A Network Analysis for Correspondence Learning via Linearly-Embedded Functions

Sharik Siddiqi[1,2][0009−0002−1929−7847] and Zorah Lähner[1][0000−0003−0599−094X]

[1] University of Siegen, Hölderlinstr. 3, 57076 Siegen, Germany
zorah.laehner@uni-siegen.de
[2] IAB – Institut für Angewandte Bauforschung, Über der Nonnenwiese 1, 99428 Weimar, Germany s.siddiqi@iab-weimar.de

Abstract. Calculating correspondences between non-rigidly deformed shapes is the backbone of many applications in 3D computer vision and graphics. The functional map approach offers an efficient solution to this problem and has been very popular in learning frameworks due to its low-dimensional and continuous nature. However, most methods rely on the eigenfunctions of the Laplace-Beltrami operator as a basis for the underlying function spaces. While these have many advantages, they are also sensitive to non-isometric deformations and noise. Recently a method to learn the basis functions along with suitable descriptors has been proposed by Marin et al.. We do an in-depth analysis of the architecture proposed, including a new training scheme to increase robustness against sampling inconsistencies and an extension to unsupervised training which still obtains results on-par with the supervised approach.

Keywords: Non-Rigid Correspondence · Unsupervised Learning · Functional Maps · 3D Descriptors · Basis Learning.

1 Introduction

The problem of shape correspondence between 3D shapes acts as the foundation in many applications, for example texture and motion transfer, statistical shape modelling or 3D medical applications. The goal is to find a meaningful map between the surfaces of shapes which can vary in shape or pose but with a semantic relation. In contrast to the 3-dimensional rigid case, where only six parameters are necessary to describe the deformation (rotation and translation), the non-rigid scenario involves degrees of freedom for every vertex on the source shape, thus, the search space becomes computationally infeasible to deal with.

To overcome this issue, the concept of functional maps was introduced in [21]. Instead of computing a correspondence between vertices, functional maps compute a correspondence between function spaces on the surface of the shapes. Using the Laplace-Beltrami eigenbasis to define these function spaces reduces the correspondence problem to a low-dimensional, continuous optimisation. Due to this, functional maps have been widely popular [19,8,22], especially in learning applications [14,11,27] (see also Section 2.2). However, the Laplace-Beltrami

eigenfunctions are sensitive to non-isometric deformations and noise associated with the surface. The most common solution is to learn the *descriptor* functions which are robust against the target deformations [14,11,27] while keeping the basis unchanged. To make the *basis* functions robust, the approach of [17] proposed to learn both the basis functions as well as the descriptors used to calculate the functional maps in a data-driven way. However, the method is supervised and relies on a large training dataset with labelled point-to-point correspondences. There is a deficit of such datasets because their generation is very expensive or relies on artificial data which is often too regular to learn robustness against noise. In this paper, we analyse the framework of [17], show points of improvement and propose strategies to overcome the problems of labelled data and sampling irregularity.

Contributions. We provide an analysis of several aspects of the framework in [17] to overcome performance bottlenecks. This includes hyper parameters and base network choice, as well as two extensions to make the training unsupervised and robust against sampling inconsistencies. In order to make the shape matching pipeline [17] sampling invariant and more general, we perform a random sampling scheme during training which provides a trade-off between high computational demand and dependence on the sampling of the input shapes. Additionally, we propose an unsupervised version for the same setup which tackles the shape correspondence task by leveraging prior geometric information associated with the compared shapes as regularisers in order to overcome the need for labelled data which is expensive to produce for the correspondence problem.

2 Related Work

We provide an introduction to learning and non-learning-based non-rigid shape correspondence approaches that are directly related to our method. For a in-depth survey of the topic the interested reader may refer to [28] and [20].

2.1 Non-Rigid Correspondence Methods

The problem of finding correspondences between two non-rigidly deformed shapes is often posed as an quadratic assignment problem (QAP) where the solution is a permutation that matches vertices in a way such that the geodesic distances between all pairs of points are as similar as possible [13]. However, this formulation is NP-hard [4] and assumes that the optimal solution can be described by a permutation which is often not the case in reality. Many non-optimal algorithms to solve the QAP problem have been proposed, for example convex relaxations [13], and heuristics for non-convex formulations [12]. To work around the permutation constraint, soft-correspondences are a popular choice [26,5], or the usage of elastic matching formulations [34,10]. Nevertheless, many of these approaches struggle with high resolution meshes. A possible work-around is to add regularization in the extrinsic embedding space, for example by restricting the motion

between the input shapes to be volume-preserving [7]. One very widely-used solution is to move from the space of vertex correspondences to functional correspondences. In [21], Ovsjanikov et al. proposed the so-called functional maps that represent correspondences as a transformation matrix between two fixed function basis spaces. Using the frequency ordering of the Laplace-Beltrami eigenspace, this means the correspondence can be approximated by a small matrix with no constraints on the entries. This is also the representation we choose in this paper.

