A common technique for map refinement in the functional maps framework is to iteratively update functional maps and the underlying point-wise maps according to different energies such as Dirichlet energy and bijectivity [32, 16, 39, 28, 37, 38]. In this work, we present a new refinement method that can robustly deal with noisy input and efficiently produce smooth maps in the functional maps framework.

Metrics for Map Quality Evaluation Different criterion have been taken into consideration to evaluate map quality, which are incorporated into map computation. The most commonly-used metric is the map accuracy, which is measured by comparing the geodesic distance between the mapped position and the pre-specified ground-truth position. Some previous work [3, 27, 19] adopt a landmark term to enforce map accuracy. To achieve a fully automatic solution, other metrics such as smoothness, bijectivity, conformality, and coverage are considered for map optimization other than accuracy which needs manually specified landmarks. For example, Reversible Harmonic Maps [16] proposes to optimize the Dirichlet energy together with the bijectivity of the pointwise maps. Smooth Shells [13] adopts the ARAP energy [45] to compute a smooth deformation field, which potentially leads to a smooth pointwise map. [23] blends across multiple maps to get a smooth one. [39] proposes heuristics to improve the bijectivity, smoothness, and coverage of the pointwise map in both spatial and spectral domain. In this work, we observe how several previous proposed approaches are closely related in formulating map smoothness. We show different variants can be compared in a coherent way within a consistent formalism.

Map Solver Previous methods adopt different search space for maps and hence need different solvers. For example, some work [17, 44, 11, 35] solve for maps that are represented by doubly stochastic matrices. Functional

maps framework [32, 31, 30, 39] usually solves a least-square system for functional maps. Quadratic-splitting technique [15, 16] is also used to solve vertex-to-point (also called precise) maps. [38] introduces a discrete solver to optimize commonly used functional map energies constrained on the proper functional maps, which is a subset of functional maps that are associated with pointwise maps. In this work, we introduce map smoothness into functional map pipeline and present an efficient algorithm to minimize the smoothness which extends the scope of discrete solver.

3. Notation & Background

Notation Given a triangle mesh $\mathcal{S} = (X, F)$ with the vertex positions X and face set F , we denote the cotangent weight matrix by W and the diagonal lumped mass matrix by A [29]. By solving the generalized eigenvalue problem $W\varphi_j = \lambda_j A\varphi_j$, we can obtain the Laplace-Beltrami basis Φ by collecting the first k eigenfunctions as columns, i.e., $\Phi = [\varphi_1 \dots \varphi_k]$ and the corresponding eigenvalues in a diagonal matrix, denoted as $\Delta = \text{diag}(\lambda_1 \dots \lambda_k)$. We then have $\Phi^T A \Phi = I$. A pointwise map is denoted as $\Pi_{ij} : \mathcal{S}_i \rightarrow \mathcal{S}_j$, where the subscript indicates the map direction. Specifically, $\Pi_{ij} \in \{0, 1\}^{n_i \times n_j}$ (n_i is the number of vertices in \mathcal{S}_i) is a binary matrix indicating the correspondences between the two shapes. For example, if the p -th vertex on \mathcal{S}_i is mapped to the q -th vertex on shape \mathcal{S}_j , we then have $\Pi_{ij}(p, q) = 1$ and $\Pi_{ij}(p, t) = 0$ for $\forall t \neq q$.

Functional Maps Framework The goal of shape matching is to find semantically meaningful and continuous pointwise map for a given shape pair. In this work, we follow the functional map framework [32] and encode a point-wise map as a linear transformation (called functional map) in the Laplace-Beltrami basis. Specifically, for a pointwise map $\Pi_{ij} : \mathcal{S}_i \rightarrow \mathcal{S}_j$, the associated functional map is given as $C_{ji} = \Phi_i^\dagger \Pi_{ij} \Phi_j$. Note that C_{ji} a pull-back linear operator that maps functions on shape \mathcal{S}_j to functions on shape \mathcal{S}_i . In the original pipeline [32], a functional map is computed by solving a least-squared system in the continuous linear operator space, i.e., $C_{21} = \arg \min_{C \in \mathbb{R}^{k_1 \times k_2}} E(C)$, where $E(\cdot)$ is a functional map energy that preserves input descriptors or landmarks, surface area or angles, multiplicative operators, or shape orientation [32, 30, 39, 22]. Solving for a function map in the unconstrained search space simplifies the optimization problem, but can lead to errors when converting the computed functional map to a point-wise one [41, 15, 38]. Thus, additional post-processing techniques are used to improve the quality of the pointwise maps [39, 16, 28, 35].

