



**HAL**  
open science

# Deformation-Guided Unsupervised Non-Rigid Shape Matching

Aymen Merrouche, João Regateiro, Stefanie Wuhler, Edmond Boyer

► **To cite this version:**

Aymen Merrouche, João Regateiro, Stefanie Wuhler, Edmond Boyer. Deformation-Guided Unsupervised Non-Rigid Shape Matching. BMVC 2023 - 34th British Machine Vision Conference, Nov 2023, Aberdeen, United Kingdom. pp.1-22. hal-04249563

**HAL Id: hal-04249563**

**<https://inria.hal.science/hal-04249563>**

Submitted on 9 Nov 2023

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

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



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Deformation-Guided Unsupervised Non-Rigid Shape Matching

Aymen Merrouche<sup>1</sup>  
aymen.merrouche@inria.fr

João Regateiro<sup>2</sup>  
joaoregateiro@gmail.com

Stefanie Wuhrer<sup>1</sup>  
stefanie.wuhrer@inria.fr

Edmond Boyer<sup>1</sup>  
edmond.boyer@inria.fr

<sup>1</sup> Inria centre  
University Grenoble Alpes  
Grenoble, France

<sup>2</sup> Interdigital  
Cesson-Sévigné, France

---

## Abstract

We present an unsupervised data-driven approach for non-rigid shape matching. Shape matching identifies correspondences between two shapes and is a fundamental step in many computer vision and graphics applications. Our approach is designed to be particularly robust when matching shapes digitized using 3D scanners that contain fine geometric detail and suffer from different types of noise including topological noise caused by the coalescence of spatially close surface regions. We build on two strategies. First, using a hierarchical patch based shape representation we match shapes consistently in a coarse to fine manner, allowing for robustness to noise. This multi-scale representation drastically reduces the dimensionality of the problem when matching at the coarsest scale, rendering unsupervised learning feasible. Second, we constrain this hierarchical matching to be reflected in 3D by fitting a patch-wise near-rigid deformation model. Using this constraint, we leverage spatial continuity at different scales to capture global shape properties, resulting in matchings that generalize well to data with different deformations and noise characteristics. Experiments demonstrate that our approach obtains significantly better results on raw 3D scans than state-of-the-art methods, while performing on-par on standard test scenarios. Our code is publicly available at <https://gitlab.inria.fr/amerrouc/deformation-guided-unsupervised-non-rigid-shape-matching>.

## 1 Introduction

Non-rigid shape matching is the problem of finding correspondences between two geometric objects which differ in their pose or shape. It provides deformation or transfer functions between shapes, including motion fields when considering a single object in motion. Shape matching is a critical part of many applications in computer vision and computer graphics. Recently, data-driven approaches that learn plausible correspondences from training datasets have shown significant improvements in matching shapes with complex deformations, for instance humans with clothing. Yet, they often require ground truth correspondences for training. We consider the unsupervised case where no ground truth is provided.

Shape matching has been studied extensively over the past decades. Recent data-driven strategies include matching shapes in a spectral basis or adversely in the spatial domain.

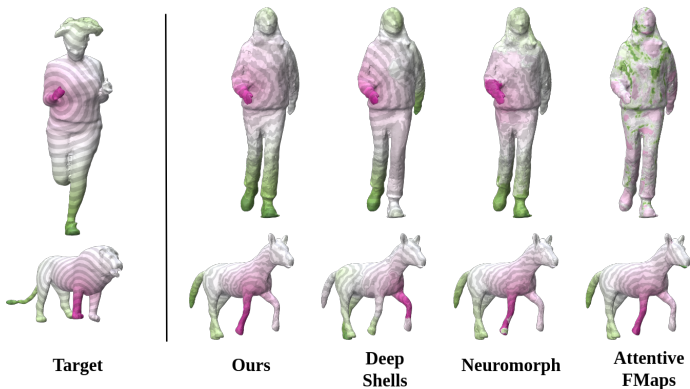


Figure 1: Overview of input, output and our results compared to Deep Shells [12], Neuromorph [13] and AttentiveFMaps [21] on challenging data. Each point on target (left) is assigned a color, which is transferred to the source (right) using correspondences computed by different methods. Our results are both globally and locally correct.

Spectral methods *e.g.* [21, 22] find global maps between real-valued functions defined in spectral shape spaces, allowing them to generalize to different deformations and sampling rates. However, correct shape topology is assumed for spectral decompositions. Spatial methods learn the alignment of shapes *e.g.* [13, 20, 24] or consider the registration to a given template by casting the problem as a vertex classification *e.g.* [12, 25]. They are robust to changes in topology, but exhibit low generalisation abilities to deformations unseen during training due to the unstructured feature space used for matching.

