

# Fast Sinkhorn Filters: Using Matrix Scaling for Non-Rigid Shape Correspondence with Functional Maps - Supplementary Material

Gautam Pai<sup>1</sup>    Jing Ren<sup>2</sup>    Simone Melzi<sup>3</sup>    Peter Wonka<sup>2</sup>    Maks Ovsjanikov<sup>1</sup>  
<sup>1</sup>LIX, Ecole Polytechnique, IP Paris    <sup>2</sup>KAUST    <sup>3</sup>Sapienza University of Rome

In this supplementary material, we provide additional theoretical insights and qualitative examples that could not be fitted into the main paper due of lack of space. Specifically, we organize the material as follows. First in Section 1, we provide the exact definitions of the geometric and functional metrics that we have used to evaluate the results in Tables 1 and 2 of the main paper. In section 2, we show qualitative examples associated to Table 2, in which we visualize the obtained maps from different methods via color transfer, error highlighting, and texture transfer (See Fig. 1) on a pair of shapes from the remeshed SCAPE [6] dataset. We further illustrate the effectiveness of our fast sinkhorn filter on two pairs of *non-isometric* shapes in Figures 2 and 3, that complement the results of Table 1 of the main paper. Furthermore, we also provide a short analysis on the effect of number of sinkhorn iterations in comparison to the nearest neighbor baseline in Fig. 4.

In Section 3, we have included additional complementary discussions to our main theory of regularized spectral embedding alignment. In Section 4, we further generalize our iterative meta algorithm (IMA) (Algorithm 1 in the main paper) with different choices of search spaces, such that the existing work, including PMF [10] and VarMin [3] also fit in this framework. Finally in section 5, we prove that the energy of PMF is actually equivalent to the energy of VarMin thereby justifying the formulation of our fast sinkhorn filters from a different perspective.

## 1. Geometric and Functional Metrics

Given two shapes  $S_1$  and  $S_2$  we evaluate the forward ( $T_{12} : S_1 \rightarrow S_2$ ) and backward ( $T_{21} : S_2 \rightarrow S_1$ ) maps respectively. We first define the *geometric metrics* of the pointwise maps as follows.

**Accuracy** We measure the geodesic distance between  $T_{12}$  and the given ground-truth correspondence for a given shape pair.

**Bijectivity** We construct the composite map  $T_{21} \circ T_{12}$  which is therefore a map from the shape  $S_1$  to itself. We measure the geodesic distance between this composite map and the identity map.

**Coverage** For a given map  $T_{12}$  is the ratio between the area of the covered vertices (i.e. the indices indexed by the map  $T_{12}$ ) and the total surface area of the target shape.

**Smoothness** We evaluate the Dirichlet energy of the transferred coordinate functions as a smoothness measure:

$$E(T_{12}) = \sum_{(u,v) \in \mathcal{E}_1} w_{uv} \|\phi_{12}(u) - \phi_{12}(v)\|_2^2 \quad (1)$$

where  $\phi_{12} = T_{12}^{-1} \circ X_2$ , and  $X_2 \in \mathbb{R}^{n_2 \times 3}$  are the coordinate functions of the shape  $S_2$ ,  $w_{uv}$  are the stiffness weights of the cotangent Laplacian for  $S_1$ .

In addition, the following functional metrics are commonly used to evaluate given *functional* maps  $C$ :

**Orthogonality** As shown in [5], the functional map between an isometric shape pair is supposed to be orthogonal. Therefore we can measure the orthogonality of a given functional map as  $\|C^T C - \mathbf{I}\|^2$ .

**Laplacian Commutativity** Let  $\Lambda_i$  be discrete Laplacian for shape  $S_i$ . The Laplacian commutativity for a functional map  $C$  is defined as:

$$\|C\|_L^2 = \frac{\|C\Lambda_1 - \Lambda_2 C\|^2}{\|C\|^2} \quad (2)$$

**ZoomOut Energy** It measures the orthonormality of each principal sub-matrix of the functional map  $C$  [4].

$$\|C\|_{ZO}^2 = \sum_k \frac{1}{k} \|C_k C_k^T - \mathbf{I}_k\|_F^2 \quad (3)$$

Here  $C_k$  is the  $k^{\text{th}}$  principal submatrix of  $C$ .

**Spectral Chamfer Distance** Let  $\mathcal{F}_1 = \Phi_1 D_{21}^T \in \mathbb{R}^{n_1 \times k}$  and  $\mathcal{F}_2 = \Phi_2 \in \mathbb{R}^{n_2 \times k}$  be the aligned basis from the spectral alignment procedure outlined in the paper and hence treated as point clouds in  $k$  dimensional space. The Chamfer distance of this embedding is defined as:

$$\begin{aligned} \text{CD}(\mathcal{F}_1, \mathcal{F}_2) = & \frac{1}{n_1} \sum_{x \in \mathcal{F}_1} \min_{y \in \mathcal{F}_2} \|x - y\|_2^2 \\ & + \frac{1}{n_2} \sum_{y \in \mathcal{F}_2} \min_{x \in \mathcal{F}_1} \|x - y\|_2^2 \end{aligned} \quad (4)$$

