



HAL
open science

Unsupervised Deep Learning for Structured Shape Matching

Jean-Michel Roufousse, Abhishek Sharma, Maks Ovsjanikov

► **To cite this version:**

Jean-Michel Roufousse, Abhishek Sharma, Maks Ovsjanikov. Unsupervised Deep Learning for Structured Shape Matching. ICCV, Oct 2019, SEOUL, South Korea. hal-02305166

HAL Id: hal-02305166

<https://hal.science/hal-02305166v1>

Submitted on 3 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Unsupervised Deep Learning for Structured Shape Matching

Jean-Michel Roufousse
LIX, École Polytechnique
jm.roufousse@gmail.com

Abhishek Sharma
LIX, École Polytechnique
kein.iitian@gmail.com

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

Abstract

We present a novel method for computing correspondences across 3D shapes using unsupervised learning. Our method computes a non-linear transformation of given descriptor functions, while optimizing for global structural properties of the resulting maps, such as their bijectivity or approximate isometry. To this end, we use the functional maps framework, and build upon the recent FMNet architecture for descriptor learning. Unlike that approach, however, we show that learning can be done in a purely unsupervised setting, without having access to any ground truth correspondences. This results in a very general shape matching method that we call SURFMNet for Spectral Unsupervised FMNet, and which can be used to establish correspondences within 3D shape collections without any prior information. We demonstrate on a wide range of challenging benchmarks, that our approach leads to state-of-the-art results compared to the existing unsupervised methods and achieves results that are comparable even to the supervised learning techniques. Moreover, our framework is an order of magnitude faster, and does not rely on geodesic distance computation or expensive post-processing.

1. Introduction

Shape matching is a fundamental problem in computer vision and geometric data analysis, with applications in deformation transfer [42] and statistical shape modeling [6] among other domains. During the past decades, a large number of techniques have been proposed for both rigid and non-rigid shape matching [44]. The latter case is both more general and more challenging since the shapes can potentially undergo arbitrary deformations (See Figure 1), which are not easy to characterize by purely axiomatic approaches. As a result, several recent learning-based techniques have been proposed for addressing the shape correspondence problem, e.g. [10, 25, 26, 51] among many others. Most of these approaches are based on the idea that the underlying correspondence model can be learned from data, typically given in the form of ground truth corre-

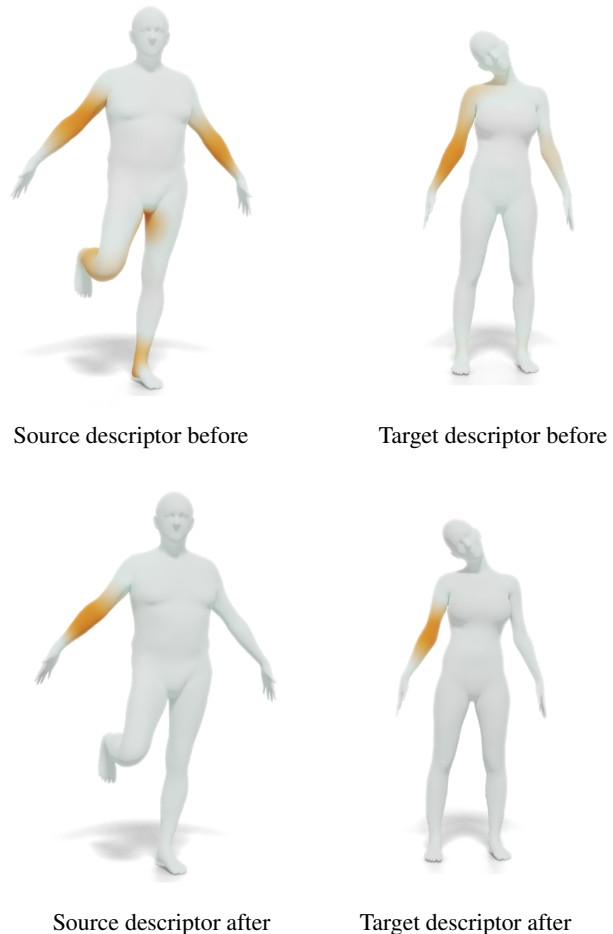


Figure 1: Given a pair of shapes with noisy descriptors (top), our approach makes them more consistent (bottom) without the knowledge of the underlying map, and automatically computes an accurate pointwise correspondence.

spondences between some shape pairs. In the simplest case, this can be formulated as a labeling problem, where different points, e.g., in a template shape, correspond to labels to be predicted [51, 27]. More recently, several methods have been proposed for *structured map prediction*, aiming to infer an entire map, rather than labeling each point in-

dependently [10, 23]. These techniques are based on learning pointwise descriptors, but, crucially, impose a penalty on the entire map, obtained using these descriptors, resulting in higher quality, globally consistent correspondences. Nevertheless, while learning-based methods have achieved impressive performance, their utility is severely limited by requiring the presence of high-quality ground truth maps between a sufficient number of training examples. This makes it difficult to apply such approaches to new shape classes for which ground truth data is not available.

In our paper, we show that this limitation can be lifted and propose a purely unsupervised strategy, which combines the accuracy of learning-based methods with the generality of axiomatic techniques for shape correspondence. The key to our approach is a bi-level optimization scheme, which optimizes for descriptors on the shapes, but imposes a penalty on the entire map, inferred from them. For this, we use the recently proposed FMNet architecture [23], which exploits the functional map representation [30]. However, rather than penalizing the deviation of the map from the ground truth, we enforce structural properties on the map, such as its bijectivity or approximate isometry. This results in a shape matching method that achieves state-of-the-art accuracy among unsupervised methods and, perhaps surprisingly, achieves comparable performance even to supervised techniques.

2. Related Work

Computing correspondences between 3D shapes is a very well-studied area of computer vision and computer graphics. Below we only review the most closely related methods and refer the interested readers to recent surveys including [46, 44, 5] for more in-depth discussions.

Functional Maps Our method is built on the functional map representation, which was originally introduced in [30] for solving non-rigid shape matching problems, and then extended significantly in follow-up works, including [2, 21, 20, 9, 15, 36] among many others (see also [31] for a recent overview).

One of the key benefits of this framework is that it allows us to represent maps between shapes as small matrices, which encode relations between basis functions defined on the shapes. Moreover, as observed by several works in this domain [30, 40, 21, 36, 9], many natural properties on the underlying pointwise correspondences can be expressed as objectives on functional maps. This includes orthonormality of functional maps, which corresponds to the local area-preservation nature of pointwise correspondences [30, 21, 40]; commutativity with the Laplacian operators, which corresponds to intrinsic isometries [30], preservation of inner products of gradients of functions, which corresponds to conformal maps [40, 9, 50]; preservation of *point-*

wise products of functions, which corresponds to functional maps arising from point-to-point correspondences [29, 28]; and slanted diagonal structure of functional map in the context of partial shapes [36, 24] among others.

Similarly, several other regularizers have been proposed, including exploiting the relation between functional maps in different directions [14], the map adjoint [18], and powerful cycle-consistency constraints [17] in shape collections to name a few. More recently constraints on functional maps have been introduced to promote map *continuity* [35, 34] and kernel-based techniques for extracting more information from given descriptors [49] among others. All these methods, however, are based on combining first-order penalties that arise from enforcing *descriptor preservation constraints* with these additional desirable structural properties of functional maps. As a result, any artefact or inconsistency in the pre-computed descriptors will inevitably lead to severe map estimation errors. Several methods have been suggested to use robust norms [21, 20], which can help reduce the influence of certain descriptors but still does not control the global map consistency properties.

