Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence

Nicolas Donati
LIX, École Polytechnique
nicolas.donati@polytechnique.edu

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

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

Abstract

We present a novel learning-based approach for computing correspondences between non-rigid 3D shapes. Unlike previous methods that either require extensive training data or operate on handcrafted input descriptors and thus generalize poorly across diverse datasets, our approach is both accurate and robust to changes in shape structure. Key to our method is a feature-extraction network that learns directly from raw shape geometry, combined with a novel regularized map extraction layer and loss, based on the functional map representation. We demonstrate through extensive experiments in challenging shape matching scenarios that our method can learn from less training data than existing supervised approaches and generalizes significantly better than current descriptor-based learning methods. Our source code is available at: https://github.com/LIX-shape-analysis/GeomFmaps.

1. Introduction

Shape correspondence is a key problem in computer vision, computer graphics and related fields with a broad range of applications, including texture or deformation transfer and statistical shape analysis [7], among many others. While classical correspondence methods have been based on handcrafted features or deformation models [49], more recent approaches have focused on *learning* an optimal model from the data either in supervised [11, 53, 25, 16] or even unsupervised settings [19, 43, 17].

Despite significant progress in recent years, however, learning-based approaches for shape correspondence typically require large amounts of training data in order to learn a model that generalizes well to diverse shape classes [53, 16]. Several existing methods address this challenge by learning a derived representation, through a non-linear transformation of pre-computed feature descriptors [11, 25, 19, 43], rather than on the geometry of the shapes themselves. Unfortunately, as we demonstrate below, this

Figure 1. Given a pair of shapes, our approach builds consistent descriptors directly from the underlying point clouds (left), and automatically computes an accurate pointwise correspondence (right).

reliance on *a priori* hand-crafted descriptors makes the resulting learned models both less robust and less accurate leading to a significant drop in generalization power to new shape classes or instances.

In this work, we propose an approach that combines the power of learning directly from the 3D shapes with strong regularization based on a novel spectral correspondence extraction layer. Our method is inspired by recent learning techniques employing the functional map representation [25, 43]; however, we extend them to learn the features from 3D geometry rather than from some pre-computed de-

scriptors. Furthermore, we introduce a regularizer into the functional map computation layer that greatly improves the speed and robustness of training. Finally, we demonstrate how the spectral loss based on the functional map representation in the reduced basis significantly reduces overfitting, while still leading to accurate correspondences coupled with recent post-processing techniques. As a result, our overall pipeline is both more robust and has greater generalization power than existing methods, while still being able to learn from limited training data.

2. Related Work

Computing point-to-point maps between two 3D discrete surfaces is a very well-studied area of computer vision. Below, we review those closest to our method, or with the best known results to serve as baselines, and refer to recent surveys [49, 6, 44] for an in-depth discussion.

Our method is built upon the functional map representation, which was originally introduced in [32] as a tool for non-rigid shape matching, and then extended in follow-up works [33]. The key property of this representation is being able to express maps as small matrices, encoded in a reduced basis, which greatly simplifies the associated optimization problems.

The original work used only a basic set of constraints on functional maps, which have been extended significantly in, e.g., [24, 2, 20, 14, 9, 41, 31, 21, 40] among many other works. These approaches both extend the generality and improve the robustness of the functional map estimation pipeline, by using regularizers, robust penalties and powerful post-processing of the computed maps.

A key challenge in all of functional map estimation techniques, however, is the strong reliance on given input *descriptors* used for computing the maps. Several approaches have suggested to use robust norms [24, 23], improved pointwise map recovery [42, 15] or more principled regularizers [39] which can help alleviate noise in the input descriptors to a certain extent but do not resolve strong inconsistencies in challenging cases.

More recent techniques have advocated learning optimal descriptors for functional map estimation directly from the data [11, 25]. These methods compute a transformation of given input descriptors so that the estimated functional maps are close to ground truth maps given during training. This idea was very recently extended to the *unsupervised* setting [19, 43] where the supervised loss was replaced with structural penalties on the computed maps.

Despite significant progress, however, in all of these cases, the descriptors are optimized through a transformation of *hand-crafted input features*, such as SHOT [48], Heat [45] or Wave kernel signatures [5]. This has two severe consequences: first, any information not present in the input features will be absent from the optimized descriptors, and

second, such approaches generalize poorly across datasets as the input features can change significantly. This is particularly true of the commonly-used SHOT descriptors [48], which are sensitive to the triangle mesh structure and, as we show below, can vary drastically across different datasets.

