DOI: 10.1111/cgf.14359
Eurographics Symposium on Geometry Processing 2021
K. Crane and J. Digne
(Guest Editors)

*Volume 40 (2021), Number 5*

# Discrete Optimization for Shape Matching

Jing Ren[1], Simone Melzi[2], Peter Wonka[1], Maks Ovsjanikov[3]

[1]KAUST [2] Sapienza University of Rome [3]LIX, École Polytechnique, CNRS

**Abstract**

*We propose a novel discrete solver for optimizing functional map-based energies, including descriptor preservation and promoting structural properties such as area-preservation, bijectivity and Laplacian commutativity among others. Unlike the commonly-used continuous optimization methods, our approach enforces the functional map to be associated with a pointwise correspondence as a hard constraint, which provides a stronger link between optimized properties of functional and point-to-point maps. Under this hard constraint, our solver obtains functional maps with lower energy values compared to the standard continuous strategies. Perhaps more importantly, the recovered pointwise maps from our discrete solver preserve the optimized for functional properties and are thus of higher overall quality. We demonstrate the advantages of our discrete solver on a range of energies and shape categories, compared to existing techniques for promoting pointwise maps within the functional map framework. Finally, with this solver in hand, we introduce a novel Effective Functional Map Refinement (EFMR) method which achieves the state-of-the-art accuracy on the SHREC'19 benchmark.*

**CCS Concepts**

• ***Computing methodologies*** → ***Shape analysis;*** • ***Theory of computation*** → *Computational geometry;*

## 1. Introduction

Non-rigid shape matching is a classical problem in geometry processing and related fields. Given a pair of 3D shapes, the objective is to find a mapping that associates points on the two shapes according to some quality criteria. Once such a high-quality mapping is computed, it can then be used in a wide variety of downstream tasks, ranging from statistical shape analysis [BRLB14] to deformation transfer [SP04].

This problem is challenging because the space of possible maps is typically non-linear and most common objectives result in complex non-convex optimization problems. For example, it is well-known that optimizing the standard geodesic distortion energy over the space of permutation matrices leads to NP-hard quadratic assignment problems [LdABN*07].

To address this challenge, most correspondence approaches use relaxation techniques replacing discrete constraints on the permutation matrices with spaces more amenable to optimization. This includes optimizing over the space of doubly stochastic matrices [FJBd13, SPKS16, DML17], using spectral relaxation, which constrains the Frobenius norm of the matrices [LH05], or using the functional map relaxation, which uses a low rank spectral representation [OBCS*12]. These methods lead to continuous (often convex) optimization problems that can be solved efficiently, but then require potentially costly or error-prone projection steps to recover an integer solution. Finally, state-of-the-art generic provably tight relaxation techniques [DML17, BTM18] are only applicable to shapes or graphs with hundreds of points making them impractical for dense meshes.

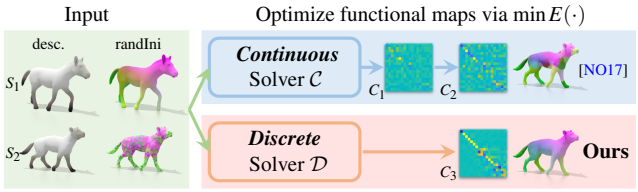


**Figure 1:** ***Overview of our approach.*** *The standard functional map framework uses a continuous solver to optimize for a functional map by minimizing a given energy $E(\cdot)$ and projecting to a pointwise map during post-processing. In contrast, our discrete solver directly optimizes in the space of "proper" functional maps and thus leads to better final pointwise correspondences.*

In this paper, we focus on the functional map relaxation for shape matching. The standard functional map pipeline first defines an energy $E(\cdot)$, typically based on preservation of descriptors coupled with promoting structural map properties. An optimal functional map is then computed by optimizing this energy with a *continuous* solver usually in an *unconstrained* setting. Finally, in most practical scenarios this functional map is then projected to recover a pointwise map. Note that the projection step is usually independent of the defined energy $E$, and can thus destroy the optimized map properties, see Fig. 1 (top right, blue box) for an illustration. In most cases, the recovered

pointwise maps are then further refined using a post-processing method [OBCS*12, VLR*17, RPWO18, MRR*19, ESBC19].

While simple and efficient, the functional map pipeline has two major issues: (1) there is no guarantee that the optimized functional map is associated with *any* point-to-point correspondence (2) although prior works have introduced powerful functional map objectives, the pointwise map conversion and refinement steps discard those carefully designed energies and adopt other heuristics/criterion to promote the pointwise maps.

In this paper, we show how to resolve these conflicts by exploiting the link between functional and pointwise maps. First, we introduce the concept of the *proper* functional maps. A functional map is called *proper* if it arises from some underlying pointwise map. We then propose to minimize functional map-based energies in the space of *proper* functional maps. This problem is much harder to solve than the unconstrained one since the search space becomes non-convex and discrete. To address this, we propose a general *discrete* solver which is simple and can optimize a range of energies with the proper functional map constraint. For example, in Fig. 1 we use our discrete solver to optimize the energy proposed in [NO17] and obtain a *proper* functional map $C_3$ which leads to a more accurate underlying pointwise map, despite the energy in [NO17] being designed to explicitly promote point-to-point correspondences.

Our main contributions include the problem formulation with the proper functional map constraint and a general discrete solver which can be used to optimize a large set of functional map based energies under that constraint. We compare our approach to existing techniques for promoting pointwise maps, e.g., [NO17], and show that our method is both more flexible and leads to higher accuracy. We also demonstrate that the recently-proposed standard and bijective ZoomOut method [MRR*19, RMOW20] are special cases of applying our discrete solver to different energies. Finally, we apply our discrete solver to a new functional map energy that combines a range of different objectives and achieves state-of-the-art accuracy on the SHREC'19 benchmark. To summarize, our main contributions include:

1. Introducing the proper functional map hard constraint to optimization that involves functional map based energies, which can lead to better and more desirable local minima.
2. A general discrete optimizer capable of optimizing a large class of functional map energies.

## 2. Related Work

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

**Shape Matching** Our framework is closely related to the problem of shape matching, and thus to methods that look for dense correspondences between non-rigid 3D shapes. For an in-depth review of this area we refer the readers to [BCBB16, TCL*13]. Several approaches to shape matching directly solve for correspondences between points on the two surfaces by minimizing an explicit energy, e.g., [BBK06, HAWG08, OMMG10]. The main limitation of these methods is that they often lead to complex combinatorial problems. An alternative is to first map the shapes to a canonical domain (e.g. a sphere), and then solve for the correspondence between these parametric representations [LF09, APL15, AL16], or blend across multiple such maps [KLF11].

One successful strategy for shape matching is to relax the search space from permutation matrices to a space more amenable to continuous optimization, e.g., doubly stochastic matrices [FJBd13, SPKS16, DML17], but also other successful relaxations exist [LH05, SNB*12, KKBL15, MDK*16, ADG19]. This, however, requires a possibly error-prone projection step to compute an integer solution, while provably tight relaxations do not scale well to complex meshes.

**Shape Matching with Functional Maps** Our approach is based on the functional map representation [OBCS*12]. The vast majority of methods that use this framework for shape matching start with a set of descriptor functions, derived from point signatures or from landmarks, and use them jointly with global map quality criteria to compute a correspondence [KBB*13, AK13, NO17] (we refer to [OCB*17] for an overview). While computing *a functional* map reduces to solving a least-squares system, the conversion from a functional map to a point-wise map is not trivial and can lead to inaccuracy and noise [RMC15, EBC17]. To improve accuracy, several desirable map attributes have been promoted via regularizers for the functional map estimation first using geometric insights [ERGB16, RCB*17, NO17, LRBB17, BDK17, WLZT18, RPWO18, WGBS18, GBKS18, NMR*18, SVBC19], and more recently using learning-based techniques [LRR*17, HLR*19, RSO19]. Nevertheless, despite significant progress, the reliance on descriptors and decoupling of continuous optimization and pointwise map conversion remains common to all existing methods.

**Map Refinement** A common strategy for improving estimated correspondences consists in iterative map refinement as a post-processing step, e.g. [SPKS16, MCSK*17, VLR*17, VLB*17]. The simplest refinement in the functional maps framework is the Iterative Closest Point algorithm in the spectral domain [OBCS*12]. Recently, other more advanced refinement methods for both functional and pointwise maps have been proposed in [ESBC19, RPWO18], that, respectively, try to minimize the bi-directional geodesic Dirichlet energy, and promote the bijectivity, smoothness and coverage of the correspondences. When shape *collections* are considered, a common strategy is to use cycle consistency constraints [WHG13, HWG14, WS13]. Most closely related to ours, is the ZoomOut method proposed in [MRR*19], and based on iterative conversion between functional and pointwise maps. This approach was recently extended and incorporated in a strategy to analyze the space of maps between shapes in [RMOW20]. Without any initialization, this method explores the space of maps between shapes exploiting their functional representations and providing as output a set of maps encoding all the different symmetries between two objects (or eventually from an object and itself). We significantly enrich this approach, generalizing the possible energies minimized during the process.

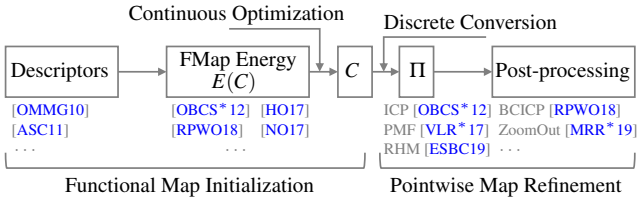


**Figure 2:** ***Standard functional map pipeline for shape matching.*** *Given some input descriptors, a functional map energy E is formulated, and the optimal functional map C is optimized for in the* continuous *setting. This functional map is then converted to a pointwise map* Π*. A post-processing step is applied to further improve the quality of the pointwise map.*

## 3. Notation, Background & Motivation

**Notation** Given a pair of shapes $S_1$ and $S_2$ represented as triangle meshes with respectively $n_1$ and $n_2$ vertices, we compute the cotangent Laplace-Beltrami operator [MDSB03] of each shape $S_i$ and collect the first $k$ eigenfunctions as columns in a matrix denoted by $\Phi_i = \left[\varphi_1^{S_i}, \varphi_2^{S_i}, \cdots, \varphi_k^{S_i}\right]$ and the corresponding eigenvalues in a diagonal matrix denoted as $\Delta_i$. The eigenfunctions are orthonormal with respect to the mass (area) matrices of the shapes: $\Phi_i^T A_i \Phi_i = Id$. A pointwise map $T_{ij} : S_i \to S_j$, associates each vertex on shape $S_i$ to a vertex on shape $S_j$ (e.g. the $p$-th vertex on $S_i$ corresponds to the $T_{ij}(p)$-th vertex on $S_j$). The map $T_{ij}$ can be represented as a binary matrix $\Pi_{ij}$, such that $\Pi_{ij}(p,q) = 1$ if $T_{ij}(p) = q$ and 0 otherwise. The subscript $(i, j)$ indicates the map direction. Throughout our discussion below we will always use $\Pi$ to denote point-to-point maps.

Our work is based on the functional map representation, introduced in [OBCS*12], that encodes a correspondence as a linear transformation in a functional basis. Given a point-to-point correspondence $\Pi_{21}$, its associated functional map is a $k_2 \times k_1$ matrix, given as: $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$. Note that $C_{12}$ maps functions in a reduced basis from $S_1$ to $S_2$.

In the standard functional map pipeline [OCB*17], *any* matrix with the appropriate size w.r.t. the reduced basis, is considered a valid functional map. This greatly simplifies many optimization problems by enabling the use of unconstrained continuous optimization techniques. In practice, however, additional regularizers are introduced to ensure that the recovered map is associated with a pointwise map [NO17]. Furthermore, in the standard pipeline, pointwise map recovery is performed *a posteriori*, potentially introducing errors and inconsistencies.

**Proper Functional Maps** In our paper, we consider a subset of the space of functional maps, restricted to those that encode *pointwise* correspondences:

**Definition 3.1** The *proper functional map space* $\mathcal{P}$ is the set of functional maps that arise from pointwise correspondences. Particularly, we call a functional map $C_{12}$ *proper* if there exists a pointwise map $\Pi_{21}$ such that $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$.

In other words, $C_{12}$ is called *proper* if and only if there exists a binary matrix $\Pi_{21}$ with exactly one value 1 per row, so that $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$. The space $\mathcal{P}_{12}$ of proper functional maps between $S_1$ and $S_2$ is denoted as:

$$\mathcal{P}_{12} = \left\{ C_{12} \mid \exists \Pi_{21}, \text{ s.t. } C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1 \right\} \tag{1}$$

Unlike the commonly used unconstrained functional map space $\mathbf{R}^{k_2 \times k_1}$, we can observe that the *proper* functional map space is *discrete* and has a *finite* size with at most ${n_1}^{n_2}$ elements.