Discrete Optimization A recent work [38] has proposed a *discrete solver* for functional map pipeline which constrains the optimization problem to the space of *proper functional maps*. Specifically the functional map, $C_{21} =$

$\arg \min_{C \in \mathcal{P}_{21}} E(C)$, is solved in a discrete search space $\mathcal{P}_{21} = \{C_{21} \mid \exists \Pi_{12} \text{ s.t. } C_{21} = \Phi_1^\dagger \Pi_{12} \Phi_2\}$, i.e., the set of functional maps arising from *some* pointwise correspondence. The general strategy to solve this constrained problem, advocated in [38] mimics the Augmented Lagrangian methods with variable splitting [18] and consists of the following two main steps: (i) reformulate the energy $E(\cdot)$ by making C_{21} and Π_{12} independent variables, and adding a *coupling* term:

$$E_{\text{couple}}(C_{21}, \Pi_{12}) = \|C_{21} - \Phi_1^\dagger \Pi_{12} \Phi_2\|_F^2, \quad (1)$$

(ii) iteratively solve for C_{21} and Π_{12} with the other variables fixed. This approach is shown to be efficient and leads to high-quality and well-regularized functional maps. Key to the success of this strategy is the ability to reformulate the given functional map energy so that the resulting optimization problems for C_{21} and Π_{12} in step (ii) can be solved in closed form. In [38], a range of energies is considered including bijectivity, landmarks preservation, orthogonality and Laplacian commutativity.

Dirichlet Energy Given two Riemannian manifolds \mathcal{S}_1 and \mathcal{S}_2 , the Dirichlet energy of a map $f : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ is defined as $E_D(f) = \frac{1}{2} \int_{\mathcal{S}_1} \|df\|^2 d\mu_{\mathcal{S}_1}$, with df the map differential, which intuitively acts as a measure of the stretch induced by the map (see, e.g., [16] for a discussion). A *smooth* map f is therefore characterized as being a minimizer of the Dirichlet energy. In the discrete setting, a map $f : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ can be seen as a function between the two surface embeddings (i.e., $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$) and is assumed to be affine on each face. We can then define the discrete Dirichlet energy [36]:

$$E_D(f) = \sum_{(x_i, x_j) \in \mathcal{E}(\mathcal{S}_1)} w_{ij} \|f(x_i) - f(x_j)\|^2, \quad (2)$$

where $\mathcal{E}(\mathcal{S}_1)$ is the set of edges on \mathcal{S}_1 and w_{ij} the cotangent weight of edge (i, j) . We can rewrite Eq. (2) in a more compact way: $E_D(f) = \text{Trace}(f^\top W_1 f) := \|f\|_{W_1}^2$, where W_1 is the cotangent weight matrix of shape \mathcal{S}_1 .

Note that in practice one only needs to store the value of f at each vertex of \mathcal{S}_1 and therefore if f is a pointwise map from \mathcal{S}_1 to \mathcal{S}_2 , we can represent it in matrix form $f = \Pi_{12} X_2$, where the value at row i gives the coordinates $f(x_i)$. We therefore define the Dirichlet energy of the map Π_{12} as the Dirichlet energy of f , which is the W -norm of the pull-back vertex coordinates:

$$E_D(\Pi_{12}) = \|\Pi_{12} X_2\|_{W_1}^2. \quad (3)$$

Note that [16] adopts a similar formulation to measure the smoothness of a given map, but pulls-back a high-dimensional embedding, in which the L^2 distance approximates the geodesic distance, and that is computed via multidimensional scaling [8].



Figure 3. Previous methods focus on improving map accuracy and do not have explicit control over the map smoothness. Here we show an example of a non-isometric pair. We report the Dirichlet energy (E_D) of maps after refinement by different methods.

While the Dirichlet energy defines a measure of distortion induced by a map, we note that mapping all vertices in \mathcal{S}_1 to a single vertex in \mathcal{S}_2 leads to zero energy, as seen by setting $f(x_i) = y$ for some fixed y in Eq. (2). The Dirichlet energy thus only contains partial information about the quality of the map, and one needs to use additional constraints to obtain a *non-trivial* smooth map.

4. Discrete Solver for Dirichlet Energy

While functional maps intrinsically represent correspondences using low frequency eigenfunctions, thus inducing some smoothness, they do not provide any explicit control over the pointwise map smoothness (see Fig. 3). The discrete solver proposed in [38] has shown that many desirable map properties can be promoted directly on the functional maps, including bijectivity, landmarks preservation or conformality, the latter being unable to effectively promote smoothness as shown in the supplementary material. In this work we therefore seek to extend this framework by introducing pointwise map smoothness constraint that can be efficiently used alongside other objectives.

4.1. Problem Formulation

As discussed in Sec. 3, the Dirichlet energy, seen as a measure of smoothness, is globally minimized by constant maps. To avoid such trivial solutions, we propose to couple a smoothness energy with bijectivity constraints, which can be enforced in the spectral domain using the discrete optimization framework [38].

