Joint Symmetry Detection and Shape Matching for Non-Rigid Point Cloud

Abhishek Sharma, Maks Ovsjanikov

Joint Symmetry Detection and Shape Matching for Non-Rigid Point Cloud

Abhishek Sharma
LIX, Ecole Polytechnique
IPParis, France
kein.iitian@gmail.com

Maks Ovsjanikov
LIX, Ecole Polytechnique
IPParis, France
maks@lix.polytechnique.fr

Abstract

Despite the success of deep functional maps in non-rigid 3D shape matching, there exists no learning framework that models both self-symmetry and shape matching simultaneously. This is despite the fact that errors due to symmetry mismatch are a major challenge in non-rigid shape matching. In this paper, we propose a novel framework that simultaneously learns both self symmetry as well as a pairwise map between a pair of shapes. Our key idea is to couple a self symmetry map and a pairwise map through a regularization term that provides a joint constraint on both of them, thereby, leading to more accurate maps. We validate our method on several benchmarks where it outperforms many competitive baselines on both tasks.

1. Introduction

Shape correspondence is a fundamental problem in computer vision, computer graphics and related fields [44], since it facilitates many applications such as texture or deformation transfer and statistical shape analysis [1] to name a few. Although shape correspondence has been studied from many viewpoints, we focus here on a functional map-based approaches [28] as this framework is quite general, scalable and thus, has been extended to various other applications such as pose estimation [26], matrix completion [42] and graph matching [46].

While recent learning based deep functional map approaches have made impressive gains in non rigid isometric full shape matching [13, 20, 37, 41], symmetry detection [25, 34] has received little attention in the learning paradigm. This is despite the fact that the two problems are inherently linked and symmetry disambiguation remains a challenge in shape matching pipeline. While there have been some attempts to learn shape matching in a noisy setup [20, 21], we are not aware of any learning setup that investigates the effect of noise for symmetry detection.

In this work, we study the use of a canonical embedding based framework that is a promising direction towards obtaining a unified shape matching framework for both partial as well as full shape matching. Instead of using predefined basis functions in the functional map framework, we learn a canonical embedding that simultaneously learns to model self-symmetry as well as a pairwise map. Learning canonical embedding for non-rigid shape matching is still a relatively open problem in terms of exploring what prior assumptions should be made on such embedding. Marin et al. [21] make the first attempt by assuming such embeddings to be linearly invariant between a pair of shapes. However, as we show later, learning such an embedding without exploiting natural priors on 3D shapes, such as their symmetry structure, leads to overfitting as no regularization or constraint is enforced on the linear transformation between a pair of shape embeddings [21].

In this paper, we advocate an orthogonal approach where we rely on canonical embedding for shape matching but simultaneously impose prior structure on a self-symmetry map to induce information transfer between the two spaces. This is advantageous for two reasons: first, it significantly simplifies the embedding learning pipeline and makes it learnable end-to-end. This is because the linearly invariant assumption can be made on a self-symmetry map and not on the pairwise map between shapes as done in [21], thereby, reducing the two stage sequential optimization scheme of [21] to single shot. Secondly, modelling a self symmetry map enables us to explicitly enforce the pointwise map between two shapes to take into account the intrinsic self-symmetry during training.

Our second contribution is a novel commutative regularization that couples the self-symmetry map with a pairwise map and thus, enables knowledge transfer between the two maps during training. This significantly improves generalization and robustness to sampling resolution as well as the size of embedding. Our method obtains competitive results on multiple shape matching benchmarks such as FAUST remesh and partial SHREC'16 when compared to recent learning-based methods while being very robust to noisy set up. We also evaluate our method on symmetry detection on various benchmarks where it shows resilience

to noise where other methods based on the Laplacian Beltrami operator fail.

To summarize, our contributions are as follows:

- We propose to learn shape matching in the canonical space with appropriate regularization that consistently outperforms the linearly invariant embedding approach.
- To the best of our knowledge, we propose a first method that simultaneously learns symmetry detection and shape matching for non-rigid point clouds.
- We propose a novel regularization that constrains the symmetry map and pairwise map which is of independent interest for future work in this direction.

2. Related Work

Functional Maps Computing point-to-point maps between two 3D discrete surfaces is a very well-studied problem. We refer to a recent survey [38] for an in-depth discussion. Our method is closely related to the functional map pipeline, introduced in [28] and then significantly extended in follow-up works (see, e.g., [29]). The key idea of this framework is to encode correspondences as small matrices, by using a reduced functional basis, thus greatly simplifying many resulting optimization problems. The functional map pipeline has been further improved in accuracy, efficiency and robustness by many recent works including [3, 11, 14, 17, 27, 35, 36]. There also exist other works [2, 24, 47] that treat shape correspondence as a dense labeling problem but they typically require a lot of data as the label space is very large.