2.2 Functional Map-Based Learning Approaches

Since functional maps provide a continuous and memory-efficient way to represent correspondence, they have been a popular choice in learning-based correspondence approaches. The first attempt to incorporate functional maps into a learning framework was done in DeepFMs [14] by learning the optimal combination of descriptors before optimizing the functional map matrix. This lead to a variety of follow-up work in which this strategy is refined [29], the functional map matrix itself is predicted [15], the framework is made unsupervised [11,2], and the properties of the functional map matrix [27] or of the deformation between the shapes are imposed [9].

However, most of these methods like original work [21], still rely on the usage of the eigenfunctions of the Laplace-Beltrami operator as basis functions. While this selection has many advantages in the setting of non-rigid shape matching, it is also sensitive to non-isometric deformations and noise which deters the performance in these settings. Alternative basis sets have been proposed in the literature, for example in [18,6,23], but most of them work best for fixed settings. Instead of using a predetermined basis set, the approach of [17] suggests to learn the optimal function basis in combination with the descriptor functions from a training set. This works very well, even in the presence of noise, but the approach requires large amounts of labelled data for a supervised training. In this work, we propose a novel way to train this kind of framework in a completely unsupervised way, inspired by the unsupervised descriptor-learning methods of [11,2], thus eliminating the need for accumulation and labelling of huge volumes of data.

3 Background

3.1 Functional Maps

Functional maps frame the correspondence problem in terms of function spaces instead of points on the surface. Let $\mathcal{F}(\mathcal{X})$, $\mathcal{F}(\mathcal{Y})$ be two comparable function spaces on the shapes $\mathcal{X}, \mathcal{Y}$ discretised with n vertices with basis sets $\{\phi_j\}_{j\geq 1} \subset \mathcal{F}(\mathcal{X})$ and $\{\psi_i\}_{i\geq 1} \subset \mathcal{F}(\mathcal{Y})$ respectively. Then, a *functional map* is a linear mapping $C_{\mathcal{XY}} : \mathcal{F}(\mathcal{X}) \to \mathcal{F}(\mathcal{Y})$ between the function spaces. In the case of finite basis sets with cardinality k, the linear mapping can be written simply as a matrix and computed either by construction through a given point-to-point correspondence P or optimised to preserve certain descriptor functions.

For a given $P \in \{0,1\}^{n \times n}$ representing the correspondence between the vertices of input shapes $(\mathcal{X}, \mathcal{Y})$ and the basis functions stacked into $\Phi \in \mathbb{R}^{n \times k}$ and $\Psi \in \mathbb{R}^{n \times k}$.
The functional map matrix C can be computed using

$$C = \Psi^{\dagger} \cdot P \cdot \Phi. \tag{1}$$

where † refers to the Pseudo-inverse which is either the transpose or mass-corrected transpose depending on how the eigenfunctions were computed.

In the case where no point-wise correspondence is given, the functional map can be approximated from a set of comparable vertex descriptor functions. Consider a set of q corresponding functions $G_{\mathcal{X}} = (f_1, \ldots, f_q) \in \mathbb{R}^{n \times q}$ and $G_{\mathcal{Y}} = (g_1, \ldots, g_q) \in \mathbb{R}^{n \times q}$ (such that $G_{\mathcal{X}} \approx T_F G_{\mathcal{Y}}$, where T_F can be thought of as some sort of functional correspondence), C is computed by solving this least-square problem with k^2 variables

$$\min_{C \in \mathbb{R}^{k \times k}} \left\| G_{\mathcal{X}}^{\top} \Phi - G_{\mathcal{Y}}^{\top} \Psi C \right\|_F^2 \tag{2}$$

which has a closed form solution given by :

$$C = (G_{\mathcal{Y}}^{\top} \Psi)^{-1} (G_{\mathcal{X}}^{\top} \Phi) \tag{3}$$

$\|.\|_F$ denotes the calculation of the Frobenius norm. The most common choice for Φ, Ψ are the first k-eigenfunctions of the Laplace-Beltrami operator, as it was proposed in [21], which provide a basis for square integrable functions L^2 on the surface.

3.2 Linearly-Invariant Embedding

Our method is based on the general network architecture proposed in Marin et al. [17] which proposes to learn both the function basis and the optimal descriptors, associated with the learned basis set, in a joint training regime. The framework uses two PointNet [24] networks, one for the basis functions called *linearly invariant embedding network*, and other for the descriptors, called *probe function network*, which are trained sequentially. An overview of the pipeline is shown in Figure 1.