Specifically, given two shapes \mathcal{S}_1 and \mathcal{S}_2 we consider functional maps C_{ij} and pointwise maps Π_{ij} from *both* directions, where $(i, j) \in \{1, 2\}^2$ indicates the map direction. The discrete solver framework [38] introduces a bijectivity energy which reads:

$$E_{\text{bij}}(\Pi, C) = \sum_{ij} \|\Pi_{ji} \Phi_i C_{ji} - \Phi_j\|_{A_j}^2 + \alpha \|\Phi_j C_{ij} - \Pi_{ji} \Phi_i\|_{A_j}^2, \quad (4)$$

where the first term is derived from a spectral bijectivity energy and the second is a coupling term between functional maps C_{ij} and pointwise maps Π_{ji} (note the change in map directions). Note that variables Π and C contain maps in *both* directions in order to simplify notations. We refer the reader to [38] for a detailed derivation.

In this work, we augment this energy using smoothness constraints, acting on the primal domain instead of the functional (dual) one, which reads:

$$\min_{C, \Pi} E_{\text{bij}}(\Pi, C) + \gamma E_{\text{smooth}}(\Pi) \quad (5)$$

where E_{smooth} penalizes non-smooth pointwise maps, its most basic version being the sum of the Dirichlet energies of the pointwise maps $E_{\text{smooth}}(\Pi) = \sum_{ij} E_D(\Pi_{ij})$ with E_D being defined in Eq.(3). In section 5, we highlight how other common energies for smoothness can be expressed as variations of this Dirichlet energy, thus enabling their straightforward introduction within our formulation.

4.2. Smoothness-promoting Discrete Solver

We aim at solving Eq. (5) using a similar algorithm to the standard discrete solver discussed in Sec. 3. However as long as the energy E_{smooth} includes quadratic terms in Π_{ij} , for instance the Dirichlet energy, this solver cannot be applied as it assumes row-separable variables (see Lemma 4.1 in [38]). Since quadratic terms in the Dirichlet energy appear as W -norms of terms $\Pi_{ij}X_j$, we introduce auxiliary variables Y_{ij} as surrogate for products $\Pi_{ij}X_j$, and add a corresponding coupling term between the two, resulting in a new *coupled* smoothness energy:

$$E_{\text{sm}}^c(\Pi, Y) = E_{\text{smooth}}(\Pi, Y) + \beta \sum_{ij} \|Y_{ij} - \Pi_{ij}X_j\|_{A_i}^2 \quad (6)$$

where the second term is a spatial coupling term and, using some abuse of notations, $E_{\text{smooth}}(\Pi, Y)$ is obtained by replacing products $\Pi_{ij}X_j$ in $E_{\text{smooth}}(\Pi)$ by Y_{ij} . In the particular case where $E_{\text{smooth}} = E_D$, the coupled smoothness energy is now *row-separable* for Π :

$$E_{\text{sm}}^c(\Pi, Y) = \sum_{ij} \|Y_{ij}\|_{W_i}^2 + \beta \|Y_{ij} - \Pi_{ij}X_j\|_{A_i}^2 \quad (7)$$

Note that this particular half-quadratic splitting was used in [16] to handle similar constraints. Furthermore we will show in Sec. 5 that multiple common energy for smoothness can benefit from this similar technique, resulting in a row-separable problem for Π in all cases.

Total energy Eventually, the initial optimization problem, Eq. (5), has been relaxed into a problem of the form $\min_{\Pi, C, Y} E_{\text{ours}}(\Pi, C, Y)$ with

$$E_{\text{ours}}(\Pi, C, Y) = E_{\text{bij}}(\Pi, C) + \gamma E_{\text{sm}}^c(\Pi, Y) \quad (8)$$

Crucially, this reformulation makes the total energy row-separable w.r.t. the pointwise maps Π . We can therefore propose a general iterative method (summarized in Algorithm 1) to minimize the total energy, in the spirit of the discrete solver, which iteratively updates each variable Π, C, Y with the other two sets fixed.

Solver The solver described in Algorithm 1 is divided in three optimization problems for which we present the solution procedure. **(1)** Computing $C^{(k+1)}$ from $\Pi^{(k)}$ reduces to a simple $K \times K$ linear system, which has actually been introduced as *bijective ZoomOut* in [37]. **(2)** Computing $Y^{(k+1)}$ from $\Pi^{(k)}$ also reduces to a sparse linear system whose form depends on the choice of smoothness energy E_{smooth} , some of which are given in section 5. In the case of $E_{\text{smooth}} = E_D$, computing Y_{ij} requires solving $(W_i + \beta I_n)Y_{ij} = \beta \Pi_{ji}X_j$ where the system can be *prefactored* to further improve efficiency. **(3)** Since introducing auxiliary variables leads to a row-separable problem for Π , computing $\Pi^{(k+1)}$ from $C^{(k+1)}$ and $Y^{(k+1)}$ reduces to a simple nearest neighbor search. Note that this step is done in a high-dimensional space obtained by concatenating several terms, and can be heavily accelerated by only using coupling terms from equations Eq. (4) and Eq. (6), which significantly reduces the embedding dimension on which to perform nearest neighbor search. Finally, following [38], we also increase the size K of the functional map as iterations grow, which has shown to be a great regularization procedure in many spectral algorithms.