**Functional Map Computation Pipeline** Fig. 2 illustrates the standard functional map pipeline, which consists of the following two major steps (see Chapter 2 in [OCB*17]):

1. Formulate an energy $E(\cdot)$ based on preservation of geometric properties, and optimize for a functional map in an unconstrained setting. E.g.,:

$$\min_{C_{12} \in \mathbf{R}^{k_2 \times k_1}} \left\| C_{12} F_1 - F_2 \right\|_F^2 + \gamma \left\| C_{12} \Delta_1 - \Delta_2 C_{12} \right\|_F^2. \tag{2}$$

   Here, the first term corresponds to preservation of descriptors, while the second promotes commutativity with the Laplacian, which is based on the common near-isometry assumption.
2. Convert the functional map computed in Step 1. to a pointwise map, potentially applying post-processing and refinement [EBC17, VLR*17, MRR*19].

The main advantage of this pipeline is that the unconstrained optimization problem in Step 1 has only $k_1 k_2$ unknowns, independently of the size of the underlying meshes, and it can be solved with a standard least squares solver. Unfortunately, this comes at a price, since there is no guarantee that the recovered functional map will be proper, which can induce significant errors during the pointwise conversion step. Previous works have aimed to remedy this problem by introducing regularizers promoting pointwise maps [NO17], considering more sophisticated functional map objectives, e.g., [HO17, PSO18, WGBS18, GBKS18], or using *learned* descriptors that lead to more accurate maps [LRR*17, HLR*19, DSO20] among many others. Unfortunately, despite significant progress, the underlying problem of decoupling functional map optimization and pointwise map conversion is present in all existing approaches. We note that this problem is exacerbated further in post-processing methods [EBC17, VLR*17, MRR*19], as the carefully crafted geometric energies used in Step 1. are typically abandoned completely and each post-processing method uses its own set of objectives or heuristics.

## 4. Method Description

Rather than solving an unconstrained optimization problem as is done in virtually all existing functional maps based methods, we propose to constrain the optimization to the space of *proper* functional maps introduced above.

**Problem Formulation** Our general approach consists in optimizing a given energy $E(C_{12})$ over the space of proper functional maps:

$$\min_{C_{12} \in \mathcal{P}_{12}} E(C_{12}), \tag{3}$$

instead of optimizing $E(C_{12})$ in an unconstrained setting. In full generality, this is a very difficult problem, and can potentially

even include quadratic assignment as a special case. Our main observation, however, is that certain energies, including the one in Eq. (2), admit a particular decomposition, in which the functional and point-to-point maps can be solved for in an alternating fashion, where each sub-problem has a simple or even closed form solution.

**Approach overview** Our overall strategy mimics the Augmented Lagrangian methods with variable splitting [GM76, BPC*11], which are commonly used to solve constrained problems. Specifically, our approach consists of the following general steps:

1. Given a functional map energy, reformulate it by replacing some terms $C_{12}$ with $\Phi_2^\dagger \Pi_{21} \Phi_1$.
2. Add a *coupling term* to the energy and make the functional map $C_{12}$ and pointwise map $\Pi_{21}$ independent free variables of the resulting problem.
3. Alternate between computing the optimal functional and point-to-point maps, while fixing the other representation.

Our main observation is that for many energies, the resulting problem in Step 3. for each representation can be solved in closed form, as long as the other representation is fixed. Crucially, unlike the standard pipeline, our pointwise map recovery in Step 3. at every iteration is *informed* by the original functional map objective.

To make this pipeline possible, the key step is to relax the original energy in a way that would allow efficient point-to-point map recovery in Step 3. For this, we will make repeated use of the following lemma:

**Lemma 4.1** Given arbitrary matrices $X, Y$, and a reduced basis $\Phi$, s.t. $\Phi^T A \Phi = Id$, then the following two problems: *(i)* $\min_\Pi \left\| \Phi^\dagger \Pi X - Y \right\|_F^2 + \left\| (Id - \Phi\Phi^\dagger)\Pi X \right\|_A^2$; *(ii)* $\min_\Pi \left\| \Pi X - \Phi Y \right\|_F^2$; are equivalent. Moreover problem ii) is row-separable and can be solved in closed form through nearest neighbor search.

The second term in problem *(i)* of this lemma is a regularizer that penalizes the image of $X$ that lies outside of the span of $\Phi$. A special case this of lemma appeared in [EBC17], and we include a proof in Appendix A for completeness. In our method, we exploit Lemma 4.1 extensively for various pairs $X, Y$, to transform any optimization problem of the form $\min_\Pi \left\| \Phi^\dagger \Pi X - Y \right\|_F^2$ to the form $\min_\Pi \left\| \Pi X - \Phi Y \right\|_F^2$ by implicitly adding the appropriate regularizer.

**Example** As an example, consider the energy in Eq. (2). First remark that $C_{12} = 0$ is a global minimizer for the Laplacian commutativity term in the unconstrained setting, while it is well-known that in the space of *proper functional maps*, in the full basis $E_{\text{lap}} = 0$, only for discrete isometries [ZGLG12, MRR*19].

**Relaxation** In our relaxation, we start by replacing some terms $C_{12}$ in the original energy using the hard constraint $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$. This leads to the following modified energy for Eq. (2):

$$E^{\text{mod}} = \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 F_1 - F_2 \right\|_F^2 + \gamma \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 \Delta_1 - \Delta_2 C_{12} \right\|_F^2, \quad (4)$$

Note that $E^{\text{mod}}(C_{12}, \Pi_{21})$ is equivalent to the problem in Eq. (2) under the hard constraint $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$. However, we can now relax $E^{\text{mod}}$ by making it a problem in two *independent variables* $C_{12}, \Pi_{21}$ and solving for them in an alternating fashion. Lemma 4.1 then allows us to optimize for $\Pi_{21}$ through the implicit use of a regularizer. This leads to the following iterative algorithm:

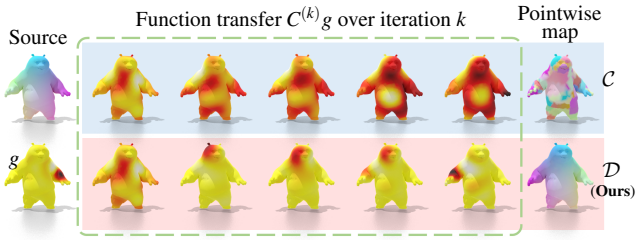


**Figure 3:** *Minimizing the orientation-reversing energy on a panda shape from random initialization. We visualize how the function $g$ gets transferred by the optimized functional map $C^{(k)}$ from the continuous solver $\mathcal{C}$ and our discrete solver $\mathcal{D}$ over iterations. We also visualize the computed pointwise map in the last column, where our discrete solver outputs a high-quality self-symmetric map.*

1. $C_{12} = \arg\min_{C_{12}} E^{\text{mod}}(C_{12} \,|\, \Pi_{21})$
2. $\Pi_{21} = \arg\min_{\Pi_{21}} \left\| \Pi_{21} \Phi_1 F_1 - \Phi_2 F_2 \right\|_F^2 + \gamma \left\| \Pi_{21} \Phi_1 \Delta_1 - \Phi_2 \Delta_2 C_{12} \right\|_F^2$.

Note that the problem in step 2 above results from applying Lemma 4.1 to the energy $E^{\text{mod}}$. Crucially, this new problem is row separable and the optimal $\Pi_{21}$ can be computed via nearest neighbor search efficiently.

**Different Coupling Strategies** In the relaxation above we proceeded in two stages: first, rewriting the original energy in modified form, that enabled the use of Lemma 4.1, and second, by removing the hard constraint and optimizing for $C_{12}$ and $\Pi_{21}$ in an alternating fashion. While this may be the simplest option, it typically does not work well in practice and a *coupling* term is necessary to link the two sets of variables and improve the overall energy optimization.

Common coupling strategies for relaxing constrained optimization problems include introducing a soft penalty or using an augmented Lagrangian approach, such as ADMM. Due to the special continuous-discrete nature of our setting that uses spectral and primal domains, the best choice of coupling is problem-dependent. See Appendix B for more detailed discussions of various coupling strategies in our setting. We have observed that for most energies that we tested, the following coupling term leads to the best results: $\left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{12}^T - Id \right\|_F^2$. Below we demonstrate that the resulting strategy is efficient, flexible that can handle different energies, recovering good pointwise maps even from random initializations.

**Applicability** Our method is based on applying Lemma 4.1 to rewrite functional maps-based energy in a way that leads to closed-form expressions for pointwise map recovery. Our overall strategy is therefore applicable to energies that can be written in a specific format (see Appendix C for full details on different energies). Interestingly, as we remark below, this covers a wide range of commonly-used functional map energies, and even allows effective approximation of energies that do not fit in that format. Below provide expressions for reformulating several commonly-used functional map-based energies

1. **Descriptor preserving energy** A standard energy in functional map computations, introduced in [OBCS*12] is based on preservation of descriptors, which can be written as: $E(C_{12}) = \left\| C_{12} F_1 - F_2 \right\|_F^2$. As mentioned above, we can write this

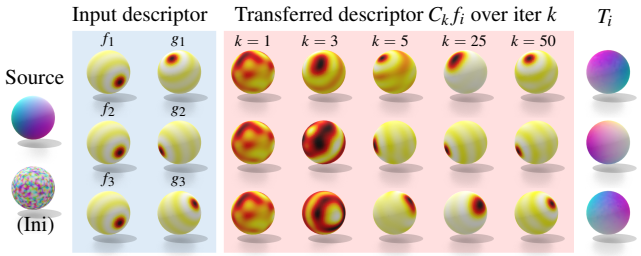


**Figure 4:** *For a sphere shape, we use our discrete solver to minimize the descriptor-preserving energy defined on $(f_i, g_i)$, a pair of descriptors we would like to enforce the correspondence. Here we show three examples of the corresponding descriptors (blue box) and visualize how the descriptor $f_i$ is transported over iterations (red box). We also show the corresponding output pointwise maps $T_i$ in the last column.*

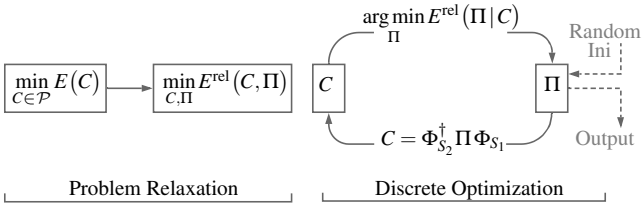


**Figure 5:** *Overview of our problem relaxation and discrete optimization through iterative conversion ($E^{rel}$ is short for $E^{relaxed}$).*

energy $E^{\text{mod}} = \|\Phi_2^{\dagger}\Pi_{21}\Phi_1 F_1 - F_2\|_F^2$. Lemma 4.1 can then be applied directly to this modified energy.

2. **Operator commutativity energy** Another commonly used energy is based on penalizing lack of commutativity of the sought functional map with some other operators $E(C_{12}) = \|C_{12}\Omega_1 - \Omega_2 C_{12}\|_F^2$. Here, $\Omega_i$ can be the Laplacian operator $\Delta_i$ which promotes isometric correspondences [OBCS*12], or multiplicative operators introduced in [NO17] promoting preservation of function products and thus pointwise maps in a soft way. $\Omega_i$ can also represent orientation-preserving/reversing operators introduced in [RPWO18]. See Fig. 3 for an example. In all of these cases, the corresponding modified energy is: $E^{\text{mod}} = \|\Phi_2\Pi_{21}\Phi_1\Omega_1 - \Omega_2 C_{12}\|_F^2$. Again, the modified energy can be optimized for both $C_{12}$ (via least squares) if $\Pi_{21}$ is fixed. Conversely, by using Lemma 4.1, $\Pi_{21}$ can be computed via nearest neighbor search if $C_{12}$ is fixed.
3. **Orthogonality energy** It is well-known that locally area-preserving correspondences lead to *orthonormal* functional maps [OBCS*12, ROA*13]. This prior has been used as a hard constraint by optimizing on the Stiefel manifold of orthonormal matrices [KGB16, LRBB17]. The standard functional map energy has the following form: $E(C_{12}) = \|C_{12}^T C_{12} - Id\|_F^2 = \|C_{12}C_{12}^T - Id\|_F^2$. We re-write it simply as$E^{\text{mod}} = \|\Phi_2^{\dagger}\Pi_{21}\Phi_1 C_{12}^T - Id\|_F^2$. This form allows efficient optimization for both $C_{12}$ and $\Pi_{21}$ whenever the other set of variables is fixed.
4. **Conformal energy** A point-to-point map is conformal if and only if the functional map preserves $H^1$ inner products of functions. As was shown in [ROA*13] this is equivalent to the following condition: $E(C_{12}) = \|C_{12}^T\Delta_2 C_{12} - \Delta_1\|_F^2$. Note that this energy does not follow the format that we used above. We therefore *approximate it* by a different surrogate energy: $E(C_{12}) = \|C_{12}\Delta_1 C_{12}^T - \Delta_2\|_F^2$. Though not equivalent to the original energy, it is strongly related to the original energy (see detailed discussion in Sec. 5.3). This latter energy can be re-written as: $E^{\text{mod}} = \|\Phi_2^{\dagger}\Pi_{21}\Phi_1\Delta_1 C_{12}^T - \Delta_2\|_F^2$, again, enabling the use of Lemma 4.1 for optimization of pointwise maps $\Pi$.
5. **Bijectivity energy** Instead of optimizing the functional map only in one direction it can also be convenient to couple maps in both directions, thereby promoting invertibility (bijectivity) of the recovered correspondences. This energy, originally introduced in [ERGB16] can be stated as: $E(C_{12}, C_{21}) = \|C_{12}C_{21} - Id\|_F^2 + \|C_{21}C_{12} - Id\|_F^2$. Note that this energy involves the functional maps from both directions. We can there introduce the corresponding pointwise maps $\Pi_{12}$ and $\Pi_{21}$ and rewrite this energy: $E^{\text{mod}} = \|\Phi_2^{\dagger}\Pi_{21}\Phi_1 C_{21} - Id\|_F^2 + \|\Phi_1^{\dagger}\Pi_{12}\Phi_2 C_{12} - Id\|_F^2$. Note that in this particular case, we have four sets of independent variables $C_{12}, C_{21}, \Pi_{12}, \Pi_{21}$ to optimize.