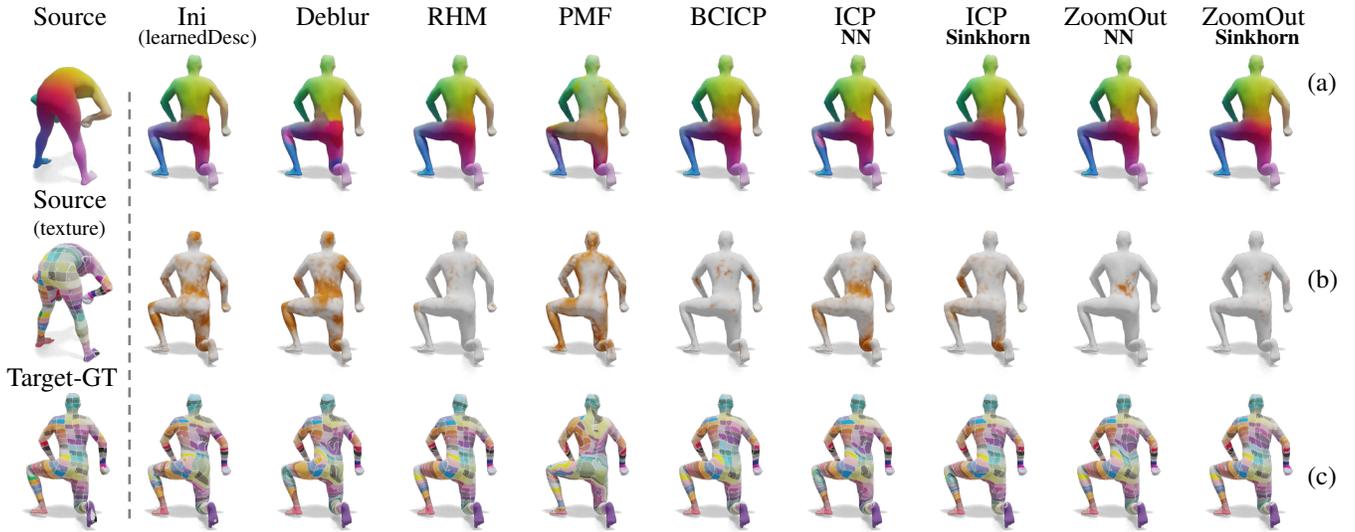


Figure 1. An example shape pair from the remeshed SCAPE dataset. (a) visualize the computed maps via color transfer (b) highlight the map errors by red (c) visualize the computed maps via texture transfer.

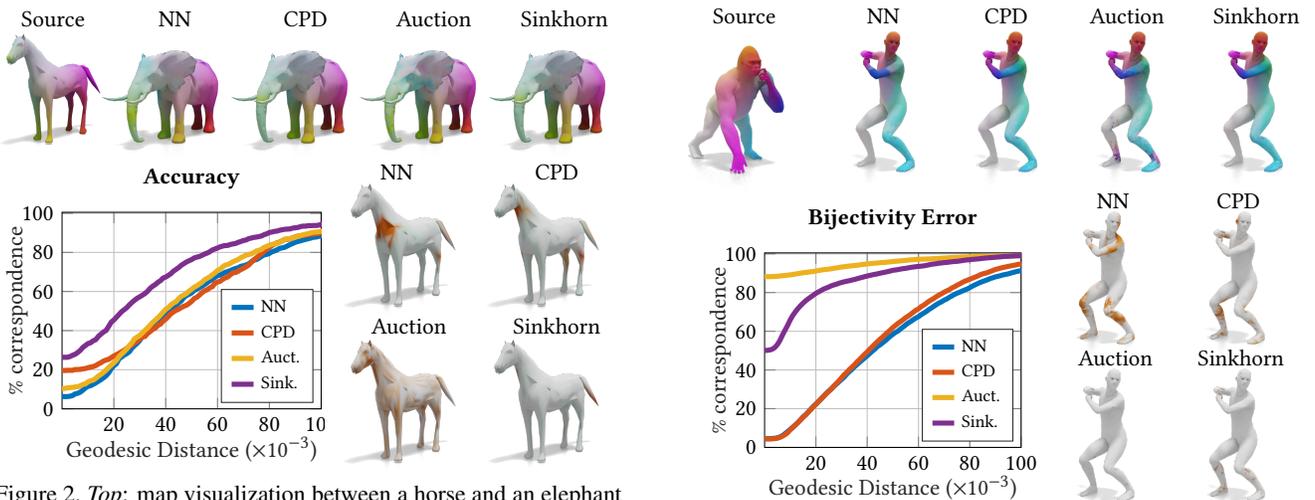


Figure 2. *Top*: map visualization between a horse and an elephant via color transfer. *Bottom*: on the left we show the accuracy summary of the four methods, and on the right we visualize the ground truth error on the horse shapes. Red corresponds to large and white corresponds to smaller errors.

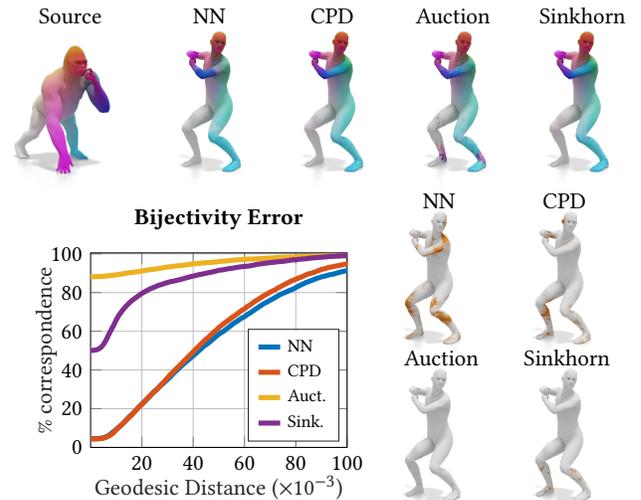


Figure 3. *Top*: map visualization between a gorilla and a human shape via color transfer. *Bottom*: on the left we show the bijection error summary of the four methods, and on the right we visualize the bijection error on the man-shape. Red corresponds to large and white corresponds to smaller errors.