5. Smoothness Analysis in Unified Framework

In this section, we formulate several existing formulations for promoting map smoothness, including non-rigid ICP (nICP) [3], as-rigid-as-possible (ARAP) [45], reversible harmonic maps (RHM) [16], and Smooth Shells [13]. Our first objective is to provide a coherent formulation of various smoothness terms in the form of the Dirichlet energy on either a map or a deformation. Secondly, we aim to show how different energy terms and solvers can ultimately be introduced in our smoothness-promoting Discrete Solver. This will form the basis for our quantitative evaluation in the next section, in which we compare different terms within our solver. We remain succinct regarding the following derivations and their incorporation in our algorithm, and refer the interested reader to the supplementary material for a more complete overview.

nICP was originally proposed to wrap a source shape \mathcal{S}_1 onto a target shape \mathcal{S}_2 via a per-vertex affine deformation

ALGORITHM 1: Meta-algorithm

Input : Initial maps $\Pi_{12}^{\text{in}}, \Pi_{21}^{\text{in}}$ and vertex positions X_1, X_2

Output: Refined pointwise maps Π_{12}, Π_{21}

Initialization : $\Pi_{ij}^{(0)} = \Pi_{ij}^{\text{in}}, Y_{ij}^{(0)} = \Pi_{ij}^{(0)}X_j$ for $i, j \in \{1, 2\}$

while *Not converged* **do**

$C^{(k+1)} = \arg \min_C E_{\text{bij}}(\Pi^{(k)}, C)$

$Y^{(k+1)} = \arg \min_Y E_{\text{sm}}^c(\Pi^{(k)}, Y)$

$\Pi^{(k+1)} = \arg \min_{\Pi} E_{\text{ours}}(\Pi, C^{(k+1)}, Y^{(k+1)})$

end

field \mathbf{D} . nICP implicitly maintains a pointwise map Π_{12} such that the deformed coordinates $\mathbf{D} \circ X_1$ approximate the pointwise map $\Pi_{12}X_2$. The total energy reads

$$E_{\text{nicp}}(\Pi_{12}, \mathbf{D}) = \|\mathbf{D}\|_{W_1}^2 + \beta \|\mathbf{D} \circ X_1 - \Pi_{12}X_2\|_{A_1}^2 \quad (9)$$

with $\|\mathbf{D}\|_{W_1}^2 = \sum_{i \sim j} w_{ij} \|D_i - D_j\|_F^2$ extends the Dirichlet energy to per-vertex matrices. In our algorithm, this energy may be used as a surrogate for E_{sm}^c given in Eq. (7).

ARAP is a commonly-used energy that aims at promoting *local rigidity* of the shape deformation by enforcing the deformation to remain locally close to a rotation. ARAP optimizes both for expected vertex coordinates Y_{12} and per-vertex rotations \mathbf{R} . The total reformulated energy reads:

$$E_{\text{arap}}(Y_{12}, \mathbf{R}) = \|Y_{12}\|_{W_1}^2 + \lambda E_{\text{arap}}^{\text{rigid}}(Y_{12}, \mathbf{R}), \quad (10)$$

where $E_{\text{arap}}^{\text{rigid}}$ is a bilinear term promoting local rigid deformations. One can augment the energy using the coupling term from Eq. (6) to use the ARAP energy in our algorithm.

Smooth Shells models the deformation \mathbf{D} as a simple per-vertex translation seen as a function $\mathcal{S}_1 \rightarrow \mathbb{R}^3$, restricted to lie in the *spectral* basis of size K , i.e., $\mathbf{D} \in \mathbb{R}^{K \times 3}$. In addition smooth shells uses the ARAP energy to enforce the smoothness of the deformation. Specifically if $Y_{12} = X_1 + \Phi_1 \mathbf{D}$ denotes the updated vertex positions, the shells energy is defined as

$$E_{\text{shells}}(\mathbf{D}, \mathbf{R}, \Pi_{12}) = E_{\text{arap}}(Y_{12}, \mathbf{R}) \quad (11)$$

which is augmented with a coupling term $\|X_1 + \Phi_1 \mathbf{D} - \Pi_{12}X_2\|_{A_1}^2$ to remain close to given correspondences.

RHM directly minimizes the Dirichlet energy of a map without manipulating deformation fields. To avoid making the map collapse the authors look for bijective maps with the lowest possible Dirichlet energy. Specifically using notations of Sec. 4, smoothness is enforced by minimizing the same energy as in Eq. (6) extended with a pointwise bijectivity term $\sum_{ij} \|\Pi_{ij}Y_{ji} - X_i\|_{A_i}$, resulting in a slower solver.

All these smoothness terms can be incorporated quickly within our solver, only affecting steps 2. and 3. of algorithm 1. Furthermore note that, for fairness of comparison, we ignored additional building blocks used in these works like normal preservation, high-dimensional embeddings, etc. More details on these two points can be found in the supplementary material.