This demonstrates that matching two shapes robustly remains difficult, especially when considering raw 3D scans. In addition to large non-rigid deformations and different sampling distributions of the shapes, raw 3D scans suffer from sensor noises including geometric noise of the vertex positions, and topological noise caused by the coalescence of spatially close surface regions. These noises distort both the extrinsic and intrinsic geometry of the shape that is being captured. Matching raw 3D scans remains a challenging problem.

We design a robust matching technique that generalizes well and is robust to noise including topological noise present in raw 3D scans. This is achieved by a spatial matching method that retains the robustness of spatial approaches while producing correspondence maps that, as in the case of spectral approaches, capture global geometric shape properties.

This is done by relying on two strategies. First, our method considers a hierarchical approach that builds correspondences at different scales of the shapes. We associate shape elements in a coarse-to-fine approach going from coarse surface patches to vertices, which has two main advantages. First, it significantly increases robustness to noise. Second, it reduces the dimensionality of the problem, which allows for efficient unsupervised learning. At coarse scales, we can represent the matching between two shapes as a small matrix, on which we can efficiently impose desirable properties such as cycle consistency or minimal distortion. This information can subsequently be leveraged efficiently at finer scales.

Second, our method enforces correspondences to agree with a deformation model that controls the spatial continuity of the produced matching. We choose a piece-wise near-rigid deformation model that can represent complex non-rigid deformations, and successfully models deformations of humans, possibly in clothing, and other vertebrates. Imposing

spatial continuity on our hierarchical maps constrains correspondences as a whole by linking the matching of individual shape elements, thus allowing to capture global shape properties.

To combine these two strategies, an association network estimating the matching between two shapes is combined with a deformation network estimating the induced alignment in 3D. Both networks operate in a hierarchical fashion, which allows for unsupervised training. Although multi-scale matching and deformation guidance were used in optimisation based matching approaches *e.g.* [4, 5, 6, 50], the novelty of our work is the design of a data-driven method that combines these two strategies in an unsupervised fashion to build a structured feature space where shapes can be embedded and matched with robustness to noise and generalisation to different deformations and sampling rates.

We demonstrate experimentally that our method performs on-par with state-of-the-art for standard test scenarios. Furthermore, our approach significantly outperforms existing methods when matching raw 3D scans after having been trained exclusively on clean data. Figure 1 shows this for challenging data with topological noise (top row).

In summary, our main contributions are as follows. Firstly, we propose a novel spatial unsupervised data-driven non-rigid shape matching approach that combines multi-scale association maps with a piecewise near-rigid deformation model. Secondly, we outperform state-of-the-art on matching raw 3D scans captured using a multi-camera platform.

## 2 Related Work

Non-rigid shape matching has been studied extensively during the past decades. We focus on unsupervised learning-based approaches as our method falls into this category. For a deeper review we refer to a recent survey [7]. Existing works can be roughly categorised into two main classes : spatial methods and spectral methods.

**Spatial Methods** Early spatial methods [8, 15] proposed supervised strategies to extract a matching between point clouds by learning the deformation in 3D space that best aligns the shapes to a common template. Groueix *et al.* [16] use an unsupervised network for point cloud matching by template-free deformation, where cycle consistency of the deformation is used as a supervising signal to define good correspondences. Recent methods embed input shapes in a feature space where the shapes are aligned guided by criteria on the alignment in 3D. CorrNet3D [34] and DPC [17] estimate an association in feature space between two point clouds. CorrNet3D guides this association by the reconstruction of the two point clouds, while DPC uses self- and cross-reconstruction of the two point clouds for this purpose. Closely related in spirit to our method is NeuroMorph [18], an unsupervised mesh matching method that simultaneously learns associations in feature space and an interpolation between two meshes in 3D. The interpolation is used as a criterion to guide the correspondence search in feature space. In our method, the deformation search is used to guide the global correspondence search in addition to the hierarchical modeling of the matching.

**Spectral Methods** Functional maps (FM) [19] introduced the idea of considering point-to-point matching as a special case of mappings between functions defined on shapes, and forms the basis of many matching algorithms. Functions defined on points are projected onto the eigenfunctions of the Laplace-Beltrami operator of the shapes, where the mapping between functions is estimated. This mapping is then used to extract a point-to-point matching. Deep functional maps were introduced in Litany *et al.* [20] where instead of using pre-defined point descriptors as the functions to match, the output of a neural network

is used. Deep functional maps were successfully extended to the unsupervised regime, by minimizing a distortion measure of the extracted point map [14, 17], or by exploiting the desired structural properties of the maps directly in the spectral domain [29, 30]. DUO-FM [9] proposes to incorporate the complex functional maps in the representation to estimate orientation preserving maps. More recently, AttentiveFMaps [21] propose a spectral attention framework to combine multiple resolution functional maps producing maps that are particularly robust at handling non-isometric shapes.