Most recently, a powerful technique BCICP, for map optimization, was introduced in [35] that combines a large number of functional constraints with sophisticated post-processing, and careful descriptor selection. As we show below our method is simpler, more efficient and achieves superior accuracy even to this recent approach.

Learning-based Methods To overcome the inherent difficulty of axiomatic techniques, several methods have been introduced to learn the correct deformation model from data with learning-based methods. Some early approaches in this direction were used to learn either optimal parameters of spectral descriptors [25] or exploited random forests [38] or metric learning [11] for learning optimal constraints given some ground truth matches.

More recently, with the advent of deep learning methods, several approaches have been proposed to learn transformations in the context of non-rigid shape matching. Most of the proposed methods either use Convolutional Neural Networks (CNNs) on depth maps, e.g. for dense human body correspondence [51] or exploit extensions of CNNs directly to curved surfaces, either using the link between convolution and multiplication in the spectral domain [7, 12], or directly defining local parametrizations, for example via the exponential map, which allows convolution in the tangent plane of a point, e.g. [26, 8, 27, 33] among others.

These methods have been applied to non-rigid shape matching, in most cases modeling it as a label prediction problem, with points corresponding to different labels. Although successful in the presence of sufficient training data, such approaches typically do not impose global consistency, and can lead to artefacts, such as outliers, requiring post-processing to achieve high-quality maps.

Learning for Structured Prediction Most closely related to our approach are recent works that apply learning for structured map prediction [10, 23]. These methods learn a transformation of given input descriptors, while optimizing for the deviation of the map computed from them using the functional map framework, from ground truth correspondences. By imposing a penalty on entire maps, and thus evaluating the ultimate use of the descriptors, these methods have led to significant accuracy improvements in practice. We note that concurrent to our work, Halimi et al. [16] also proposed an unsupervised deep learning method that computes correspondences without using the ground truth. This approach is similar to ours, but is based on computation of geodesic distances, while our method operates purely in the spectral domain making it extremely efficient.

Contribution Unlike these existing methods, we propose an *unsupervised* learning-based approach that transforms given input descriptors, while optimizing for structural map properties, without any knowledge of the ground truth or geodesic distances. Our method, which can be seen as a bi-level optimization strategy, allows to explicitly control the interaction between pointwise descriptors and global map consistency, computed via the functional map framework. As a result, our technique is scalable with respect to shape complexity, leads to significant improvement compared to the standard unsupervised methods, and achieves comparable performance even to supervised approaches.

3. Background & Motivation

3.1. Shape Matching and Functional Maps

Our work is based on the functional map framework and representation. For completeness, we briefly review the basic notions and pipeline for estimating functional maps, and refer the interested reader to a recent course [31] for a more in-depth discussion.

Basic Pipeline Given a pair of shapes, S_1, S_2 represented as triangle meshes, and containing, respectively, n_1 and n_2 vertices, the basic pipeline for computing a map between them using the functional map framework, consists of the following main steps (see Chapter 2 in [31]):

1. Compute a small set of k_1, k_2 of basis functions on each shape, e.g. by taking the first few eigenfunctions of the respective Laplace-Beltrami operators.
2. Compute a set of descriptor *functions* on each shape that are expected to be approximately preserved by the unknown map. For example, a descriptor function can correspond to a particular dimension (e.g. choice of time parameter of the Heat Kernel Signature [43]) computed at every point. Store their coefficients in the respective bases as columns of matrices $\mathbf{A}_1, \mathbf{A}_2$.

3. Compute the optimal *functional map* \mathbf{C} by solving the following optimization problem:

$$\mathbf{C}_{\text{opt}} = \arg \min_{\mathbf{C}_{12}} E_{\text{desc}}(\mathbf{C}_{12}) + \alpha E_{\text{reg}}(\mathbf{C}_{12}), \quad (1)$$

where the first term aims at the descriptor preservation: $E_{\text{desc}}(\mathbf{C}_{12}) = \|\mathbf{C}_{12}\mathbf{A}_1 - \mathbf{A}_2\|^2$, whereas the second term regularizes the map by promoting the correctness of its overall structural properties. The simplest approach penalizes the failure of the unknown functional map to commute with the Laplace-Beltrami operators:

$$E_{\text{reg}}(\mathbf{C}_{12}) = \|\mathbf{C}_{12}\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2\mathbf{C}_{12}\|^2 \quad (2)$$

where $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are diagonal matrices of the Laplace-Beltrami eigenvalues on the two shapes.

4. Convert the functional map \mathbf{C} to a point-to-point map, for example using nearest neighbor search in the spectral embedding, or using other more advanced techniques [37, 15].

One of the strengths of this pipeline is that typically Eq. (1) leads to a simple (e.g., least squares) problem with $k_1 k_2$ unknowns, independent of the number of points on the shapes. This formulation has been extended using e.g. manifold optimization [22], descriptor preservation constraints via commutativity [29] and, more recently, with kernelization [49] among many others (see also Chapter 3 in [31]).

3.2. Deep Functional Maps

Despite its simplicity and efficiency, the functional map estimation pipeline described above is fundamentally dependent on the initial choice of descriptor functions. To alleviate this dependence, several approaches have been proposed to learn the optimal descriptors from data [10, 23]. In our work, we build upon a recent deep learning-based framework, called FMNet, introduced by Litany et al. [23] that aims to transform a given set of descriptors so that the optimal map computed using them is as close as possible to some ground truth map given during training.

Specifically, the approach proposed in [23] assumes, as input, a set of shape pairs for which ground truth point-wise maps are known, and aims to solve the following problem:

$$\min_T \sum_{(S_1, S_2) \in \text{Train}} l_F(\text{Soft}(\mathbf{C}_{\text{opt}}), GT_{(S_1, S_2)}), \quad \text{where} \quad (3)$$

$$\mathbf{C}_{\text{opt}} = \arg \min_{\mathbf{C}} \|\mathbf{C}\mathbf{A}_{T(D_1)} - \mathbf{A}_{T(D_2)}\|. \quad (4)$$

Here T is a non-linear transformation, in the form of a neural network, to be applied to some input descriptor functions D , Train is the set of training pairs for which ground truth correspondence $GT_{(S_1, S_2)}$ is known, l_F is the *soft error loss*, which penalizes the deviation of the computed functional map \mathbf{C}_{opt} , after converting it to a soft map $\text{Soft}(\mathbf{C}_{\text{opt}})$

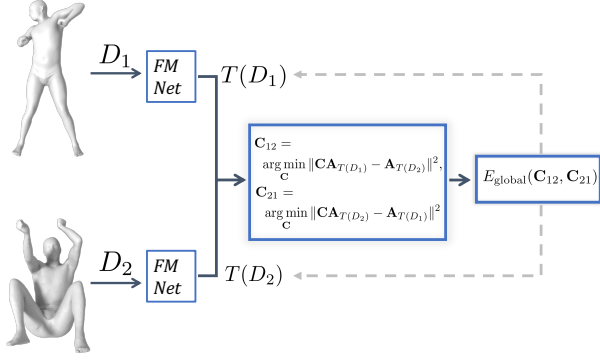


Figure 2: Overview of our SURFMNet approach: given a pair of shapes and their descriptors D_1, D_2 , we optimize for a non-linear transformation T using the FMNet architecture so that the transformed descriptors lead to functional maps that best satisfy the structural constraints.

from the ground truth correspondence, and $\mathbf{A}_{T(D_1)}$ denotes the transformed descriptors D_1 written in the basis of shape 1. In other words, the FMNet framework [23] aims to learn a transformation T of descriptors, so that the transformed descriptors $T(D_1), T(D_2)$, when used within the functional map pipeline result in a soft map that is as close as possible to some known ground truth correspondence. Unlike methods based on formulating shape matching as a labeling problem this approach evaluates the quality of the *entire map*, obtained using the transformed descriptors, which as shown in [23] leads to significant improvement compared to several strong baselines.