Learning from raw 3D shape Although early approaches in functional maps literature used hand-crafted features [29], more recent methods directly aim to *learn* either the optimal transformations of hand crafted descriptors [20, 37] or even features directly from 3D geometry itself [6, 41]. Initial efforts in this direction used classical optimisation techniques [4]. In contrast, Deep Functional Maps [19] proposed a deep learning architecture called FMNet to optimize a non-linear transformation of SHOT descriptors [45], that was further extended to unsupervised setting [8, 13, 37]. To alleviate the sensitivity to SHOT descriptor, recent works including [6, 12, 41] learn shape matching directly from the *raw 3D data* without relying on pre-defined descriptors, thus leading to improvements in both robustness and accuracy. However, all these works are aimed at *full* (complete) shape correspondence and do not handle partial shape matching effectively.

Learning Basis from Data Most of the functional map frameworks can not handle partiality in data as they rely on Laplacian eigenfunctions that are shown to be unstable under partial data. [20, 36, 48] deal with partiality but they are based on hand-crafted features and require an expensive optimization scheme and are instance specific. While [41] proposes to learn a suitable alignment of pre-computed Laplacian Eigen basis functions, the approach still relies on the Laplacian basis and can therefore be unstable. [21] proposed a two stage architecture to learn a linear transformation invariant shape embedding to bypass the difficulties associated with LBO. However, as we demonstrate later in experiments, the two stage architecture is suboptimal due to the lack of adequate regularization.

Self Supervised Learning Self supervised learning has been exploited for learning representations and embedding in various domains where a proxy task is used to learn the representation. e.g. [40] uses an autoencoder to complete the partial shapes and uses the resulting representation of shape completion for the shape classification task. [10] learns to predict image rotations and uses the resulting representation for image classification. Our formulation is in the same spirit as we learn to inject the symmetry information in a 3D shape and use the resulting representation for 3D shape matching. However, we choose symmetry learning as a proxy task for embedding learning for a principled reason as described in detail in the methodology section.

Symmetry for Non Rigid Shape Matching Matching shapes with intrinsic symmetries involves dealing with symmetric ambiguity problem which has been very well studied and explored in axiomatic methods [18, 23, 25, 30, 33, 34]. More recently, [9, 43] proposes an end to end method to learn extrinsic 3D symmetries from a RGB-D image. However, none of the existing learning based non-rigid shape matching method models or learn symmetry explicitly as a regularizer for shape matching.

Joint Learning of similar tasks Computer vision literature is full of problems that are inherently linked [7, 16, 39] and thus, should be learned simultaneously. In 3D shape analysis, Neuromorph [7] simultaneously learns shape correspondence and interpolation. Our work also falls in a similar direction as we aim to learn shape matching and symmetry detection simultaneously. Our work is most similar in spirit to [39] that couples image segmentation and detection via linear constraints and thus, induces information transfer/sharing between the segmentation map and detection map via these constraints. In our formulation, we enable this information transfer during training via a commutative loss that couples the self-symmetry and pairwise map.

Rest of the paper is structured as follows: In the next section, we briefly cover the necessary background on the functional map. Afterwards, we propose our method to learn canonical embedding for joint shape matching and symmetry detection and introduce our novel regularization term that constrains self-symmetry and pairwise map. Lastly, we validate our framework on three benchmark datasets by comparing it to various state-of-the-art methods and providing ablation studies.

3. Background

Before describing our method, we provide a brief overview of the basic pipeline to compute a functional map [28].

Functional Map Computation The typical functional map pipeline [28] assumes that we are given a source and a target shape, $\mathbf{X}, \mathbf{Y}$, containing, respectively, n_x and n_y vertices, a small set of k basis functions, e.g. of the respective Laplace-Beltrami operators (LBO). We are also given a set of descriptors on each shape, to be preserved by the unknown map, whose coefficients in the basis functions are stored as columns of matrices $\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}}$. The optimal *functional map* $\mathbf{C}_{\text{opt}}$ is 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 $E_{\text{desc}}(\mathbf{C}) = \left\|\mathbf{C}\Phi_{\mathbf{X}} - \Phi_{\mathbf{Y}}\right\|^2$ aims at the descriptor preservation whereas the second term acts as a regularizer on the map by enforcing its overall structural properties, such as bijectivity of the map. The optimization problem in equation 1 can be solved with any convex solver. Once the optimal functional map $\mathbf{C}_{\text{opt}}$ is computed, one can use nearest neighbor search in the spectral embedding to convert it to a point to point correspondence.