## 2. Additional Results

### 2.1. Qualitative Evaluation

In Fig. 1, we show a pair of human shapes from the remeshed SCAPE dataset [6], corresponding to the quantitative results of Table 2 in the main paper. The last two columns visualize the maps corresponding to the *sinkhornized* versions of ICP[5] and Zoomout[4] over the original ones. We can see that the addition of the sinkhorn filter into the iterative schemes lead to lesser errors (visually evidenced by the reduced error profile at the middle-back of the man) as well as a smoother result, visually apparent in the texture transfer in the bottom row.

Figures 2 and 3 visualize *one step* of pointwise conversion from a functional map on a non-isometric example. All algorithms were imputed with the *same* aligned spectral basis and then registered to a pointwise correspondence using the different methods (identical to the setup in Table 1 of the main paper). Figure 2 focuses on the accuracy with respect to the ground truth and Figure 3 on the bijection properties of different algorithms. As evidenced in Figure 2, the sinkhorn algorithm shows an improved accuracy in comparison to the others for this non-isometric shape pair with 1.2k points per shape. In Figure 3, we can see that even though the auction algorithm has a lower bijection error

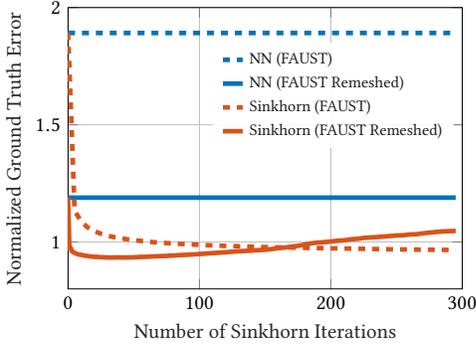


Figure 4. **Sinkhorn Filtering:** Effect of number of Sinkhorn iterations on the mean ground truth error of the original and remeshed versions of the FAUST dataset. A few iterations ( $\sim 10$ ) show a decrease the error in comparison to the nearest neighbor solution, which is equivalent to no Sinkhorn filtering (0 iterations).

(since it approximately solves for a permutation), it shows a significantly poor accuracy (highlighted by the considerable map distortion on the legs of the man). In contrast to these shortcomings, and re-emphasizing the main result in Table 1 of the main paper, our fast Sinkhorn filter demonstrates higher accuracy *and* bijectivity with a significantly smaller runtime: NN: 0.54, CPD: 18.73, Auction: 16.9, Sinkhorn: 0.95, all in seconds. Each shape has around 8k points with different connectivity.

## 2.2. Analysis of the Fast Sinkhorn Filter

We study the effect of number of iterations of the Sinkhorn procedure on the quality of the map on the FAUST and FAUST remeshed datasets. See Figure 4. Firstly, we remark that the output of the fast Sinkhorn procedure with 0 iterations yields the same solution as a nearest neighbor approach. The matrix scaling procedure outlined in our Sinkhorn filter has a *filtering* effect on the nearest neighbor map, by injecting bijectivity and as a result improving the overall map accuracy. Secondly, only a few iterations of the filter are enough to yield a significant gain in the accuracy. An excess filtering with a large number of iterations could lead to a degradation of performance especially in scenarios where the sampling and connectivity are not identical. Relating to our earlier discussion about regularization for pointwise map recovery, the number of Sinkhorn iterations intuitively *control* the extent of bijective regularization imposed onto the map.

## 2.3. Runtime analysis

In Fig. 5, we make a runtime analysis of the three prominent linear-assignment registration schemes: nearest neighbor, auction and our fast sinkhorn method. In contrast to an exponential growth in runtime as seen using the auction, both the nearest neighbor and our fast sinkhorn method show a linear growth with respect to the mesh resolution.

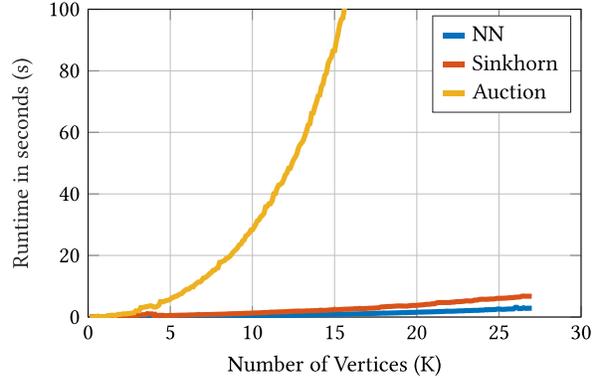


Figure 5. Runtime comparison w.r.t. mesh resolution

This further illustrates the advantages of our method that can achieve superior accuracy and bijectivity while maintaining a modest computational complexity.