A number of other techniques have also been proposed for shape correspondence learning without using the functional map representation. These include approaches that exploit novel convolutional layers on triangle meshes [27, 8, 30, 36] and more general methods that use learning from depth-maps [53] or in some feature space [46, 10] among many others. Remarkably, relatively few methods aim to learn directly from the raw 3D shape geometry for shape correspondence, with the notable exceptions of [16, 12]. In large part this is due to the complexity of the correspondence problem, where unlike, e.g., shape segmentation, the number of labels can be unbounded. As a result, existing techniques address this either by learning from precomputed features, or relying on template-based matching and large training sets [16, 12], that might even require manual curation. Although PointNet [37] and its variants [38, 4, 47] achieve impressive results from raw point clouds for classification tasks, they are not yet competitive for shape correspondence task.

Contribution

In this paper we show that feature learning for shape matching can be done directly from the raw 3D geometry even in the presence of relatively little training data, and without relying on a template or an *a priori* parametric (e.g., human body) model. Our main contribution is a end-to-end learnable pipeline that computes features from the 3D shapes and uses them for accurate dense point-to-point correspondence. We achieve this by introducing a novel map extraction layer using the functional map representation in a reduced basis, which provides a very strong regularization. Finally, we demonstrate that recent refinement techniques adapted to small functional maps [29], combined with our efficient learning pipeline jointly result in accurate dense maps at the fraction of the cost of existing methods.

3. Shape Matching and Functional Maps

One of the building blocks in our pipeline work is based on the functional map framework and representation. For completeness, we briefly review the basic notions for estimating functional maps, and refer the interested reader to a recent course [33] for a more in-depth discussion.

Basic Pipeline Given a pair of 3D shapes, $\mathcal{M}, \mathcal{N}$ represented in a discrete setting as triangle meshes, and containing respectively m and n vertices, this pipeline aims at computing a map between them.

It consists in four main steps. First, the first few eigen-

functions of the discrete Laplace-Beltrami operator are computed on each shape, namely $k_{\mathcal{M}}$ and $k_{\mathcal{N}}$ functions respectively.

Second, a set of descriptor *functions* on each shape that are expected to be approximately preserved by the unknown map. For instance, a descriptor function can correspond to a particular dimension of the Heat or Wave Kernel Signatures [45, 5] computed at each point. Their coefficients are stored in the respective basis as columns in matrices $\mathbf{A}, \mathbf{B}$.

Third, the optimal *functional map* $\mathbf{C}$ is then computed by solving the following optimization problem:

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

where the first term aims at preserving the descriptors: $E_{\text{desc}}(\mathbf{C}) = \left\| \mathbf{CA} - \mathbf{B} \right\|^2$, whereas the second term regularizes the map by promoting the correctness of its overall structural properties. It is common to use Fröbenius norm to compute the distance between these matrices. This Eq. (1) leads to a simple least-squares problem with $k_{\mathcal{M}} \times k_{\mathcal{N}}$ unknowns, independent of the number of points on the shapes.

As a last step, the estimated functional map $\mathbf{C}$, which maps across the spectral domains and converted to a point-to-point map. As a post processing step, called refinement, a number of advanced techniques are available [42, 15, 40, 29]. Most of them iteratively take the map from spectral to spatial domain, until it reaches a local optimum.

3.1. Deep Functional Maps

Despite its simplicity and efficiency, being a sequential framework, the functional map estimation pipeline described above is fundamentally error prone, due to the initial choice of descriptor functions. To alleviate this dependence, several approaches have been proposed to learn an optimal transformation of initial descriptors from data [11, 25, 43]. These works aim at transforming a given set of descriptors so that the optimal computed map satisfies some desired criteria during training. This transformation can be learned with a supervised loss, as in [11, 25], as well as with an unsupervised loss as in the more recent works of [19, 43].

More specifically, the FMNet approach proposed in [25] assumes to have as input, a set of shape pairs for which ground truth point-wise maps are known, and aims to solve the following problem:

$$\min_T \sum_{(S_1, S_2) \in \text{Train}} l_F(Soft(\mathbf{C}_{\text{opt}}), GT_{(S_1, S_2)}), \text{ where} \quad (2)$$

$$\mathbf{C}_{\text{opt}} = \arg\min_{\mathbf{C}} \| \mathbf{CA}_{T(D_1)} - \mathbf{A}_{T(D_2)} \|. \qquad (3)$$

Here, adopting the notation from [43] 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 $\mathbf{C}_{\text{opt}}$, after converting it to a soft map $Soft(\mathbf{C}_{\text{opt}})$ from the ground truth correspondence, and $\mathbf{A}_{T(D_1)}$ denotes the transformed descriptors D_1 written in the basis of shape 1. Thus, the FMNet framework [25] learns a transformation T of descriptors $T(D_1)$, $T(D_2)$ based on a supervised loss that minimizes the discrepancy between the resulting soft map and the known ground truth correspondence.

A related recent approach, SURFMNet [43] follows a similar strategy but replaces l_F with an *unsupervised loss* that enforces the desired structural properties on the resulting map, such as its bijectivity, orthonormality and commutativity with the Laplacian.