Other spectral approaches combine the alignment in a spectral shape space with an alignment in ambient space. DeepShells [12] is a functional map based method that also aligns shapes in 3D. The learning criterion of the map measures the alignment tightness between the deformed source and the target shape in ambient space. Spectral Teacher Spatial Student (STS) [11] proposes a student teacher mechanism where one of the spatial methods DPC [20] or CorrNet3D [34] is used as a student network to embed shapes in a feature space where the point-to-point mapping is estimated via feature similarity. These features are then used as functions to match in a teacher standard functional maps mechanism. STS uses the spectral teacher to build a feature space that captures global shape properties. We use the deformation model constraint at multiple scales to achieve the same goal.

### 3 Method

This section details our unsupervised<sup>1</sup> data-driven matching approach. Given as input two 3D meshes  $\mathcal{X}$  and  $\mathcal{Y}$ , our method outputs a hierarchical mapping between  $\mathcal{X}$  and  $\mathcal{Y}$  as well as the deformation in 3D that this mapping induces.

Our method represents  $\mathcal{X}$  and  $\mathcal{Y}$  hierarchically with a hierarchy of surface patches using a greedy approach based on furthest point sampling inspired by [28]. We select  $L + 1$  patch resolutions where patch 0 represents the vertex level (where each patch is restricted to the vertex itself) and  $L$  represents the coarsest patch level. In the following we denote the patches as  $(P_i^l)_{1 \leq i \leq n_l}$  for  $l = 0, \dots, L$  and their centers as  $\mathcal{C}_l = (c_i^l \in \mathbb{R}^3)_{1 \leq i \leq n_l}$ .

Our method is implemented by a composition of two networks. First, an association network that aligns  $\mathcal{X}$  and  $\mathcal{Y}$  in a hierarchical feature space as detailed in Sec. 3.1. Second, a deformation network that constrains these hierarchical maps by aligning  $\mathcal{X}$  and  $\mathcal{Y}$  in 3D space by fitting a hierarchical patch-wise near-rigid deformation model as detailed in Sec. 3.2. Fig. 2 shows the architecture of our network.

#### 3.1 Association Network

This network estimates association maps hierarchically, from coarse patches to vertices. Association maps are computed as matrices  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$  and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l$  mapping from surface patches of  $\mathcal{X}$  to surface patches of  $\mathcal{Y}$  and vice versa. Element  $(i, j)$  of  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$  contains a matching score between the  $i$ -th patch of  $\mathcal{X}$  and the  $j$ -th patch of  $\mathcal{Y}$  at hierarchy level  $l$ , estimated via feature similarity. The shape features are extracted by a convolutional graph neural network, based on the FeaStConv [62] operator. At hierarchy level  $l$ , FeaStConv acts on patch neighborhoods. On the vertex scale (hierarchy level 0), where convolutions act on vertex neighborhoods, we use vertex coordinates and normals as input features. We use patch-wise max-pooling to go to a coarser hierarchy level. This allows to compute patch-wise

<sup>1</sup>We assume training shapes without correspondence information that are rigidly pre-aligned in  $\mathbb{R}^3$  to have roughly the same front-back orientation. This is sometimes referred to as weakly supervised in the literature.

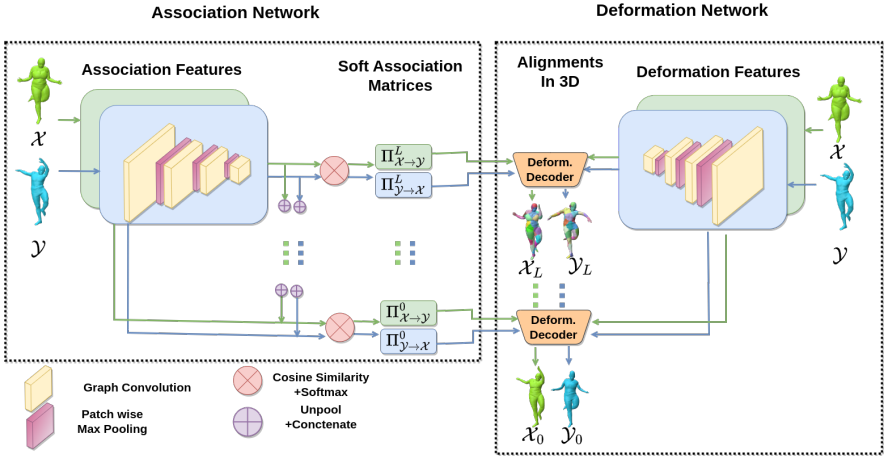


Figure 2: Network architecture. The network takes as input  $\mathcal{X}, \mathcal{Y}$  decomposed into a hierarchy of surface patches. It is decomposed into two networks that work from coarse to fine levels, sequentially per level. An association network extracts coarse-to-fine correspondences between  $\mathcal{X}$  and  $\mathcal{Y}$  as inter-patch association matrices (from  $\Pi^L$  to  $\Pi^0$ ). A deformation network outputs deformations that respect these associations in 3D (from  $\mathcal{X}_L, \mathcal{Y}_L$  to  $\mathcal{X}_0, \mathcal{Y}_0$ ).