Note that when the basis functions are neural network-based, instead of optimizing over $\mathbf{C}$, we are optimizing the functional in equation 1 over $\mathbf{C}$, $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$. In this case, joint optimization over $\mathbf{C}$, $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ is challenging as $\mathbf{C}$ is computed via an iterative solver itself.

4. Joint Shape Matching and Symmetry Detection

In the previous section, we outlined a basic mechanism to compute a functional map given a set of basis functions. Due to the instability of Laplace-Beltrami operator, LBO, on partial 3D shapes [15] and noise [21], our main goal is to avoid using its eigenfunctions and instead aim to *learn* an embedding that can replace the spectral embedding given by the LBO. This section details how to learn such an embedding whilst working in the symmetric space.

Input Shape Representation In contrast to several recent works [13, 41] that assume to be given a mesh representation of 3D shapes in terms of LBO operator, we do not impose any such constraint and directly work the with point cloud representation without LBO. Our work is most closely related to a recent work [21] that proposes to replace the Laplace-Beltrami basis by learning embeddings that are related by a linear transformation across pairs of shapes. Intuitively, this formulation aims to embed a shape from the 3D space, in which complex non-rigid deformations could occur, to another higher-dimensional space, in which transformations across shapes are linear. However, using a supervised loss to learn this embedding without enforcing any structural properties on the underlying linear transform provides little guarantee that the learned transform will generalize from the train to test setting.

Notations We denote a map between a pair of shapes $\mathbf{X}$ and $\mathbf{Y}$ by $T_{\mathbf{XY}} : \mathbf{X} \rightarrow \mathbf{Y}$ such that $T_{\mathbf{XY}}(x_i) = y_j$, $\forall i \in \{1, \ldots, n_{\mathbf{X}}\}$ and some $j \in \{1, \ldots, n_{\mathbf{Y}}\}$. This map can be represented by a matrix $\Pi_{\mathbf{XY}} \in \mathbb{R}^{n_{\mathbf{X}} \times n_{\mathbf{Y}}}$ such that $\Pi_{\mathbf{XY}}(i, j) = 1$ if $T_{\mathbf{XY}}(x_i) = y_j$ and 0 otherwise. We use the same notation T for self symmetry map $\mathbf{T}_{\mathbf{XX_f}}$ as well. We use $P_{\mathbf{X}}$ to denote the 3D coordinates of $\mathbf{X}$.

Our network takes a shape $\mathbf{X}$ as input. We then perform a reflection (flip) of each shape along one axis resulting in a shape denoted as $\mathbf{X_f}$. The original and flipped shapes are then forwarded to a Siamese architecture, based on a PointNet [32] feature extractor, that embeds these two shapes into some fixed k dimensional space. The intuition behind this operation is to help the network learn representation that can disambiguate left from right in shape matching.

Let $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{X_f}}$ denote the matrices, whose rows can be interpreted as embeddings of the points of $\mathbf{X}$ and $\mathbf{X_f}$. In the functional map framework, there exists a functional map $C_{\mathbf{XX_f}}$ that aligns the corresponding embeddings. Given a self symmetry ground truth pointwise map $\mathbf{T}_{\mathbf{XX_f}}$, we can estimate $C_{\mathbf{XX_f}}$ by solving the following optimization problem:

$$C_{\mathbf{XX_f}} = \arg\min_{C} \|\Phi_{\mathbf{X}} C^T - \mathbf{T}_{\mathbf{XX_f}} \Phi_{\mathbf{X_f}}\|_2 \qquad (2)$$

The optimal symmetry map $C_{\mathbf{XX_f}}$ is given by: $C_{\mathbf{XX_f}} = (\Phi_{\mathbf{X}}^{+} \mathbf{T}_{\mathbf{XX_f}} \Phi_{\mathbf{X_f}})^T$, that is differentiable using the closed-form expression of derivatives of matrix inverses, as also mentioned in Section 3. Similarly, we can compute $C_{\mathbf{YY_f}}$ for shape $\mathbf{Y}$. Note that one can also estimate a functional map, e.g. $C_{\mathbf{XX_f}}$, even without using a point to point map, as shown in Roufosse et al. [37] if there are sufficient structural constraints to enforce on $C_{\mathbf{XX_f}}$.

4.1. Loss functions

Given a set of pairs of shapes $\mathbf{X}, \mathbf{Y}$ for which ground truth correspondences $\mathbf{T}^{gt}_{\mathbf{XY}}$ are known, our network com-

putes an embedding $\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}}$ for each shape as well as a self symmetry functional map $C_{\mathbf{X}\mathbf{X}_\mathbf{f}}$ and $C_{\mathbf{Y}\mathbf{Y}_\mathbf{f}}$ respectively as described above. We then optimize the sum of three loss functions, one each defined for linearly invariant self symmetry embedding, nearest neighbour based loss for pairwise (shape pair) embedding and a commutativity loss for explicitly enforcing the coupling between a self symmetry map and a pointwise map during training.