Motivation Similarly to other supervised learning methods, although FMNet [23] can result in highly accurate correspondences, its applicability is limited to shape classes for which high-quality ground truth maps are available. Moreover, perhaps less crucially, the soft map loss in FMNet is based on the knowledge of geodesic distances between all pairs of points, making it computationally expensive. Our goal, therefore, is to show that a similar approach can be used more widely, without any training data, while working purely in the spectral domain.

4. SURFMNet

4.1. Overview

In this paper, we introduce a novel approach, which we call **SURFMNet** for Spectral Unsupervised FMNet. Our method aims to optimize for non-linear transformations of descriptors, in order to obtain high-quality functional, and thus pointwise maps. For this, we follow the general strategy proposed in FMNet [23]. However, crucially, rather than penalizing the deviation of the computed map from the known ground truth correspondence, we evaluate the

structural properties of the inferred functional maps, such as their bijectivity or orthogonality. Importantly, we express all these desired properties, and thus the penalties during optimization, purely in the spectral domain, which allows us to avoid the conversion of functional maps to soft maps during optimization as was done in [23]. Thus, in addition to being purely unsupervised, our approach is also more efficient since it does not require pre-computation of geodesic distance matrices or expensive manipulation of large soft map matrices during training.

To achieve these goals, we build on the FMNet model, described in Eq. (3) and (4) in several ways: first, we propose to consider functional maps in both directions, i.e. by treating the two shapes as both source and target; second, we remove the conversion from functional to soft maps; and, most importantly, third, we replace the soft map loss with respect to ground truth with a set of penalties on the computed functional maps, which are described in detail below. Our optimization problem can be written as:

$$\min_T \sum_{(S_1, S_2)} \sum_{i \in \text{penalties}} w_i E_i(\mathbf{C}_{12}, \mathbf{C}_{21}), \text{ where} \quad (5)$$

$$\mathbf{C}_{12} = \arg \min_{\mathbf{C}} \|\mathbf{C}\mathbf{A}_{T(D_1)} - \mathbf{A}_{T(D_2)}\|, \quad (6)$$

$$\mathbf{C}_{21} = \arg \min_{\mathbf{C}} \|\mathbf{C}\mathbf{A}_{T(D_2)} - \mathbf{A}_{T(D_1)}\|. \quad (7)$$

Here, similarly to Eq. (3) above, T denotes a non-linear transformation in the form of a neural network, (S_1, S_2) is a set of pairs of shapes in a given collection, w_i are scalar weights, and E_i are the penalties, described below. Thus, we aim to optimize for a non-linear transformation of input descriptor functions, such that functional maps computed from transformed descriptors possess certain desirable structural properties and are expressed via penalty minimization. Figure 2 illustrates our proposed method where we denote the total sum of all penalty terms in Eq. (5) as E_{global} and back-propagation via grey dashed lines.

When deriving the penalties used in our approach, we exploit the links between properties of functional maps and associated pointwise maps, that have been established in several previous works [30, 40, 14, 29]. Unlike all these methods, however, we *decouple the descriptor preservation constraints from structural map properties*. This allows us to optimize for descriptor functions, and thus, gain a very strong resilience in the presence of noisy or uninformative descriptors, while still exploiting the compactness and efficiency of the functional map representation.

4.2. Deep Functional Map Regularization

In our work, we propose to use four regularization terms, by including them as a penalties in the objective function, all inspired by desirable map properties.

Bijectivity Given a pair of shapes and the functional maps

in both directions, perhaps the simplest requirement is for them to be inverses of each other, which can be enforced by penalizing the difference between their composition and the identity map. This penalty, used for functional map estimation in [14], can be written, simply as:

$$E_1 = \|\mathbf{C}_{12}\mathbf{C}_{21} - \mathbf{I}\|^2 + \|\mathbf{C}_{21}\mathbf{C}_{12} - \mathbf{I}\|^2 \quad (8)$$

Orthogonality As observed in several works [30, 40] a point-to-point map is locally area preserving if and only if the corresponding functional map is *orthonormal*. Thus, for shape pairs, approximately satisfying this assumption, a natural penalty in our unsupervised pipeline is:

$$E_2 = \|\mathbf{C}_{12}^\top \mathbf{C}_{12} - \mathbf{I}\|^2 + \|\mathbf{C}_{21}^\top \mathbf{C}_{21} - \mathbf{I}\|^2 \quad (9)$$

Laplacian commutativity Similarly, it is well-known that a pointwise map is an intrinsic isometry if and only if the associated functional map commutes with the Laplace-Beltrami operator [39, 30]. This has motivated using the lack of commutativity as a regularizer for functional map computations, as mentioned in Eq. (2). In our work, we use it to introduce the following penalty:

$$E_3 = \|\mathbf{C}_{12}\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2\mathbf{C}_{12}\|^2 + \|\mathbf{C}_{21}\mathbf{\Lambda}_2 - \mathbf{\Lambda}_1\mathbf{C}_{21}\|^2 \quad (10)$$

where $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are diagonal matrices of the Laplace-Beltrami eigenvalues on the two shapes.

Descriptor preservation via commutativity The previous three penalties capture desirable properties of pointwise correspondences when expressed as functional maps. Our last penalty promotes functional maps that arise from point-to-point maps, rather than more general soft correspondences. To achieve this, we follow the approach proposed in [29] based on preservation of pointwise products of functions. Namely, it is known that a non-trivial linear transformation \mathcal{T} across function spaces corresponds to a point-to-point map if and only if $\mathcal{T}(f \odot h) = \mathcal{T}(f) \odot \mathcal{T}(h)$ for any pair of functions f, h . Here \odot denotes the pointwise product between functions [41], i.e. $(f \odot h)(x) = f(x)h(x)$. When f is a descriptor function on the source and g is the corresponding descriptor on the target, the authors of [29] demonstrate that this condition can be rewritten in the reduced basis as follows: $\mathbf{C}\mathbf{M}_f = \mathbf{M}_g\mathbf{C}$, where $\mathbf{M}_f = \Phi^\top \text{Diag}(f)\Phi$, and $\mathbf{M}_g = \Psi^\top \text{Diag}(g)\Psi$. This leads to the following penalty, in our setting:

$$E_4 = \sum_{(f_i, g_i) \in \text{Descriptors}} \left(\|\mathbf{C}_{12}\mathbf{M}_{f_i} - \mathbf{M}_{g_i}\mathbf{C}_{12}\|^2 + \|\mathbf{C}_{21}\mathbf{M}_{g_i} - \mathbf{M}_{f_i}\mathbf{C}_{21}\|^2 \right), \quad (11)$$

$$\mathbf{M}_{f_i} = \Phi^\top \text{Diag}(f_i)\Phi, \mathbf{M}_{g_i} = \Psi^\top \text{Diag}(g_i)\Psi.$$

In this expression, f_i and g_i are the *optimized* descriptors on source and target shape, obtained by the neural network,

and expressed in the full (hat basis), whereas Φ, Ψ are the fixed basis functions on the two shapes, and $+$ denotes the Moore-Penrose pseudoinverse.