features for hierarchy level  $l$  on  $\mathcal{X}$  and  $\mathcal{Y}$ , which we denote by  $\mathcal{X}^l = (\mathbf{x}_i^l \in \mathbb{R}^{d_l})_{1 \leq i \leq n_l}$  and  $\mathcal{Y}^l = (\mathbf{y}_i^l \in \mathbb{R}^{d_l})_{1 \leq i \leq m_l}$ , where  $n_l, m_l$  are the number of patches at level  $l$  for  $\mathcal{X}$  and  $\mathcal{Y}$ .

Features are computed in a fine-to-coarse manner. To allow for coarse-to-fine hierarchical matching, we combine the feature of a patch at level  $l$  with the features of all its parent patches in coarser levels  $l+1, \dots, L$ . We denote these combined features by  $\mathcal{X}^{L,l} = (\mathbf{x}_i^{L,l} \in \mathbb{R}^{d_l})_{1 \leq i \leq n_l}$ . At the coarsest level  $L$ , we set  $\mathcal{X}^{L,L} = \mathcal{X}^L$ . At level  $l$ , we unpool  $\mathcal{X}^{L,l+1}$  to level  $l$ . This is done by unpooling to the vertex level, and employing max-pooling to all vertices of  $(P_i^l)$ , leading to features  $\text{Unpool}_{l+1,l}(\mathcal{X}^{L,l+1})$ . We concatenate  $\text{Unpool}_{l+1,l}(\mathcal{X}^{L,l+1})$  and  $\mathcal{X}^l$ . This concatenated feature may be non-smooth on the surface along patch boundaries in hierarchy level  $l+1$ . To remedy this, we perform one FeaStConv convolution to get  $\mathcal{X}^{L,l}$ . The same computations yield  $\mathcal{Y}^{L,l} = (\mathbf{y}_i^{L,l} \in \mathbb{R}^{d_l})_{1 \leq i \leq m_l}$ .

The association matrices between patches of  $\mathcal{X}$  and  $\mathcal{Y}$  at hierarchy level  $l$  are computed using cosine similarity as a distance measure in feature space, similar to previous works [10, 20]. We then employ softmax to get normalized similarity scores:

$$(\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l)_{ij} := \frac{\exp(s_{ij}^l)}{\sum_{k=1}^{m_l} \exp(s_{ik}^l)} \quad (1) \quad (\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l)_{ij} := \frac{\exp(s_{ji}^l)}{\sum_{k=1}^{n_l} \exp(s_{ki}^l)} \quad (2)$$

with  $s_{ij}^l := \left( \langle \mathbf{x}_i^{L,l}, \mathbf{y}_j^{L,l} \rangle_2 \right) / \left( \|\mathbf{x}_i^{L,l}\|_2 \|\mathbf{y}_j^{L,l}\|_2 \right)$ .

## 3.2 Deformation Network

The deformation network deforms  $\mathcal{X}$  to  $\mathcal{Y}$  and  $\mathcal{Y}$  to  $\mathcal{X}$  while respecting association matrices  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$  and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l$ , respectively, for every level  $l$  of the hierarchy.

**Deformation Model** We use a patch-based deformation model that represents a shape deformation as a collection of patch-wise near-rigid deformations [8]. It allows to represent

complex deformations while drastically reducing the deformation parameters. We summarize how to deform hierarchy level  $l$  and omit subscript  $l$  to simplify notation. Given  $\mathcal{X}$  with its decomposition into surface patches  $(P_i)_{1 \leq i \leq n}$  along with their centers  $\mathcal{C} = (c_i \in \mathbb{R}^3)_{1 \leq i \leq n}$ , each patch  $P_i$  is associated with a rotation matrix  $R_i \in \mathbb{R}^{3 \times 3}$  and a new center position  $u_i \in \mathbb{R}^3$ . If we denote by  $x_0(v) \in \mathbb{R}^3$  the original position of vertex  $v$  in  $\mathcal{X}$ , the rigid deformation of a vertex  $v$  according to  $P_i$  can be written as  $x_i(v) = R_i(x_0(v) - c_i) + u_i \in \mathbb{R}^3$ . The rigid deformations of all patches  $(P_i)_{1 \leq i \leq n}$  are blended at the vertex level using weighting functions  $\alpha_i(v) \in \mathbb{R}$  defined as a Gaussian of the geodesic distance<sup>2</sup> of  $v$  to  $c_i$ .