Cosine Similarity Our loss functions are based on a soft-correspondence matrix, also used in [19] and [21]. To define it for self symmetry map, we *transform* each shape embedding $\widehat{\Phi}_{\mathbf{X}} = \Phi_{\mathbf{X}} C^T_{\mathbf{X}\mathbf{X}_\mathbf{f}}$ by applying the optimal symmetry map. We then compare the rows of $\widehat{\Phi}_{\mathbf{X}}$ to those of $\Phi_{\mathbf{X}_\mathbf{f}}$ to obtain the *soft* correspondence matrix $S_{\mathbf{X}\mathbf{X}_\mathbf{f}}$ that approximates the self-symmetry map in a differentiable way as follows:

$$(S_{\mathbf{X}\mathbf{X}_\mathbf{f}})_{ij} = \frac{e^{\widehat{\Phi}^i_{\mathbf{X}}{}^T \Phi^j_{\mathbf{X}_\mathbf{f}}/\tau}}{\sum_j e^{\widehat{\Phi}^j_{\mathbf{X}}{}^T \Phi^j_{\mathbf{X}_\mathbf{f}}/\tau}} \tag{3}$$

where $\widehat{\Phi}^i_{\mathbf{X}}{}^T \Phi^j_{\mathbf{X}_\mathbf{f}}$ measures the similarity between any two pointwise embedding and is defined as their inner product where the scalar τ is set to .3.

Nearest Neighbour Loss Our Nearest Neighbour loss links the embeddings of the two shapes and is designed to preserve the given ground truth mapping. Specifically, we first compute the soft correspondence matrix $S_{\mathbf{X}\mathbf{Y}}$ between a pair of shapes, by comparing the rows of $\Phi_{\mathbf{X}}$ to those of $\Phi_{\mathbf{Y}}$ in a differentiable way as done in equation 3. We then evaluate the computed soft map, again, by evaluating how well it transfers the coordinate functions, compared to the given ground truth mapping.

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{NN.} = \sum \|S_{\mathbf{X}\mathbf{Y}} P_{\mathbf{Y}} - \mathbf{T}^{gt}_{\mathbf{X}\mathbf{Y}} P_{\mathbf{Y}}\|_2^2. \tag{4}$$

Note that unlike the linearly invariant loss, this loss is based on comparing $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ directly, without computing any linear transformations. This significantly simplifies the learning process and in particular, reduces the computation of the correspondence at test time to a simple nearest-neighbor search. Despite this, as we show below, due to our strong regularization, our approach achieves superior results compared to the method of [21], based on computing an optimal linear transformation at test time.

Symmetry Commutativity Loss Our next loss aims to link the symmetry map computed for each shape and the correspondence across the two shapes. We achieve this by using the algebraic properties of the functional representation, and especially using the fact that map composition can simply be expressed as matrix multiplication.

Specifically, given a self-symmetry pointwise map on shapes $\mathbf{X}$ and shape $\mathbf{Y}$, we aim to promote the *consistency* between the computed correspondence and the symmetries on each shape. We do this by imposing the following commutativity loss during training:

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{comm.} = \|S_{\mathbf{X}_\mathbf{f}\mathbf{X}} S_{\mathbf{X}\mathbf{Y}} - S_{\mathbf{X}\mathbf{Y}} S_{\mathbf{Y}\mathbf{Y}_\mathbf{f}}\|_2 \tag{5}$$

Intuitively, this loss considers the difference between mapping from $\mathbf{X}$ to $\mathbf{Y}$ and applying the symmetry map on $\mathbf{Y}$, as opposed to applying the symmetry on $\mathbf{X}$ and then mapping from $\mathbf{X}$ to $\mathbf{Y}$. Note that this is similar to the commonly used *Laplacian* commutativity in the functional maps literature. However, rather than promoting isometries, our loss enforces that the computed map respects the self-symmetry structure of each shape, which holds regardless of the deformation class, and is not limited to isometries.

Linearly Invariant Loss Our final loss is optional as we only utilize it to maximize the performance for shape matching and demonstrate the benefits of shape matching in canonical space. This loss was introduced in [21] but we enforce it on the self-symmetry map in an unsupervised way. The linearly invariant assumption on self-symmetry map makes the inference of symmetry map harder at test time. We therefore only use it for ablation study in subsection 5.2. The loss described here considers the embedding of each shape independently and aims to promote the structural property of this embedding: i.e., that the symmetry map should be linear in the embedding space. We define our loss that uses the soft-map to transfer the Euclidean coordinates and compares the result to transferring the coordinates using the ground truth map.