4.3. Optimization

As mentioned in Section 4.1, we incorporate these four penalties into the energy in Eq. (5). Importantly, the only unknowns in this optimization are the parameters of the neural network applied to the descriptor functions. The functional maps \mathbf{C}_{12} and \mathbf{C}_{21} are fully determined by the optimized descriptors via the solution of the optimization problems in Eq. (6) and Eq. (7). Note that although stated as optimization problems, both Eq. (6) and Eq. (7) reduce to solving a linear system of equations. This is easily differentiable using the well-known closed-form expression for derivatives of matrix inverses [32]. Moreover, the functionality of differentiating a linear system of equations is implemented in TensorFlow [1] and we use it directly, in the same way as it was used in the original FMNet work. Finally, all of the penalties E_1, E_2, E_3, E_4 are differentiable with respect to the functional maps $\mathbf{C}_{12}, \mathbf{C}_{21}$. This means that the gradient of the total energy can be back-propagated to the neural network T in Eq. (5), allowing us to optimize for the descriptors while penalizing the structural properties of the functional maps.

5. Implementation & Parameters

Implementation details We implemented ¹ our method in TensorFlow [1] by adapting the open-source implementation of FMNet [23]. Thus, the neural network T used for transforming descriptors in our approach, in Eq. (5) is exactly identical to that used in FMNet, as mentioned in Eq. (3). Namely, this network is based on a residual architecture, consisting of 7 fully connected residual layers with exponential linear units, without dimensionality reduction. Please see Section 5 in [23] for more details.

Following the approach of FMNet [23], we also sub-sample a random set of 1500 points at each training step, for efficiency. However, unlike their method, sub-sampling is done independently on each shape, without enforcing consistency. Remark that our network is fully connected on the dimensions of the descriptors, not across vertices themselves. For example, the first layer has 352×352 weights (not 1500×352 weights) where 352 and 1500 are the dimensions of the SHOT descriptors, and no. of sampled vertices respectively. Indeed, in exactly the same way as in FMNet, our network is applied on the descriptors of each point independently, using the same (learned) weights, and different points on the shape only communicate through the functional map estimation layer, and not in the MLP layers. This

¹Code available at <https://github.com/LIX-shape-analysis/SURFMNet>.

Methods	E1+E2+E3+E4	E3	E1	E2	E4
Geodesic Error	0.020	0.073	0.083	0.152	0.252

Table 1: Ablation study of penalty terms in our method on the FAUST benchmark.

ensures invariance to permutation of shape vertices. We also randomly sub-sample 20% of the optimized descriptors for our penalty E_4 at each training step to avoid manipulating a large set of operators. We observed that this sub-sampling not only helps to gain speed but also robustness during optimization. Importantly, we do not form large diagonal matrices explicitly, but rather define the multiplicative operators \mathbf{M} in objective E_4 directly via pointwise products and summation using contraction between tensors.

Finally, we convert functional maps to pointwise ones with nearest neighbor search in the spectral domain, following the original approach [30].

Parameters Our method takes two types of inputs: the input descriptors, and the scalar weights w_i in Eq. (5). In all experiments below, we used the same SHOT [45] descriptors as in FMNet [23] with the same parameters, which leads to a 352-dimensional vector per point, or equivalently, 352 descriptor functions on each shape. For the scalar weights, w_i , we used the same four fixed values for all experiments below (namely, $w_1 = 10^3$, $w_2 = 10^3$, $w_3 = 1$ and $w_4 = 10^5$), which were obtained by examining the relative penalty values obtained throughout the optimization on a small set of shapes, and setting the weights inversely proportionally to those values. We train our network with a batch size of 10 for 10 000 iterations using a learning rate of 0.001 and ADAM optimizer [13].

6. Results

Datasets We evaluate our method on the following datasets: the original FAUST dataset [6] containing 100 human shapes in 1-1 correspondence and the remeshed versions of SCAPE [3] and FAUST [6] datasets, made publicly available recently by Ren et al. [35]. These datasets were obtained by independently re-meshing each shape to approximately 5000 vertices using the LRVD re-meshing method [52], while keeping track of the ground truth maps within each collection. This results in meshes that are no longer in 1-1 correspondence, and indeed can have different number of vertices. The re-meshed datasets therefore offer significantly more variability in terms of shape structures, including e.g. point sampling density, making them more challenging for existing algorithms. Let us note also that the SCAPE dataset is slightly more challenging since the shapes are less regular (e.g., there are often reconstruction artefacts on hands and feet) and have fewer features than those in FAUST.

We stress that although we also evaluated on the original FAUST dataset, we view the remeshed datasets as more realistic, providing a more faithful representation of the accuracy and generalization power of different techniques.

Ablation study We first evaluated the relative importance of the different penalties in our method on the FAUST shape dataset [6]. We evaluated the average correspondence geodesic error with respect to the ground truth maps.

Table 1 summarizes the quality of the computed correspondences between shapes in the test set, using different combination of penalties. We observe that the combination of all four penalties significantly out-performs any other subsets. Besides, among individual penalties used independently, the Laplacian commutativity gives the best result. For more combinations of penalty terms, we refer to a more detailed ablation study in the supplementary material.

Baselines We compared our method to several techniques, both supervised and fully automatic. For conciseness, we refer to SURFMNet as **Ours** in the following text. For a fair comparison with FMNet, we evaluate our method in two settings: **Ours-sub** and **Ours-all**. For Ours-sub, we split each dataset into training and test sets containing 80 and 20 shapes respectively, as done in [23]. For Ours-all, we optimize over all the dataset and apply the optimized network on the same test set as before. We stress that unlike FMNet, our method does not use any ground truth in either setting. We use the notation Ours-sub only to emphasize the split of dataset into train and test since the “training set” was only used for descriptor optimization with the functional map penalties introduced above without any ground truth.

Since the original FMNet work [23] already showed very strong improvement compared to existing supervised learning methods we primarily compare to this approach. For reference, we also compare to the Geodesic Convolutional Neural Networks (GCNN) method of [26] on the remeshed datasets, which were not considered in [23]. GCNN is a representative supervised method based on local shape parameterization, and as FMNet assumes, as input, ground truth maps between a subset of the training shapes. For supervised methods, we always split the datasets into 80 (resp. 60) shapes for training and 20 (resp. 10) for testing in the FAUST and SCAPE datasets respectively.

Among fully automatic methods, we use the Product Manifold Filter method with the Gaussian kernel [48] (PMF Gauss) and its variant with the Heat kernel [47] (PMF Heat). We also compare to the recently proposed BCICP [35], which achieved state-of-the-art results among axiomatic methods. With a slight abuse of notation, we denote these non-learning methods as Unsupervised in Figure 4 since none of these methods use ground truth. Finally, we also evaluated the basic functional map approach, based on directly optimizing the functional maps as outlined in Sec-

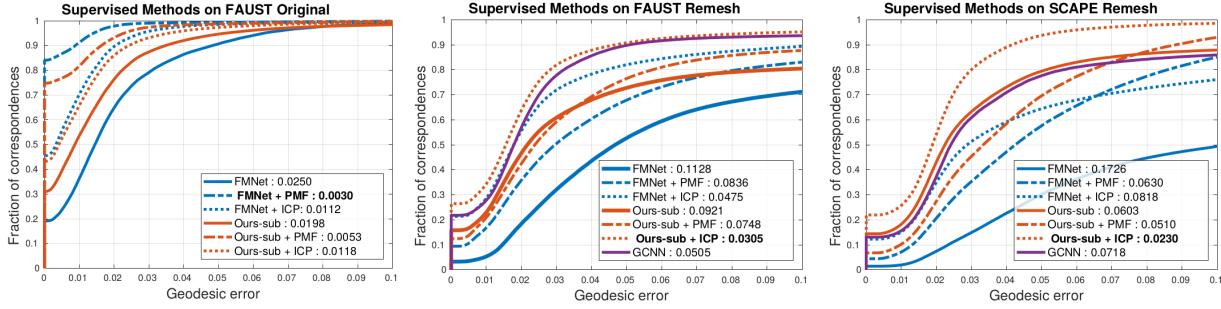


Figure 3: Quantitative evaluation of pointwise correspondences comparing our method with Supervised Methods.