## 3. Regularized Spectral Alignment

Here we will provide more details about the theoretical background of the problem of regularized spectral alignment, which is complementary to Sec. 3 in the main paper. Recall that in Sec. 3, we first introduced the definition of the functional maps and adjoint operators in the smooth setting. We also proved that the adjoint operators always map Dirac deltas to Dirac deltas (see Theorem 1). We then discussed how to discretize these two operators in discrete settings with full or reduced basis, where Theorem 1 holds exactly or approximately respectively.

### 3.1. Why use the adjoint for map conversion?

The original functional map paper [5] discussed a possible way to recover a pointwise map  $T_{\mathcal{X}\mathcal{Y}}$  from the functional map  $T_{\mathcal{X}\mathcal{Y}}^{\mathcal{F}}$  (see Remark 4.1 in [5]), that is to use the functional map to map *indicator functions*, i.e., functions that equal 1 at some point and zero elsewhere. Unfortunately, this has a major problem: in  $L^2$ , such indicator functions are equivalent to the zero function almost everywhere. This means that in an orthonormal basis, such as the Laplace-Beltrami eigenfunctions, such functions will be represented as vectors of zeros. As a result, we cannot apply such a method in practice directly.

A more principled approach can be obtained by using the adjoint operators as justified by Theorem 1. Note that the adjoint operator of functional maps has been considered, e.g., in [2] although its role in pointwise map conversion was not explicitly addressed in that work. Specifically, instead of transporting ill-defined indicator functions, we can use the adjoint operator to map well-defined Dirac delta functions in  $L^2$ , which is guaranteed to get Dirac deltas after mapping.

---

**Algorithm 1: IMA variant: ICP**

---

**Input** : A pair of shapes  $\mathcal{X}, \mathcal{Y}$  with basis  $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$   
**Initialization** : An initial guess of  $\mathbf{C}_{\mathcal{Y}\mathcal{X}}$   
**while** *Not converged* **do**  
    (1)  $T_{\mathcal{X}\mathcal{Y}} = \text{NNsearch}(\Phi_{\mathcal{Y}}\mathbf{C}_{\mathcal{Y}\mathcal{X}}^T, \Phi_{\mathcal{X}})$   
    (2)  $\mathbf{C}_{\mathcal{Y}\mathcal{X}} = \Phi_{\mathcal{X}}^\dagger \Pi_{\mathcal{Y}\mathcal{X}} \Phi_{\mathcal{Y}}$   
    (3) Project all singular values of  $\mathbf{C}_{\mathcal{Y}\mathcal{X}}$  to 1  
**end**

---

### 3.2. Another derivation for Eq. (5)

We could equivalently derive Eq. (5) in the main paper using the definition of the adjoint operators directly. The adjoint operators in the smooth setting is defined as

$$\langle T_{\mathcal{Y}\mathcal{X}}^{\mathcal{F}}(f), g \rangle_{\mathcal{X}} = \langle T_{\mathcal{X}\mathcal{Y}}^{\mathcal{A}}(g), f \rangle_{\mathcal{Y}}.$$

In the reduced basis it would read:

$$\langle \mathbf{C}_{\mathcal{Y}\mathcal{X}} \mathbf{c}_{\mathbf{f}}, \mathbf{c}_{\mathbf{g}} \rangle_{\mathcal{X}} = \langle \mathbf{D}_{\mathcal{X}\mathcal{Y}} \mathbf{c}_{\mathbf{g}}, \mathbf{c}_{\mathbf{f}} \rangle_{\mathcal{Y}},$$

where  $\mathbf{c}_{\mathbf{f}}(\mathbf{c}_{\mathbf{g}})$  is the discrete representation of  $f(g)$  in the reduced basis  $\Phi_{\mathcal{Y}}(\Phi_{\mathcal{X}})$ . Therefore, we know there exists  $\mathbf{c}_1(\mathbf{c}_2)$  such that  $\mathbf{c}_{\mathbf{f}} = \Phi_{\mathcal{Y}}\mathbf{c}_1$  ( $\mathbf{c}_{\mathbf{g}} = \Phi_{\mathcal{X}}\mathbf{c}_2$ ). Since in the reduced basis inner products of functions simply correspond to standard inner products, we have

$$\begin{aligned} \text{LHS} &= \langle \mathbf{C}_{\mathcal{Y}\mathcal{X}} \mathbf{c}_{\mathbf{f}}, \mathbf{c}_{\mathbf{g}} \rangle_{\mathcal{X}} = \mathbf{c}_{\mathbf{g}}^T \mathbf{C}_{\mathcal{Y}\mathcal{X}} \mathbf{c}_{\mathbf{f}} \\ &= (\Phi_{\mathcal{X}}\mathbf{c}_2)^T \mathbf{C}_{\mathcal{Y}\mathcal{X}} \Phi_{\mathcal{Y}}\mathbf{c}_1 = \mathbf{c}_2^T \Phi_{\mathcal{X}}^T \mathbf{C}_{\mathcal{Y}\mathcal{X}} \Phi_{\mathcal{Y}}\mathbf{c}_1 \\ \text{RHS} &= \langle \mathbf{D}_{\mathcal{X}\mathcal{Y}} \mathbf{c}_{\mathbf{g}}, \mathbf{c}_{\mathbf{f}} \rangle_{\mathcal{Y}} = \mathbf{c}_1^T \Phi_{\mathcal{Y}}^T \mathbf{D}_{\mathcal{X}\mathcal{Y}} \Phi_{\mathcal{X}}\mathbf{c}_2 \\ \text{LHS} &= \text{RHS} \Rightarrow \mathbf{D}_{\mathcal{X}\mathcal{Y}} = \mathbf{C}_{\mathcal{Y}\mathcal{X}}^T \end{aligned} \quad (5)$$