6. Experiment

6.1. DEFORMTHINGS4D-MATCHING Dataset

We propose DEFORMTHINGS4D-MATCHING (Fig. 2), a new dataset based on the DEFORMTHINGS4D dataset [25],

a rich synthetic dataset with significant variations in both identities and types of motions, containing 1,972 animation sequences spanning 31 categories of humanoids and animals. However, using DEFORMTHINGS4D to evaluate shape matching is difficult since: (1) most shape models are disconnected; (2) the meshes belong to the same category are in the same triangulation, which provides perfect ground-truth but can lead to over-fitting issues for matching algorithms [39], while cross-category ground truth is missing; (3) some meshes of the synthesized poses have unrealistic distortions such as large self-intersections and unnatural twists. We therefore select 56 animal categories and 8 humanoid categories from DEFORMTHINGS4D, each containing 15-50 poses selected from different motion clips while ensuring large enough pose variations. We then apply LRVD [51] to *independently* remesh all the meshes in the same category. For the humanoid shapes, we further use the commercial software ¹R3DS to non-rigidly fit one shape into another to get *cross-category* correspondences. See Fig. 2 for some examples, where the corresponding vertices are assigned the same color. See supplementary materials for more details of how we construct the dataset and obtain the ground-truth correspondences between the remeshed shapes with different triangulations. The dataset is available at https://github.com/llorz/3DV22_DeformingThings4DMatching_dataset.

6.2. Comparison on Smoothness Formulation

We evaluate our method on the standard benchmark for non-isometric shape matching TOSCA non-Isometric Dataset [7], and the cross-category humanoid shape pairs from our DEFORMTHINGS4D-MATCHING Dataset. Note that on standard benchmarks like the FAUST dataset [5], existing methods already perform well as shapes remain near-isometric. We provide some results in Table 2 to show our method performs similarly in these simple cases, and refer to supplementary material for additional discussions.

Evaluation Metrics We follow [39] to measure the *accuracy*, *bijectivity*, *coverage* and *runtime* to compare different methods. Additionally, We apply Eq. (3) to compute the Dirichlet energy on the obtained pointwise maps to evaluate the *smoothness*. See supplementary materials for detailed definitions and discussions.

Initialization & Baselines Since the tested shape pairs are highly non-isometric and challenging, standard shape descriptors failed to produce reasonable initialization as shown in supplementary. We therefore compute each initial map from a 5×5 functional map obtained by using 5 landmarks. Our baselines can be categorized into three groups: (1) We compare to ZoomOut (ZO) [28] and Discrete Solver (DO) [38], the current-state-of-the-art refinement methods

¹<https://www.russian3dscanner.com/>

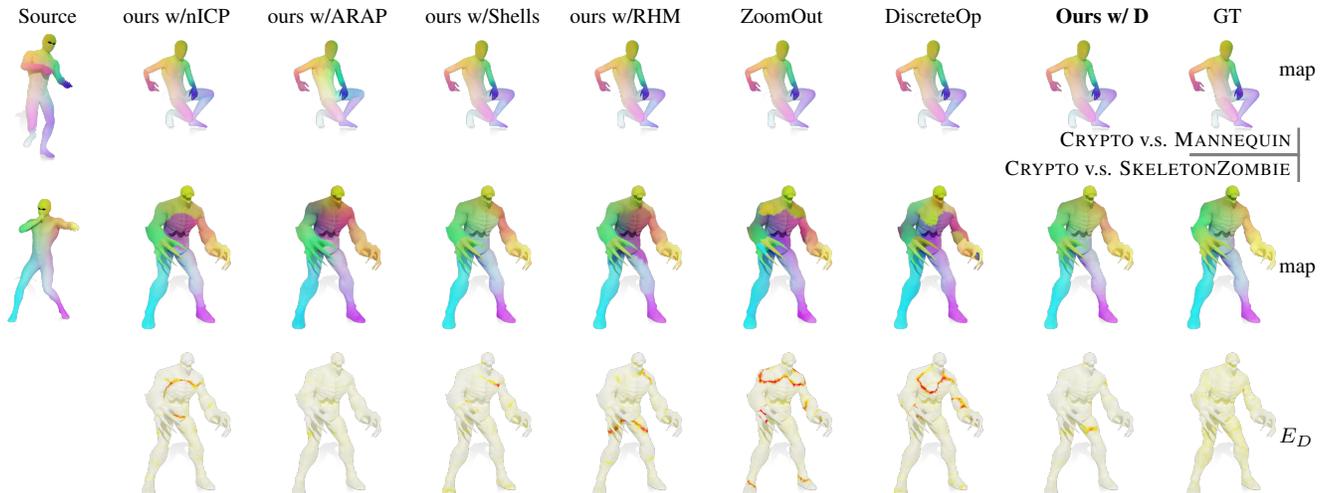


Figure 4. Qualitative evaluation on two pairs from DEFORMTHINGS4D-MATCHING. For a near-isometric shape pair shown on the *top*, all methods achieve smooth maps. For a shape pair that is far from isometry shown on *bottom*, nICP, ARAP, RHM, and Shells achieve relatively smooth maps but contain large patch of back-to-front ambiguity. The maps obtained by ZoomOut and Discrete Solver are locally smooth due to their spectral representation, but fail to maintain global smoothness. As a comparison, our methods can be robustly generalized to non-isometric shape maps and achieve globally smooth maps.