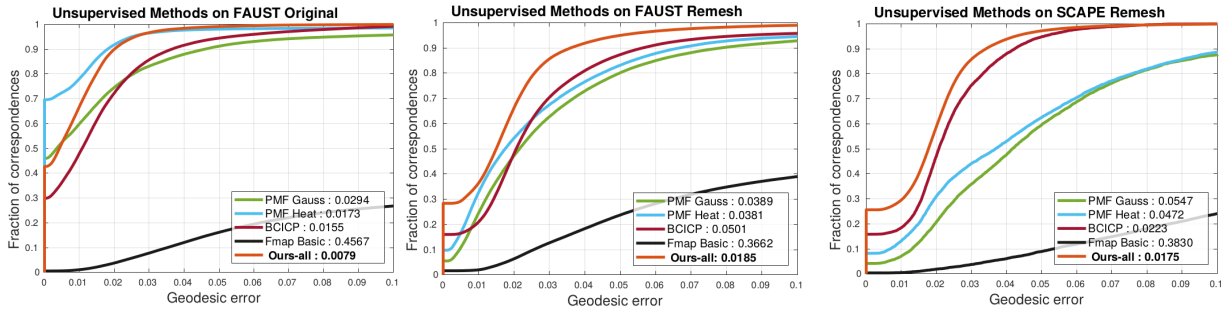


Figure 4: Quantitative evaluation of pointwise correspondences comparing our method with Unsupervised Methods.

tion 3.1, but using all four of our energies for regularization. This method, which we call “Fmap Basic” can be viewed as a combination of the approaches of [14] and [28], as it incorporates functional map coupling (via energy E_1) and descriptor commutativity (via E_4). Unlike our technique, however, it operates on fixed descriptor functions, and uses descriptor preservation constraints with the original and noisy descriptors.

For fairness of comparison, we used SHOT descriptors [45] as input to all methods, except BCICP [35], which uses carefully curated WKS [4] descriptors. Furthermore, we consider the results of FMNet [23] before and after applying the PMF-based post-processing as suggested in the original article. We also report results with ICP post-processing introduced in [30]. Besides the accuracy plots shown in Figures 3 and 4, we also include statistics such as maximum and 95th percentile in supplementary material.

6.1. Evaluation and Results

Figure 3 summarizes the accuracy obtained by supervised methods on the three datasets whereas Figure 4 compares with unsupervised methods, using the evaluation protocol introduced in [19]. Note that in all cases, our network SURFMNet, (Ours-all), when optimized on all shapes achieves the best results even compared to the recent state-of-the-art method in [35]. Furthermore, our method is comparable even to supervised learning techniques, GCNN [7] and FMNet [23] despite being purely unsupervised.

Remark that the remeshed datasets are significantly harder for both supervised and unsupervised methods, since the shapes are no longer identically meshed and in 1-1 correspondence. We have observed this difficulty also while training supervised FMNet and GCNN techniques with very slow convergence during training. On both of these datasets, our approach achieves the lowest average error, reported in Figure 3 and 4. Note that on the remeshed FAUST dataset, as shown in Figure 3, only GCNN [7] produces a similarly large fraction of correspondences with a small error. However, this method is *supervised*. On the remeshed SCAPE dataset, our method leads to the best results across all measures, despite being purely unsupervised.

Postprocessing Results As shown in Figures 3 and 4 our method can often obtain high quality results even without any post-processing. Nevertheless, in the challenging cases such as the SCAPE remeshed dataset, when trained on a subset of shapes, it can also benefit from an efficient ICP-based refinement. This refinement, does not require computing geodesic distances and does not require the shapes to have the same number of points, thus maintaining the flexibility and efficiency of our pipeline.

Correlation with actual Geodesic loss We further investigated if there is a correlation between the value of our loss and the quality of correspondence. Specifically, whether minimizing our loss function, mainly consisting of regularization terms on estimated functional maps, corresponds to

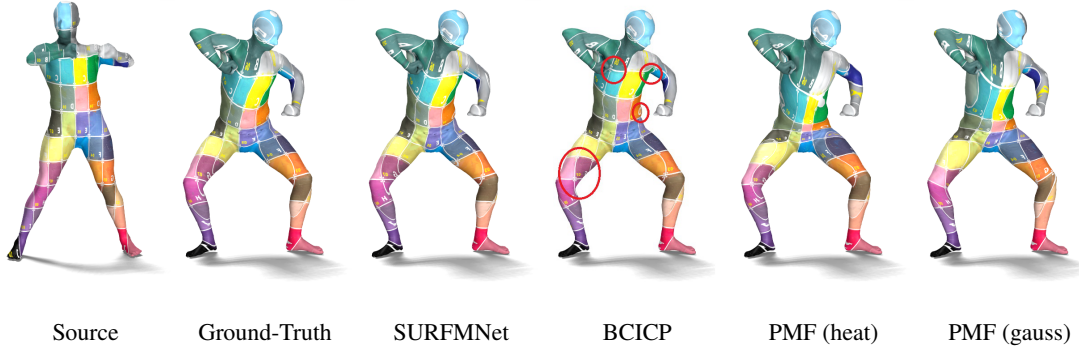


Figure 5: Comparison of our method with *Unsupervised* methods for texture transfer on the SCAPE remeshed dataset. Note that BCICP is roughly 7 times slower than our method and its shortcomings are marked with red circles.

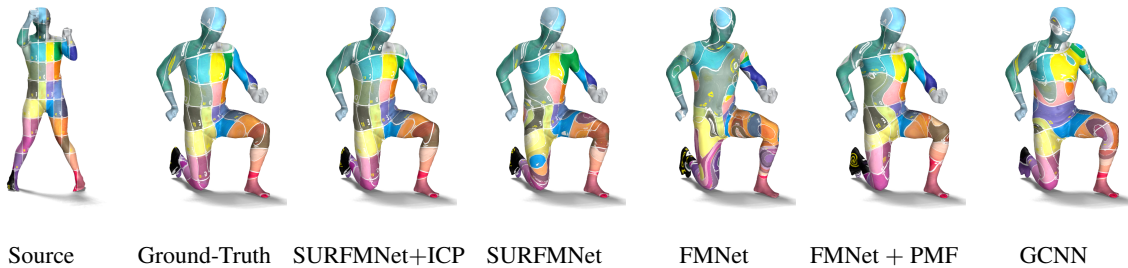


Figure 6: Comparison of our method with *Supervised* method for texture transfer on the SCAPE remeshed dataset.

Methods	Runtime				Total
	Pre-processing	Training	Testing	Post-processing	
FMNet	60s	1500s	0.3s	N/As	1650s
FMNet + PMF	60s	1500s	0.3s	30s	1680s
Fmap Basic	10s	N/A	60s	N/A	120s
BCICP	N/A	N/A	60s	180s	240s
SURFMNet	10s	25s	0.3s	N/A	35s
SURFMNet + ICP	10s	25s	0.3s	10s	45s

Table 2: Runtime of different methods averaged over 190 shape pairs.

minimizing the geodesic loss with respect to the unknown ground truth map. We found strong correlation between the two and share a plot in the supplementary material.

Qualitative and Runtime Comparison Figures 5 and 6 show examples shape pairs and maps obtained between them using different methods, visualized via texture transfer. Note the continuity and quality of the maps obtained using our method, compared to other techniques (more results in supplementary material). One further advantage of our method is its efficiency, since we do not rely on the computation of geodesic matrices and operate entirely in the spectral domain. Table 2 compares the run-time of the best performing methods on an Intel Xeon 2.10GHz machine with an NVIDIA Titan X GPU. Note that our method is over an order of magnitude faster than FMNet and significantly faster than the currently best unsupervised BCICP.