$$\begin{aligned} L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{X}_\mathbf{f}}, \Phi_{\mathbf{Y}}, \Phi_{\mathbf{Y}_\mathbf{f}})_{lin.} = &\sum \|S_{\mathbf{X}\mathbf{X}_\mathbf{f}} P_{\mathbf{X}_\mathbf{f}} - \mathbf{T}^{gt}_{\mathbf{X}\mathbf{X}_\mathbf{f}} P_{\mathbf{X}_\mathbf{f}}\|_2^2 \\ &+ \sum \|S_{\mathbf{Y}\mathbf{Y}_\mathbf{f}} P_{\mathbf{Y}_\mathbf{f}} - \mathbf{T}^{gt}_{\mathbf{Y}\mathbf{Y}_\mathbf{f}} P_{\mathbf{Y}_\mathbf{f}}\|_2^2 \end{aligned} \tag{6}$$

Note that this does not aim for Euclidean coordinates to correspond. Instead, this loss measures how well the predicted map transfers a particular set of functions, compared to the ground truth map.

Overall training Loss We combine the two embedding losses defined in equation 6 and equation 4 with that of commutativity loss defined in equation 5 and define the training loss as follows:

$$L_{tot.} = L_{NN.} + \lambda * L_{lin.} + \gamma * L_{comm.} \tag{7}$$

The scalars γ allows us to weigh the symmetry information differently in a supervised setting where we assume to be given a self-symmetry map and in an unsupervised setting where we work without a symmetry map. Naturally,

we set them higher for the supervised case where enforcing symmetry structure makes more sense than unsupervised case where symmetry is induced by a pairwise matching NN loss and transferred by commutativity loss. We set γ and λ to 1 for supervised setting and .2 for unsupervised setting.

Test Phase At test time, once the network is trained, we simply compute the embedding $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ and do a nearest neighbour search between them to find correspondence between the two shapes. Similarly, to estimate a self-symmetry map, we compute the embedding $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{X_f}}$ and do a nearest neighbour search between them.

Implementation Details We implemented our method in Pytorch [31]. All experiments are run on a Nvidia RTX 2080 graphics processing card and require 16 GB of GPU memory. We learn a $k = 20$ dimensional embedding (basis) for each point cloud. Following [21,41], our feature extractor is also based on the architecture of PointNet. We use a batch size of 8 and learning rate of $1e-4$ and optimize our objective with Adam optimizer in Pytorch [31]. During training, we randomly sample 3000 points from the point cloud and obtain an embedding of 20 dimensions. Our results are not sensitive to small changes in these two parameters. We experimented with an embedding size of $20, 40, 60$ and obtained an average geodesic error in the range $33-36$ on FAUST-R. Similarly, in addition to the 3000 point cloud resolution during training, we also tried a point cloud resolution in the range $2k-4k$ and found almost negligible drop in performance. This can be explained by Pointnet resilience to change in point cloud density.

5. Results

This section is divided into three subsections. First subsection 5.1 shows the experimental comparison of our approach with state-of-the art methods for shape matching and tests our method on a wide spectrum of datasets: the remeshed versions [35] of FAUST dataset [1] and SHREC'16 Partial Correspondence dataset [5]. These experiments validate the promising direction of our embedding based method as it obtains competitive performance on these two benchmarks and especially outperforms LBO based methods on benchmarks with noise. The next subsection5.2 ablates the overall performance and experimentally validates our first claim that shape matching with canonical embedding with appropriate regularization outperforms the linearly invariant embeddings proposed in Marin et. al [21]. We demonstrate this with both symmetry supervision as well as without symmetry supervision. Lastly, section5.3 shows the effectiveness of our method on the symmetry detection task in the presence of noise. We evaluate all results by reporting the per-point-average geodesic distance between the ground truth map and the computed map. All results are multiplied by 100 for the sake of readability. We conclude with an illustration showing a failure case of our method. Due to space constraints, we provide more qualitative results in supplement.

5.1. Shape Matching

We present our results on a full shape matching benchmark dataset FAUST remesh [1, 35], denoted in future subsections as FAUST-R. We also use its two other versions used previously: the Faust aligned dataset used in [41], denoted as FAUST-A and noisy Faust version [21] denoted as FAUST-N. All these dataset contains 100 shapes of 10 different subjects in different poses where each point cloud contains roughly 5000 points. Following prior work, we use the last 20 shapes as a test set and report the performance on this test set. We compare our results with various LBO based methods [6, 8, 41] in Table 1 as well as embedding based methods [12, 21] as they are applicable, in principle, to both partial and complete shape matching.