## 5. Practical Algorithm of Discrete Optimization

As mentioned above, our approach starts with a given functional map-based energy. We then re-write it in a modified form as described above, make the functional and pointwise maps independent variables that can be optimized in an alternating fashion.

In practice, we make several modifications that we have observed improve the speed and robustness of the resulting algorithm (1) we add a coupling term $\|\Phi_2^{\dagger}\Pi_{21}\Phi_1 C_{12}^T - Id\|_F^2$ that links the pointwise and functional maps and, as regularizes the optimization especially in difficult settings; (2) instead of optimizing the functional map from the relaxed energy, we simply compute it from the optimized pointwise map by setting $C_{12} = \Phi_2^{\dagger}\Pi_{21}\Phi_1$ after optimizing for $\Pi_{21}$, (3) we adopt the progressive upsampling technique proposed in [MRR*19] in our solver. In the following, we introduce our discrete solver with full details.

### 5.1. Discrete Solver

Compared to its unconstrained setting (Eq. (2)), Eq. (3) is hard to solve since the energy $E(\cdot)$ can be non-convex, and the search space constrained by $\mathcal{P}$ is discrete and finite. Our discrete solver tackles these challenges by relaxing the problem and utilizing an alternating scheme to optimize the energy and promote the proper functional map property in an efficient way.

**Reformulation by Replacement** For a given energy $E(C)$ (that might contain multiple terms), we first reformulate it in the form of $E(C) = \sum_i w_i \|CA_i - B_i\|_F^2$ as described in the previous section. Note that $A_i$ and $B_i$ can be constant matrices or be dependent on $C$. We then rewrite this energy by replacing $C$ with $\Phi_2^{\dagger}\Pi_{21}\Phi_1$ to obtain:

$$E^{\text{mod}}(C_{12}, \Pi_{21}) = \sum_i w_i \|\Phi_2^{\dagger}\Pi_{21}\Phi_1 A_i - B_i\|_F^2 \qquad (5)$$

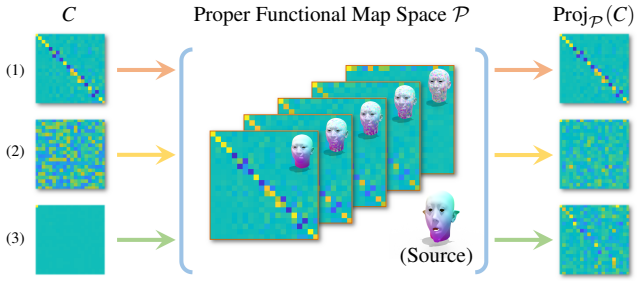


**Figure 6:** ***Projection into proper functional map space****. We show the proper functional map space $\mathcal{P}$ of a pair of head meshes in the middle, where each function map in $\mathcal{P}$ has an underlying pointwise map (visualized on top right at each functional map). For any arbitrary functional map $C$ we can project them into $P$ to find its nearest proper functional map $Proj_{\mathcal{P}}(C)$. Here we show three example functional maps before (on the left) and after (on the right) projection: (1) the ground-truth functional map; (2) a random matrix; (3) the optimized functional map as proposed in [RPWO18].*

**Relaxation with Coupling** The reformulation by replacement step helps to simplify the optimization with independent variables. To address the hard constraint, we add another coupling term between $C_{12}$ and $\Pi_{21}$ to the modified energy while maintaining the simplicity of the energy:

$$E^{\text{relax}}(C_{12},\Pi_{21}) = E^{\text{mod}}(C_{12},\Pi_{21}) + \alpha\left\| \Phi_2^{\dagger}\Pi_{21}\Phi_1 C_{12}^T - Id \right\|_F^2 \quad (6)$$

Instead of using a soft penalty $\left\| C_{12} - \Phi_2^{\dagger}\Pi_{21}\Phi_1 \right\|_F^2$, we used the coupling term $\left\| \Phi_2^{\dagger}\Pi_{21}\Phi_1 C_{12}^T - Id \right\|_F^2$ that is based on orthogonality, which in practice can lead to better results when we have poor initializations or the original energy is under-constrained (e.g., the descriptor-preserving energy as shown in Fig. 4).

**Alternating Optimization with Progressive Upsampling** With the relaxed energy in hand, our approach for solving the problem in Eq. (3) consists of the following simple procedure: alternatively optimizing a pointwise map $\Pi_{21}$ and constructing a proper functional map $C_{12}$ until convergence or reaching other stopping criterion. Throughout the discussion above, we have assumed that the functional maps are of fixed size, determined by the number of basis functions on each shape. In most previous work this number is chosen *a priori* and ranges between 60-200 [OBCS*12, KBB*13, NO17, EBC17]. However, a recent work has shown that progressively increasing the dimensionality of functional maps can be very beneficial for improving accuracy and robustness [MRR*19]. We therefore incorporate the progressive upsampling technique into our discrete solver:

1. Construct the basis with size $k$: $\Phi_1 = \Phi_1^{(k)}, \Phi_2 = \Phi_2^{(k)}$
2. Optimize the pointwise map: $\Pi_{21}^* = \arg\min_{\Pi_{21}} E^{\text{relax}}(\Pi_{21}|C_{12})$;
3. Construct proper functional map: $C_{12} = \Phi_2^{\dagger}\Pi_{21}^*\Phi_1$.
4. Repeat step 2-3 for $N$ times
5. $k \leftarrow k+1$, go to step 1.

See Fig. 5 for an overview of our discrete solver. Thanks to the energy reformulation and the relaxed energy construction step, the optimization in Step 1 can be reduced to a nearest-neighbor searching in a closed form according to Lemma 4.1.

**Table 1:** *Comparing our solution to the exact solution and the approximate solution on minimizing the orthogonality energy on the synthetic shapes with size $n = 3,4,5,6$ (one per row).*

| $n$ | Runtime (s) | | | $E_{\text{ortho}}(C^*)$ | | |
|---|---|---|---|---|---|---|
| | Exact | Approx. | **Ours** | Exact | Approx. | **Ours** |
| 3 | 0.0039 | 0.0233 | 0.0024 | 0.4722 | 1.4727 | 0.4722 |
| 4 | 0.0129 | 0.0089 | 0.0039 | 0.0139 | 1.4283 | 1.4005 |
| 5 | 0.1252 | 0.0105 | 0.0049 | 1.0058 | 2.4786 | 1.0058 |
| 6 | 3.0243 | 0.0275 | 0.0067 | 0.1664 | 1.7843 | 1.0275 |

### 5.2. Baselines

**Projection into Proper Functional Map Space** Before we discuss alternative solutions to solve Eq. (3), we first introduce a useful technique: a proper projector $\text{Proj}_{\mathcal{P}_{12}}(\cdot)$ that maps an arbitrary functional map in $\mathbf{R}^{k_2\times k_1}$ to a proper functional map in $\mathcal{P}_{12} \subset \mathbf{R}^{k_2\times k_1}$:

**Definition 5.1** The proper projector $\text{Proj}_{\mathcal{P}_{12}}(\cdot)$ maps an arbitrary functional map to its nearest neighbor in $\mathcal{P}_{12}$ in Frobinus-norm distance metric. Specifically, for any $C_{12}$ we have $\text{Proj}_{\mathcal{P}_{12}}(C_{12}) = \arg\min_{C\in\mathcal{P}_{12}} \left\| C - C_{12} \right\|_F^2$.

We use the Frobenius norm to compare functional maps as done in many previous works, e.g., [OCB*17, RPWO19], and since it can be shown, e.g., that the difference between two functional maps in that norm can be used to bound the geodesic difference between the corresponding underlying pointwise maps (See Theorem 4.1. [RMOW20]).

**Remark 5.1** With an additional regularizer $\left\|(Id - \Phi_1\Phi_1^{\dagger})\Pi_{21}C_{12}\right\|_A^2$ as discussed in Lemma 4.1, we can redefine the proper projector to have a closed form solution:

$$\text{Proj}_{\mathcal{P}_{12}}(C_{12}) = \Phi_2^{\dagger}\left(\arg\min_{\Pi_{21}} \left\| \Phi_2 C_{12} - \Pi_{21}\Phi_1 \right\|_F^2\right)\Phi_1 \quad (7)$$

See Appendix A for a simple proof. Fig. 6 shows some examples of projecting functional maps into the proper functional map space $\mathcal{P}$. With this technique in hand, we individuate two straight-forward baselines to solve Eq. (3):

1. **Exact solution.** We can plug in the hard constraint into the original energy and obtain $E_p(\Pi_{21}) = E(\Phi_2^{\dagger}\Pi_{21}\Phi_1)$, where the new energy $E_p$ is solely defined on a pointwise map $\Pi_{21}$. We can then first solve for $\Pi_{21}^* = \arg\min_{\Pi_{21}} E_p(\Pi_{21})$ using a discrete optimizer. Finally, we can obtain the $C_{21}^* = \Phi_2^{\dagger}\Phi_{21}^*\Phi_1$
2. **Approx. solution.** We can first solve $C_{12} = \arg\min_{C_{12}} E(C_{12})$ in unconstrained setting, then project $C_{12}$ to $\mathcal{P}_{12}$ to satisfy the hard constraint. I.e., $C_{12}^* = \text{Proj}_{\mathcal{P}_{12}}(C_{12})$.

We can see that the solution 1 gives the exact minimizer of Eq. (3). However, for most of the commonly-used functional map energies, it is very hard to find the global minimizer of $\Pi_{21}^* = \arg\min_{\Pi_{21}} E_p(\Pi_{21})$. Take the orthogonality for an example, we have $E(C_{12}) = \left\| C_{12}C_{12}^T - Id \right\|_F^2$ where $Id$ is an identity matrix with

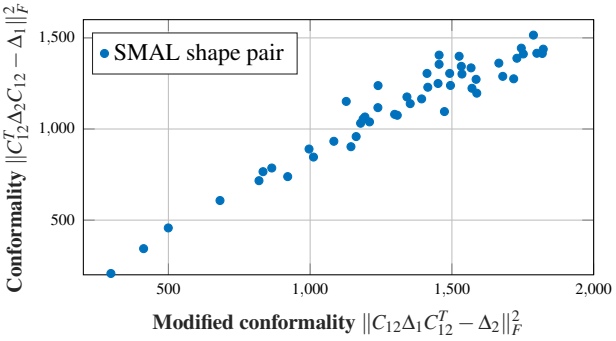


**Figure 7:** *We compare the* modified *conformality energy to the* standard *conformality energy on the proper functional maps of 50 SMAL shape pairs (obtained by minimizing the modified conformality using our discrete optimizer).*

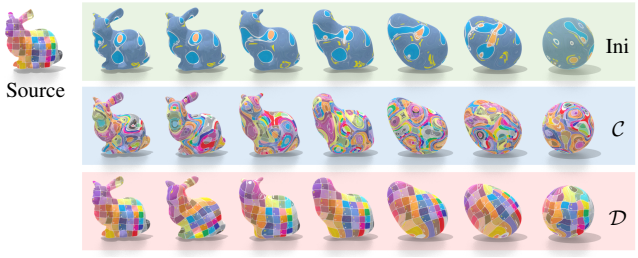


**Figure 8:** *Minimizing conformality using the continuous solver $\mathcal{C}$ and our discrete solver $\mathcal{D}$ on deformed bunny pairs from random initializations. We visualize the obtained pointwise map via texture transfer. See Fig. 20 for the corresponding color transfer visualization.*

proper size. As discussed in the solution 1, we can plug in the hard constraint to obtain an energy defined on the point-wise map $\Pi_{21}$, i.e., $E_p(\Pi_{21}) = \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 \left(\Phi_2^\dagger \Pi_{21} \Phi_1\right)^T - Id \right\|_F^2$. It is hard to find the global minimizer over the large and discrete search space of all possible pointwise maps. As a comparison, the solution 2 is much easier to obtain. However, as illustrated in Fig. 6, the projected proper functional map $C_{12}^*$ can be far from the optimized $C_{12}$ in the unconstrained setting. Therefore, it is very likely that the solution $C_{12}^*$ is far from a local minimizer of the original $E(C_{12})$.