3D-CODED In contrast to the the methods described above that primarily operate in the *spectral* domain, there are also some approaches that never leave the spatial domain. With the recent works on point clouds neural networks, pioneered by PointNet [37], and significantly extended by [4, 47], to name a few, it is now possible to learn 3D features directly from point clouds. 3D-CODED [16, 12] is based on this approach, as it is a method built on a variational auto-encoder with a PointNet architecture for the encoder. Their method relies on a template that is supposed to be deformable in a non-rigid but isometric way to any of the shape of the datasets. It is a supervised method, and requires the knowledge of all ground-truth correspondences between any shape of the dataset and the deformable template. 3D-CODED is trained on 230K shapes, introduced in SURREAL [50], and generated with SMPL [26].

Motivation The two main classes of existing approaches have their associated benefits and drawbacks. On the one hand, spectral methods are able to use small matrices instead of all the points of the shape, and operate on *intrinsic* properties of the 3D surfaces, making them resilient to a change in pose, and allowing them to train on really small datasets. However, due to their use of input descriptors (typically SHOT [48]), they tend to overfit to the connectivity of the training set, which can lead to catastrophic results even in apparently simple cases. On the other hand, 3D-CODED shows extreme efficiency when trained on enough data, regardless of the connectivity, but with a small dataset, it is prone to overfitting and fails to generalize the training poses to predict the different poses of the test set.

Our method is a mix of the two approaches, and, as we show below, can obtain accurate results with little training data leading to state-of-the-art accuracy on a challenging recent benchmark of human shapes in different poses and with different connectivity [28].

4. Method

4.1. Overview

In this paper, we introduce a novel approach to learning descriptors on shapes in order to get correspondences through the functional map framework. Our method is composed of two main parts, labeled as *Feat* and *FMReg* in Figure 2. The first aims at optimizing point cloud convolutional filters [4, 47] to extract features from the raw geometry of the shapes. These filters are learned using a Siamese network on the source and a target shapes by using shared learnable parameters Θ, in a similar way as in [25]. However, unlike that approach and follow-up works [19, 43] we learn the features directly from the geometry of the shapes rather than computing a transformation of some pre-defined existing descriptors. These learned descriptors are projected in the spectral bases of the shapes and fed to the second block of the method, which uses them in a novel regularized functional map estimation layer. Finally, we use a spectral loss, based on the difference between the computed and the ground truth functional maps. This makes our approach very efficient as it operates purely in the spectral domain, avoiding expensive geodesic distance matrix computations as in [25, 19] and moreover allows us to handle functional or soft ground truth input maps without requiring the training shapes to have the same number of points or fixed mesh connectivity.

We stress again that the two components: learning features directly from the shapes and using the functional map representation both play a crucial role in our setup. The former allows us to learn robust and informative features independently from the mesh structure, while the latter allows us to strongly regularize correspondence learning, resulting in a method that generalizes even in the presence of a relatively small training set.

4.2. Architecture

The novelty of our architecture lies in its hybrid character. The first part, which we will refer to as the *feature extractor* in the following, aims at computing point-wise features on the input shapes. It corresponds to the *Feat* block in Figure 2, and takes as input only the point clouds making it robust towards changes in connectivity.

The purpose of the second part is to recover robust functional maps using these learned features. This block is built according to the pipeline of [32], first taking the features to the spectral domain over the two shapes (which corresponds to the dot products blocks after the *Feat* blocks in Figure 2), and then computing the map by minimizing an energy. However, since our method is based on a neural network, this operation should be differentiable with respect to the features over the shapes for the back-propagation algorithm to work. We extend the previously proposed functional map
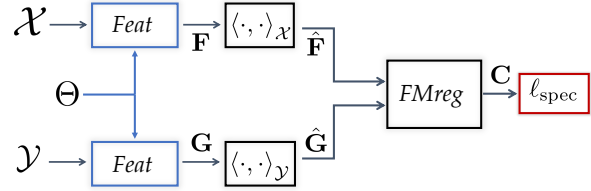

Figure 2. Overview of our approach: given a pair of shapes, we optimize for a point cloud convolutional model to get point-wise features for each shape, that we convert to a functional map using our FMReg block. The loss that we put forward penalizes maps according to their distance to the ground-truth map between the two shapes.

layers [25] to also incorporate a differentiable regularizer, which results in the very robust map extraction, represented as *FMReg* in Figure 2.

4.3. The feature extractor

The goal of this block is to learn functional characterizations of point clouds that will later be used to compute spectral descriptors and then functional maps. To this end, this network must be applied with the same weights to the source and target shapes, as represented in Figure 2, and must result in informative descriptors, extracted from the point clouds of the two shapes.