Baselines We compare with the following two broad approaches that are shown to outperform existing competitors:

LBO based Methods. Such baselines [6, 8, 41] assume to be given as input a mesh representation of a shape as they rely on LBO. While [6,41] directly learn features from raw 3D data similar to our method, they project them into LBO basis. [8] refines pre-computed shot descriptors [45] to learn shape matching. We provide results after refining the point to point map with ZoomOut [22] where applicable for all the methods. Note that in presence of outliers and noise, such a refinement makes the resulting point to point map worse and thus, for FAUST-N, we do not apply it. [8] already has a refinement built in their architecture.

Embedding based Methods. 3D-Coded [12] and Marin et al. [21] are considered state-of-the-art in learning correspondence directly from point cloud representation without relying on LBO. Note that the baseline [21] is somewhat different from others since it requires and thus, learns both basis functions and probe functions (feature descriptors). For fair comparison, we also refine the resulting output of [21] with Zoomout where possible.

Ours. For all results in this paper, we denote our method with symmetry supervision as Ours-sym-Sup and without symmetry supervision as Ours-sym-Unsup. Here symmetry supervision means the access to the ground truth self-symmetry map that is publicly available for Faust-R point

Table 1. Avg. Geodesic Error for Shape Matching on FAUST

Method \ Dataset	FAUST-R	FAUST-N
GeomFM [6]+Zo	**19**	320
DeepShell [8]	**17**	240
Sharma-Ovsjanikov [41]+Zo	50	280
3D-Coded [12]	25	**68**
Marin et al. [21]	70	90
Marin et al. [21] +Zo	50	-
Ours-sym-Sup.	33	**58**
Ours-sym-Sup.+Zo	**18**	-
Ours-sym-Unsup.	48	**69**
Ours-sym-Unsup.+Zo	**18**	-

clouds. While our method already achieves good performance without symmetry ground truth during training, we include Ours-sym-Sup to show the additional gain brought in by additional symmetry supervision during training.

Results and Discussion. As evident in Table 1, we obtain competitive performance on FAUST-R especially when compared to LBO based approaches. LBO eigen functions already form a good basis for shapes and thus, prior work based on it obtains impressive performance. However, performance of this line of work degrades significantly under noise, as shown in the Table 1 and also in [21]. Thus, our method is significantly more resilient to noise than LBO based methods. Compared to embedding based approaches, we obtain slightly better accuracy. In particular, our symmetry-unsupervised version, Ours-sym-Unsup, obtains slightly better performance than our main baseline [21]. We also provide a qualitative example to show comparison with [21] in Figure1. Note that the right foot is mismatched in Marin et al. whereas we transfer it comparatively well without left-right ambiguity. We note that 3D-Coded is also resilient to noise in point clouds and achieves competitive performance in both scenarios.

Figure 1. On the left, we show the source shape. In the middle, we transfer a color function on a target shape using Marin et al. whereas on the right, we show the transfer using our results.

Table 2. Avg. Geodesic Error on partial SHREC benchmarks

Method \ Dataset	Holes	Cuts
Litany et al. [20]	16	13
Sharma-Ovsjanikov [41]	14	16
Marin et al. [21]	12	15
Ours-sym-UnSup.	**10**	**12**

Partial Shape Matching. For a fair comparison with [20, 41], we follow the same experimental setup and test our method on the challenging SHREC'16 Partial Correspondence dataset [5]. The dataset is composed of 200 partial shapes, each containing about few hundreds to 9000 vertices, belonging to 8 different classes (humans and animals), undergoing nearly-isometric deformations in addition to having missing parts of various forms and sizes. Each class comes with a "null" shape in a standard pose which is used as the full template to which partial shapes are to be matched. The dataset is split into two sets, namely cuts (removal of a few large parts) and holes (removal of many small parts). We use the same test set following [41]. Overall, this test set contains 20 shapes each for cuts and holes datasets chosen randomly from the two sets respectively. In addition to [21], we compare with the following two baselines:

Sharma & Ovsjanikov [41]. This baseline relies on learning LBO alignment and thus, is dependent on class and needs to be retrained for each of the 8 classes. We include their results even though our results are class agnostic and thus, significantly more robust and efficient. We obtain these results by running the code provided by the authors.

Litany et al. [20]. This baseline is not learning based and relies on hand crafted features and an expensive optimization scheme on the Stiefel manifold for every pair of shapes at test time. Thus, in terms of computation and ground truth map requirement, it is most expensive.

Results and Discussion We present our findings on partial shape matching in Table 2 where we obtain superior performance on both benchmark datasets for partial shape matching. We would like to stress that baseline such as [41] are class specific and need to be trained each time whereas our method is class agnostic and can obtain good results with a fraction of computational time. Similarly, [21] trains a similar network as ours two times. First, it learns an embedding with a network similar to ours, followed by a similar network training to compute the optimal linear transformation between the two embeddings. Moreover, the test phase also requires running the network twice. Therefore, our method is at least twice faster than this baseline in com-

Figure 2. Shape Matching on a Dog where two legs are missing

Figure 3. Shape Matching on a Cat where the right half is missing.