To ensure consistent deformations between neighboring patches  $P_i$  and  $P_j \in \mathcal{N}(P_i)$ , transformations are constrained by minimizing

$$l_{\text{rig}}(\mathcal{X}) = \sum_{(P_i)_{1 \leq i \leq n}} \sum_{P_j \in \mathcal{N}(P_i)} \sum_{v \in P_i \cup P_j} E_v^{ij} \text{ with} \quad (3)$$

$$E_v^{ij} = (\alpha_i(v) + \alpha_j(v)) \times \|x_i(v) - x_j(v)\|_2^2. \quad (4)$$

**Architecture** Our network deforms both  $\mathcal{X}$  to  $\mathcal{Y}$  and  $\mathcal{Y}$  to  $\mathcal{X}$  symmetrically. We detail  $\mathcal{X}$  to  $\mathcal{Y}$  in what follows. At level  $l$  of the hierarchy, we know matrix  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$  from the association network. The deformation network consists of a feature extractor and a deformation decoder. The feature extractor is a convolutional neural network identical to the one used in the association network. It extracts patch-wise features for  $\mathcal{X}$  and  $\mathcal{Y}$  at level  $l$  called  $\tilde{\mathcal{X}}^l = (\tilde{x}_i^l \in \mathbb{R}^{d_l})_{1 \leq i \leq n_l}$  and  $\tilde{\mathcal{Y}}^l = (\tilde{y}_i^l \in \mathbb{R}^{d_l})_{1 \leq i \leq m_l}$ . We choose to decouple the features used for association and deformation to allow the networks to learn optimal features independently for each task.

The deformation decoder consists of a graph convolutional network followed by an MLP. It outputs rotation parameters  $(R_i \in \mathbb{R}^6)_{1 \leq i \leq n_l}$  and new center positions  $(u_i \in \mathbb{R}^3)_{1 \leq i \leq n_l}$  for every patch of level  $l$  of  $\mathcal{X}$ . It takes as input patch centers  $C_l^{\mathcal{X}} \in \mathbb{R}^{n_l \times 3}$ , patch-wise features  $\tilde{\mathcal{X}}^l \in \mathbb{R}^{n_l \times d_l}$ , patch centers of  $\mathcal{X}$  projected in  $\mathcal{Y}$  using the association matrix  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l C_l^{\mathcal{X}} \in \mathbb{R}^{n_l \times 3}$ , which represent spatial targets for the patch centers, and patch-wise features of  $\mathcal{X}$  projected in patch-wise features of  $\mathcal{Y}$  using the association matrix  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l \tilde{\mathcal{X}}^l \in \mathbb{R}^{n_l \times d_l}$ . We use the 6D representation of rotation matrices introduced in [65] to allow for efficient learning. Applying the resulting transformations leads to the deformed shape  $\mathcal{X}_l$ .

### 3.3 Learning Criteria

We train our model using five self-supervised criteria at each hierarchy level  $l$

$$l_{\text{network}} = \sum_{l=0}^L (\lambda_{\text{g}}^l l_{\text{geodesic}}^l + \lambda_{\text{c}}^l l_{\text{cycle}}^l + \lambda_{\text{r}}^l l_{\text{rec}}^l + \lambda_{\text{m}}^l l_{\text{match}}^l + \lambda_{\text{ri}}^l l_{\text{rigidity}}^l), \quad (5)$$

where the individual loss terms for each level  $l_{\text{geodesic}}^l, l_{\text{cycle}}^l, l_{\text{rec}}^l, l_{\text{match}}^l, l_{\text{rigidity}}^l$  are weighted by corresponding weights  $\lambda_{\text{g}}^l, \lambda_{\text{c}}^l, \lambda_{\text{r}}^l, \lambda_{\text{m}}^l, \lambda_{\text{ri}}^l$  and detailed below.

**Geodesic Distance Distortion Criterion** The first criterion favours maps that preserve geodesic distances, and is commonly used for shape matching *e.g.* [10, 11]. We implement this by minimizing  $l_{\text{geodesic}}^l = \|\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l D_{\mathcal{Y}}^l \Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l{}^T - D_{\mathcal{X}}^l\|_2^2 + \|\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l D_{\mathcal{X}}^l \Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l{}^T - D_{\mathcal{Y}}^l\|_2^2$ , where  $D_{\mathcal{X}}^l$  and  $D_{\mathcal{Y}}^l$  are the geodesic distance matrices of  $\mathcal{X}$  and  $\mathcal{Y}$ , restricted to the patch

<sup>2</sup>In the original paper [10], the Euclidean distance is used to define the blending functions. We use the geodesic distance instead to account for points that might be extrinsically close to and intrinsically far from  $c_i$ .



centers of level  $l$ . This criterion measures the geodesic distortion of maps  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$  and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l$ . We only use this loss on coarse levels because it is computationally prohibitive for finer levels.