### 3.3. Additional discussion on IMA

The iterative meta algorithm (IMA) outlined in Sec. 3.4 in the main paper performs iterative alignment between the spectral embeddings given some initial guess. It is easy to see that this method is guaranteed to converge to some (possibly local) minimum. The reason is that at every iteration (step 1-2 in IMA Algorithm), the overall distance between spectral embeddings is guaranteed to not increase, since  $\mathbf{D}_{\mathcal{X}\mathcal{Y}}$  is the *optimal* linear transformation that aligns the embeddings given the point-to-point map, and  $T_{\mathcal{X}\mathcal{Y}}$  is an *optimal* point-to-point map given a linear functional transformation. Algorithms 1, 2 and 3, provide a pseudocode of the meta algorithm procedure corresponding to the iterative algorithms: ICP, PMF and ZoomOut, emphasizing the underlying common principle of the meta algorithm in their action.

## 4. Pointwise Conversion as Linear Assignment

### 4.1. Search Spaces for Pointwise Maps

As suggested by IMA (Algorithm 1 in the main paper), in order to minimize the spectral embedding alignment

---

**Algorithm 2: IMA variant: PMF**

---

**Input** : A pair of shapes  $\mathcal{X}, \mathcal{Y}$  with basis  $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$   
**Initialization** : Initial pointwise map  $\Pi_{\mathcal{Y}\mathcal{X}}$   
**while** *Not converged* **do**  
    (1) set  $\mathbf{X} = \Phi_{\mathcal{X}}\Phi_{\mathcal{X}}^\dagger \Pi_{\mathcal{Y}\mathcal{X}}\Phi_{\mathcal{Y}}\Phi_{\mathcal{Y}}^\dagger$   
    (2) compute  $\Pi_{\mathcal{Y}\mathcal{X}} = \arg \min_{\Pi \text{ is bijective}} \|\Pi - \mathbf{X}\|$   
    (3)  $k_{\mathcal{X}} \leftarrow k_{\mathcal{X}} + 1, k_{\mathcal{Y}} \leftarrow k_{\mathcal{Y}} + 1$   
**end**

---

---

**Algorithm 3: IMA variant: ZoomOut**

---

**Input** : A pair of shapes  $\mathcal{X}, \mathcal{Y}$  with basis  $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$   
**Initialization** : Initial pointwise map  $T_{\mathcal{X}\mathcal{Y}}$   
**while** *Not converged* **do**  
    (1)  $\mathbf{C}_{\mathcal{Y}\mathcal{X}} = \Phi_{\mathcal{X}}^\dagger \Pi_{\mathcal{Y}\mathcal{X}} \Phi_{\mathcal{Y}}$   
    (2)  $T_{\mathcal{X}\mathcal{Y}} = \text{NNsearch}(\Phi_{\mathcal{Y}}\mathbf{C}_{\mathcal{Y}\mathcal{X}}^T, \Phi_{\mathcal{X}})$   
    (3)  $k_{\mathcal{X}} \leftarrow k_{\mathcal{X}} + 1, k_{\mathcal{Y}} \leftarrow k_{\mathcal{Y}} + 1$   
**end**

---

error, we can iterate between extracting a pointwise map from a functional/adjoint map, and then estimate the functional/adjoint map from the obtained pointwise maps. As shown in section 3.4 of the main paper, the *adjoint* operator is the optimal linear operator that aligns the spectral embeddings of two shapes given a pointwise correspondence between them.

The spectral embedding alignment error is given by:

$$\Pi_{\mathcal{Y}\mathcal{X}} = \arg \min_{\Pi \in \mathcal{Q}} \|\Phi_{\mathcal{X}}\mathbf{D}_{\mathcal{X}\mathcal{Y}}^T - \Pi\Phi_{\mathcal{Y}}\| \quad (6)$$

where  $\mathcal{Q}$  is the regularized search space of the pointwise map  $\Pi_{\mathcal{Y}\mathcal{X}}$ . We remark that the spectral embedding alignment error given by equation 6 can be reformulated as a *linear assignment* cost:

$$\Pi_{\mathcal{Y}\mathcal{X}} = \arg \min_{\Pi \in \mathcal{Q}} \langle d, \Pi \rangle_F \quad (7)$$

where  $\langle \cdot, \cdot \rangle_F$  is the matrix Frobenius inner product, defined as  $\langle A, B \rangle_F = \sum_i \sum_j A_{ij}B_{ij}$  for matrix  $A, B$  with the same size, and  $d$  is a cost matrix with size  $n_{\mathcal{X}} \times n_{\mathcal{Y}}$  which is defined as  $d = [d_{ij}]_{i=1, \dots, n_{\mathcal{X}}, j=1, \dots, n_{\mathcal{Y}}}$  where

$$d_{ij} = \|\Phi_{\mathcal{X}}(i, :)D_{\mathcal{X}\mathcal{Y}}^T - \Phi_{\mathcal{Y}}(j, :)\| \quad (8)$$

Therefore, different algorithms for pointwise conversion (like nearest neighbor search, auction etc) can be viewed as different *search spaces*  $\mathcal{Q}$  for the linear assignment problem of equation 7 which we enumerate below:

**Vertex-to-Vertex maps** This is the simplest setting where we only enforce that each vertex on the source shape can be mapped to a single vertex on the target:

$$\mathcal{Q}_1 = \left\{ \Pi \mid \Pi \in \{0, 1\}^{n_{\mathcal{X}} \times n_{\mathcal{Y}}}, \Pi \mathbf{1}_{n_{\mathcal{Y}}} = \mathbf{1}_{n_{\mathcal{X}}} \right\}$$

where  $\mathbb{1}_n$  is a vector with size  $n$  where all the entries are 1. For this type of search space, a pointwise map can be typically solved by a nearest-neighbor search.