For this part, we chose the state of the art point cloud learning method KPConv [47], by extending the segmentation network proposed in that work. Our feature extractor is thus a Siamese version of the segmentation network described in KPConv, which we review for completeness in the supplementary materials.

4.4. The regularized FMap layer

This block provides a novel fully differentiable way to compute a robust functional map from potentially low dimensional spectral descriptors.

The main goal is, as in Section 3, to recover the ground-truth bijection between $\mathcal{M}$ and $\mathcal{N}$, on which we have the computed raw-data features $\mathbf{F}$ and $\mathbf{G}$.

For this we first express the computed feature functions in the respective spectral basis, which we denote by $\Phi^{\mathcal{M}}$ and $\Phi^{\mathcal{N}}$. This leads to the spectral descriptors $\mathbf{A} = (\Phi^{\mathcal{M}})^{\dagger}\mathbf{F}$ and $\mathbf{B} = (\Phi^{\mathcal{M}})^{\dagger}\mathbf{G}$, with $\Phi^{\dagger}$ the Moore pseudo-inverse of Φ. We stress again that this step is where we shift focus from the spatial to the spectral domain, and corresponds to the dot product blocks in Figure 2.

In the pipeline first introduced in [32] and then widely used in the follow-up works [34], the standard strategy is to compute the functional map $\mathbf{C}$ that optimizes the following energy:

$$\min_{\mathbf{C}} \left\| \mathbf{CA} - \mathbf{B} \right\|^2 + \lambda \left\| \mathbf{C}\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}\mathbf{C} \right\|^2, \qquad (4)$$

where λ is a scalar regularization parameter.

Remark that the optimization problem in Eq. (4.4) is quadratic in terms of $\mathbf{C}$ and can be solved e.g. via standard convex optimization techniques. However, in the learning context, we need to differentiate the solution with respect to the spectral features $\mathbf{A}, \mathbf{B}$, which is challenging when $\mathbf{C}$ is computed via an iterative solver. Alternatively, the problem in Eq. can be written directly in terms of a large least squares system, by vectorizing the matrix $\mathbf{C}$ as was suggested in [32]. However, for a $k \times k$ functional map, this leads to a system of size $k^2 \times k^2$ which becomes prohibitive even for moderate values of k. To avoid these issues, previous learning-based approaches based on functional maps [25, 19, 43] have only optimized for $\mathbf{C}$ using the first part of the energy in Eq. (4.4): $\left\| \mathbf{CA} - \mathbf{B} \right\|^2$. This results in a simple linear system for which the derivatives can be computed in closed form. This has two major limitations, however: first the linear system is only invertible if there are at least k linearly independent feature functions. This condition can easily be violated in practice, especially in the early stages of learning, potentially resulting in a fatal error. Furthermore, the lack of regularization makes the solved-for functional map very sensitive to inconsistencies in the computed descriptors, which leads to an overall loss of robustness.

In our work we address this problem by using the full energy in Eq. (4.4) in a fully differentiable way. In particular, we use the fact that the operators $\Delta_{\mathcal{M}}$ and $\Delta_{\mathcal{N}}$ are diagonal when expressed in their own eigen-basis.

Indeed we remark that the gradient of the energy in Eq. (4.4) vanishes whenever $\mathbf{CAA}^T + \lambda\boldsymbol{\Delta} \cdot \mathbf{C} = \mathbf{BA}^T$, where the operation $\cdot$ represents the element-wise multiplication, and $\boldsymbol{\Delta}_{ij} = (\mu_j^{\mathcal{M}} - \mu_i^{\mathcal{N}})^2$, where $\mu_l^{\mathcal{M}}$ and $\mu_l^{\mathcal{N}}$ respectively correspond to the eigenvalues of $\Delta_{\mathcal{M}}$ and $\Delta_{\mathcal{N}}$. It is then easy to see that this amounts to a separate linear system *for every row c_i of $\mathbf{C}$* :

$$(\mathbf{AA}^T + \lambda\mathrm{diag}((\mu_j^1 - \mu_i^2)^2))c_i = \mathbf{A}b_i \qquad (5)$$

where b_i stands for i^{th} row of B.

In total, if k is the number of eigenvectors used for representing the functional map, this operation amounts to inverting k different $k \times k$ matrices. Since inverting a linear system is a differentiable operation, which is already implemented e.g. in TensorFlow, this allows us to estimate the functional map in a robust way, while preserving differentiability.

4.5. The supervised spectral loss

Our method also uses a loss with respect to the ground truth functional map in the spectral domain. This is similar to the energy used in [11], but is different from the loss of the original FMNet work [25], which converted a functional map to a soft correspondence matrix and imposed a loss with respect to the ground truth point-wise map, relying on expensive geodesic distance matrix computation.