Table 1. DEFORMTHINGS4D-MATCHING Dataset: Summary over 433 shape pairs. We highlight the best two in blue, except those of Shells and RHM (see text for details).

methods	<i>accuracy</i>	<i>bijectivity</i>	<i>smoothness</i>	<i>coverage</i>	<i>runtime (s)</i>
Init	12.71	11.70	3.60	24.57%	-
RHM	11.8	1.6	0.50	56.6%	
Shells	11.4	5.1	1.50	50.8%	
Ours w/ ARAP	12.16	11.70	0.71	31.0%	25.3
Ours w/ nICP	9.56	3.89	1.72	40.4%	100.8
Ours w/ Shells	8.41	2.59	2.18	51.7%	48.2
ZO	8.57	7.14	4.02	67.0%	17.5
DO	9.01	1.78	3.21	62.4%	40.9
Ours w/ D	8.19	2.63	1.56	50.4%	21.4
Ours w/ RHM	8.10	2.18	1.47	56.0%	42.1

in functional maps pipeline. (2) We compare the standard Dirichlet Energy with the different variants presented in Sec. 5, namely nICP [3], ARAP [45], Shells [13] and RHM [16], all using the same algorithm. We highlight the Dirichlet energy (ours w/ D) and the RHM energy (ours w/ RHM) as respectively the simplest and globally best performing energies within our algorithm, which we both advocate. (3) We also include the results using original implementations of RHM and Shells for reference only since additional regularizers besides smoothness are included.

DeformThings4D-Matching Dataset We report the average metrics over 433 cross-category shape pairs from the humanoid shapes from our DEFORMTHINGS4D-MATCHING dataset in Tab. 1. Among all the baseline methods, our method achieves the best accuracy. Compared to ZoomOut (ZO) and Discrete Solver (DO), our two se-

Table 2. Results on a random subset of 200 pairs of the FAUST dataset. We highlight the best two in blue.

methods	<i>accuracy</i>	<i>bijectivity</i>	<i>smoothness</i>	<i>coverage</i>
Init	6.45	5.51	2.67	38.47 %
ZO	3.95	2.16	0.79	82.16 %
DO	4.07	1.08	0.86	77.96 %
Ours w/ D	4.43	1.83	0.64	67.47 %
Ours w/ RHM	3.94	1.11	0.71	79.26 %

lected energies achieve $3\times$ better smoothness on average with comparable bijectivity and coverage. It suggests that, our method, as an extended algorithm of discrete solver by adding a smoothness term, is indeed effective to promote map smoothness. In supplementary, we also report *per-category* map evaluation. We show two qualitative examples in Fig. 4, where the obtained maps are visualized by color transfer. For the pair between CRYPTO and SKELETONZOMBIE, we also visualize the per-vertex smoothness error for each map. We additionally display texture transfer for a difficult pair in Fig. 5, using [15] to obtain a vertex-to-point map for each method to improve visualization. While this figure shows that our maps clearly outperform standard spectral method starting from poor initialization, there is room for improvement for all energies.

TOSCA Non-Isometric Dataset contains cross-category correspondences among one gorilla shape (with 5 different poses), one male shape (with 7 different poses), and one female shape (with 12 different poses). We use all 95 non-isometric shape pairs between the gorilla shapes and the human (male and female) shapes. The summary evaluation is shown in Tab. 3. See supplementary for qualitative exam-

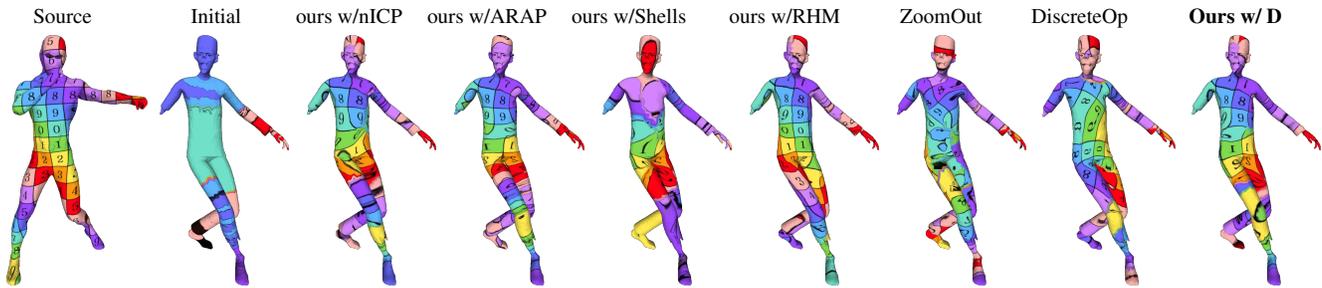


Figure 5. Starting from a poor initial map, our method can produce a more smooth and accurate map compared to the baseline methods.