**Permutations** When the two shapes  $\mathcal{X}, \mathcal{Y}$  have the same resolution, we can naturally enforce the pointwise map to be bijective. i.e.,  $\Pi$  is a permutation matrix:

$$\mathcal{Q}_2 = \left\{ \Pi \mid \Pi \in \{0, 1\}^{n \times n}, \Pi \mathbb{1}_n = \mathbb{1}_n, \Pi^T \mathbb{1}_n = \mathbb{1}_n \right\}$$

where  $n = n_{\mathcal{X}} = n_{\mathcal{Y}}$  by assumption. Usually, such a search space is computationally intensive to implement in practice, and typically approximation algorithms like the auction [1] are employed.

**Transport plans** We can also enforce  $\Pi$  to be a transport plan that maps the distribution  $\mu_{\mathcal{Y}}$  on shape  $\mathcal{Y}$  to the distribution  $\mu_{\mathcal{X}}$  on shape  $\mathcal{X}$

$$\mathcal{Q}_3 = \left\{ \Pi \mid \Pi \in \mathbb{R}_{\geq 0}^{n_{\mathcal{X}} \times n_{\mathcal{Y}}}, \Pi \mathbb{1}_{n_{\mathcal{Y}}} = \mu_{\mathcal{X}}, \Pi^T \mathbb{1}_{n_{\mathcal{X}}} = \mu_{\mathcal{Y}} \right\}$$

where  $\mu_{\mathcal{X}}$  and  $\mu_{\mathcal{Y}}$  are initial masses for  $\mathcal{X}$  and  $\mathcal{Y}$  predefined on each vertex.

## 5. Additional Links to Spectral Alignment

In this section we provide an insight into another prominent correspondence method: [3], which can be seen under the general principle of spectral alignment promoted in this paper. Similar in spirit to [7, 8], correspondence is modelled by a probabilistic mixture of gaussians, in which the variance is iteratively refined in order to localize the uncertainty and achieve dense correspondence.

In order to prove a link between the optimization cost in [3] and the Iterative Meta Algorithm of this paper, we establish two steps: First we argue that the form of Eq. (7) is similar to the iterations of PMF [9, 10] (PMF is a special case of the IMA as shown earlier). Secondly, the cost function of the variance minimizing transport plan in [3] can be written in terms of Eq. (7) by modelling a different cost matrix  $d$ .

**Variance-Minimization Optimal Transport (VarMin)** Given some initial correspondences (named "the projected centers" in the paper), the goal is to *minimize* the sum of *local variances* defined on all the vertices. The energy (defined in equation (3) in [3]) is:

$$\begin{aligned} C(\pi, \eta) = & \sum_{i=1}^{n_1} \mathbf{m}_i \text{var} \left[ \pi_{\mathbf{X}} \left( \frac{\mathbf{W}_{\mathbf{x}_i \mu}}{\sum_k \mathbf{m}_k \mathbf{W}_{\mathbf{x}_i}(\mathbf{x}_k)} \right), \eta_{\mathbf{x}_i} \right] + \\ & \sum_{j=1}^{n_2} \mathbf{n}_j \text{var} \left[ \pi_{\mathbf{Y}} \left( \frac{\mathbf{W}_{\mathbf{y}_j \nu}}{\sum_k \mathbf{n}_k \mathbf{W}_{\mathbf{y}_j}(\mathbf{y}_k)} \right), \eta_{\mathbf{y}_j} \right] \end{aligned} \quad (9)$$

where  $\pi_{\mathbf{X}}, \pi_{\mathbf{Y}}$  are transport plans from the opposite directions,  $\mathbf{m}, \mathbf{n}$  are initial mass defined on the two shapes, the weighting function  $\mathbf{W}_{\mathbf{x}_i}, \mathbf{W}_{\mathbf{y}_j}$  are defined in Eq.(3) and

discussed in detail in Section 3.2. in [3]. The goal is to solve a transport plan that minimize the transport cost that defined from the local variance:

$$\min_{\eta, \pi \in \mathcal{Q}_3} E(\pi, \eta) = \langle \mathbf{C}(\pi, \eta), \pi \rangle_F \quad (10)$$