Specifically, calling $\mathbf{C}$ the functional map obtained by the FMap block, and $\mathbf{C}^{gt}$ the ground truth spectral map, our loss is defined as:

$$l_{\mathrm{spec}} = \left\| \mathbf{C} - \mathbf{C}^{gt} \right\|^2$$

As mentioned above, we use a Fröbenius norm to compute the distance between matrices.

It is important to note that whenever a pointwise ground truth map is given it is trivial to convert it to the functional map representation. Conversely, the ground truth spectral map is more general than the point-wise ground truth correspondence. Indeed, with just a few precise landmarks one can recover a functional map accurate enough to make this loss efficient, for instance through the original pipeline of [32], but also with more recent follow-up works, such as [40] or [29], which we will further describe as baselines to our method in Section 5.

This is useful, e.g., in the case of re-meshed datasets. Indeed, complete ground truth correspondences between two shapes of these datasets are not fully known. One can only have access to the (often partial and not bijective) ground truth pointwise map from a template mesh $\mathcal{T}$ to each re-meshed shape $\mathcal{S}_i$. Ecah such map can be converted to a functional map $\mathbf{C}_i$ and a very good approximation of the spectral ground truth $C_{i \to j}^{gt}$ between $\mathcal{S}_i$ and $\mathcal{S}_j$ can be expressed as $\mathbf{C}_j^{\dagger}\mathbf{C}_i$.

4.6. Postprocessing

Once our model is trained, we can then test it on a pair of shapes and get a functional map between these shapes. This map can either directly be converted to a point to point map, or refined further. We use a very recent and efficient refining algorithm, called ZoomOut [29] based on navigating between spatial and spectral domains while progressively inceasing the number of spectral basis functions. This efficient postprocessing technique allows us to get state-of-the-art results, as described in Section 5.

4.7. Implementation

We implemented our method in TensorFlow [1] by adapting the open-source implementation of SURFMNet [43] and KPConv [47].

Our feature extraction network is based on a residual convolutional architecture of [47], consisting of 4 convolutional blocks with leaky linear units, with successive poolings and dimension augmentation from 128 to 2048, followed by a 4 up-sampling blocks with shortcuts from corresponding pooling layers, and dimension reduction from

2048 back to 128. Please see the Supplementary materials, part A, in [47] for more details. Following the pipeline of KPConv, we start with a sub-sampled version of our point clouds with a grid subsampling of step 0.03. The pooling layers are therefore obtained with grid samplings of parameters 0.06, 0.12, 0.24 and 0.48.

Similarly to FMNet [25] and SURFMNet [43], our network is applied in a Siamese way on the two shapes, using the same learned weights for the feature extractor.

In the case of fully automatic spectral methods such as BCICP [40] and ZoomOut [29], or the deep learning based FMNet [25, 19] (supervised or unsupervised) and SURFM-net [43], all results are invariant by any rigid transformation of the input shapes. However, in the case of methods using the 3D coordinates of the points to generate information about the input shape, this does not remain true. Consequently, both 3D-CODED [16] and our method avoid this dependency through data augmentation to be as close as possible to the generality of fully spectral methods. To that end, assuming the shapes are all aligned on one axis (e.g. on human the natural up axis), both 3D-CODED and our method perform data augmentation by randomly rotating the input shapes along that axis. Our implementation is publicly available at https://github.com/LIX-shape-analysis/GeomFmaps.

4.8. Parameters

In addition to the architecture above, our method has two key hyper-parameters: the size of the functional basis and the regularizer λ in Equation 5. For the size of the basis, we discovered if this number is too high, for instance, with 120 eigenvectors as in FMNet and SURFMNet, it can easily lead to overfitting. However, by reducing this number to 30, the results of SURFMNet on FAUST re-meshed (here reported in Table 1) go from 0.15 to 4.5. As a consequence, we choose the number of eigenvectors to be 30 in all of our experiments on our method. Regarding the weight λ in Equation (5), we observed that setting it to $\lambda = 10^{-3}$ helps getting good results while drastically reducing the number of training steps, as pointed out in the ablation study. We use this value throughout all experiments.

We train our network with a batch size of 4 shape pairs for a number of epochs depending on the number of shapes in the dataset. We use a learning rate of .001 and gradually decreasing it to 0.0001 with ADAM optimizer [13].

5. Results

Datasets

We test our method on a wide spectrum of human datasets: first, the re-meshed versions of FAUST dataset [7] containing 100 human shapes in 1-1 correspondence, and of SCAPE [3], made publicly available by Ren et al. [40].

Method \ Dataset	F	S	F on S	S on F
BCICP	15.	16.	*	*
ZoomOut	6.1	7.5	*	*
SURFMNet	15.	12.	32.	32.
SURFMNet+icp	7.4	6.1	19.	23.
Unsup FMNet	10.	16.	29.	22.
Unsup FMNet+pmf	5.7	10.	12.	9.3
FMNet	11.	17.	30.	33.
FMNet+pmf	5.9	6.3	11.	14.
3D-CODED	2.5	31.	31.	33.
Ours	3.1	4.4	11.	6.0
Ours+zo	**1.9**	**3.0**	**9.2**	**4.3**

Table 1. Comparative results ($\times 100$) of the different methods on Experiment 1.