7. Conclusion & Future Work

We presented an unsupervised method for computing correspondences between shapes. Key to our approach is a bi-level optimization formulation, aimed to optimize descriptor functions, while promoting the structural properties of the entire map, obtained from them via the functional maps framework. Remarkably, our approach achieves similar, and in some cases superior performance even to supervised correspondence techniques.

In the future, we plan to incorporate other penalties on functional maps, e.g., those arising from recently-proposed kernalization approaches [49], or for promoting orientation preserving maps[35] and also incorporate cycle consistency constraints [17]. Finally, it would be interesting to extend our method to partial and non-isometric shapes and matching other modalities, such as images or point clouds, since it opens the door to linking the properties of local descriptors to global map consistency.

Acknowledgements Parts of this work were supported by the ERC Starting Grant StG-2017-758800 (EXPROTEA), KAUST OSR Award No. CRG-2017-3426, and a gift from Nvidia. We are grateful to **Jing Ren**, Or Litany, Emanuele Rodolà and Adrien Poulenard for their help in performing quantitative comparisons and producing qualitative results.

References

- [1] Martín Abadi, Ashish Agarwal, and Paul Barham et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. [5](#)
- [2] Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision*, 118(3):380–392, 2016. [2](#)
- [3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. [6](#)
- [4] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. 31(4), Nov. 2011. [7](#)
- [5] Silvia Biasotti, Andrea Cerri, A Bronstein, and M Bronstein. Recent trends, applications, and perspectives in 3d shape similarity assessment. In *Computer Graphics Forum*, volume 35, pages 87–119, 2016. [2](#)
- [6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE. [1](#), [6](#)
- [7] Davide Boscaini, Jonathan Masci, Simone Melzi, Michael M Bronstein, Umberto Castellani, and Pierre Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In *Computer Graphics Forum*, volume 34, pages 13–23. Wiley Online Library, 2015. [2](#), [7](#)
- [8] Davide Boscaini, Jonathan Masci, Emanuele Rodola, and Michael M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Proc. NIPS*, pages 3189–3197, 2016. [2](#)
- [9] Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. Embedding shapes with Green’s functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017. [2](#)
- [10] Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. Supervised descriptor learning for non-rigid shape matching. In *Proc. ECCV Workshops (NORDIA)*, 2014. [1](#), [2](#), [3](#)
- [11] Luca Cosmo, Emanuele Rodola, Jonathan Masci, Andrea Torsello, and Michael M Bronstein. Matching deformable objects in clutter. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 1–10. IEEE, 2016. [2](#)
- [12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016. [2](#)
- [13] J. Ba D.P. Kingma. Adam: A method for stochastic optimization. In *ICLR*, 2015. [6](#)
- [14] Davide Eynard, Emanuele Rodola, Klaus Glashoff, and Michael M Bronstein. Coupled functional maps. In *3D Vision (3DV)*, pages 399–407. IEEE, 2016. [2](#), [4](#), [5](#), [7](#)
- [15] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. In *Computer Graphics Forum*, volume 36, pages 165–174. Wiley Online Library, 2017. [2](#), [3](#)
- [16] Oshri Halimi, Or Litany, Emanuele Rodol’a, Alex Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019. [3](#)
- [17] Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)*, 33(4):36, 2014. [2](#), [8](#)
- [18] Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. In *Computer Graphics Forum*, volume 36, pages 151–163. Wiley Online Library, 2017. [2](#)
- [19] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. [7](#)
- [20] Artiom Kovnatsky, Michael M Bronstein, Xavier Bresson, and Pierre Vandergheynst. Functional correspondence by matrix completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 905–914, 2015. [2](#)
- [21] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448, 2013. [2](#)
- [22] Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Proc. ECCV*, pages 680–696. Springer, 2016. [3](#)
- [23] Or Litany, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5660–5668, 2017. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [24] Or Litany, Emanuele Rodolà, Alex M Bronstein, and Michael M Bronstein. Fully spectral partial shape

- matching. In *Computer Graphics Forum*, volume 36, pages 247–258. Wiley Online Library, 2017. 2
- [25] Roei Litman and Alexander M Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2014. 1, 2
- [26] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 1, 2, 6
- [27] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5425–5434. IEEE Computer Society, 2017. 1, 2
- [28] Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, M Bronstein, and Maks Ovsjanikov. Improved functional mappings via product preservation. In *Computer Graphics Forum*, volume 37, pages 179–190. Wiley Online Library, 2018. 2, 7
- [29] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017. 2, 3, 4, 5
- [30] 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, 2012. 2, 4, 5, 6, 7
- [31] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, SIGGRAPH '17, pages 5:1–5:62, 2017. 2, 3
- [32] K. B. Petersen and M. S. Pedersen. *The matrix cookbook*, 2012. 5
- [33] Adrien Poulencard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *arXiv preprint arXiv:1810.02303*, 2018. 2
- [34] Adrien Poulencard, Primoz Skraba, and Maks Ovsjanikov. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, volume 37, pages 13–25. Wiley Online Library, 2018. 2
- [35] Jing Ren, Adrien Poulencard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. 2, 6, 7, 8
- [36] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017. 2
- [37] Emanuele Rodolà, M Moeller, and Daniel Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)*, 2015. 3
- [38] Emanuele Rodolà, Samuel Rota Bulo, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4184, 2014. 2
- [39] Steven Rosenberg. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*, volume 31. Cambridge University Press, 1997. 5
- [40] Raif Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frederic Chazal, and Leonid Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Trans. Graphics*, 32(4):72:1–72:12, July 2013. 2, 4, 5
- [41] R K Singh and J S Manhas. *Composition Operators on Function Spaces*. ELSEVIER, 1993. 5
- [42] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 399–405. ACM, 2004. 1
- [43] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392, 2009. 3
- [44] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2013. 1, 2
- [45] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV)*, pages 356–369, 2010. 6, 7
- [46] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707, 2011. 2
- [47] Matthias Vestner, Zorah Löhner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola,

Alex Bronstein, Michael Bronstein, Ron Kimmel, and Daniel Cremers. Efficient deformable shape correspondence via kernel matching. In *Proc. 3DV*, 2017. 6

- [48] M. Vestner, R. Litman, E. Rodolà, A. Bronstein, and D. Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proc. CVPR*, pages 6681–6690, 2017. 6
- [49] Larry Wang, Anne Gehre, Michael M Bronstein, and Justin Solomon. Kernel functional maps. In *Computer Graphics Forum*, volume 37, pages 27–36. Wiley Online Library, 2018. 2, 3, 8
- [50] Yuexuan Wang, B Liu, K Zhou, and Yu Tong. Vector field map representation for near conformal surface correspondence. In *Computer Graphics Forum*, volume 37, pages 72–83. Wiley Online Library, 2018. 2
- [51] Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016. 1, 2
- [52] Dong-Ming Yan, Guanbo Bao, Xiaopeng Zhang, and Peter Wonka. Low-resolution remeshing using the localized restricted voronoi diagram. *IEEE transactions on visualization and computer graphics*, 20(10):1418–1427, 2014. 6

8. Supplement

A. Correlation with actual geodesic loss

To support the claim made in the subsection 'Evaluation and Results', we include a plot here to visualize the correlation between our loss and the actual geodesic loss. As evident in Figure 7, there is a strong correlation between our loss value and the quality of correspondence as measured by average geodesic error.

B. Detailed Tabular Quantitative Comparison

Besides the average geodesic error reported for quantitative comparison in Figures 3 and 4, we provide detailed statistics in Table 3. Note that Table 3 also includes 'Fmap Ours Opt' which is equivalent to "Fmap Basic" but uses the learned descriptors instead of original ones. Its competitive performance across all datasets proves quantitatively the utility of learning descriptors. Figures 13 and 14 illustrate this further. For completeness, in Table 4, we also provide a detailed ablation study with different combinations of penalties.