Let  $\mathbf{M}^i$  denote the distance matrix, that stores the pairwise squared distances on shape  $i$ . With the definition of  $\text{var}(\cdot)$  (Eq. (2) in [3]), it is easy to check that the *cost matrix* for VarMin equals to:

$$\begin{aligned} \mathbf{C}(\pi, \eta) = & \sum_i \sum_j \mathbf{m}_i \left( \mathbf{W}^1 \text{diag}(\mathbf{a}) \mathbf{M}^2 \right)_{ij} \pi_{ij} + \\ & \sum_i \sum_j \mathbf{n}_j \left( \mathbf{M}^1 \text{diag}(\mathbf{b}) \mathbf{W}^2 \right)_{ij} \pi_{ij} \\ = & \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (\mathbf{C}_{ij}^1 + \mathbf{C}_{ij}^2) \pi_{ij} \\ = & \langle \mathbf{C}^1 + \mathbf{C}^2, \pi \rangle_F \end{aligned} \quad (11)$$

where  $\mathbf{W}^i$  is a weighting matrix that is supposed to be negatively proportional to the distance matrix, and

$$\begin{aligned} \mathbf{a}_i = & \frac{\mathbf{m}_i}{\left( \sum_k \mathbf{m}_k \mathbf{W}_{ik}^1 \right) \left( \sum_j \pi_{ij} \right)} \in \mathbb{R}^{n_1}, i = 1, \dots, n_1 \\ \mathbf{b}_j = & \frac{\mathbf{n}_j}{\left( \sum_k \mathbf{n}_k \mathbf{W}_{jk}^2 \right) \left( \sum_i \pi_{ij} \right)} \in \mathbb{R}^{n_2}, j = 1, \dots, n_2 \\ \mathbf{C}^1 = & \text{diag}(\mathbf{m}) \mathbf{W}^1 \text{diag}(\mathbf{a}) \mathbf{M}^2 \in \mathbb{R}^{n_1 \times n_2} \\ \mathbf{C}^2 = & \mathbf{M}^1 \text{diag}(\mathbf{b}) \mathbf{W}^2 \text{diag}(\mathbf{n}) \in \mathbb{R}^{n_1 \times n_2} \end{aligned} \quad (12)$$

**Product Manifold Filter** The goal is to find a permutation matrix  $\Pi \in \mathbb{R}^{n \times n}$  (with the assumption that two meshes have the same resolution, i.e.,  $n_1 = n_2 = n$ ), such that the *form* does not change too much after being smoothed by the diffusion kernels. Therefore, we aim to *maximize* the *similarity* before and after the diffusion, which is defined as  $\langle \mathbf{K}^1 \Pi \mathbf{K}^2, \Pi \rangle_F$ . Therefore, the goal is to maximize:

$$\max_{\Pi \in \mathcal{Q}_2} \langle \mathbf{K}^1 \Pi \mathbf{K}^2, \Pi \rangle_F \quad (13)$$

**Equivalence between VarMin and PMF** We claim that the two energies, Eq. (10) and (13) are related to each other. More specifically, the two problems are *equivalent* to each other (up to first order) under the following assumptions: (1) The input two shapes have the same resolution. (2) Two problems are initialized with the same set of correspondences. (3) Two problems are solved in the same search space. Recall that the original search space for VarMin is  $\mathcal{Q}_3$  and for PMF is  $\mathcal{Q}_2$ . (4) The weighting matrix and distance matrix in VarMin are consistent with the diffusion kernel in PMF. (5) The VarMin adopts unweighted local variances.

Note that, for the assumption (4), in general, for any distance measure, there exists a corresponding weighting matrix in VarMin setting, and a corresponding kernel in PMF setting. Therefore, the two problems are equivalent w.r.t.

the choices of the distance measure, weighting function and the kernel. Here we only show the equivalence when both formulations use *geodesic distances*, whereas the case of using other distance measure such as diffusion distance can be similarly proved.

**Proof of the equivalence** With assumption (3) as discussed above, where two problems are solved in the same search space, to prove the equivalence, it is sufficient to show the energy functions have the same characterizations. Moreover, since two algorithms are designed in an alternating fashion, given assumption (2), we only need to show that the two problems are equivalent with fixed projected centers  $\eta$  for VarMin and fixed correspondences  $\mathbf{P}_{12}$  and  $\mathbf{P}_{21}$  for PMF. Specifically we need to prove that the following two problems have the same optimizer:

$$\begin{aligned} \pi^* &= \arg \min_{\pi \in \mathcal{Q}} \langle \mathbf{C}(\eta), \pi \rangle_F \\ \Pi^* &= \arg \max_{\Pi \in \mathcal{Q}} \langle \mathbf{B}(\mathbf{P}_{12}, \mathbf{P}_{21}), \Pi \rangle_F \end{aligned} \quad (14)$$

where the *benefit matrix* for PMF is modified according to assumption (4) as  $\mathbf{B}(\mathbf{P}_{21}, \mathbf{P}_{12}) = \mathbf{K}^1(\mathbf{P}_{12} + \mathbf{P}_{21}^T)\mathbf{K}^2$ .

To prove the equivalence, we only need to show that for any given  $(\eta, \mathbf{P}_{12}, \mathbf{P}_{21})$  (derived from the same set of correspondences), the summation of the cost matrix  $\mathbf{C}$  and the benefit matrix  $\mathbf{B}$  is a constant matrix (i.e., independent of the initial correspondences). In this case, for any input correspondences, it would be equivalent to minimize over the cost matrix or to maximize over the benefit matrix.

According to Assumption (4), we have that the distance in VarMin is set to geodesic distance:  $\mathbf{d}_X = \mathbf{d}^{S_1}, \mathbf{d}_Y = \mathbf{d}^{S_2}$ . We then have

$$\begin{aligned} \mathbf{M}_{pq}^1 &= \left( \mathbf{d}_{p, T_{21}(q)}^{S_1} \right)^2 \Rightarrow \mathbf{M}^1 = (\mathbf{d}^{S_1})^2 \mathbf{P}_{12} \\ \mathbf{M}_{pq}^2 &= \left( \mathbf{d}_{T_{12}(p), q}^{S_2} \right)^2 \Rightarrow \mathbf{M}^2 = \mathbf{P}_{21}^T (\mathbf{d}^{S_2})^2 \end{aligned} \quad (15)$$

For simplicity, we set the weighting function  $\mathbf{W}^i = -(\mathbf{d}^{S_i})^2$ , which is supposed to be negatively proportional to the distance matrix.