As a comparison, our solution combines the advantages of both two baselines. See Table 1 for a comparison between our solution and the two baselines on minimizing the orthogonality. We test on synthetic data (see Fig. 14 in Appendix C) where we can explore the complete proper functional map space and find the exact solution. We can see that, the exact solution can find the global minimum, but the computation complexity is exponential w.r.t. the mesh size. Note that it is impossible to enumerate the search space $\mathcal{P}_{12}$ when $n > 10$. The approximate solution is much more efficient. However, the output functional map still has large error. As a comparison, our discrete solver, is more efficient and accurate than the approximate solution. For $n = 3,5$ we even obtain the global minimum from a random initialization.

**Table 2:** *For the tested bunny pairs in Fig. 8, we measure the conformality of the optimized functional maps and of the recovered pointwise maps obtained from the continuous solver $\mathcal{C}$ and our discrete solver $\mathcal{D}$.*

| Conformality Metric | Solver | Shape Pair ID 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Functional map | $\mathcal{C}$ | 275.9 | 281.2 | 280.8 | 315.4 | 282.1 | 307.4 | 411.2 |
| | $\mathcal{D}$ | 118.2 | 262.9 | 257.7 | 258.1 | 268.0 | 251.7 | 276.7 |
| Pointwise map | $\mathcal{C}$ | 21.05 | 21.09 | 20.59 | 24.03 | 20.95 | 25.84 | 22.31 |
| | $\mathcal{D}$ | 0.308 | 1.049 | 0.753 | 0.471 | 0.731 | 1.150 | 1.742 |

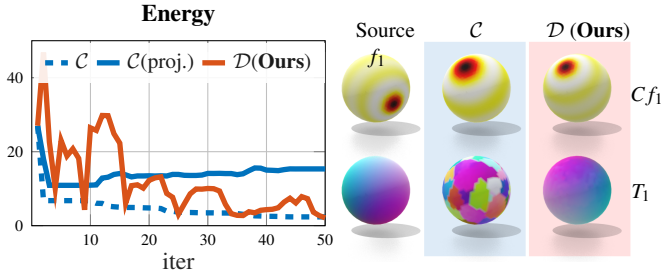


**Figure 9:** *For the input descriptor $(f_1, g_1)$ in Fig. 4, we compare our discrete solver $(\mathcal{D})$ to the continuous solver $(\mathcal{C})$. We visualize the transported descriptor and the recovered pointwise map on the right. See Fig. 15 and 16 in Appendix C for the corresponding results of the other two pairs of descriptors.*

### 5.3. Examples

In this section, we discuss how to apply our discrete solver to optimize the conformality energy as an example. Please see Appendix C for detailed discussions on other functional map based energies such as descriptor preserving energy, (multiplicative, orientation preserving/reversing, Laplacian) operator commutativity energy, orthogonality, and bijectivity energy.

**Conformality** We can construct the relaxed version for the modified conformality energy as discussed in Sec. 4: $E^{\text{relax}} = \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 \Delta_1 C_{12}^T - \Delta_2 \right\|_F^2 + \alpha \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{12}^T - Id \right\|_F^2$
We can then update the pointwise map by (1) $\Pi_{21} = \arg\min_{\Pi_{21}} \left\| \Pi_{21} \Phi_1 \left[\Delta_1 C_{12}^T, \alpha C_{12}^T\right] - \left[\Phi_2 \Delta_2, \alpha \Phi_2\right] \right\|_F^2$, and (2) $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$. We show that the modified conformal energy is strongly related to the standard conformal energy in Fig. 7. Specifically, we use our method to optimize for proper functional maps on 50 SMAL animal shapes [ZKJB17], obtained by minimizing the modified conformal energy. We then evaluate the standard and the modified conformal energy on these obtained proper functional maps. We can see that these two energies are nearly linearly dependent: for the maps (the blue points) with small modified conformal energy, their standard conformal energy is also small. We show another example in Fig. 8, where we apply the continuous solver to minimize the *standard* conformality energy, and apply our discrete solver to minimize the *modified* conformality energy on deformed bunnies. In Table 2 we report

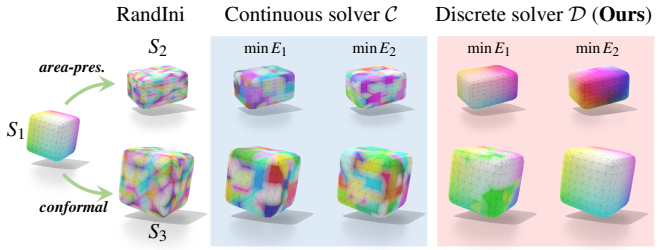


**Figure 10:** *Comparing our discrete solver $\mathcal{D}$ to the continuous solver $\mathcal{C}$ on minimizing the area-preserving energy $E_1 = \|CC^T - I\|_F^2$ and the conformal energy $E_2 = \|C^T\Delta_1 C - \Delta_2\|_F^2$ on two shape pairs $(S_1, S_2)$ and $(S_1, S_3)$, where $S_2$ has the same area as $S_1$ and $S_3$ is conformal to $S_1$.*

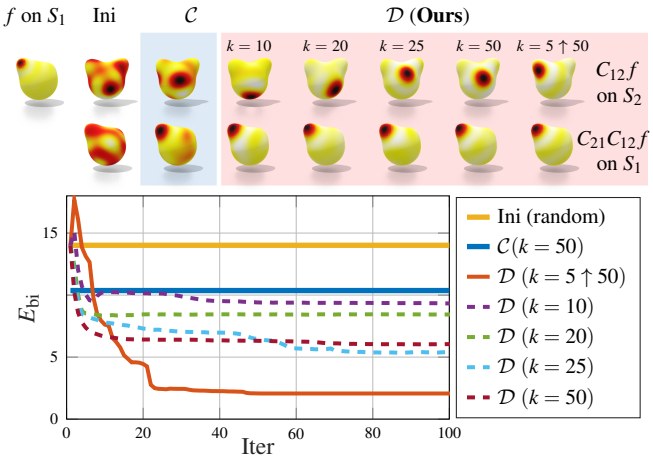


**Figure 11:** Top*: for a pair of deformed spheres, we compare our discrete solver $\mathcal{D}$ with different functional map size $k$ to the continuous solver $\mathcal{C}$ on minimizing the bijectivity energy from random initialization. Here $k = 5 \uparrow 50$ means running our solver with the upsampling technique from size 5 to 50.* Bottom*: We show the bijective energy over iterations of different results.*

the standard conformality measured on both the functional maps and the pointwise maps. We can see that the pointwise maps from our discrete solver are more conformal and smooth.

**Energy-aware optimization** Fig. 4 already shows that our discrete solver is energy-aware. With different input descriptors, the pointwise maps obtained by our discrete solver satisfy the descriptor preserving constraint respectively. We compare our discrete solver to the continuous solver on preserving the descriptors $(f_1, g_1)$ in Fig. 9. We show another example in Fig. 10. We use the continuous solver to minimize the area-preserving energy $E_1$ and the *standard* conformal energy $E_2$ (shown in blue box). We then use our discrete solver to minimize area-preserving energy $E_1$ and the *modified* conformal energy $E_2$. We test on two shape pairs, where the first pair $(S_1, S_2)$ have the same surface area, and the second shape pair $(S_1, S_3)$ are conformal to each other. We can see that, minimizing the area-preserving energy on the shape pair $(S_1, S_2)$ that have the same surface area leads to a better and smoother map than minimizing the conformal energy. Similarly, minimizing the conformal energy on the shape pair $(S_1, S_3)$ that the two shapes are conformal leads to a better and smoother map than minimizing the area-preserving energy.

**Table 3:** *Comparing our discrete solver $\mathcal{D}$ to the continuous solver $\mathcal{C}$ for minimizing different functional map energies on non-isometric animal shapes from the SMAL dataset. We report the statistics over 50 shape pairs including the minimum, average, maximum, and standard deviation of the energy values $E_i, i = 1, 2, 3, 4$.*

| Energies \ Stats. | | min. | avg. | max. | std. |
|---|---|---|---|---|---|
| $E_1 = \|CC^T - I\|_F$ | $\mathcal{C}$ | 4.9924 | 5.1932 | 5.4530 | 0.0976 |
| | $\mathcal{D}$ | 0.6789 | **2.2557** | 3.0707 | 0.4801 |
| $E_2 = \|C\Delta_1 - \Delta_2 C\|_F$ | $\mathcal{C}$ | 1.0615 | 1.2261 | 1.4205 | 0.0900 |
| | $\mathcal{D}$ | 0.1575 | **0.8053** | 1.2197 | 0.2229 |
| $E_3 = \|C\Delta_1 C^T - \Delta_2\|_F$ | $\mathcal{C}$ | 1.9219 | 2.1302 | 2.2884 | 0.1005 |
| | $\mathcal{D}$ | 0.2972 | **1.3173** | 1.8223 | 0.3694 |
| $E_4 = \|C_{12}C_{21} - I\|_F$ | $\mathcal{C}$ | 12.531 | 13.059 | 13.948 | 0.2645 |
| $+ \|C_{21}C_{12} - I\|_F$ | $\mathcal{D}$ | 0.9355 | **4.1369** | 5.3564 | 1.0138 |


**Figure 12:** *We visualize the pointwise maps between SMAL shapes obtained from optimizing the Laplacian Commutativity energy by using the continuous solver (middle) or our discrete solver (bottom) from random initialization (top). In the top row, we also visualize the topology of the meshes.*

**Progressive upsampling** In Fig. 11 we compare our discrete solver with fixed dimension $k$ (dashed lines) to the setting with upsampling technique(red solid line) on minimizing the bijective energy. We can see that our discrete solver outperforms the continuous solver (blue solid line) with different choices of $k$. Applying the upsampling technique can further improve the performance of our discrete solver.

**Combined Functional Map Energy** We also propose a new energy $E_{\text{new}}(C_{12}, C_{21})$ that combines the orthogonality, Laplacian Commutativity, and bijectivity on functional maps from *both* directions of a shape pair.

$$\begin{aligned} E_{\text{new}}(C_{12}, C_{21}) =& w_1 E_{\text{bi}}(C_{12}, C_{21}) + w_2 E_{\text{ortho}}(C_{12}) + w_2 E_{\text{ortho}}(C_{21}) \\ &+ w_3 E_{\text{lap}}(C_{12}) + w_3 E_{\text{lap}}(C_{21}) \end{aligned} \tag{8}$$

Applying our discrete solver to this combined energy, we obtain a new refinement method called Effective Functional Map refinement (EFMR). In the supplementary materials, we discuss the ad-

**Table 4:** *Comparing our discrete solver ($\mathcal{D}$) to the standard continuous solver ($\mathcal{C}$) on minimizing the multiplicative energy ($E_1$) and the orientation-preserving energy ($E_2$) proposed in [NO17] and [RPWO18] respectively. We measure the final energy on the output functional maps ("Energy"), and measure the quality of recovered pointwise maps including the smoothness, conformal distortion, and map accuracy w.r.t. the ground-truth. We report the statistics on 18 FAUST shapes, where the first 9 shapes are the same person with different poses, and the rest shapes are different persons with different poses. See Fig. 17 in Appendix C for the test shapes and the computed maps.*

| Energies \stats | | min. | avg. | max. | std. |
|---|---|---|---|---|---|
| $E_1$ ($\times 10^{-3}$) | $\mathcal{C}(E_1)$ | 69.0 | 330 | 642 | 0.1696 |
| | $\mathcal{D}(E_1)$ | 2.40 | **130** | 256 | 0.0777 |
| | $\mathcal{C}(E_2)$ | 19.8 | 240 | 463 | 0.1360 |
| | $\mathcal{D}(E_2)$ | 2.50 | **134** | 247 | 0.0780 |
| $E_2$ ($\times 10^{-3}$) | $\mathcal{C}(E_1)$ | 217 | 1017 | 1653 | 0.2855 |
| | $\mathcal{D}(E_1)$ | 18.8 | 1722 | 1394 | 3.1232 |
| | $\mathcal{C}(E_2)$ | 35.0 | **433** | 662 | 0.1574 |
| | $\mathcal{D}(E_2)$ | 18.7 | **379** | 585 | 0.1434 |
| Smoothness (Dirichlet) | $\mathcal{C}(E_1)$ | 13.8 | 40.4 | 65.5 | 16.407 |
| | $\mathcal{D}(E_1)$ | 8.25 | **9.78** | 11.9 | 0.9051 |
| | $\mathcal{C}(E_2)$ | 9.13 | 25.1 | 48.9 | 12.304 |
| | $\mathcal{D}(E_2)$ | 7.87 | **9.56** | 13.3 | 1.3858 |
| Conformal | $\mathcal{C}(E_1)$ | 2.78 | 6.83 | 10.7 | 2.1250 |
| | $\mathcal{D}(E_1)$ | 1.17 | **2.18** | 3.21 | 0.4637 |
| | $\mathcal{C}(E_1)$ | 2.11 | 5.55 | 8.32 | 1.6330 |
| | $\mathcal{D}(E_1)$ | 1.20 | **2.06** | 3.36 | 0.5518 |
| Accuracy (direct) ($\times 10^{-3}$) | $\mathcal{C}(E_1)$ | 87.6 | 337 | 570 | 160.75 |
| | $\mathcal{D}(E_1)$ | 15.4 | 309 | 649 | 278.81 |
| | $\mathcal{C}(E_2)$ | 40.4 | **147** | 370 | 92.401 |
| | $\mathcal{D}(E_2)$ | 16.8 | **169** | 656 | 243.30 |
| Accuracy (direct or symmetry) ($\times 10^{-3}$) | $\mathcal{C}(E_1)$ | 48.1 | 118 | 268 | 52.468 |
| | $\mathcal{D}(E_1)$ | 15.4 | **47.3** | 86.2 | 21.349 |
| | $\mathcal{C}(E_2)$ | 40.0 | 87.6 | 208 | 35.824 |
| | $\mathcal{D}(E_2)$ | 16.8 | **45.3** | 93.2 | 22.851 |

vantages of the new energy and EFMR method with a thorough investigation on synthetic datasets. For example, we show that the new energy can better constrain a proper functional map with much less local minima over the complete proper functional map search space than other commonly used functional map energies. At the same time, EFMR has a much stronger convergence power from random initializations, and the terminating (converged) points of EFMR are indeed local minima of the new energy. Please see the supplementary materials for more details.