Linearly Invariant Embedding. The first step is to train a Siamese network which outputs the embedding functions for each input shape $\mathcal{X}$, $\mathcal{Y}$. Given a fixed function set $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ and a ground-truth correspondence $\Pi_{\mathcal{XY}}^{gt}$, The generation of $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ is learned through the loss function

$$L(\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}) = \frac{1}{n_{\beta}} \sum \| S_{\mathcal{XY}} P_{\mathcal{X}} - \Pi_{\mathcal{XY}}^{gt} P_{\mathcal{X}} \|_2^2 \tag{4}$$

where n_{β} represents the total correspondence cases and $S_{\mathcal{XY}}$ denotes a soft permutation matrix which provides an approximate point to point mapping between

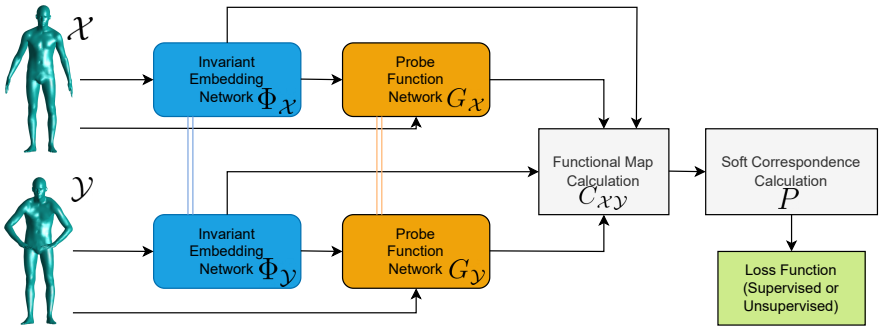

Fig. 1. Overview of the framework. The invariant embedding network generates a basis set $\Phi.$ which is then used to generate descriptor functions $G.$ for this basis. Both are used to compute a functional map $C_{\mathcal{XY}}$ and soft correspondence P for either the loss function or solution. Networks in the same colour share weights.

the defined shapes in a matrix format given by :

$$(S_{\mathcal{XY}})_{ij} = \frac{e^{-\left\|\widehat{\Phi}^i_{\mathcal{X}} - \Phi^j_{\mathcal{Y}}\right\|_2}}{\sum_{k=1}^{n_{\mathcal{Y}}} e^{-\left\|\widehat{\Phi}^i_{\mathcal{X}} - \Phi^k_{\mathcal{Y}}\right\|_2}} \tag{5}$$

Eq. 4 aims to preserve the coordinate function $P_{\mathcal{X}}$ of $\mathcal{X}$ through the soft correspondence induced by $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$.

Probe Functions. After completely training the embedding network, the next step is to generate the probe functions based on which the functional map between $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ can be predicted with the ground-truth knowledge. The probe functions are also generated by a Siamese network which is trained using the ground-truth functional maps $C^{gt}_{\mathcal{YX}} = \Phi^{\dagger}_{\mathcal{X}} \Pi^{gt}_{\mathcal{YX}} \Phi_{\mathcal{Y}}$ and $C_{\mathcal{YX}} = (\Phi^{\dagger}_{\mathcal{X}} G_{\mathcal{X}})(\Phi^{\dagger}_{\mathcal{Y}} G_{\mathcal{Y}})^{-1}$. The loss function simply compares $C^{gt}_{\mathcal{YX}}$ and $C_{\mathcal{YX}}$:

$$L(G_{\mathcal{X}}, G_{\mathcal{Y}}) = \|C^{gt}_{\mathcal{YX}} - C_{\mathcal{YX}}\|_2 \tag{6}$$

4 Method

In this section we describe two major changes we made to the training procedure in order to make it robust to sampling (Section 4.2) and shift to unsupervised domain (Section 4.1). Additional experiments on single aspects of the architecture are described in Section 5.

For two shapes $\mathcal{X}, \mathcal{Y}$, we denote the basis which is the output of the embedding network as $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}} \in \mathbb{R}^{n \times k}$, where n and k are the number of vertices and the number of basis functions, respectively. The descriptor functions generated by the probe network are $G_{\mathcal{X}}, G_{\mathcal{Y}} \in \mathbb{R}^{n \times m}$ where m is the descriptor dimension which is a hyperparameter. Finally, $C_{\mathcal{XY}}$ indicates the functional map from the

function space on $\mathcal{X}$ to $\mathcal{Y}$. The function space is spanned by $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ unless indicated otherwise. Notice that with given $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ and $G_{\mathcal{X}}, G_{\mathcal{Y}}$, a functional map can always be computed using Eq. (3).

4.1 Unsupervised Training

We adapt the training strategy of [17] to work without the need for labelled ground-truth correspondences. As described in Section 3.2 and Figure 1, the pipeline has two networks that are trained sequentially and both use the ground-truth as part of the loss function. In Eq. (4) the true correspondence is directly included as $\Pi^{gt}_{\mathcal{XY}}$ and in Eq. (6) it is used to generate $C^{gt}_{\mathcal{YX}}$ using the computed embedding functions. Both networks have to be made unsupervised separately.

Invariant Embedding Network. To make the embedding network unsupervised, we need to remove the need for the ground-truth permutation from the calculation of the functional map $C_{\mathcal{YX}}$ in Eq. (4). To achieve this, we utilise handcrafted descriptors for the computation of $C_{\mathcal{YX}}$ as is done in the original functional maps framework and explained in Eq. (3). This is only done during training when the learned probe functions are not available yet. To make up for the robustness of the ground-truth, we can impose different meaningful properties of the optimal solution, like orthonormality, to guide the optimisation [27].