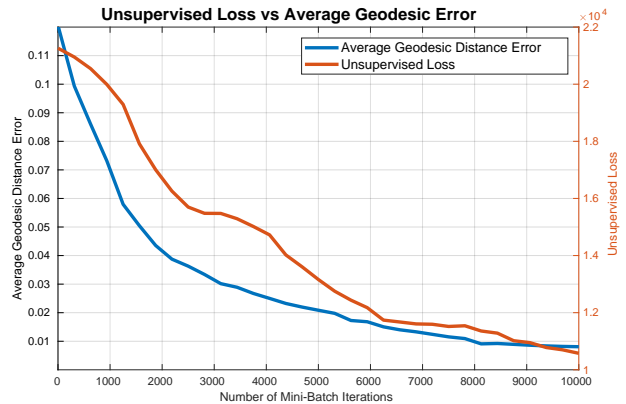


Figure 7: Correlation with average geodesic loss computed from ground truth correspondences.

C. Sensitivity to number of basis functions

Figure 8 shows the sensitivity of our network SURFM-Net on the SCAPE remeshed dataset as the number of eigenfunctions are varied from 20 to 150. We train the network each time with 10000 mini batch steps. As evident, we obtain best result using 120. However, when trained on an individual dataset and tested on a different one, we see over-fitting when using a large eigen-basis. We attribute this phenomenon to the initialization of our descriptors with SHOT which is a very local descriptor and is not robust to very strong mesh variability. However, over-fitting is minimal when we train together on a relatively larger subset of SCAPE and FAUST and test on a different subset of shapes from both datasets, with smaller eigen basis.

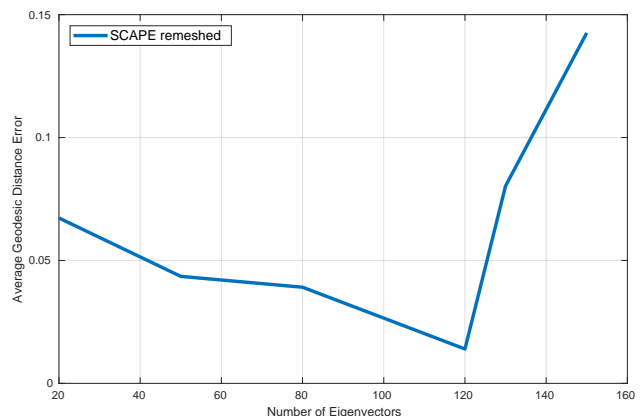


Figure 8: Accuracy of our method on the SCAPE remeshed dataset as the number of eigenfunctions is varied from 20 to 150.

(Results are $\times 10^{-3}$)									
Supervised Methods	FAUST 7k			FAUST 5k			SCAPE 5k		
	Mean	95th Percentile	Maximum	Mean	95th Percentile	Maximum	Mean	95th Percentile	Maximum
FMNet	25.01	63.11	1207.8	112.8	451.8	1280.6	172.6	543.8	1399.6
SURFMNet Subset	19.83	52.11	1204.0	92.09	493.6	1279.4	60.32	329.8	1068.7
FMNet + PMF	2.98	14.10	1222.7	83.61	395.7	1576.4	63.00	159.8	1561.5
SURFMNet-sub + PMF	5.33	22.90	1302.4	74.80	408.5	1619.3	51.03	111.5	1555.6
FMNet + ICP	11.16	27.91	1206.8	47.53	237.3	1348.6	81.76	341.4	1226.5
SURFMNet-sub + ICP	11.79	35.76	1088.4	30.47	95.64	1277.3	23.00	54.76	73.18
GCNN	-	-	-	50.49	206.3	1578.2	71.85	374.2	1523.7
Unsupervised Methods									
BCICP	15.46	53.27	572.4	31.08	64.51	1149.9	22.28	50.60	107.5
PMF (Gaussian Kernel)	29.42	83.80	1168.1	75.13	236.9	1632.7	54.68	156.9	465.1
PMF (Heat Kernel)	17.26	25.06	1168.1	31.08	64.51	1150.0	47.23	133.4	802.1
Fmap Basic	457.56	1171.4	1568.4	366.2	1159.0	1549.1	383.0	1043.7	1280.3
Fmap Ours Opt	9.75	30.02	420.2	20.19	53.24	1169.5	13.98	31.16	86.45
SURFMNet-all	7.89	26.01	572.4	18.56	50.25	1156.3	17.50	42.50	228.8

Table 3: Quantitative comparison on all three benchmark datasets for shape correspondence problem.

Methods	E1+E2+E3+E4	E3	E1+E2+E3	E1+E3+E4	E1	E2+E3+E4	E1+E2+E4	E2	E4	FMNet	Ours-Sub	Ours-all
Mean Geodesic Error	0.044	0.073	0.081	0.077	0.111	0.079	0.126	0.135	0.330	0.025	0.020	0.008

Table 4: Ablation study of penalty terms in our method and comparison with the supervised FMNet on the FAUST benchmark.

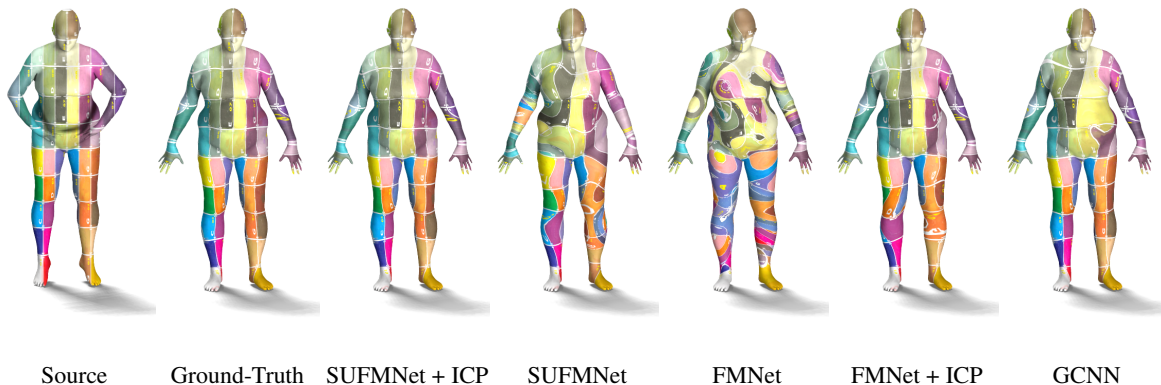


Figure 9: Comparison of our method with *Supervised* methods for texture transfer on the FAUST remeshed dataset.

D. More Qualitative Comparison

In Figures 9 and 12, we provide more qualitative comparisons of SURFMNet on the FAUST remeshed datasets whereas Figures 10 and 11 provide a comparison on the SCAPE remeshed dataset. In all cases, our method produces the highest quality maps.