## 6. Results

In this section, we demonstrate the advantages of our discrete solver over the continuous solver on different energies. We show that our discrete solver is an alternative to the commonly used multiplicative operator in the standard functional map pipeline. We also propose a new energy that combines some standard functional map regularizers including the orthogonality, Laplacian commutativity, and bijectivity. We then use our discrete solver to optimize this new energy, which leads to a novel refinement method that achieves the state-of-the-art accuracy on shape matching task on SHREC'19 benchmark.

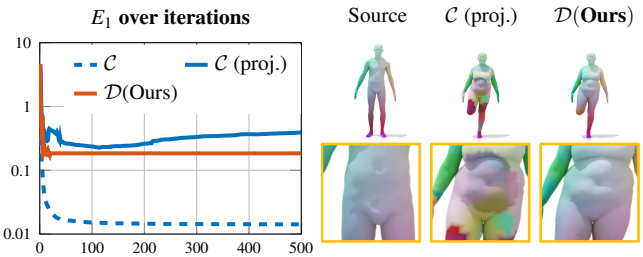


**Figure 13:** *Example shape pair in Table 4. We compare our discrete solver to the continuous solver with multiplicative terms without postprocessing.* Left*: the energy $E_1$ on functional maps over iterations;* Right*: we visualize the obtained pointwise maps via color transfer.*

### 6.1. Evaluation of Our Discrete Solver

We compare our discrete solver to the standard continuous solver on some commonly used energies on 50 animal shape pairs from SMAL dataset. To remove the bias present due to identical mesh connectivity within the dataset, we use LRVD algorithm [YBZW14] to remesh the shapes independently and each remeshed shape contains approximately 5k vertices. We investigate four functional map based energies including orthogonality (area-preserving), Laplacian commutativity (isometric), modified conformality (angle-preserving), and bijectivity. We report different statistics of the corresponding objective values on the optimized proper functional map using the continuous solver ($\mathcal{C}$) or our discrete solver ($\mathcal{D}$) in Table 3 on 50 shape pairs where we highlight the best average value with bold text. Fig. 12 shows some qualitative results, where we visualize the output pointwise maps obtained from the continuous solver and our discrete solver on minimizing the Laplacian commutativity initialized by random maps. In summary, our discrete solver not only achieves smaller objective values for different energies, but also outputs more desirable pointwise map than the continuous solver.

### 6.2. Application 1: Alternative of Multiplicative Operators

Our discrete solver can also benefit the state-of-the-art functional map optimization pipeline proposed in [NO17, RPWO18]. Specifically, [NO17] also observed the limitation of the functional map pipeline that the solved-for functional maps do not necessarily correspond to a pointwise map. To address this issue, [NO17] proposed multiplicative operators to by extracting extra information from the input descriptors to guide the optimization towards functional maps that are closer to pointwise maps, i.e., to proper functional maps in our terms.

We quantitatively demonstrate that our discrete solver outperforms the multiplicative operators on finding proper functional maps from input descriptors introduced in [NO17]. Specifically, we

**Table 5:** ***SHREC'19 Challenge.*** *We compare our method to different shape matching techniques including ICP, PMF, BCICP, and* ZOOMOUT *on 430 shape pairs in SHREC'19. For PMF and BCICP, we pick the sampling size (1k and 5k respectively) with the best accuracy shown in [MRR*19] as our baselines.*

| Methods \ Metrics | Accuracy $(\times 10^{-3})$ | Bijectivity $(\times 10^{-3})$ | Runtime (s) |
|---|---|---|---|
| Initialization | 60.4 | 95.1 | - |
| ICP | 47.0 | 47.4 | 87.3 |
| PMF (1k) | 51.8 | **11.8** | 118.1 |
| BCICP (5k) | 30.1 | 12.7 | 437.9 |
| RHM | 42.6 | 13.5 | 2313 |
| ZOOMOUT | 28.8 | 26.1 | **1.5** |
| **Ours** | **27.3** | 15.1 | 8.17 |

test the state-of-the-art energies used in [NO17] and [RPWO18] which involve the multiplicative operators constructed from the wave kernel signatures [ASC11]. We then compare our discrete solver to the continuous solver in the following ways: (1) we use the standard *continuous solver* to optimize the originally proposed energy as in [NO17, RPWO18]; (2) we use our *discrete solver* to optimize the energies without the multiplicative energy term. We then compare (1) and (2) with the same initialization and the same parameters (as used in the original paper). We report the statistics of the energy values on the proper functional maps solved by (1) and (2) and evaluate the corresponding recovered pointwise maps including the accuracy and smoothness metrics in Table 4 (where we highlight the best two average values with bold text). Fig. 13 shows the optimization process on an example pair and Fig. 17 in Appendix C shows the recovered pointwise maps of the tested shape pairs. We can see that our discrete solver can achieve better objective values and better pointwise maps (i.e., more accurate and smoother) than the multiplicative operators with continuous solver.

### 6.3. Application 2: Effective Functional Map Refinement

Here we show that our new refinement method EFMR, obtained from applying our discrete solver to the new energy (Eq. (8)) that combines the orthogonality, Laplacian Commutativity, and bijectivity, achieves state-of-the-art accuracy on SHREC'19 benchmark composed of human shapes with different connectivities and that contains complex meshes obtained from real scans.

We run EFMR on SHREC'19 [MMR*19] for shape matching in the same setting as reported in [MRR*19]. Specifically, we compare our EFMR method to the other refinement methods on 430 shape pairs from the same initializations obtained from 5 landmarks. In Table 5, we report the average map accuracy, map bijectivity error, and runtime. We pick the best tested parameters for the baseline methods PMF and BCICP, as reported in [MRR*19]. Compared with the best baseline ZOOMOUT, our method improves the map accuracy by 5%, and the map bijectivity by 40% with a comparable runtime complexity. Note that, PMF and BCICP obtained better map bijectivity since they have explicit bijectivity constraints on the pointwise maps, while our method only adds the bijectivity to the functional maps with no extra constraints added to the pointwise maps. See the supplementary materials to find more justifications and evaluations of our EFMR method.

**Table 6:** *The average runtime (sec) over 50 shape pairs of using the continuous ($\mathcal{C}$) or our discrete ($\mathcal{D}$) solver to optimize different energies as shown in Table 3.*

| Method \ Energy | $E_1$ | $E_2$ | $E_3$ | $E_4$ |
|---|---|---|---|---|
| $\mathcal{C}$ | 9.214 | 11.50 | 13.03 | 32.76 |
| $\mathcal{D}$ **(Ours)** | 5.786 | 5.898 | 6.090 | 22.89 |

### 6.4. Implementation, Parameters & Runtime

We implemented the discrete solver and EFMR method in MATLAB. We use the MATLAB build-in function "fminunc" as the continuous solver $\mathcal{C}$ to optimize different energies in our experiments. In most of our tests, we set $k_1 = k_2 = 20$ for both the continuous solver and our discrete solver. In [RPWO19], it was shown that with the same input descriptors, increasing the size of the functional map can degrade the quality of the optimized solution, since the number of variables increases. At the same time, it was shown in [MRR*19] that a functional map of size at least 15-20 is necessary to resolve the main symmetries of most organic shapes (e.g., humans and animals) . These two reasons motivate our choice for $k_1$ and $k_2$. For the synthetic examples with the number of vertices $n < 10$, we mainly set $k_i = n_i$, i.e., setting the basis size to equal to the number of vertices to avoid an empty feasibility region. We mainly set $N$ (the number of inner loops with fixed basis) to 5 or 10. For the application of EFMR, we used the same set of parameters, i.e., $w_1 = w_2 = w_3 = 1, N = 10$ for all the tests ( except the ablation study in the supplementary materials). In Table 6, we report the average runtime of the continuous solver and our discrete solver on minimizing different functional map energies. We will release our complete implementation for full reproducibility. †

## 7. Conclusion, Limitations & Future Work

In this paper, we propose to optimize functional map based energies in a constrained setting to obtain a functional map that has a high-quality underlying pointwise map. To solve such a constrained problem, we propose a general and novel discrete solver that is energy-aware and efficient to solve for a proper functional map. Our discrete solver is easy to use and can be adapted to different functional map based energies. We present a large variety of experiments demonstrating that our discrete solver is better than the continuous solver in both terms of obtaining proper functional maps with lower objective values and recovering pointwise maps with better quality and accuracy. We further show two practical applications of our discrete solver: (1) as an alternative choice of the commonly-used multiplicative operators in standard functional

† Demo Code: https://github.com/llorz/SGP21_discreteOptimization

map optimization pipeline. (2) a new refinement method that has a strong convergence power and can produce high-quality pointwise map from random initialization. It also achieves state-of-the-art accuracy on SHREC'19 benchmark.

Our method still has some limitations. First, our discrete solver with the practical modifications has few theoretical guarantees on the performance of optimization. For example, our discrete solver is not guaranteed to monotonically decrease the original energy (See Fig. 9 for an example). However, we observe that it works well in practice. Second, using a Laplace-Beltrami discretization such as [BSW09], our method could be directly generalized to point clouds, but we did not investigate the performance in this setting. Finally, one important step to maintain the applicability of our discrete solver is to modify the energy in a way that fits our formulation using Lemma 4.1. We have shown the efficiency and feasibility of our discrete solver on a range of common functional map energies. However, for some functional map energies with complicated formulations, e.g., using higher order terms, this strategy might not work directly and more advanced solvers might be needed. Moreover, as shown in [PRM*21], other procedure such as Sinkhorn algorithm can be used to solve for the pointwise map in the first step of our discrete solver. We leave the exploration of different pointwise map recovery methods as future work. In the future, we plan to investigate the best coupling strategies, explore the full potential of the augmented Lagrangian formulations, and extend our solver to larger set of energies.

**Acknowledgement** The authors thank the anonymous reviewers for their valuable comments. Parts of this work were supported by the KAUST OSR Award No. CRG-2017-3426, the ERC Starting Grants No. 758800 (EXPROTEA) and No. 802554 (SPECGEO), and the ANR AI Chair AIGRETTE.