In the end, our solution involves aligning pairwise features instead of comparing to the ground-truth and imposing orthonormality both on the resulting functional map as well on the basis functions. Then, the loss function has the following structure:

$$L_{\text{embed}}(\mathcal{X}, \mathcal{Y}) = \ell_{\text{dist}}(\mathcal{X}, \mathcal{Y}) + \alpha \ell_{\text{orth-func}}(\mathcal{X}, \mathcal{Y}) + \beta \ell_{\text{orth-basis}}(\mathcal{X}, \mathcal{Y}) \tag{7}$$

where α and β weight the significance of the orthonormality conditions w.r.t the overall optimisation landscape. We obtained good results with $\alpha = \beta = 0.1$. The terms are defined below.

Distortion Minimisation Term $\ell_{dist}(\mathcal{X}, \mathcal{Y})$. Instead of comparing whether the soft-correspondence induced by $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ imposes the same transformation on $P_{\mathcal{X}}$ as the ground-truth, we will compare whether the soft-correspondence aligns pair-wise properties of $\mathcal{X}$ and $\mathcal{Y}$. Given two matrices $\mathbf{D}_{\mathcal{X}}, \mathbf{D}_{\mathcal{Y}} \in \mathbb{R}^{n \times n}$ that define comparable pair-wise properties on $\mathcal{X}, \mathcal{Y}$, the loss reads as

$$\ell_{\text{dist}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{Y}|^2} \left\| \mathbf{D}_{\mathcal{Y}} - \mathbf{Q}^\top \mathbf{D}_{\mathcal{X}} \mathbf{Q} \right\|^2_{\text{F}} \tag{8}$$

where $\mathbf{Q} = \mathbf{P} \circ \mathbf{P}$ is the Hadamard product of $\mathbf{P} = \left| \boldsymbol{\Phi}_{\boldsymbol{\mathcal{Y}}} \mathbf{C}_{\mathcal{XY}} \boldsymbol{\Phi}_{\boldsymbol{\mathcal{X}}}{}^\top \mathbf{A}_{\mathcal{X}} \right|^{\wedge}$ which is the column normalised ($|\cdot|^{\wedge}$) reversal of Eq. (1) to get a soft correspondence from a given functional map. $\mathbf{A}_{\mathcal{X}}$ is the mass matrix of $\mathcal{X}$. The Hadamard product

means that the loss function is derived from the expected deviation from the optimal solution when the columns of Q are interpreted as probabilities of the correspondence of one point. In this step, the $G_{\mathcal{X}}, G_{\mathcal{Y}}$ in the computation of $\mathbf{C}_{\mathcal{XY}}$ are manually chosen. We show experiments on this choice in Section 5.5. As has been proposed previously in the literature, we try the geodesic distances [11] and heat kernel [2] for $D_{\cdot}$. The results can be found in Section 5.5. This term ensures that the geometric structure of the shapes is preserved through the functional map imposed by the embedding functions, which is a property of any good solution.

Orthonormality of the Basis Functions $\ell_{orth\text{-}basis}(\mathcal{X}, \mathcal{Y})$. Orthonormality of the basis functions is desirable as it allows for the efficient calculation and regularisation of the functional map. We impose orthonormality by penalising deviation from the identity in the following form:

$$\ell_{\text{orth-basis}}(\mathcal{X}, \mathcal{Y}) = \left\| \mathbf{\Phi}_{\mathcal{X}}{}^{\top} A_{\mathcal{X}} \mathbf{\Phi}_{\mathcal{X}} - \mathbb{I} \right\|_{\text{F}}^2 + \left\| \mathbf{\Phi}_{\mathcal{Y}}{}^{\top} A_{\mathcal{Y}} \mathbf{\Phi}_{\mathcal{Y}} - \mathbb{I} \right\|_{\text{F}}^2 \tag{9}$$

where $\mathbb{I}$ is the identity matrix and $A_{\cdot}$ again is the mass matrix of the respective shape which is used to weight the inner product on the surface.

Orthonormality of the Functional Map $\ell_{orth\text{-}func}(\mathcal{X}, \mathcal{Y})$. Orthonormality of the functional map matrix C is associated to area preservation [21] and has been shown to work well as a regularisation term when learning of functional maps [27]. If $\mathbf{C}_{\mathcal{XY}}$ is the functional map obtained through Eq. (3), the orthonormality can be enforced as follows:

$$\ell_{\text{orth-func}}(\mathcal{X}, \mathcal{Y}) = \left\| \mathbf{C}_{\mathcal{XY}}{}^{\top} \mathbf{C}_{\mathcal{XY}} - \mathbb{I} \right\|_{\text{F}}^2 \tag{10}$$

where $\mathbb{I}$ represents the identity matrix

Probe Function Network. Once the embedding network has been trained, the next step is to train the probe network to generate optimal descriptors that work with new embedding functions. Again, we only need to replace the loss function with an unsupervised counterpart. We adjust the loss function from the embedding network (Eq. (7)) to make it suitable for the new task by dropping the orthonormality of the basis functions which is not needed in this case. The final loss function is then defined as