**Cycle Consistency Criterion** This criterion asks  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$  and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l$  to be cycle consistent, i.e. every point going through a length two cycle is mapped back to itself. It is implemented by minimizing  $l_{\text{cycle}}^l = \|\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l(\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l C_i^{\mathcal{X}}) - C_i^{\mathcal{X}}\|_2^2 + \|\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l(\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l C_i^{\mathcal{Y}}) - C_i^{\mathcal{Y}}\|_2^2$ .

**Self Reconstruction Criterion** The third criterion aims to identify each patch in feature space, to avoid many patches being mapped to the same patch. This criterion helps handling intrinsically symmetric shapes, where different patches share the same intrinsic geometry. It is implemented by minimizing  $l_{\text{rec}}^l = \|\Pi_{\mathcal{X} \rightarrow \mathcal{X}}^l C_i^{\mathcal{X}} - C_i^{\mathcal{X}}\|_2^2 + \|\Pi_{\mathcal{Y} \rightarrow \mathcal{Y}}^l C_i^{\mathcal{Y}} - C_i^{\mathcal{Y}}\|_2^2$ , where  $\Pi_{\mathcal{X} \rightarrow \mathcal{X}}^l$  and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{Y}}^l$  are self association matrices computed similar to Eq. 1 and 2.

**Matching Criterion** The matching criterion aims to deform one shape to the other using the computed associations. It serves two roles. First, ensuring that the deformation realised by the deformation network matches the correspondences computed by the association network. Second, allowing for extrinsic geometric control on the correspondences. It is implemented by minimizing  $l_{\text{match}}^l = \|C_i^{\mathcal{X}^l} - \Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l C_i^{\mathcal{Y}}\|_2^2 + \|C_i^{\mathcal{Y}^l} - \Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l C_i^{\mathcal{X}}\|_2^2$ , where  $C_i^{\mathcal{X}^l}$  and  $C_i^{\mathcal{Y}^l}$  are the deformed cluster centers of  $\mathcal{X}_l$  and  $\mathcal{Y}_l$  respectively.

**Rigidity Criterion** The rigidity criterion aims for consistent transformations between neighboring patches at all hierarchy levels as  $l_{\text{rigidity}}^l = l_{\text{rig}}(\mathcal{X}^l) + l_{\text{rig}}(\mathcal{Y}^l)$ , with  $l_{\text{rig}}$  defined in Eq. 3. The matching and rigidity criteria encourage the network to explain the similarity in feature space on the deformed shapes in 3D at every level of the patch hierarchy. At every level of the hierarchy, the deformation for every vertex on the shapes is evaluated even though the matching criterion evaluates the deformation result only on the patch centers of the deformed shapes. This means that even if the association matrices are sparse at coarse levels they consider the full correspondence between the two shapes. This acts as a strong inductive bias in our network, which promotes spatially continuous maps.

**Fine Tuning** Our full network is trained in an end-to-end fashion. After training, the network produces good quality matchings in a single forward pass. We improve these matchings by optimizing the network for each pair  $\mathcal{X}$  and  $\mathcal{Y}$  only, with the same architecture and hyperparameters. This is possible as all of our losses are unsupervised and do not require any ground truth information. We refer to this specialisation as fine tuning. We extract final dense correspondences by applying argmax on vertex associations  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^0$  and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^0$ .

## 4 Experiments

This section presents results of our method, including comparative evaluations, for human and animal shapes. For human shapes, we provide an extensive evaluation protocol to demonstrate that our method generalizes well to new body shapes and poses, sampling densities, topological noise, and raw acquisition data. In all of our experiments we fix the number of patches to 50, 200 and 800. More details are provided in the appendices; Appendix A provides implementation details, Appendix B provides details on data processing, Appendix C provides an ablation study proving the benefits of the main parts of our method i.e. hierarchy in feature space, use of a deformation model and fine tuning, and Appendix D provides additional quantitative and qualitative comparisons and qualitative deformation results.

**Evaluation metrics** We consider two metrics that compare the ground truth mapping



$\Pi_{\rightarrow}^{gt}$ , with an estimated mapping  $\Pi_{\rightarrow}$ . The mean geodesic error (MGE):

$$\text{MGE}(\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}) := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} d_{\mathcal{Y}}(\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}(x), \Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^{gt}(x)), \quad (6)$$

and the cycle geodesic error (CycleGE):

$$\text{CycleGE}(\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}) := \frac{1}{|\mathcal{X}|^2} \sum_{\substack{x_1, x_2 \\ \in \mathcal{X} \times \mathcal{X}}} \left| 1 - \frac{d_{\mathcal{X}}(x_1, x_2)}{d_{\mathcal{X}}(\tilde{x}_1, \tilde{x}_2)} \right| \text{ with } \tilde{x}_i := \Pi_{\mathcal{Y} \rightarrow \mathcal{X}}(\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}(x_i)), \quad (7)$$

where  $d_{\mathcal{X}}$ ,  $d_{\mathcal{Y}}$  are the geodesic distances on meshes  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively, normalised by the square root of their area. These metrics allow to evaluate the distance between the true and estimated correspondences of a point  $x$  on  $\mathcal{Y}$ , and the metric distortion induced by matching a pair of points using a length two cycle.