## References

[ADG19] AZENCOT O., DUBROVINA A., GUIBAS L.: Consistent shape matching via coupled optimization. *Computer Graphics Forum 38*, 5 (2019), 13–25. 2

[AK13] AFLALO Y., KIMMEL R.: Spectral multidimensional scaling. *PNAS 110*, 45 (2013), 18052–18057. 2

[AL16] AIGERMAN N., LIPMAN Y.: Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics 35*, 6 (Nov. 2016), 217:1–217:14. 2

[APL15] AIGERMAN N., PORANNE R., LIPMAN Y.: Seamless surface mappings. *ACM Transactions on Graphics (TOG) 34*, 4 (July 2015). 2

[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (2011), IEEE, pp. 1626–1633. 3, 10, 14

[BBK06] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences 103*, 5 (2006), 1168–1172. 2

[BCBB16] BIASOTTI S., CERRI A., BRONSTEIN A., BRONSTEIN M.: Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum 35*, 6 (2016), 87–119. 2

[BDK17] BURGHARD O., DIECKMANN A., KLEIN R.: Embedding shapes with Green's functions for global shape matching. *Computers & Graphics 68* (2017), 1–10. 2

[BPC*11] BOYD S., PARIKH N., CHU E., PELEATO B., ECKSTEIN J., ET AL.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning 3*, 1 (2011), 1–122. 4

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR* (Columbus, Ohio, 2014), IEEE, pp. 3794–3801. 1

[BSW09] BELKIN M., SUN J., WANG Y.: Constructing laplace operator from point clouds in rd. In *Proc. Symposium on Discrete Algorithms (SODA)* (2009), pp. 1031–1040. 11

[BTM18] BERNARD F., THEOBALT C., MOELLER M.: Ds*: Tighter lifting-free convex relaxations for quadratic matching problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 4310–4319. 1

[DML17] DYM N., MARON H., LIPMAN Y.: Ds++: a flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 184. 1, 2

[DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8592–8601. 3

[EBC17] EZUZ D., BEN-CHEN M.: Deblurring and denoising of maps between shapes. *Computer Graphics Forum 36*, 5 (2017), 165–174. 2, 3, 4, 6, 13

[ERGB16] EYNARD D., RODOLA E., GLASHOFF K., BRONSTEIN M. M.: Coupled functional maps. In *2016 Fourth International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 399–407. 2, 5

[ESBC19] EZUZ D., SOLOMON J., BEN-CHEN M.: Reversible harmonic maps between discrete surfaces. *ACM Trans. Graph. 38*, 2 (2019), 15:1–15:12. 2, 3

[FJBd13] FOGEL F., JENATTON R., BACH F., D'ASPREMONT A.: Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems* (2013), pp. 1016–1024. 1, 2

[GBKS18] GEHRE A., BRONSTEIN M., KOBBELT L., SOLOMON J.: Interactive curve constrained functional maps. *Computer Graphics Forum 37*, 5 (2018), 1–12. 2, 3

[GM76] GABAY D., MERCIER B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications 2*, 1 (1976), 17–40. 4

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. *Computer Graphics Forum 27*, 5 (2008), 1449–1457. 2

[HLR*19] HALIMI O., LITANY O., RODOLA E., BRONSTEIN A. M., KIMMEL R.: Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4370–4379. 2, 3

[HO17] HUANG R., OVSJANIKOV M.: Adjoint map representation for shape analysis and matching. *Computer Graphics Forum 36*, 5 (2017), 151–163. 3

[HWG14] HUANG Q., WANG F., GUIBAS L.: Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 36. 2

[KBB*13] KOVNATSKY A., BRONSTEIN M., BRONSTEIN A., GLASHOFF K., KIMMEL R.: Coupled quasi-harmonic bases. *Computer Graphics Forum 32*, 2pt4 (2013), 439–448. 2, 6

[KGB16] KOVNATSKY A., GLASHOFF K., BRONSTEIN M. M.: MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Proc. ECCV* (2016), Springer, pp. 680–696. 5

[KKBL15] KEZURER I., KOVALSKY S. Z., BASRI R., LIPMAN Y.: Tight relaxation of quadratic matching. *Computer Graphics Forum 34*, 5 (2015), 115–128. 2

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 79. 2

[LdABN*07] LOIOLA E. M., DE ABREU N. M. M., BOAVENTURA-NETTO P. O., HAHN P., QUERIDO T.: A survey for the quadratic assignment problem. *European journal of operational research 176*, 2 (2007), 657–690. 1

[LF09] LIPMAN Y., FUNKHOUSER T.: MÖbius voting for surface correspondence. *ACM Trans. Graph. 28*, 3 (July 2009), 72:1–72:12. 2

[LH05] LEORDEANU M., HEBERT M.: A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (2005), vol. 2, IEEE, pp. 1482–1489. 1, 2

[LRBB17] LITANY O., RODOLÀ E., BRONSTEIN A., BRONSTEIN M.: Fully spectral partial shape matching. *Computer Graphics Forum 36*, 2 (2017), 247–258. 2, 5

[LRR*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE, pp. 5659–5667. 2, 3

[MCSK*17] MANDAD M., COHEN-STEINER D., KOBBELT L., ALLIEZ P., DESBRUN M.: Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics 36* (2017), 14. 2

[MDK*16] MARON H., DYM N., KEZURER I., KOVALSKY S., LIPMAN Y.: Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 73. 2

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*. Springer, New York, NY, 2003, pp. 35–57. 3

[MMR*19] MELZI S., MARIN R., RODOLÀ E., CASTELLANI U., REN J., POULENARD A., WONKA P., OVSJANIKOV M.: SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval* (2019), The Eurographics Association. 10

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG) 38*, 6 (Nov. 2019), 155:1–155:14. 2, 3, 4, 5, 6, 10, 13, 16

[NMR*18] NOGNENG D., MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M., OVSJANIKOV M.: Improved functional mappings via product preservation. *Computer Graphics Forum 37*, 2 (2018), 179–190. 2

[NO17] NOGNENG D., OVSJANIKOV M.: Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum 36*, 2 (2017), 259–267. 1, 2, 3, 5, 6, 9, 10, 14, 15

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 30:1–30:11. 1, 2, 3, 4, 5, 6

[OCB*17] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLÀ E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.: Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses* (2017), pp. 5:1–5:62. 2, 3, 6

[OMMG10] OVSJANIKOV M., MERIGOT Q., MEMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. *CGF 29*, 5 (2010), 1555–1564. 2, 3

[PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *CVPR* (2021). 11

[PSO18] POULENARD A., SKRABA P., OVSJANIKOV M.: Topological function optimization for continuous shape matching. *Computer Graphics Forum 37*, 5 (2018), 13–25. 3

[RCB*17] RODOLÀ E., COSMO L., BRONSTEIN M., TORSELLO A., CREMERS D.: Partial functional correspondence. *Computer Graphics Forum 36*, 1 (2017), 222–236. 2

[RMC15] RODOLÀ E., MOELLER M., CREMERS D.: Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)* (2015). 2

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph. 39*, 6 (Nov. 2020). 2, 6, 16

[ROA*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 72. 5

[RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG) 37*, 6 (2018). 2, 3, 5, 6, 9, 10, 14, 15

[RPWO19] REN J., PANINE M., WONKA P., OVSJANIKOV M.: Structured regularization of functional map computations. *Computer Graphics Forum 38*, 5 (2019), 39–53. 6, 10

[RSO19] ROUFOSSE J.-M., SHARMA A., OVSJANIKOV M.: Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 1617–1627. 2

[SNB*12] SOLOMON J., NGUYEN A., BUTSCHER A., BEN-CHEN M., GUIBAS L.: Soft maps between surfaces. *Computer Graphics Forum 31*, 5 (2012), 1617–1626. 2

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 399–405. 1

[SPKS16] SOLOMON J., PEYRÉ G., KIM V. G., SRA S.: Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 72. 1, 2

[SVBC19] SHOHAM M., VAXMAN A., BEN-CHEN M.: Hierarchical Functional Maps between Subdivision Surfaces. *Computer Graphics Forum* (2019). 2

[TCL*13] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE TVCG 19*, 7 (2013), 1199–1217. 2

[VLB*17] VESTNER M., LÄHNER Z., BOYARSKI A., LITANY O., SLOSSBERG R., REMEZ T., RODOLÀ E., BRONSTEIN A., BRONSTEIN M., KIMMEL R.: Efficient deformable shape correspondence via kernel matching. In *3D Vision (3DV), 2017 International Conference on* (2017), IEEE, pp. 517–526. 2

[VLR*17] VESTNER M., LITMAN R., RODOLÀ E., BRONSTEIN A., CREMERS D.: Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proc. CVPR* (2017), pp. 6681–6690. 2, 3

[WGBS18] WANG L., GEHRE A., BRONSTEIN M., SOLOMON J.: Kernel functional maps. *Computer Graphics Forum 37*, 5 (2018), 27–36. 2, 3

[WHG13] WANG F., HUANG Q., GUIBAS L. J.: Image co-segmentation via consistent functional maps. In *Proc. ICCV* (2013), pp. 849–856. 2

[WLZT18] WANG Y., LIU B., ZHOU K., TONG Y.: Vector field map representation for near conformal surface correspondence. *Computer Graphics Forum 37*, 6 (2018), 72–83. 2

[WS13] WANG L., SINGER A.: Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA 2*, 2 (2013), 145–193. 2

[YBZW14] YAN D.-M., BAO G., ZHANG X., WONKA P.: Low-resolution remeshing using the localized restricted voronoi diagram. *IEEE transactions on visualization and computer graphics 20*, 10 (2014), 1418–1427. 9

[ZGLG12] ZENG W., GUO R., LUO F., GU X.: Discrete heat kernel determines discrete riemannian metric. *Graphical Models 74*, 4 (2012), 121–129. 4

[ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D., BLACK M. J.: 3D menagerie: Modeling the 3D shape and pose of animals. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017* (Piscataway, NJ, USA, July 2017), IEEE, pp. 5524–5532. 7

## Appendix A: Proofs

### Proof of Lemma 4.1

**Lemma A.1** Given arbitrary matrices $X, Y$, and a reduced basis $\Phi$, s.t. $\Phi^T A \Phi = Id$, then problem (1): $\min_\Pi \|\Phi^+ \Pi X - Y\|_F^2 + \|(Id - \Phi\Phi^+)\Pi X\|_A^2$ is equivalent to problem (2): $\min_\Pi \|\Pi X - \Phi Y\|_F^2$.

We follow the proof idea, used for a special case of this statement shown in [EBC17] and adapted in [MRR*19]. We include the more general result below for completeness, although the proof is virtually identical.

*Proof* First note that if basis $\Phi$ is orthonormal with respect to a matrix $A$, i.e. $\Phi^T A \Phi = Id$, so that $\Phi^+ = \Phi^T A$, and $\|W\|_A^2 = tr(W^T A W)$, then for any matrix $W$: $\|W\|_A^2 = \|\Phi^+ W\|_F^2 + \|(I - \Phi\Phi^+)W\|_A^2$. This is simply because $\|\Phi^+ W\|_F^2 = tr(W^T A \Phi \Phi^T A W)$ while

$$\begin{aligned}\|(I - \Phi\Phi^+)W\|_A^2 &= tr\left(W^T (I - A\Phi\Phi^T) A (I - \Phi\Phi^T A) W\right) \\ &= tr\left(W^T (A - A\Phi\Phi^T A) W\right).\end{aligned}$$

Now we use this result with $W = \Pi X - \Phi Y$ to get

$$\begin{aligned}\|\Pi X - \Phi Y\|_A^2 &= \|\Phi^+(\Pi X - \Phi Y)\|_F^2 + \|(I - \Phi\Phi^+)(\Pi X - \Phi Y)\|_A^2 \\ &= \|\Phi^+ \Pi X - Y)\|_F^2 + \|(I - \Phi\Phi^+)\Pi X)\|_A^2.\end{aligned}$$

Finally, to show that $\|\Pi X - \Phi Y\|_F^2$ is equivalent to $\|\Pi X - \Phi Y\|_A^2$ note that both problems reduce to nearest neighbor search whenever $A$ is diagonal. □

### Proof of Remark 5.1

*Proof* By definition 5.1 we have $\text{Proj}_{\mathcal{P}_{12}}(C_{12}) = \arg\min_{C \in \mathcal{P}_{12}} \|C - C_{12}\|_F^2$. To derive its close-form formulation, we are supposed to solve the following problem:

$$\min_{C \in \mathcal{P}_{12}} \|C - C_{12}\|_F^2, \tag{9}$$

which can be equivalently reformulated as an unconstrained problem: $\min_{\Pi_{21}} \|\Phi_2^\dagger \Pi_{21} \Phi_1 - C_{12}\|_F^2$. According to Lemma 4.1, we can rewrite this equation (with an extra regularizer) as $\Pi_{21}^* = \arg\min_\Pi \|\Pi_{21}\Phi_1 - \Phi_2 C_{12}\|_F^2$, which has *row separable* variables [EBC17]. Thus it can be solved in close form solution, as a nearest-neighbor searching problem. Then the minimizer of Eq. (9) is the proper functional map that arises from $\Pi_{21}^*$. Therefore, $\text{Proj}_{\mathcal{P}_{12}}(C_{12}) = \Phi_2^\dagger \Pi_{21}^* \Phi_1 = \Phi_2^\dagger \left(\arg\min_{\Pi_{21}} \|\Phi_2 C_{12} - \Pi_{21}\Phi_1\|_F^2\right)\Phi_1$. □

## Appendix B: Different Coupling Strategies

There are different solutions of coupling the pointwise maps with the functional maps to relax the original problem with the hard constraint. Note that we are mainly interested in the coupling strategies that can be solved efficiently using alternating scheme for optimization. Here we give some example coupling strategies with algorithmic details to optimize the basic functional map energy Eq. (2) with the proper functional map constraint.