$$L_{\text{probe}}(\mathcal{X}, \mathcal{Y}) = \ell_{\text{dist}}(\mathcal{X}, \mathcal{Y}) + \alpha \ell_{\text{orth-func}}(\mathcal{X}, \mathcal{Y}) \tag{11}$$

where $\ell_{\text{dist}}(\mathcal{X}, \mathcal{Y})$ and $\ell_{\text{orth-func}}(\mathcal{X}, \mathcal{Y})$ can be obtained from Eqs. (8) and (10) respectively. Here, instead of fixing $G_{\mathcal{X}}, G_{\mathcal{Y}}$, we fix $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ within the loss functions. We use $\alpha = 0.1$.

4.2 Subsampling Scheme

In the proposed approach of [17] all shapes are subsampled to a fixed amount of vertices to make the training feasible. The sampled vertices are the same for all shapes to preserve the ground-truth information from the full shape to make the supervised training possible. However, this leads to incentives for the network to rely on the fixed vertex sampling instead of learning robustness against different vertex distributions which will be present in realistic datasets.

We propose to use a flexible training scheme that simulates varying and non-uniform vertex sampling on the surface. To that end, our training uses a different subset of vertices of a full shape in each training epoch. To achieve a full coverage of the surface we choose a subset $F \subset \mathcal{X}$ with farthest point sampling that stays the same in all epochs – similar to how the training set is chosen in [17]. Further, we add a certain percentage of randomly chosen vertices to the training set. This set changes in each epoch and forces the network to generalise to a wide variety of vertex positions. Unfortunately, this cannot be simply combined with the unsupervised scheme from Section 4.1 because it requires either storing a huge geodesic distance matrix with n^2 entries, or computing geodesics on-the-fly which is very slow. However, we show that this strategy vastly improves the performance of the original pipeline when evaluating on full resolution shapes.

5 Experiments

In the following section we describe the experiments done for this paper. Section 5.3 gives a comparison of the supervised approach of [17] with our unsupervised training scheme. In Section 5.4 we describe how the random sampling training improves the results of the supervised approach. And finally, in Section 5.5 we do an ablation study of several aspects of the original architecture. The mean geodesic error results and hyper parameters of all experiments can be found in the supplementary material.

5.1 Datasets

We use two datasets: the SURREAL [32] dataset consisting of 1000 human shapes with varying pose and body shape. We use this dataset as the training and validation set during the training phase in a ratio of 49:1. For testing we use the registration of the FAUST [3] dataset which contains 100 human shapes of 10 individuals in 10 poses each. We call the setting with two shapes from the same individual *isometric* and from two different individuals *non-isometric*. In order to keep the problem size tractable, all shapes are downsampled to 1000 and 2100 vertices (as indicated in the experiment description).

Both datasets originally use the SMPL [16] mesh connectivity and, thus, have a compatible sampling which makes the task easier. However, we introduce different downsampling strategies in the experiments in Section 5.4 to generate inconsistent samplings with increased complexity.

For pre-processing, we zero-mean the coordinate functions of all shapes and add random rotations in the range of $(-\pi/2, \pi/2)$ around the y-axis for augmenting the training data.

5.2 Evaluation

We evaluate using the mean geodesic error and cumulative geodesic error plots. For a pair of shapes $(\mathcal{A}, \mathcal{B})$, let $a \in \mathcal{A}$ be the source point, $b \in \mathcal{B}$ the calculated match for a, and $b^* \in \mathcal{B}$ the ground-truth match of a. We measure the relative geodesic error of the correspondence (a, b) as:

$$\epsilon(a, b) = \frac{\text{dist}_{\text{geo}}(b, b^*)}{\text{diam}(\mathcal{B})} \tag{12}$$

where $\text{dist}_{\text{geo}}(b, b^*)$ represents the geodesic distance between points b and b^* on surface $\mathcal{B}$ and $\text{diam}(\mathcal{B})$ is the geodesic diameter associated with $\mathcal{B}$. The mean geodesic error is the average of $\epsilon(a, b)$ for all calculated matches (a, b).

The cumulative geodesic error curve plots the percentage of correspondences that are below the threshold given on the x-axis in percentage of the geodesic diameter of $\mathcal{B}$. The performance comparison is done on pairs of isometric and non-isometric shapes.

Qualitative Results. We visualise the results by colour transfer. For a correspondence between shapes $\mathcal{X}$ and $\mathcal{Y}$, we plot a fixed, smooth colour function based on the 3D coordinates of $\mathcal{X}$ and then use the computed correspondence to transfer the colourmap to shape $\mathcal{Y}$. Wrong correspondences are visible through wrong colours and non-smooth areas. See Figure 2.