**Competing methods** We compare our results against four state-of-the-art methods. FM based AttentiveFMaps [24]. Spatial mesh matching method Neuromorph [14] that guides the correspondences search using an interpolation while we search for a deformation. Deep Shells [14] an FM method that combines a spectral alignment with a spatial alignment. Point cloud matching method STS [14] that uses an FM pipeline to extract global geometric properties, while we leverage spatial continuity at multiple scales. All methods that support supervised and unsupervised training were trained in the unsupervised regime.

## 4.1 Evaluation on human data

**Data** We train our method on clean pre-processed data, and test it both on (possibly degraded) pre-processed data and raw 3D acquisitions. The pre-processed data we use is the extended FAUST dataset [2], which extends FAUST [9] with synthetically generated body shapes and poses. Extended FAUST contains 561 naked human meshes with 21 body shapes in roughly 27 poses. We select 451 meshes for training and 110 meshes for testing, where the test set contains body shapes and poses not used for training. We re-mesh each mesh independently and uniformly to about 5k vertices using [53] to remove vertex density as a discriminative feature. We design test sets to evaluate different types of generalization.

**Generalization to unseen body shape and pose** The first test set considers pairs of models of the extended FAUST test set that differ in body shape and pose. It allows to evaluate how well methods generalize to body shapes and poses not observed during training.

**Robustness to sampling density** To evaluate robustness to different sampling strategies, we consider two test sets using resampled versions of the extended FAUST test set. The first one resamples the test set uniformly to contain  $\approx 15k$  vertices. The second one resamples the test set with curvature adapted triangles [26] to contain  $\approx 5k$  vertices.

**Generalization to different topology** Our method is particularly robust to changes in topology. To demonstrate this, we design a test set by altering the topology of the extended FAUST test set. Extended FAUST contains many shapes with near-contacts between different body parts, and we detect these parts and replace them by gluing the parts in contact. In practice, we achieve this by detecting self-intersections, deleting mesh parts located inside the mesh, and by filling the resulting holes using Poisson surface reconstruction [13]. The final reconstructed surface contains  $\approx 6k$  vertices.

**Generalization to raw 3D acquisitions** Our goal is to have a method that can be applied to raw scan data without pre-processing. Hence, we test our method on two test sets of raw acquisition data. All these data are plagued by acquisition noise, which alters the geometry

or topology. The first one contains 17 scans (136 pairs) of minimally dressed humans of CHUM [24] of two female subjects and one male subject, all in different poses. Each mesh contains about 15k vertices. The second one contains 12 scans (66 pairs) of humans in everyday clothing of two female subjects and two male subjects of the dataset 4DHumanOutfit [10], all in different poses. These scans were captured in a multi-view acquisition platform with 68 synchronized cameras.

**Comparison to state-of-the-art** We first provide a quantitative evaluation with Neuromorph, AttentiveFMaps and Deep Shells, where all methods were trained on the extended FAUST training set and tested on the different test sets. Tab. 1 shows the results. We could not include STS in this comparison, the code for pre-processing new data is not provided, making training or testing on our data difficult.

Method	Pre-processed data						Raw 3D acquisitions					
	Body shape and pose		Sampling density				Topology		Naked		Clothed	
			Uniform 15K vert.	Non-uniform 5K vert.								
Deep Shells	13.39	0.0694	9.11	0.0564	17.10	0.1409	20.48	0.1608	16.40	0.0775	26.49	0.2275
Neuromorph	11.61	0.1315	14.45	0.1527	37.87	0.5389	11.92	0.1231	13.83	0.1407	16.01	0.1429
AttentiveFMaps	<b>2.65</b>	<b>0.0171</b>	<b>2.47</b>	<b>0.0160</b>	<b>2.82</b>	<b>0.0385</b>	18.93	0.3426	<u>11.38</u>	0.1608	49.19	0.7701
Ours	<u>4.09</u>	<u>0.0259</u>	4.15	<u>0.0304</u>	<u>5.85</u>	<u>0.0552</u>	<b>8.22</b>	<b>0.0441</b>	<b>3.74</b>	<b>0.0349</b>	<b>6.02</b>	<b>0.0463</b>

Table 1: Comparison to state-of-the-art on human data. Each column shows MGE (Eq. 6) on left and CycleGE (Eq. 7) on right. Best scores in bold, second best scores underlined.