**a. ADMM** We can introduce the dual variables $Z, U$ to make the variables independent of each other. We also add a soft penalty term and obtain the following relaxed problem:

$$\begin{aligned}\min_{\Pi_{21}, C_{12}, Z, U} &\|C_{12}F_1 - F_2\|_F^2 + \gamma\|C_{12}\Delta_1 - \Delta_2 Z\|_F^2 \\ &+ \mu\|C_{12} - Z + U\|_F^2 + \alpha\|Z - \Phi_2^\dagger \Pi_{21} \Phi_1\|_F^2\end{aligned} \tag{10}$$

We can then alternative update the four set of variables $C_{12}, \Pi_{21}, Z, U$ with the following updating rules:

$$\begin{aligned}C_{12} &= \left[F_2, \gamma\Delta_2 Z, \mu(Z - U)\right] / \left[F_1, \gamma\Delta_1, \mu Id\right] \\ \Pi_{21} &= \text{knn}\left(\Phi_1, \Phi_2 C_{12}\right) \\ U &= Z - C_{12} \\ Z &= \left[\gamma\Delta_2, \mu Id, \alpha Id\right] \setminus \left[X\Delta_1, \gamma(X - U), \alpha\Phi_2^\dagger \Pi_{21}\Phi_1\right]\end{aligned}$$

where we use $A \setminus B$ to denote the minimizer of $\|AX - B\|_F^2$, $B/A$ to denote the minimizer of $\|XA - B\|_F^2$, $\text{knn}(A, B)$ to return the index of the nearest neighbor in $A$ for each row in $B$. We can iteratively update these four sets of variables until convergence.

**b. Coupling by replacement** We simply replace the functional map $C_{12}$ by the hard constraint for coupling and obtain the following relaxed energy:

$$\min_{C_{12}, \Pi_{21}} \|\Phi_2^\dagger \Pi_{21} \Phi_1 f_1 - F_2\|_F^2 + \gamma\|\Phi_2^\dagger \Pi_{21}\Phi_1\Delta_1 - \Delta_2 C_{12}\|_F^2 \tag{11}$$

We then have the following updating rules that alternatively solve the functional map $C_{12}$ and the pointwise map $\Pi_{21}$:

$$\begin{aligned}C_{12} &= \Delta_2 \setminus \left(\Phi_2^\dagger \Pi_{21} \Phi_1 \Delta_1\right) \\ \Pi_{21} &= \text{knn}\Big(\left[\Phi_1 F_1, \gamma\Phi_1\Delta_1\right], \left[\Phi_2 F_2, \gamma\Phi_2\Delta_2 C_{12}\right]\Big)\end{aligned}$$

**c. Replacement with soft penalty** We add a soft penalty to the solution b. and obtain:

$$\begin{aligned}\min_{C_{12}, \Pi_{21}} &\|\Phi_2^\dagger \Pi_{21}\Phi_1 F_1 - F_2\|_F^2 + \gamma\|\Phi_2^\dagger \Pi_{21}\Phi_1\Delta_1 - \Delta_2 C_{12}\|_F^2 \\ &+ \alpha\|C_{12} - \Phi_2^\dagger \Pi_{21}\Phi_1\|_F^2\end{aligned} \tag{12}$$

We can similarly derive the updating rules:

$$\begin{aligned}C_{12} &= \left[\gamma\Delta_1, \alpha Id\right] \setminus \left[\gamma\Phi_2^\dagger \Pi_{21}\Phi_1\Delta_1, \alpha\Phi_2^\dagger \Pi_{21}\Phi_1\right] \\ \Pi_{21} &= \text{knn}\Big(\left[\Phi_1 F_1, \gamma\Phi_1\Delta_1, \alpha\Phi_1\right], \left[\Phi_2 F_2, \gamma\Phi_2\Delta_2 C_{12}, \alpha\Phi_2 C_{12}\right]\Big)\end{aligned}$$

**d. Replacement with orthogonal regularizer** We add the orthogonal regularizer based penalty to the solution b. and obtain:

$$\begin{aligned}\min_{C_{12}, \Pi_{21}} &\|\Phi_2^\dagger \Pi_{21}\Phi_1 F_1 - F_2\|_F^2 + \gamma\|\Phi_2^\dagger \Pi_{21}\Phi_1\Delta_1 - \Delta_2 C_{12}\|_F^2 \\ &+ \alpha\|\Phi_2^\dagger \Pi_{21}\Phi_1 C_{12}^T - Id\|_F^2\end{aligned} \tag{13}$$

Analogously, the updating rules for solving $C_{12}$ and $\Pi_{21}$ are:

$$\begin{aligned}C_{12} &= \left[\gamma\Delta_2\Phi_2^\dagger \Pi_{21}\Phi_1, \alpha Id\right] / \left[\gamma\Delta_1, \alpha\left(\Phi_2\Pi_{21}\Phi_1\right)^T\right] \\ \Pi_{21} &= \text{knn}\Big(\left[\Phi_1 F_1, \gamma\Phi_1\Delta_1, \alpha\Phi_1 C_{12}^T\right], \left[\Phi_2 F_2, \gamma\Phi_2\Delta_2 C_{12}, \alpha\Phi_2\right]\Big)\end{aligned}$$

**ALGORITHM 1:** Algorithm Outline for Discrete Optimization

**Goal** : $\min_{C_{12}\in\mathcal{P}_{12}} E(C_{12})$

**Output:** A proper functional map $C_{12}$ that is a local minimum of $E$

**Preprocess**: Construct the relaxed energy $E^{\text{relax}}(\Pi_{21}, C_{12})$

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

  $\Phi_1 = \Phi_1^{(k)}, \Phi_2 = \Phi_2^{(k)}$

  **for** *iter = 1:N* **do**

    $\Pi_{21}^* = \arg\min_{\Pi_{21}} E^{\text{relax}}(\Pi_{21}\,|C_{12})$

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

  **end**

  $k \leftarrow k+1$

**end**

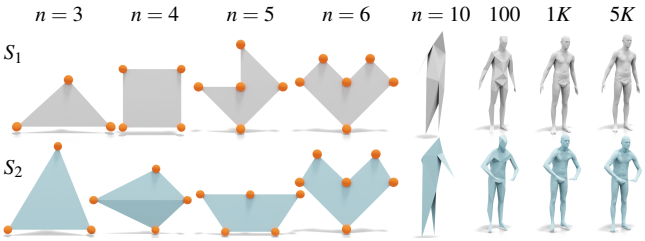


**Figure 14:** *Synthetic shapes for quantitative evaluation that are used in Table 1, Fig. 18, and Fig. 19.*

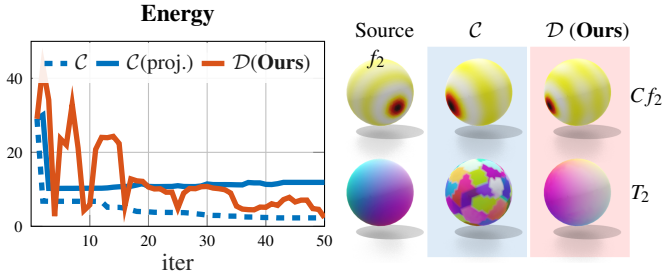


**Figure 15:** *For the input descriptor $(f_2, g_2)$ in Fig. 4, we compare our discrete solver $(\mathcal{D})$ to the continuous solver $(\mathcal{C})$.*

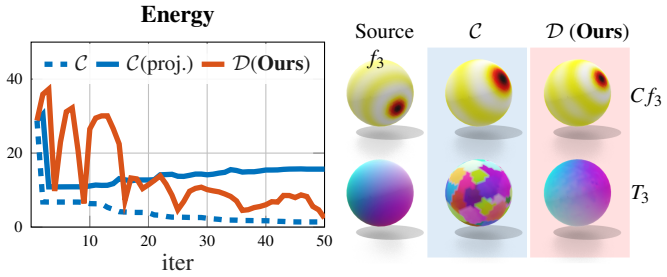


**Figure 16:** *For the input descriptor $(f_3, g_3)$ in Fig. 4, we compare our discrete solver $(\mathcal{D})$ to the continuous solver $(\mathcal{C})$.*

Note that here, when we optimize for $C_{12}$, we replaced the $C_{12}$ that besides $\Delta_1$ in the Laplacian commutativity energy to make it easier for optimization.

## Appendix C: Examples: Discrete Functional Map Optimization

In Sec. 5 we discussed our discrete solver with practical modifications. See Algorithm 1 for the general outline of minimizing a given functional map energy $E(C)$ over the proper functional map space using our discrete solver. In this appendix, we show in details how to construct the relaxed energy by coupling with replacement and adding additional orthogonality regularizer for different functional map based energies. We compare our discrete solver to the continuous solver. We also compare to the exact solution which is feasible only on synthetic shapes with small number of vertices. Fig. 14 shows some synthetic shape pairs we used. For example, for the shapes with $n = 3,4,5,6$ vertices, we can enumerate the complete proper functional map search space with size $n^n$ and find the exact global minima of any given functional map energies.

### Descriptor Preserving Energy

We first discuss the descriptor preserving energy which is commonly used in standard functional map computation pipeline. Specifically, given a pair of corresponding descriptors such as wave-kernel signatures [ASC11], $(f_{S_1}, f_{S_2})$ on shape pair $(S_1, S_2)$, we would like to find a functional map that preserves the corresponding descriptors. This leads to the following energy $E(C_{12}) = \|C_{12}\mathbf{f}_{S_1} - \mathbf{f}_{S_2}\|_F^2$, where $\mathbf{f}_{S_i}$ is the corresponding coefficient vector of the descriptor $f_{S_i}$ in the reduced basis $\Phi_{S_i}$, i.e., $\mathbf{f}_{S_i} = \Phi_{S_i}^{\dagger} f_{S_i}$. Given multiple pairs of input descriptors, we can define the energy in a similar way $E(C_{12}) = \|C_{12}F_{S_1} - F_{S_2}\|_F^2$, where $F_{S_i}$ is a matrix that stores as columns the coefficient of different descriptors. To apply our discrete solver, we first construct the relaxed energy as discussed in Sec. 5:

$$\begin{aligned} E^{\text{relax}}(C_{12},\Pi_{21}) &= \|\Phi_2^{\dagger}\Pi_{21}\Phi_1 F_{S_1} - F_{S_2}\|_F^2 + \alpha\|\Phi_2^{\dagger}\Pi_{21}\Phi_1 C_{12}^T - Id\|_F^2 \\ &= \|\Phi_2^{\dagger}\Pi_{21}[\Phi_1 F_{S_1}, \alpha\Phi_1 C_{12}^T] - [F_{S_2}, \alpha Id]\|_F^2, \end{aligned}$$

where $[A,B]$ denotes concatenating two matrices $A$ and $B$ side-by-side. According Lemma 4.1, we can obtain our algorithm for optimizing the descriptor preserving energy: alternatively updating $\Pi_{21}$ and $C_{12}$ with the upsampling technique.

$$\begin{aligned} \Pi_{21} &= \arg\min_{\Pi_{21}} \|\Pi_{21}[\Phi_1 F_{S_1}, \alpha\Phi_1 C_{12}^T] - [\Phi_2 F_{S_2}, \alpha\Phi_2]\|_F^2 \\ &= \text{knn}\Big([\Phi_1 F_{S_1}, \alpha\Phi_1 C_{12}^T], [\Phi_2 F_{S_2}, \alpha\Phi_2]\Big) \\ C_{12} &= \Phi_2^{\dagger}\Pi_{21}\Phi_1 \end{aligned} \tag{14}$$

In the following examples, we only give the updating rule for the pointwise map since updating $C_{12}$ is trivial.

Fig. 15 and Fig. 16 show the results of optimizing the descriptor preserving energy with different input descriptors $(f_2, g_2)$ and $(f_3, g_3)$ as shown in Fig. 4. We compare our discrete solver $\mathcal{D}$ to the continuous solver $\mathcal{C}$. We can see that our discrete solver can achieve lower objective values and obtain better pointwise maps.

### Operation Commutative Energy

Operation Commutativity term is also commonly used in functional map computation, including Laplacian Commutativity, multiplicative operation energy [NO17], and orientation preserving/reversing energy [RPWO18].