5.3 Correspondence Accuracy

We evaluate the performance of our unsupervised setup against the original implementation of [17] on both isometric (from the same class in FAUST) and non-isometric (from different classes) pairs. We report the results and some qualitative examples in Figure 2. Even though we do not use any ground-truth information, we were able to achieve results on-par with the supervised approach. Interestingly, there is nearly no difference in results between the isometric and non-isometric cases, especially for the unsupervised training, which is an indicator that the usage of learned basis functions instead of isometry-invariant Laplace-Beltrami eigenfunctions is indeed more robust against deviations from the isometry assumption.

5.4 Different Sampling

The SURREAL and FAUST template datasets have the same mesh topology due to the usage of the same SMPL model. This is not a realistic assumption in many settings since a registration to a joint mesh already assumes a correspondence

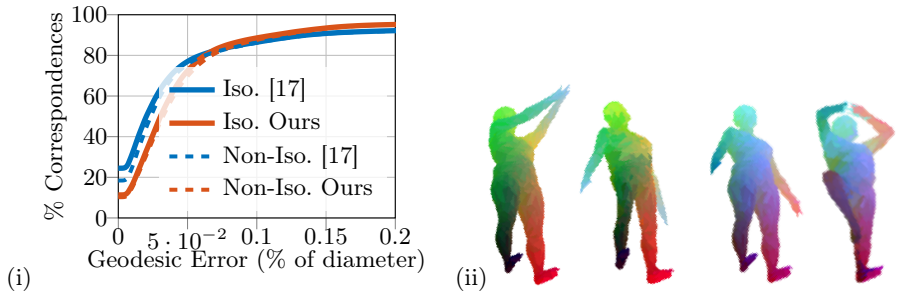

Fig. 2. (Left) Comparison of results on the FAUST test dataset. (Right) Qualitative examples of the results on the FAUST dataset with 1000 vertices. The results are overall accurate except some small noise, see for example the right hand in (ii).

was computed through the registration process. Instead real-world scans have varying connectivity and different sampling density on different mesh parts due to the acquisition process. We simulate this effect by downsampling our shapes with farthest point sampling (FPS) plus a certain percentage of random points instead of a fixed, consistent subset for all shapes as explained in Section 4.2.

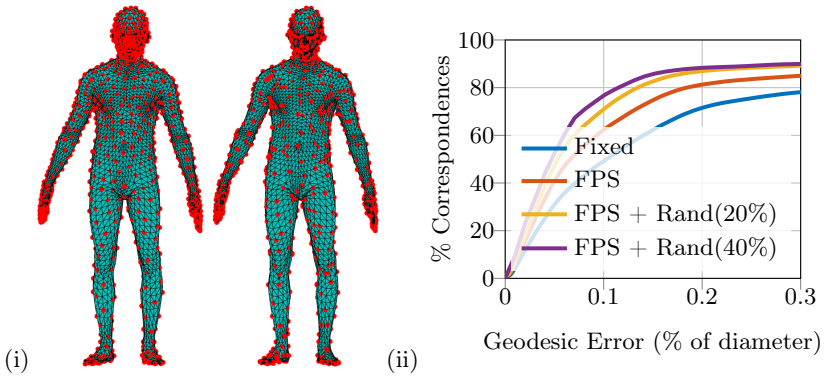

Fig. 3. (i) Examples of the same shape with different sampling. (Left) 1000 of 6980 vertices sampled via FPS. (Right) 600 vertices sampled with FPS and 400 randomly. (ii) Effect of different sampling methods on the results. Using FPS and adding randomly sampled points prevents the network from overfitting on the point distribution and makes the results more robust.

We report the results and an example of the sampling in Fig. 3. Our results show that introducing randomness in the sampling process during training significantly improves the results due to the network being less dependent on the fixed point distribution. However, there needs to be a trade-off between computational load due to addition of more points and improvement in performance.

5.5 Ablation Study

We tested the influence of the different design choices made and justify our decisions with the following ablation study. The mean geodesic errors of all experiments are reported in the supplementary material.

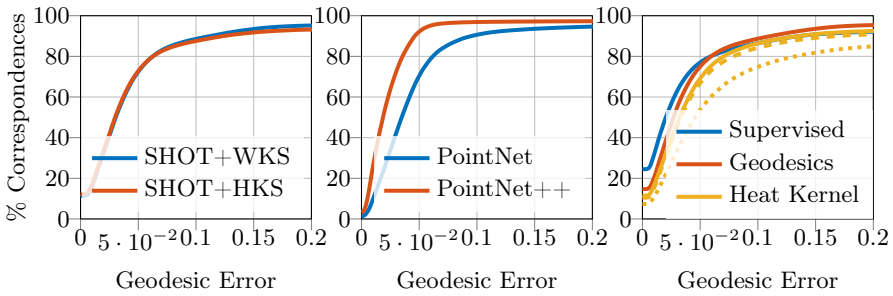

Fig. 4. Cumulative error curves of different settings and architectures. (Left) We test the influence of the different descriptor selection in training of the basis network. The performance difference is minimal. (Middle) Using PointNet++ [25] instead of PointNet [24] as the base architecture significantly improves the performance. (Right) Comparison of the supervised approach in [17], our unsupervised approach with geodesics, and our unsupervised approach with heat kernels. For heat kernel we try a fixed time ($t = 0.01$, dotted), one time reduction ($t = 0.01 \to 0.007$, dashed), and two time reductions ($t = 0.01 \to 0.007 \to 0.005$, solid).