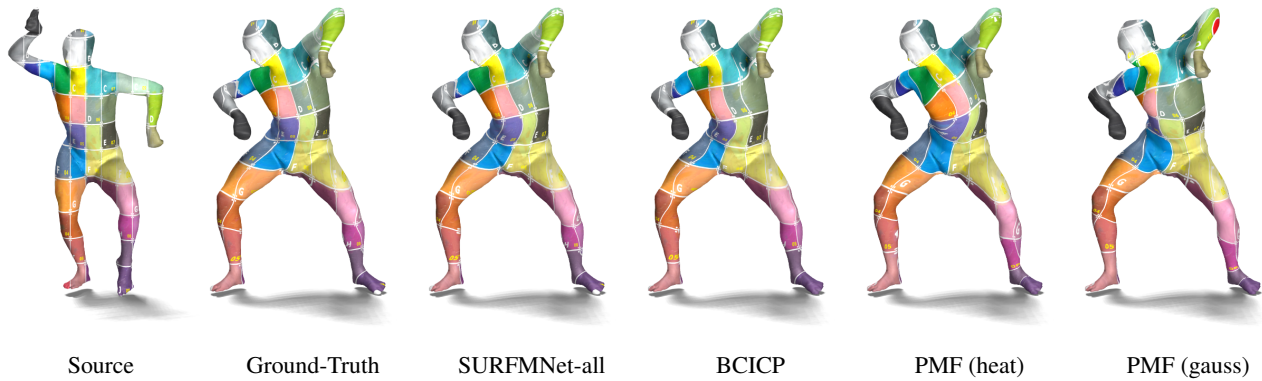


Figure 10: Comparison of our method with *Unsupervised* methods for texture transfer on the SCAPE remeshed dataset.

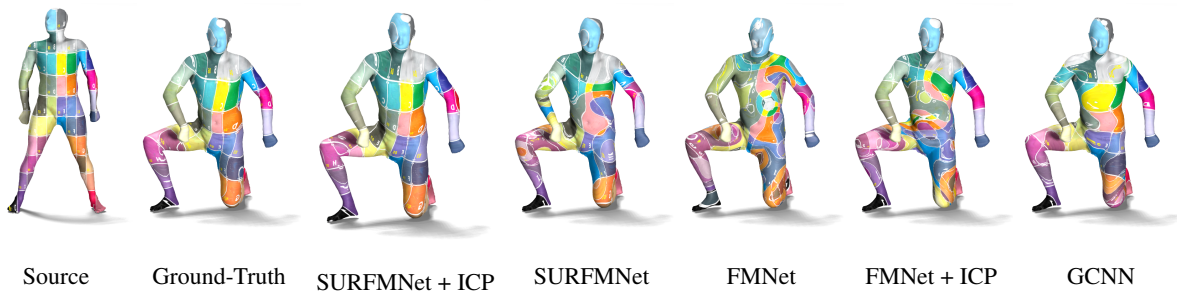


Figure 11: Comparison of our method with *Supervised* methods for texture transfer on the SCAPE remeshed dataset.

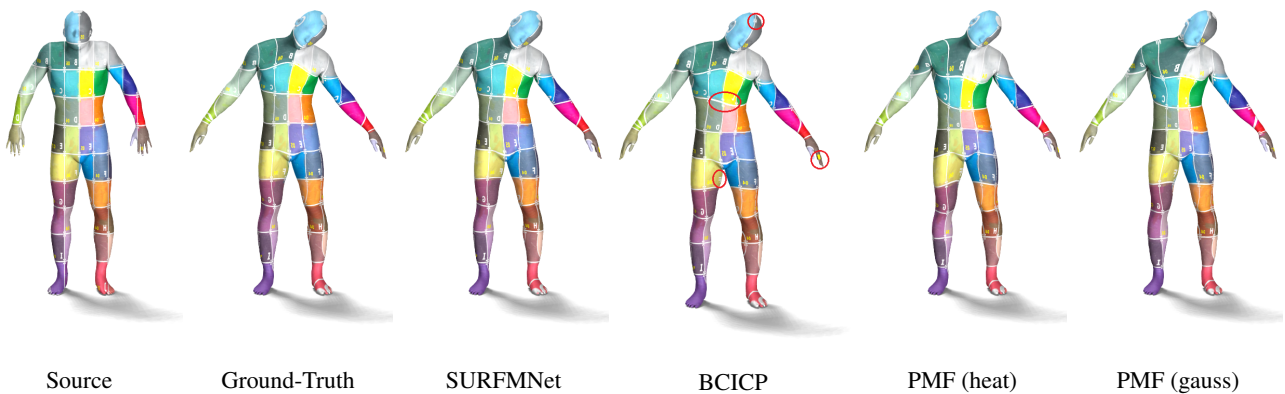


Figure 12: Comparison of our method with *Unsupervised* methods for texture transfer on the FAUST remeshed dataset. Note that BCICP is roughly 7 times slower when compared to our method. We highlight the shortcomings of BCICP matching with red circles.

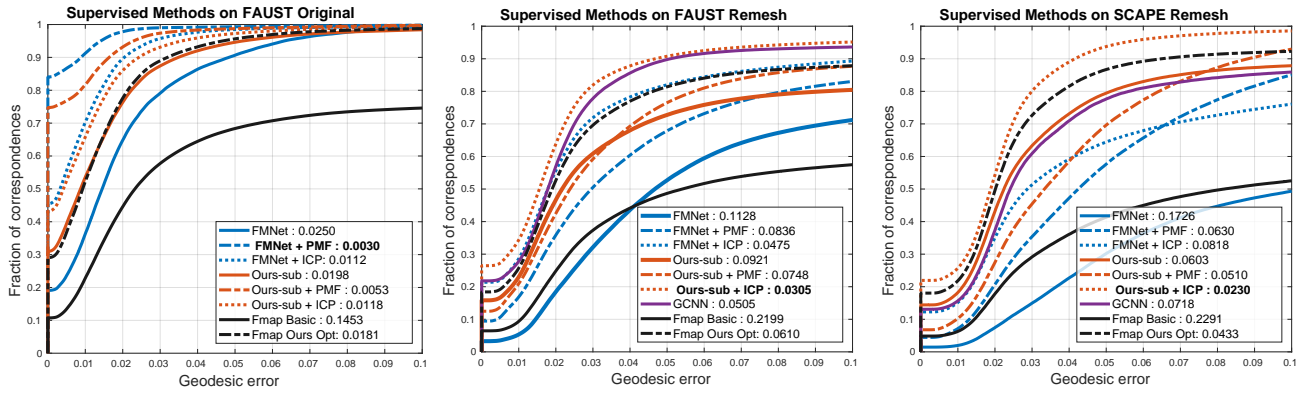


Figure 13: Quantitative evaluation of pointwise correspondences comparing our method with Supervised Methods.

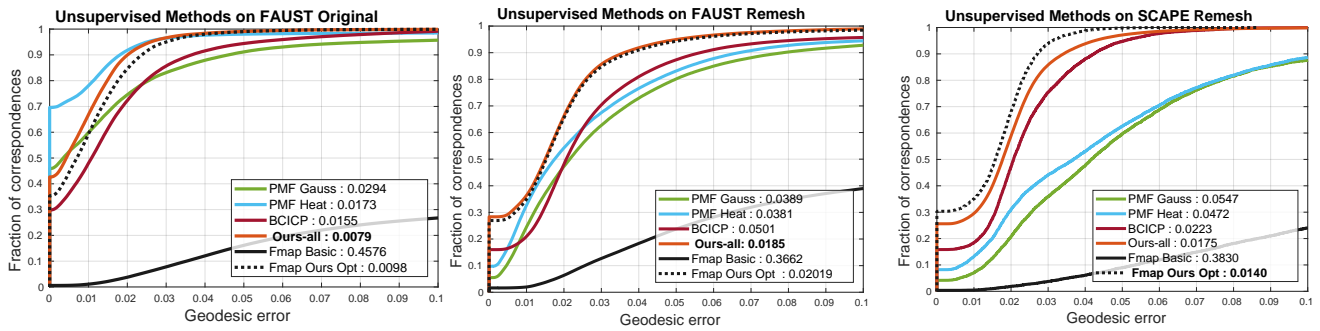


Figure 14: Quantitative evaluation of pointwise correspondences comparing our method with Unsupervised Methods.