Table 3. TOSCA Non-Isometric Dataset: Summary over 95 shape pairs. We highlight the best two in blue, except those of Shells and RHM (see text for details).

methods	accuracy	bijectivity	smoothness	coverage	runtime (s)
Init	7.51	7.23	1.94	26.9%	-
RHM	9.20	1.37	1.55	54.3 %	818
Shells	10.20	6.72	5.58	45.6 %	29.0
ours w/ ARAP	7.55	8.35	0.83	48.6%	42.8
Ours w/ nICP	7.78	3.63	1.16	40.2%	178
Ours w/ Shells	11.85	7.40	1.18	37.8%	72.5
ZO	12.47	8.17	6.53	56.8%	33.7
DO	13.30	1.90	5.51	53.4%	79.2
Ours w/ D	7.25	3.02	1.22	42.2%	33.3
Ours w/ RHM	6.26	1.87	1.39	53.1%	40.1

ples. Enforcing the smoothness of the pointwise maps via Dirichlet energy (Ours + D) help us achieve much more accurate and $5\times$ smoother maps. We additionally highlight that adding extra pointwise bijectivity (ours w/ RHM) has a positive effect on the metrics, but results in a slower solver. Finally, while ARAP and nICP energies perform quite well, the Shells energy seems to suffer from the high level of non-isometry in the dataset as it mainly relies on spectral quantities.

6.3. Implementation & Parameters

We implemented all the baselines (based on their released code) and our methods in Python to guarantee a fair comparison. We follow the discrete solver [38] to adopt the progressive upsampling technique into our algorithm, which is introduced in [28], and gradually increase the spatial coupling term weight γ to avoid over-smoothing in the earlier iterations. Detailed parameters can be found in supplementary, or in the released version of the code at <https://github.com/RobinMagnet/smoothFM>.

7. Conclusion, Limitations & Future Work

In this work, we extended the discrete solver formulation from [38] to optimize the Dirichlet energy to promote map smoothness. We then proposed an efficient algorithm that can produce high-quality and smooth maps from noisy initial maps for between non-isometric surfaces. Further-

more, we demonstrated that multiple previously proposed methods for computing smooth maps, including nICP [3], ARAP [45], RHM [16], and Smooth Shells [13], can be reformulated in a coherent framework. This allowed us to compare and analyze different formulations for smoothness using a single algorithm. Finally, to address the scarcity of evaluation data, we proposed a new dataset based on DEFORMTHINGS4D, with variable mesh structure, and dense ground truth cross-category correspondences for eight challenging categories. We believe both our framework and this dataset can be helpful for the shape matching community.

Our method still has some limitations. First, optimizing the Dirichlet energy can indeed greatly improve the smoothness compared spectral methods. This, however, can come at the expense of loss of coverage, and we observe that our maps can still collapse locally, as seen from the texture transfer of Figure 5. It would be interesting to investigate techniques that to further prevent *local collapse* and obtain a smooth maps with high coverage. Second, our results show that the proposed method improves significantly results from ZoomOut and discrete solver on complete shapes, even for non-isometric cases. However, for the partial matching setting, though our maps still outperform ZoomOut and discrete solver, there is still a lot room for further improvement. Finally, our energy is a weighted sum of a bijectivity and a smoothness term, which can become hard to balance across all initialization quality.

In the future, we would like to study different energies for partial matching and to ways prevent local map collapse. It will also be interesting to apply our approach for computing dense correspondences in other domains, such as point clouds, graphs or even 2D images.

Acknowledgments The authors thank the anonymous reviewers for their valuable comments and suggestions. Parts of this work were supported by the ERC Starting Grant No. 758800 (EXPROTEA), the ERC Consolidator Grant No. 101003104 (MYCLOTH), and the ANR AI Chair AIGRETTE.

References

- [1] Noam Aigerman and Yaron Lipman. Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics (TOG)*,