Network Type. The embedding and probe networks from [17] are realised through a PointNet architecture [24]. However, PointNet is known to be well-suited for global feature learning but less to learn distinctive local features due to the global max-pooling. PointNet++ [25] is an extension of PointNet which overcomes these shortcomings and should be able to learn point-wise descriptors better. The replacement of PointNet with PointNet++ results in performance improvement shown in Figure 4.

Interestingly, the usage of PointNet++ also solves the problem of front-back symmetry flips in the solution. The human shape has very few features that can distinguish the front from the back side (e.g. the face and feet facing forward) and none of them are present on global scale. Due to focus being only on the global scale features in the case of a simple PointNet, the solution often mixes up front and back sides of the shape-pairs but with the usage of PointNet++ this problem is solved. See Figure 5 for an example.

Learning of Probe Functions. In the unsupervised scenario, the absence of the ground-truth prompts the utilisation of handcrafted descriptors in Eq. 3 to simplify the functional map generation as explained in Section 4.1. We propose the use of well-tried HKS [30], WKS [1], and SHOT [31] as $G_{\mathcal{X}}, G_{\mathcal{Y}}$ for training

the basis network. The HKS and WKS are purely intrinsic while SHOT includes extrinsic information. A combination of intrinsic and extrinsic descriptors often leads to the best results and, eventually, we decided to choose WKS and SHOT based on the results in Fig. 4.

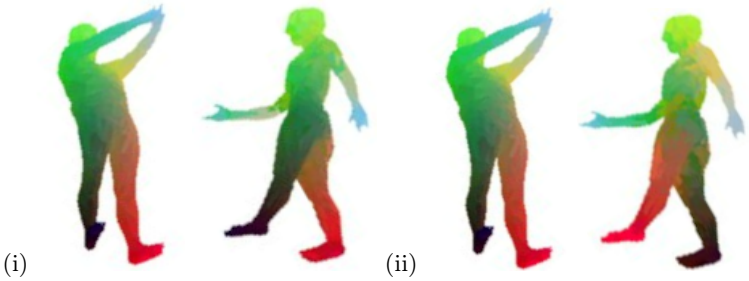

Fig. 5. (i) Front-back symmetry swap when using PointNet due to its global properties. (ii) Correct solution on the same pair when using PointNet++ instead.

Pair-Wise Descriptors. Two pair-wise descriptors are popular choices in the context of non-rigid shape correspondence, namely, the geodesic distances and heat kernel which have for example been used in [11] and [2]. While the geodesics are more expensive to compute, the heat kernel has a time parameter t that has to be chosen. The lower t the more local the heat kernel which leads to more accurate correspondences but a less convex energy landscape [33]. Therefore, reducing the time parameter after a certain amount of epochs can improve the results by applying a sort of coarse-to-fine scheme [2]. We show the results of all choices including the supervised method of [17] in Figure 4. While the time reduction helps the heat kernel approach, the geodesic distances still perform better.

Network Dimensions. Both the output dimensions of the embedding and the probe network are variables to be chosen. A larger basis and descriptor size allows for more information about the shapes to be processed but at the same time increases the network size and, thus, the complexity of training. The effect on the result is minimal as compared to the associated computational load. Our choice is $d_{out} = 20$ for the embedding network and $d_{out} = 40$ for the probe network due to better trade-off between the complexity and performance.

6 Conclusion

In this paper, we analysed several aspects of the linear embedding framework of [17] for functional maps in the context of non-rigid shape correspondence. In addition to network hyper parameters, we proposed an extension to make

	40-d probe		20-d embedd			final
	20-d E	30-d E	40-d P	50-d P	60-d P	30-d E, 50-d P
after basis train.	0.0204	0.0157	0.0204	0.0204	0.0204	0.0157
after probe train. (full)	**0.1099**	0.1126	**0.1099**	0.1119	0.1152	**0.1051**

Table 1. Effects of the network output sizes on the embedding and probe networks. We test the effect after the training of the embedding and probe network separately. E stands for embedding and P for probe.

the whole pipeline unsupervised and show that the same performance can be achieved without the use of labelled training data. Additionally, we devised a training strategy based on random sampling during training that improves the robustness of the fixed sampling training in [17] against sampling related artefacts. Overall, we achieved several improvements over the original pipeline which can be used to boost the performance in many applications.

Acknowledgements. Zorah Lähner is supported by a KI-Starter grant of the Ministry of Culture and Science of the State of North Rhine-Westphalia.