Table 3. Ablation Study for Shape Matching

Method \ Dataset	FAUST-R
NN	61
$NN + NN_{sym}$	108
NN+ comm.(sup)	33
NN+ comm.(unsup)	50
NN+ comm.(unsup) + Lin.(unsup)	48

putational complexity. We provide qualitative results of our method in Figure 2 and 3.

5.2. Ablation Study

In Table 3, we ablate the overall performance and validates our first claim of learning canonical embedding for shape matching as opposed to the linearly invariant one. NN: This baseline ablates the overall performance of our method and quantifies the gain brought in by the nearest neighbour loss alone during training. It shows the performance if we learn an embedding by just projecting the shapes into the canonical space.

$NN+NN_{sym}$: This baseline shows the results obtained for shape matching with the symmetry supervision simply by optimizing the nearest neighbour loss in symmetry space in addition to the nearest neighbour loss in pairwise setting. This baseline is most important to quantify the coupling effect brought in by our commutative loss.

NN + comm.: This baseline combines the above baseline with the commutativity loss with symmetry supervision and quantifies the gain brought in by symmetry supervision in shape matching. NN + comm.(unsup) +Lin. shows the gain brought in by coupling with a symmetry map in an unsupervised way and represents Ours-sym-Unsup.

Discussion Our ablation study shows the individual importance of the three loss functions. We note that the performance gains brought in by commutative loss on self-symmetry embeddings are significant. More specifically, as evident in Table 3, using just the nearest neighbour loss on a self-symmetry map and a pairwise map, denoted as NN + NN_{sym} in Table 3, overfits badly as there is no explicit information transfer or constraint between the two maps.

5.3. Symmetry Detection

This subsection evaluates our method on the task of symmetry detection in non-rigid shapes. We evaluate it on FAUST aligned dataset (FAUST-A), SCAPE-A as well as its noisy version. We use the usual train-test split where we test on the last 20 shapes for FAUST-A and last 12 shapes for SCAPE-A. We show the comparative results in Table 4 where we compare with multiple baselines. In particular, Ren et al. [34] is considered state-of-the-art and heavily relies on LBO to estimate self-symmetry maps. We show our results with both symmetry supervision, denoted as Ours-sym-Sup as well as without symmetry supervision denoted as Ours-sym-Unsup in Table 4. Similar to Ren et al. [34], we also refine our point to point map by applying zoomout to initial maps. For the noise setting, we simply show results as such and do not apply zoomout refinement as it is based on LBO which is unreliable in a noisy setup. We provide a qualitative example from SCAPE-A in Figure 4 and from FAUST-A in Figure 5 to illustrate our results. Due to space constraints, we provide the rest of the illustrations in supplement.

Figure 4. On the left, we show the source shape and on the right, we show our self-symmetry map color coded.

Discussion Table 4 shows that axiomatic approach of Ren et al. [34] obtains slightly better performance than us on both FAUST-A and SCAPE-A. However, in the presence of noise, its performance suffers significantly. We also remark that we are not aware of any other work that investigates the performance of axiomatic approach for symmetry detection in the presence of outliers. We also see a decrease in accuracy. However, our approach is still resilient to noise and performs significantly better than Ren et al.

Table 4. Avg. Geodesic Error for self-symmetry maps

Method \ Dataset	Faust-A	Scape-A	Faust-N	Scape-N
Nagar-Raman [25]	34	60	-	-
Ren et al.+Zo [34]	**19**	**54**	166	193
Our-sym-Sup.+Zo	**29**	**63**	58	88
Our-sym-Unsup+Zo	50	75	66	95

Figure 5. On the left, we show the source shape and on the right, we show our self-symmetry map color coded.

Limitation of Our Method We show a failure case from SCAPE-A in Figure 6 where our method finds it challenging to disambiguate symmetry. As shown in Figure 6, it maps the right foot of source shape to the left foot on target shape. Marin et al. still performs worse than us as it fails to disambiguate the lower leg of source shape from the lower right leg of target shape. Human poses are quite diverse and symmetry detection in such challenging poses remain a challenge for extrinsic approaches like ours and Marin et al.

Figure 6. On the left, we show the source shape. In the middle, we transfer a color function on a target shape using Marin et al. whereas on the right, we show our result.

6. Conclusion

In shape correspondence literature, partial shape matching and full shape matching are generally tackled by two different sets of methods which obtain impressive results in one of the two respective domains. Similarly, symmetry detection and shape matching are also learned or modelled separately. We presented a simple, general but promising approach that provides a unifying framework and reduces pairwise as well as self-symmetry map estimation to a nearest neighbour search in a canonical embedding. Our approach is significantly more resilient to noise than methods based on predefined basis/embedding functions. We believe our key idea of coupling a self-symmetry and a pairwise map via commutativity will encourage future work to explore similar constraints in unsupervised or weakly supervised learning of canonical embeddings. In future, we plan to extend and explore our work on non-human datasets.