These re-meshed datasets offer significantly more variability in terms of shape structures and connectivity, including for instance point sampling density, making them harder to match for existing algorithms. We also highlight that the SCAPE dataset is slightly more challenging since the shapes are less regular, and two shapes never share the same pose. This is not true for FAUST, wherein all the poses present in the test set also exist in the training set, with the variation coming from body type only, making the pose recovery easier at test time.

We also use the re-meshed version of the more recent SHREC'19 dataset [28], which, in theory, is the most challenging of the test sets, because of stronger distortions in the poses, the presence of an incomplete shape, and the number of test pairs (430 in total, so two times the number of test pairs of FAUST or SCAPE). At last, we also use the generic training dataset of 3D-CODED [16], originally consisting in 230K synthetic shapes generated using Surreal [50], with the parametric model SMPL introduced in [26]. We use it only for training purposes in our second experiment, to show that our method can generalize well to changes in connectivity, being able to train on a synthetic, very smooth, identical triangulation for the whole training set, and still produce results of excellent quality on re-meshed datasets.

Ablation study

Our method is built with a number of building blocks, all of which we consider essential to achieve optimal performance. To illustrate this, in the supplementary materials we provide an extensive ablation study of all the key components of our algorithm.

Baselines

We compare our method to several state of the art methods: the first category includes fully automatic methods without any learning component [40, 29]. These methods are simply evaluated on the test sets without any training. The second category includes FMNet [25] and its unsu-
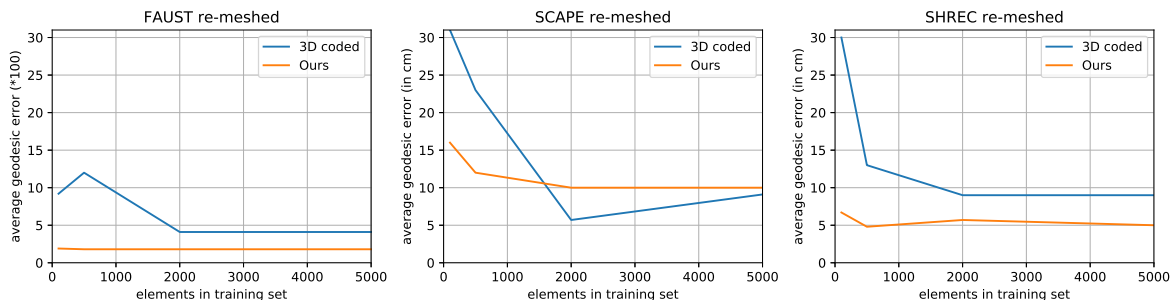
Figure 3. Comparison with 3D-CODED while varying training size of SURREAL dataset and simultaneously testing on other datasets.

pervised versions, referred to as Unsup FMNet [19] and SURFMNet [43], with and without post-processing (PMF [52] for FMNet, and standard functional map refinement [32], referred to as ICP, for SURFMNet). All these variants of FMNet give similar results, but SURFMNet is the only one to train within a few hours, without requiring too much space. This is due to the fact SURFMNet only operates in the spectral domain, in contrast to other methods. Lastly, we compare to the supervised 3D-CODED [16], described earlier in more details in Section 3. For conciseness, we refer to our method as Ours in the following text. We show our results with and without ZoomOut [29] refinement, referred to as ZO, in order to prove that our method stands out even without post processing. We compare these different methods in two main settings named Experiment 1 and Experiment 2 below.

Experiment 1 consists of evaluating the different methods in the following setting: we split FAUST re-meshed and SCAPE re-meshed into training and test sets containing 80 and 20 shapes for FAUST, and 51 and 20 shapes for SCAPE. We obtain results for training and testing on the same dataset, but also by testing on a different dataset. For instance, by training on SCAPE re-meshed train set and testing on FAUST re-meshed test set. This experiment aims at testing the generalization power of all methods to small re-meshed datasets, as well as its ability to adapt to a different dataset at test time.

Experiment 2 consists of sampling 100, 500, 2000, and 5000 shapes from the SURREAL dataset to be used for training. We then test the trained models on the test sets of FAUST re-meshed, SCAPE re-meshed, and SHREC19 re-meshed. This experiment aims at testing the robustness and generalization power of the different methods in the presence of varying amounts of training data, as well as adaptability to train on a perfect synthetic triangulations and still get results on challenging re-meshed shapes.