References

1. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In: International Conference on Computer Vision (ICCV) (2011)
2. Aygün, M., Lähner, Z., Cremers, D.: Unsupervised dense shape correspondence using heat kernels. In: Conference on 3D Vision (3DV) (2020)
3. Bogo, F., Romero, J., Loper, M., Black, M.J.: FAUST: Dataset and evaluation for 3D mesh registration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
4. Burghard, O., Klein, R.: Efficient lifted relaxations of the quadratic assignment problem. In: Vision, Modeling & Visualization (VMV) (2017)
5. Cao, D., Roetzer, P., Bernard, F.: Unsupervised learning of robust spectral shape matching. In: Transactions on Graphics (Proc. of SIGGRAPH) (2023)
6. Colombo, M., Boracchi, G., Melzi, S.: Pc-gau: Pca basis of scattered gaussians for shape matching via functional maps. In: Smart Tools and Applications in Graphics (STAG) (2022)
7. Eisenberger, M., Lähner, Z., Cremers, D.: Divergence-free shape correspondence by deformation. Computer Graphics Forum (CGF) **38**(5) (2019)
8. Eisenberger, M., Lähner, Z., Cremers, D.: Smooth shells: Multi-scale shape registration with functional maps. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
9. Eisenberger, M., Toker, A., Leal-Taixé, L., Cremers, D.: Deep shells: Unsupervised shape correspondence with optimal transport. In: 34th Conference on Neural Information Processing Systems (NeurIPS) (2020)
10. Ezuz, D., Heeren, B., Azencot, O., Rumpf, M., Ben-Chen, M.: Elastic correspondence between triangle meshes. Computer Graphics Forum (CGF) (2019)

11. Halimi, O., Litany, O., Rodolà, E., Bronstein, A.M., Kimmel, R.: Unsupervised learning of dense shape correspondence. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
12. Holzschuh, B., Lähner, Z., Cremers, D.: Simulated annealing for 3d shape correspondence. In: Conference on 3D Vision (3DV) (2020)
13. Kezurer, I., Kovalsky, S.Z., Basri, R., Lipman, Y.: Tight relaxation of quadratic matching. In: Computer Graphics Forum (CGF). vol. 34 (2015)
14. Litany, O., Remez, T., Rodolà, E., Bronstein, A., Bronstein, M.: Deep functional maps: Structured prediction for dense shape correspondence. In: International Conference on Computer Vision (ICCV) (2017)
15. Liu, S., Xu, H., Yan, D.M., Hu, L., Liu, X., Li, Q.: Wtfm layer: An effective map extractor for unsupervised shape correspondence. Computer Graphics Forum **41**(7), 51–61 (2022)
16. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
17. Marin, R., Rakotosaona, M.J., Melzi, S., Ovsjanikov, M.: Correspondence learning via linearly-invariant embedding. In: Conference on Neural Information Processing Systems (NeurIPS) (2020)
18. Melzi, S., Rodolà, E., Castellani, U., Bronstein, M.M.: Localized manifold harmonics for spectral shape analysis. Computer Graphics Forum (CGF) **37**(6) (2018)
19. Melzi, S., Ren, J., Rodolà, E., Sharma, A., Wonka, P., Ovsjanikov, M.: Zoomout: Spectral upsampling for efficient shape correspondence. Computer Graphics Forum (CGF) (2019)
20. Monji-Azad, S., Hesser, J., Löw, N.: A review of non-rigid transformations and learning-based 3d point cloud registration methods. ISPRS Journal of Photogrammetry and Remote Sensing (2023)
21. 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) (Proc. SIGGRAPH) (2012)
22. Pai, G., Ren, J., Melzi, S., Wonka, P., Ovsjanikov, M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with funcitonal maps. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
23. Panine, M., Kirgo, M., , Ovsjanikov, M.: Non-isometric shape matching via functional maps on landmark-adapted bases. Computer Graphics Forum (CGF) (2022)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Neural Information Processing Systems (NeurIPS) (2017)
26. Rodolà, E., Bronstein, A., Albarelli, A., Bergamasco, F., Torsello, A.: A game-theoretic approach to deformable shape matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
27. Roufosse, J.M., Sharma, A., Ovsjanikov, M.: Unsupervised deep learning for structured shape matching. In: International Conference on Computer Vision (ICCV) (2019)
28. Sahillioğlu, Y.: Recent advances in shape correspondence. The Visual Computer (2020)
29. Sharp, N., Attaiki, S., Crane, K., Ovsjanikov, M.: Diffusionnet: Discretization agnostic learning on surfaces. Transactions on Graphics (ToG) (2022)

30. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Symposium on Geometry Processing (SGP) (2009)
31. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision (ECCV) (2010)
32. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: CVPR (2017)
33. Vestner, M., Lähner, Z., Boyarski, A., Litany, O., Slossberg, R., Remez, T., Rodolà, E., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Cremers, D.: Efficient deformable shape correspondence via kernel matching. In: International Conference on 3D Vision (3DV) (2017)
34. Windheuser, T., Schlickewei, U., Schmidt, F.R., Cremers, D.: Large-scale integer linear programming for orientation-preserving 3d shape matching. Computer Graphics Forum (CGF) **30**(5) (2011)