7. Acknowledgement

Parts of this work were supported by the ERC Starting Grant StG-2017-758800 (EXPROTEA), and ANR AI chair AIGRETTE. We thank anonymous reviewers for their comments and suggestions. We also thank Jing Ren for providing code for color map visualization.

References

[1] 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, 5

[2] 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

[3] Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. Embedding shapes with Green's functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017. 2

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

[5] 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. 5, 6

[6] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, 2020. 2, 5, 6

[7] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *CVPR*, pages 7473–7483, 2021. 2

[8] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. In *NeurIPS*, volume 34, 2020. 2, 5, 6

[9] Clara Fernandez-Labrador, Ajad Chhatkuli, Danda Pani Paudel, Jose J Guerrero, Cédric Demonceaux, and Luc Van Gool. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *ECCV*, 2020. 2

[10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[11] Dvir Ginzburg and D. Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *ECCV*, 2020. 2

[12] 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. 2, 5, 6

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

[14] 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

[15] Maxime Kirgo, Simone Melzi, Giuseppe Patanè, Emanuele Rodolà, and Maks Ovsjanikov. Wavelet-based heat kernel derivatives: Towards informative localized shape analysis. In *Computer Graphics Forum*. Wiley Online Library, 2020. 3

[16] I. Kokkinos. Ubernet: Training universal cnn for low mid and high level vision with diverse datasets and limited memory. In *CVPR*, 2017. 2

[17] 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

[18] Y. Lipman, Xiaobai Chen, I. Daubechies, and T. Funkhouser. Symmetry factored embedding and distance. In *SIGGRAPH*, 2010. 2

[19] Or Litany, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, pages 5660–5668, 2017. 2, 4

[20] 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. 1, 2, 6

[21] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. In *NeurIPS*, volume 33, pages 1608–1620, 2020. 1, 2, 3, 4, 5, 6

[22] 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. 5

[23] Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report*, 2012. 2

[24] 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

[25] Rajendra Nagar and Shanmuganathan Raman. Fast and accurate intrinsic symmetry detection. In *ECCV*, 2018. 1, 2, 8

[26] Natalia Neverova, David Novotny, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. In *NeurIPS*, 2020. 1

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

[28] 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. 1, 2, 3

[29] 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. 2

[30] Maks Ovsjanikov, Quentin Mérigot, Viorica Pătrăucean, and Leonidas Guibas. Shape matching via quotient spaces. In *SGP*, page 1–11, 2013. 2

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 5

[32] 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. 3

[33] D. Raviv, A. Bronstein, M. Bronstein, and R. Kimmel. Full and partial symmetries of non-rigid shapes. *International Journal of Computer Vision*, 89:18–39, 2010. 2

[34] Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph.*, 39(6), Nov. 2020. 1, 2, 7, 8

[35] 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, 5

[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] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, pages 1617–1627, 2019. 1, 2, 3

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

[39] Abhishek Sharma. Foreground clustering for joint segmentation and localization in videos and images. In *NeurIPS*, volume 31, 2018. 2

[40] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconvdae: Deep volumetric shape learning without object labels. In *ECCV*, 2016. 2

[41] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. In *NeurIPS*, volume 33, 2020. 1, 2, 3, 5, 6

[42] Abhishek Sharma and Maks Ovsjanikov. Matrix decomposition on graphs: A functional view. *arXiv*, 2021. 1

[43] Yifei Shi, Junwen Huang, Hongjia Zhang, Xin Xu, Szymon Rusinkiewicz, and Kai Xu. Symmetrynet: Learning to predict reflectional and rotational symmetries of 3D shapes from single-view RGB-D images. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 39, 2020. 2

[44] Oshane O. Thomas, Hongyu Shen, Ryan L. Raaum, William E.H. Harcourt-Smith, John D. Polk, and Mark Hasegawa-Johnson. Automated morphological phenotyping using learned shape descriptors and functional maps: A novel approach to geometric morphometrics. *bioRxiv*, 2021. 1

[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. 2, 5

[46] Fudong Wang, Gui-Song Xia, Nan Xue, Yipeng Zhang, and M. Pelillo. A functional representation for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2737–2754, 2020. 1

[47] 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. 2

[48] Yan Wu, Jun Yang, and Jinlong Zhao. Partial 3d shape functional correspondence via fully spectral eigenvalue alignment and upsampling refinement. *Comput. Graph.*, 92:99–113, 2020. 2