Quantitative results

To evaluate the results, we use the protocol introduced in [22], where the per-point-average geodesic distance between the ground truth map and the computed map is reported. All results are multiplied by 100 for the sake of readability.

As we can see in Table 1, our method performs the best overall on Experiment 1. Fully automatic methods do not provide competitive results compared to the learning methods (except on crossed settings because they did not train on anything and are thus not influenced by the training shapes). As reported in the Section 3.1, this highlights that hand-crafted features can easily fail. It is noticeable that spectral methods (FMNet variations, and Ours as a hybrid method) get reasonable, or even good results in our case, with these small datasets. In comparison, 3D-CODED seems to fail in almost all cases. It is remarkable that it can learn on such a small dataset as the training set of FAUST re-meshed. One explanation for that is that FAUST contains the same set of poses in the test set as in the train set.

Contrary to other baselines, our method gives good results on all settings, *even without refinement*, showing good resilience to a really low number of shapes, even with re-meshed geometry. We would like to stress that no other method is able to achieve such a generalization with this low number of shapes.

For a fair comparison with 3D-CODED, we complete our study with a second experiment, in which the training set is now made of the same shapes 3D-CODED uses for training in their paper, namely SURREAL dataset. The aim of this experiment is to further showcase the generalization power of our method when compared to 3D-CODED. First, by training on a very smooth synthetic dataset, on which previous fully spectral methods tend to easily overfit due to the obvious mismatch in triangulation in training and test set. Our second goal is to observe the dependence of different methods on size of the training set.

We report the results (multiplied by 100) of 3D-CODED and Our method in Figure 3, as they are the only two competitive algorithms in Experiment 2. These results once
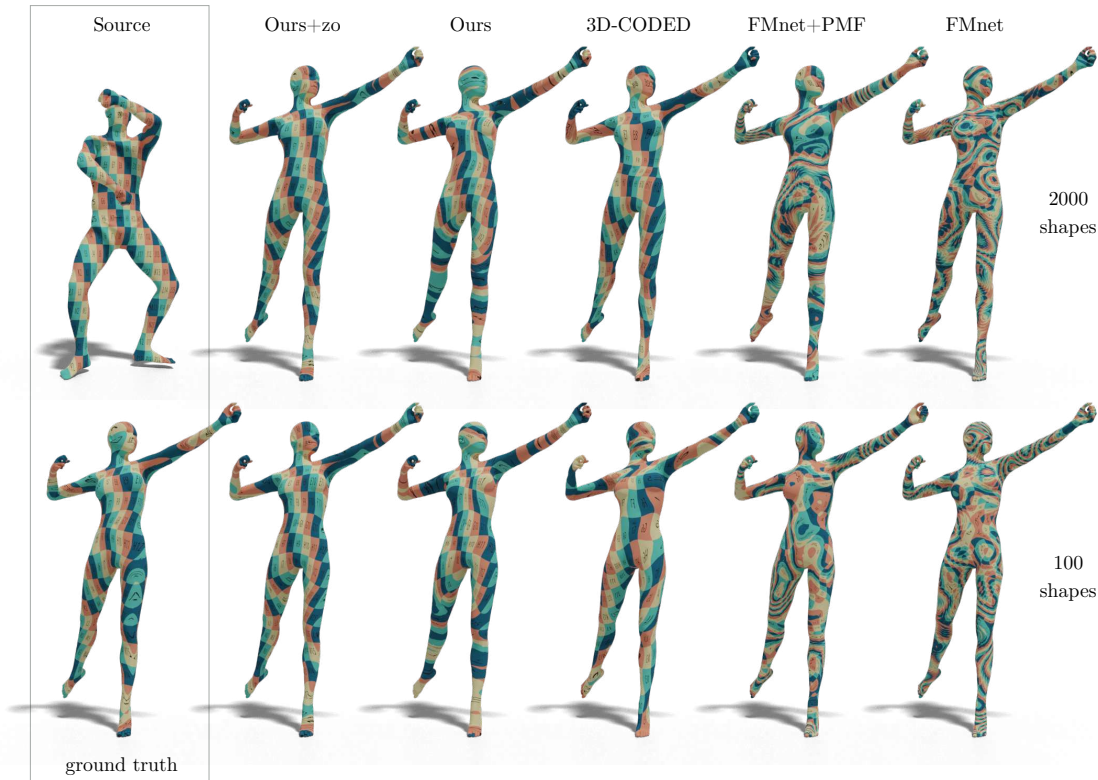
Figure 4. Qualitative results obtained with texture transfers for the different methods on Experiment 2, training on two different numbers of shapes in the SURREAL dataset, and testing on SHREC re-meshed shapes.

again demonstrate that our method can achieve impressive results even with a low number of training shapes. On SHREC re-meshed, we achieve state of the art results with an average error of **0.048** with only 500 training shapes. We provide additional quantitative comparisons in the supplementary materials.