- 35(6):217:1–217:14, Nov. 2016. 3
- [2] Noam Aigerman, Roi Poranne, and Yaron Lipman. Seamless surface mappings. *ACM Transactions on Graphics (TOG)*, 34(4), July 2015. 3
- [3] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 2, 3, 5, 7, 8
- [4] Silvia Biasotti, Andrea Cerri, Alex Bronstein, and Michael Bronstein. Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum*, 35(6):87–119, 2016. 2
- [5] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, Columbus, Ohio, 2014. IEEE. 6
- [6] Alex Bronstein, Michael Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006. 3
- [7] Alex Bronstein, Michael Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY, 2008. 6
- [8] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008. 4
- [9] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. *arXiv preprint arXiv:2203.07858*, 2022. 1, 2
- [10] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8592–8601, 2020. 2
- [11] Nadav Dym, Haggai Maron, and Yaron Lipman. Ds++: a flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics (TOG)*, 36(6):184, 2017. 3
- [12] Marvin Eisenberger, Zorah Löhner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38, pages 1–12. Wiley Online Library, 2019. 1
- [13] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12265–12274, 2020. 1, 2, 3, 5, 7, 8
- [14] Davide Eynard, Emanuele Rodola, Klaus Glashoff, and Michael M Bronstein. Coupled functional maps. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 399–407. IEEE, 2016. 2
- [15] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. *Computer Graphics Forum*, 36(5):165–174, 2017. 3, 7
- [16] Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. Reversible harmonic maps between discrete surfaces. *ACM Transactions on Graphics (TOG)*, 38(2):15:1–15:12, 2019. 1, 2, 3, 4, 5, 7, 8
- [17] Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d’Aspremont. Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems*, pages 1016–1024, 2013. 3
- [18] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. 4
- [19] Anne Gehre, Michael Bronstein, Leif Kobbelt, and Justin Solomon. Interactive curve constrained functional maps. *Computer Graphics Forum*, 37(5):1–12, 2018. 3
- [20] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27(5):1449–1457, 2008. 3
- [21] Ruqi Huang, Panos Achlioptas, Leonidas Guibas, and Maks Ovsjanikov. Limit shapes—a tool for understanding shape differences and variability in 3d model collections. In *Computer Graphics Forum*, volume 38, pages 187–202. Wiley Online Library, 2019. 2
- [22] Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. *Computer Graphics Forum*, 36(5):151–163, 2017. 3
- [23] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics (TOG)*, 30(4):79, 2011. 3
- [24] Artiom Kovnatsky, Michael Bronstein, Alex Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32(2pt4):439–448, 2013. 2
- [25] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021. 2, 6
- [26] Yaron Lipman and Thomas Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)*, 28(3):72:1–72:12, July 2009. 3
- [27] Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics (TOG)*, 36:14, 2017. 3
- [28] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155:1–155:14, Nov. 2019. 1, 3, 6, 8
- [29] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, New York, NY, 2003. 3
- [30] Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael Bronstein, and Maks Ovsjanikov. Improved functional mappings via product preservation. *Computer Graphics Forum*, 37(2):179–190, 2018. 3
- [31] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017. 2, 3

- [32] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012. 2, 3
- [33] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, pages 5:1–5:62, 2017. 2, 3
- [34] Maks Ovsjanikov, Quentin Merigot, Facundo Memoli, and Leonidas Guibas. One point isometric matching with the heat kernel. *CGF*, 29(5):1555–1564, 2010. 3
- [35] Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Computer Vision and Pattern Recognition (CVPR)*, pages 384–393, 2021. 3
- [36] Ulrich Pinkall and Konrad Polthier. Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics*, 2(1):15–36, 1993. 4
- [37] Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. Maptree: Recovering multiple solutions in the space of maps. *ACM Transactions on Graphics (TOG)*, 39(6), Nov. 2020. 3, 5
- [38] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching. In *Computer Graphics Forum*, volume 40, pages 81–96. Wiley Online Library, 2021. 2, 3, 4, 5, 6, 8
- [39] Jing Ren, Adrien Poulenc, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. 2, 3, 6
- [40] Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017. 2
- [41] Emanuele Rodolà, Michael Moeller, and Daniel Cremers. Point-wise map recovery and refinement from functional correspondence. In *Vision, Modeling, and Visualization*, 2015. 3
- [42] Raif M Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)*, 32(4):72, 2013. 2
- [43] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020. 1, 2
- [44] Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):72, 2016. 3
- [45] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Computer Graphics Forum*, volume 4, pages 109–116. Wiley Online Library, 2007. 2, 3, 5, 7, 8
- [46] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 1
- [47] Matthias Vestner, Zorah Löhner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alex Bronstein, Michael Bronstein, and Ron Kimmel. Efficient deformable shape correspondence via kernel matching. In *International Conference on 3D Vision (3DV)*, pages 517–526. IEEE, 2017. 3
- [48] Matthias Vestner, Roei Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6681–6690, 2017. 3
- [49] Larry Wang, Anne Gehre, Michael Bronstein, and Justin Solomon. Kernel functional maps. *Computer Graphics Forum*, 37(5):27–36, 2018. 3
- [50] Y Wang, B Liu, K Zhou, and Y Tong. Vector field map representation for near conformal surface correspondence. *Computer Graphics Forum*, 37(6):72–83, 2018. 3
- [51] Dong-Ming Yan, Guanbo Bao, Xiaopeng Zhang, and Peter Wonka. Low-resolution remeshing using the localized restricted voronoi diagram. *IEEE Transactions on Visualization and Computer Graphics*, 20(10):1418–1427, 2014. 6