Similarly, the kernel for PMF that is defined from the geodesic distance satisfies:  $\mathbf{K}_{pq}^i = \exp -\frac{(\mathbf{d}_{pq}^{S_i})^2}{2\sigma_i^2}$  which can be expanded as:

$$\mathbf{K}^i = \exp -\frac{(\mathbf{d}^{S_i})^2}{2\sigma_i^2} = \sum_{k=0}^{\infty} \frac{1}{k!} \left( -\frac{(\mathbf{d}^{S_i})^2}{2\sigma_i^2} \right)^k \quad (16)$$

With Assumption (5) where the local variances are unweighted, the cost matrix for VarMin can be simplified as:

$$\begin{aligned} \mathbf{C} &= \mathbf{W}^1 \mathbf{M}^2 + \mathbf{M}^1 \mathbf{W}^2 \\ &= -(\mathbf{d}^{S_1})^2 \mathbf{P}_{21}^T (\mathbf{d}^{S_2})^2 - (\mathbf{d}^{S_1})^2 \mathbf{P}_{12} (\mathbf{d}^{S_2})^2 \\ &= -(\mathbf{d}^{S_1})^2 (\mathbf{P}_{21}^T + \mathbf{P}_{12}) (\mathbf{d}^{S_2})^2 \end{aligned} \quad (17)$$

Define new kernels  $\bar{\mathbf{K}}^i = \mathbf{K}^i - \mathbf{I}_{n_i}$ . The benefit matrix derived from kernels  $\bar{\mathbf{K}}^i$  is:

$$\begin{aligned} \bar{\mathbf{B}} &= (\mathbf{K}^1 - \mathbf{I}_{n_1}) (\mathbf{P}_{12} + \mathbf{P}_{21}^T) (\mathbf{K}^2 - \mathbf{I}_{n_2}) \\ &= \left( -\mathbf{I}_{n_1} + \mathbf{I}_{n_1} - \frac{(\mathbf{d}^{S_1})^2}{2\sigma_1^2} + \dots \right) (\mathbf{P}_{12} + \mathbf{P}_{21}^T) \left( -\mathbf{I}_{n_2} + \mathbf{I}_{n_2} - \frac{(\mathbf{d}^{S_2})^2}{2\sigma_2^2} + \dots \right) \\ &= \frac{1}{4\sigma_1^2 \sigma_2^2} (\mathbf{d}^{S_1})^2 (\mathbf{P}_{21}^T + \mathbf{P}_{12}) (\mathbf{d}^{S_2})^2 + H.O.T \end{aligned}$$

Therefore, we have  $\mathbf{C} + 4\sigma_1^2 \sigma_2^2 \bar{\mathbf{B}} \rightarrow 0$ .

When the initial correspondence is close to the ground-truth, the two benefits matrices  $\mathbf{B}$  and  $\bar{\mathbf{B}}$  will give the same maximum. In this case, the minimizer of VarMin is the maximizer of PMF.

Let's call the energies in Equation (14) PMF-VarMin energy. Given the equivalence (up to some assumptions) discussed above, we can see that, PMF proposes to optimize PMF-VarMin energy in the search space of  $\mathcal{Q}_2$ , while VarMin tries to optimize the same energy in the search space of  $\mathcal{Q}_3$ . A very natural trial is to optimize PMF-VarMin in the search space of  $\mathcal{Q}_1$  where no additional constraints are added to the pointwise map. Unfortunately this approach does not yield satisfactory results, since the global optimizer of the unconstrained PMF-VarMin energy is a degenerated pointwise map, where all the points from the source are map to the same vertex on the target shape. Therefore, it is interesting to investigate how the spectral alignment energy can be optimized w.r.t. different search spaces. Our Sinkhorn filter can be viewed in a similar light, where we identify a more suitable search space (with the objective of achieving higher bijectivity and leading to improved accuracy) in order to achieve optimal spectral alignment.

## References

- [1] Dimitri P Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, 14(1):105–123, 1988. 5
- [2] Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. *Computer Graphics Forum*, 36(5):151–163, 2017. 3
- [3] Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics*, 36:14, 2017. 1, 5
- [4] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155:1–155:14, Nov. 2019. 1, 2
- [5] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012. 1, 2, 3
- [6] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. 1, 2

- [7] Emanuele Rodolà, Michael Moeller, and Daniel Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)*, 2015. 5
- [8] Emanuele Rodolà, Michael Möller, and Daniel Cremers. Regularized pointwise map recovery from functional correspondence. In *Computer Graphics Forum*, volume 36, pages 700–711. Wiley Online Library, 2017. 5
- [9] Matthias Vestner, Zorah Löhner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alex Bronstein, Michael Bronstein, and Ron Kimmel. Efficient deformable shape correspondence via kernel matching. In *3D Vision (3DV), 2017 International Conference on*, pages 517–526. IEEE, 2017. 5
- [10] Matthias Vestner, Roei Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proc. CVPR*, pages 6681–6690, 2017. 1, 5