It can be observed in Figure 3 that our results are consistent and unaltered even with the drop in number of training shapes. 3D-CODED, on the other hand, always suffers from a reduced training set.

Qualitative results

In Figure 4 we show the results of our method (with and without ZoomOut refinement [29]), 3D-CODED [16], FM-Net [25] (with and without PMF refinement [51]), trained on respectively 2000 and 100 shapes, as presented in Experiment 2, via texture transfer.

With 2000 training shapes, both our method and 3D-CODED lead to good or even excellent texture transfers, while fully spectral methods fail due to the change of connectivity from training to test set. However, with only 100 training shapes, 3D-CODED fails to get a good reconstruction in many cases, leading to bad texture transfer as in Figure 4. This highlights the fact that our method performs better than any other existing method when only a few training

shapes are provided.

6. Conclusion, Limitations & Future Work

We presented a method for improving the robustness and reducing overfitting in learning shape correspondences. Key to our approach is a hybrid network structure, made of a raw-data feature extractor that learns descriptors on a pair of shapes, and a novel robust functional map layer. Our network can thus operate in both the spectral and the spatial domain, thus taking advantages of both representations.

Our approach has several limitations: first, as a supervised method it requires at least partial correspondences (as discussed in Section 4.5) between the training shapes. Also, it requires data augmentation to be able to predict non-aligned shapes, which can be costly and unstable.

In the future, we plan to work towards an unsupervised spectral loss, similar in spirit to SURFMNet [43], while avoiding the symmetry ambiguity problem. We also plan to try other, invariant feature extractors such as [18], or [35] to avoid data augmentation.

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] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics*, 37, 03 2018. 2, 3, 4

[5] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis. In *Proc. ICCV Workshops*, pages 1626–1633. IEEE, 2011. 2, 3

[6] 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

[7] 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

[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] Mingjia Chen, Changbo Wang, and Hong Qin. Jointly learning shape descriptors and their correspondence via deep triplet cnns. *Computer Aided Geometric Design*, 62:192–205, 2018. 2

[11] Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. Supervised descriptor learning for non-rigid shape matching. In *Proc. ECCV Workshops (NORDIA)*, 2014. 1, 2, 3, 5

[12] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *arXiv preprint arXiv:1908.04725*, 2019. 2, 3

[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

[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] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 1, 2, 3, 6, 7, 8

[17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. In *Computer Graphics Forum*, volume 38, pages 123–133. Wiley Online Library, 2019. 1

[18] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. PCPNet: Learning local shape properties from raw point clouds. *Computer Graphics Forum*, 37(2):75–85, 2018. 8

[19] Oshri Halimi, Or Litany, Emanuele Rodol'a, Alex Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7

[20] 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

[21] 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

[22] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. 7

[23] 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

[24] 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

[25] 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. 1, 2, 3, 4, 5, 6, 8

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, Oct. 2015. 3, 6

[27] 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. 2

[28] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Matching Humans with Different Connectivity. In Silvia Biasotti, Guillaume Lavoué, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 3, 6

[29] Simone Melzi, Jing Ren, Emanuele Rodola, Maks Ovsjanikov, and Peter Wonka. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 2019. 2, 3, 5, 6, 7, 8

[30] 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. 2

[31] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017. 2

[32] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 2, 4, 5, 7

[33] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, SIGGRAPH '17, pages 5:1–5:62, 2017. 2

[34] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, pages 5:1–5:62, 2017. 4

[35] Adrien Poulenard, Marie-Julie Rakotosaona, Yann Ponty, and Maks Ovsjanikov. Effective rotation-invariant point cnn with spherical harmonics kernels. *arXiv preprint arXiv:1906.11555*, 2019. 8

[36] Adrien Poulenard, 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

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 652–660, 2017. 2, 3

[38] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5105–5114, USA, 2017. Curran Associates Inc. 2

[39] Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. Structured regularization of functional map computations. In *Computer Graphics Forum*, volume 38, pages 39–53. Wiley Online Library, 2019. 2

[40] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. 2, 3, 5, 6

[41] 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

[42] Emanuele Rodolà, M Moeller, and Daniel Cremers. Pointwise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)*, 2015. 2, 3

[43] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1627, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[44] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pages 1–17, 2019. 2

[45] 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. 2, 3

[46] Zhiyu Sun, Yusen He, Andrey Gritsenko, Amaury Lendasse, and Stephen Baek. Deep spectral descriptors: Learning the point-wise correspondence metric via siamese deep neural networks. *arXiv preprint arXiv:1710.06368*, 2017. 2

[47] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2, 3, 4, 5, 6

[48] 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. 2, 3

[49] 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. 1, 2

[50] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 3, 6

[51] Matthias Vestner, Zorah Lähner, 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. 8

[52] 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. 7

[53] 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