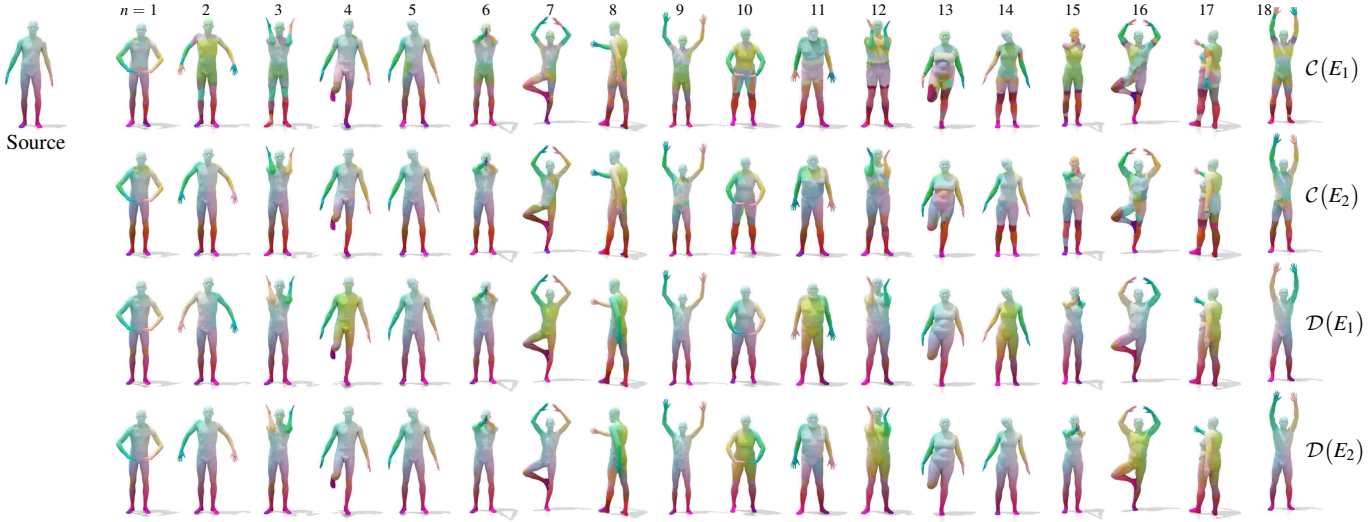


**Figure 17:** *The recovered pointwise maps by using the standard continuous solver ($\mathcal{C}$) and our discrete solver ($\mathcal{D}$) to minimize the multiplicative energy ($E_1$) [NO17] and the orientation-preserving energy ($E_2$) [RPWO18]. We report the quantitative evaluation of each recovered pointwise in Table 4. The shapes are from FAUST dataset, where the source shape is tr_reg_000. The first 9 shapes are tr_reg_00k($k = 1, \cdots, 9$), i.e., the same person as the source but with different poses; the last 9 shapes are tr_reg_0kk($k = 1, \cdots, 9$), i.e., different person with different pose.*

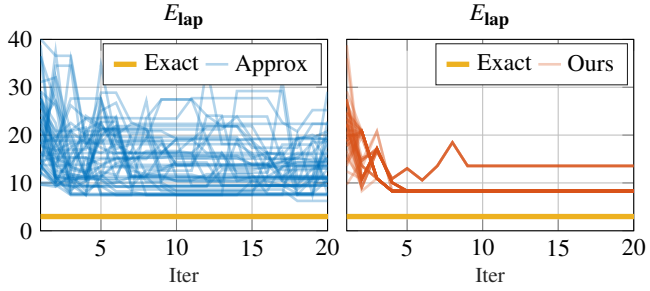


**Figure 18:** *Convergence rate. Starting from 50 random initial maps, we compare the approximate solution (left, blue lines) and ours (right, red lines), in the minimization of the Laplacian Commutativity energy. We can see that our solver converge much faster. Meanwhile, starting from different initializations, our solver terminates at two unique solutions.*

**Orientation-Reversing Energy** Given a pair of symmetric descriptor $(f_1, f_2)$ on shape pair $(S_1, S_2)$, we can construct an orientation operator $\Omega_{f_i}$ (in the reduced basis $\Phi_{S_i}$) on shape $S_i$, we can then define the energy $E_{\text{orient}}(C_{12}) = \left\| C_{12}\Omega_{f_1} + \Omega_{f_2}C_{12} \right\|_F^2$, which regularizes the functional map $C_{12}$ to reverse the symmetry that is encoded in the descriptors. We can apply our discrete solver to minimize this energy to obtain symmetric map. We first construct the relaxed energy:$E_{\text{orient}}^{\text{rel}} = \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 \Omega_{f_1} + \Omega_{f_2} C_{12} \right\|_F^2 + \alpha \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{12}^T - Id \right\|_F^2$. Then we have the updating rules for the pointwise map $\Pi_{21} = \text{knn}\Big(\Phi_1 \left[\Omega_{f_1}, \alpha C_{12}^T\right], \left[-\Phi_2 \Omega_{f_2} C_{12}, \alpha \Phi_2\right]\Big)$.In Fig. 3 we show an example of computing the self-symmetric map on a panda shape from one WKS descriptor by minimizing the orientation-reversing energy.

**Multiplicative & Orientation-Preserving Energy** Similarly, we can construct the multiplicative operator or the orientation-preserving operator $\Omega_{f_i}$ from input descriptors $(f_1, f_2)$ on shape $S_i$ respectively. We then have the operator commutativity energy as $E(C_{12}) = \left\| C_{12}\Omega_{f_1} - \Omega_{f_2} C_{12} \right\|_F^2$, with the relaxed energy $E_{\text{op}}^{\text{rel}} = \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 \Omega_{f_1} - \Omega_{f_2} C_{12} \right\|_F^2 + \alpha \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{12}^T - Id \right\|_F^2$. The corresponding update rule for the pointwise map is: $\Pi_{21} = \text{knn}\Big(\Phi_1 \left[\Omega_{f_1}, \alpha C_{12}^T\right], \left[\Phi_2 \Omega_{f_2} C_{12}, \alpha \Phi_2\right]\Big)$. Fig. 17 shows the pointwise maps computed by our discrete solver and the continuous solver on minimizing the energy proposed in [NO17] that involves the multiplicative operator commutativity energy, and [RPWO18] that involves the orientation-preserving energy.

**Laplacian Commutativity** Laplacian Commutativity is one of the most popular functional map energies that can promote map isometry. We can get the updating rules for the pointwise maps easily by replace the operator $\Omega_{f_i}$ by the Laplacian operator $\Delta_{S_i}$ directly: $\Pi_{21} = \text{knn}\Big(\Phi_1 \left[\Delta_1, \alpha C_{12}^T\right], \left[\Phi_2 \Delta_2 C_{12}, \alpha \Phi_2\right]\Big)$. In Fig. 18, we compare our solution to the approximate solution on the synthetic shape pair with $n = 6$ vertices. Specifically, we start with 50 different random initializations, and compare our discrete solver to the approximate solution on minimizing the Laplacian commutativity.

**Orthogonality: Area Preservation**

Another popular functional map energy is the orthogonality energy, which promotes the area-preserving property of the underlying map: $E_{\text{ortho}}(C_{12}) = \left\| C_{12}^T C_{12} - Id \right\|_F^2 = \left\| C_{12} C_{12}^T - Id \right\|_F^2$. We can construct the relaxed energy as: $E_{\text{ortho}}^{\text{relax}} = (1+\alpha) \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{12}^T - Id \right\|_F^2$. Then we have the updating rules for our discrete solver (1) $\Pi_{21} = \text{knn}(\Phi_1 C_{12}^T, \Phi_2)$ (2) $C_{12} = \Phi_2^\dagger \Pi_{21} \Phi_1$. In table 1, starting with the same *random* initialization, we compare our discrete solver to the two baselines, the exact solution and the continuous solver, on

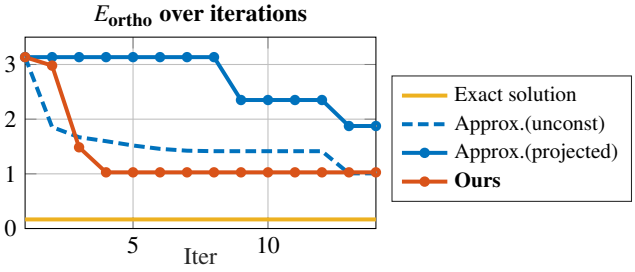


**Figure 19:** *For a pair of heart shapes with 6 vertices, we compare our method (red solid line) to the approximate solution (blue solid line) over iterations. The global minimum, i.e., the exact solution is demonstrated by the yellow line. The blue dashed line shows the orthogonality error of the intermediate functional maps during the unconstrained optimization. We can see that our method converges faster and achieves better result than the approximate solution.*

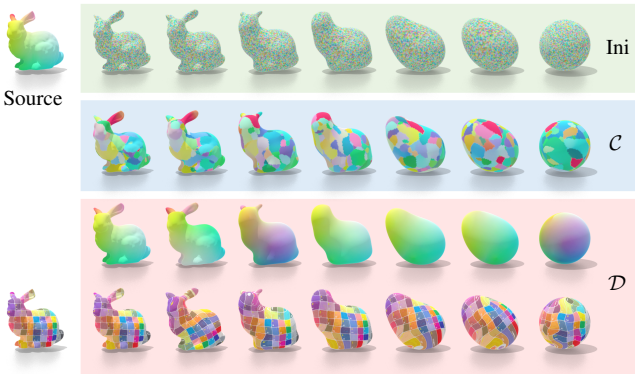


**Figure 20:** *Minimizing conformality using the continuous solver $\mathcal{C}$ and our discrete solver $\mathcal{D}$ on deformed bunny pairs from random initializations.*

synthetic shapes to minimize the orthogonality energy. In Fig. 19 we visualize the optimization process of minimizing the orthogonality over iterations on the shape pair with 6 vertices. Specifically, the exact solution is found among $6^6 \approx 46K$ proper functional maps (color in yellow). We then find the approximate solution by minimizing the orthogonality error in an unconstrained setting from a random proper functional map, and the optimizing process is demonstrated by the blue dashed line. Note that, these intermediate functional maps are not proper. To satisfy the hard constraint, we project these functional maps into the proper functional map space $\mathcal{P}$ (blue solid line). We can see that there is a big gap between the continuous optimization and the corresponding discrete conversion. As a comparison, our discrete solver (red solid line) can quickly converge to a better local minimum (500 times faster than the exact solution), and each intermediate functional map is proper by construction.

### Conformality

As discussed in Sec. 5.3, we reformulate the conformality energy from $E(C_{12}) = \left\| C_{12}^T \Delta_1 C_{12} - \Delta_2 \right\|_F^2$ to $E^{\text{mod}} = \left\| C_{12} \Delta_2 C_{12}^T - \Delta_1 \right\|_F^2$. This reformulation can help to simplify the optimization of the pointwise map. Specifically, if we construct a relaxed energy for the original conformality energy directly, we need to replace one of the $C_{12}$ by $\Phi_2^\dagger \Pi_{21} \Phi_1$ and add the soft regularizer. However, in this particular case, no matter which one of the two $C_{12}$ besides $\Delta_2$ gets replaced, the resulting problem on $\Pi_{21}$ is not easy to solve (i.e., we do not have close-form solution of the minimizer $\Pi_{21}$). As a comparison, after reformulation, the update rule for the pointwise map $\Pi_{21}$ has close-form solution and is efficient to obtain. In Fig. 20, we compare our discrete solver on minimizing the *modified* conformality to the continuous solver on minimizing the *original* conformality.

### Bijectivity

We remark that our approach can also be applied to the setting that involves multiple maps such as enforcing bijectivity of the correspondences. Specifically, we consider the bijectivity of the functional maps $C_{12}$ and $C_{21}$ from both directions: $E_{bi}(C_{12}, C_{21}) = \left\| C_{12} C_{21} - Id \right\|_F^2 + \left\| C_{21} C_{12} - Id \right\|_F^2$. The goal is to minimize the bijectivity error with the constraint that both $C_{12}$ and $C_{21}$ are proper functional maps: $\min_{C_{12} \in \mathcal{P}_{12}, C_{21} \in \mathcal{P}_{21}} E_{\text{bi}}(C_{12}, C_{21})$. We can similarly construct the relaxed energy:

$$\begin{aligned} E_{\text{bi}}^{\text{rel}} = &\left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{21} - Id \right\|_F^2 + \alpha \left\| \Phi_2^\dagger \Pi_{21} \Phi_1 C_{12}^T - Id \right\|_F^2 \\ &+ \left\| \Phi_1^\dagger \Pi_{12} \Phi_2 C_{12} - Id \right\|_F^2 + \alpha \left\| \Phi_1^\dagger \Pi_{12} \Phi_2 C_{21}^T - Id \right\|_F^2 \end{aligned} \tag{15}$$

where we add two soft regularizers to address the hard constraints. We then have the following bidirectional updating rules:

$$\begin{aligned} \Pi_{12} &= \arg\min_{\Pi_{12}} \left\| \Pi_{12} \Phi_2 \left[ C_{12}, \alpha C_{21}^T \right] - \left[ \Phi_1, \alpha \Phi_1 \right] \right\|_F^2 \\ \Pi_{21} &= \arg\min_{\Pi_{21}} \left\| \Pi_{21} \Phi_1 \left[ C_{21}, \alpha C_{12}^T \right] - \left[ \Phi_2, \alpha \Phi_2 \right] \right\|_F^2 \\ C_{12} &= \Phi_2^\dagger \Pi_{21} \Phi_1, \quad C_{21} = \Phi_1^\dagger \Pi_{12} \Phi_2 \end{aligned} \tag{16}$$

Fig. 11 shows an example of optimizing the bijectivity error from random initialization using our discrete solver in two different settings (1) applying the updating rules in fixed basis with size $k$; (2) applying the updating rules with the progressive upsampling technique. In both settings, our discrete solver outperforms the continuous solver.

### Summary

Our method can be applied to different types of functional map based energies. We demonstrate the superior accuracy, convergence power and efficiency of our discrete solver over the the approximate solution using continuous solver. We illustrate that the relaxed energy is trivial to construct, and the corresponding discrete solver is both efficient and effective with close-form solution. Also note that, the ZoomOut [MRR*19] and the bidirectional ZoomOut [RMOW20] method can be regarded as a special case of the algorithm with parameters $N = 1$. In our work, we extend this idea to larger set of functional map based energies.