For all test sets, our method outperforms Neuromorph and Deep Shells. Furthermore, our method performs on-par overall with AttentiveFMaps for pre-processed data. While AttentiveFMaps performs slightly better than our method when applied to clean pre-processed data (unseen body shape and pose and different sampling density columns in Tab. 1), it degrades significantly in the presence of topology changes. We believe that this is due to the global nature of spectral shape decompositions that naturally favours clean data. Our method significantly outperforms AttentiveFMaps in this case as our method generalizes well to topological errors. Our approach builds on a more local strategy, though also considering global geometric shape properties. This results in rare failure cases with important symmetry ambiguities but provides a strongly increased robustness to noise present in real situations, with raw acquisition data for example.

For raw acquisition data, our method significantly outperforms all other methods, demonstrating its practical value. This is thanks to its robustness to geometric and topological noise which characterize raw 3D acquisitions. Fig. 1 (top row) visualizes an example result of a raw 3D data acquisition in everyday clothing. The target is color coded and the colors are transferred using the correspondences computed by the different methods. Our method is correct both globally and locally, while competing methods tend to fail in this case. Our method is designed for complete shape matching, thus we only report results on shape acquisitions that are complete. Our method fails locally on partial shapes.

To compare to STS, we run our method on their training and test sets of FAUST [9], with their evaluation protocol considering point-to-point accuracy<sup>3</sup> and MGE<sup>4</sup>. While STS reports 50.5% and 9.5, our method’s results are 22.2% and 1.6 for point-to-point accuracy and MGE, respectively. Hence, our method outperforms STS.

<sup>3</sup>This metric measures the proportion of points for which the geodesic error in Eq. 6 is equal to zero.

<sup>4</sup>In this case the geodesic error is normalised by the geodesic diameter rather than the mesh surface.

## 4.2 Evaluation on animal data

**Data** We consider pre-processed animal models sampled from the SMAL animal model [56]. We consider 49 animal meshes re-meshed independently to about 5k vertices. We use 32 meshes for training and 17 (136 pairs) for testing.

**Comparison to state-of-the-art** Tab. 2 shows the results for Deep Shells, Neuromorph, AttentiveFMaps and our method. Our method’s performance is better than Deep Shells and Neuromorph, and slightly lower than AttentiveFMaps. Fig. 1 (bottom row) visualizes an example result. Our method is visually on par with AttentiveFMaps. This shows that our method successfully applies to data beyond human models.

Method	Mean Geodesic Error	Cycle Geodesic Error
Deep Shells	21.40	0.1984
Neuromorph	12.34	0.1695
AttentiveFMaps	<b>4.40</b>	<u>0.0601</u>
Ours	<u>5.23</u>	<b>0.0562</b>

Table 2: Comparison on animal shapes w.r.t. MGE (Eq. 6) on left and CycleGE (Eq. 7) on right. Best score in bold, second-best underlined.

## 5 Conclusions

We have presented a data-driven unsupervised approach to solve for non-rigid shape matching. Our approach uses an association network that embeds the shapes in a feature space where hierarchical maps are extracted. It then constrains these maps by extracting the induced alignment in 3D using a deformation network that fits a piece-wise near-rigid deformation model. Our method retains the robustness of spatial methods while enforcing global geometric constraints on the associations, as with spectral methods. We demonstrate experimentally that our approach performs on-par with state-of-the-art for pre-processed data and significantly outperforms existing methods when applied directly to the raw output of multi-view 3D reconstructions.

## 6 Acknowledgements

We thank Abdelmoutaleb Dakri for providing us with SMPL fittings for our experiments. This work was funded by the ANR project Human4D (ANR-19-CE23-0020).

## References

- [1] Matthieu Armando, Laurence Boissieux, Edmond Boyer, Jean-Sébastien Franco, Martin Humenberger, Christophe Legras, Vincent Leroy, Mathieu Marsot, Julien Pansiot, Sergi Pujades, et al. 4dhumanoutfit: a multi-subject 4d dataset of human motion sequences in varying outfits exhibiting large displacements. *arXiv preprint arXiv:2306.07399*, 2023.
- [2] Jean Basset, Adnane Boukhayma, Stefanie Wuhrer, Franck Multon, and Edmond Boyer. Neural human deformation transfer. In *2021 International Conference on 3D Vision (3DV)*, pages 545–554. IEEE, 2021.

- [3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Conference on Computer Vision and Pattern Recognition*, 2014.
- [4] Francesco Bonarrigo, Alberto Signoroni, and Mario Botsch. Deformable registration using patch-wise shape matching. *Graphical Models*, 76(5):554–565, 2014.
- [5] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Free-form mesh tracking: a patch-based approach. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1339–1346. IEEE, 2010.
- [6] Van-Toan Cao, Trung-Thien Tran, and Denis Laurendeau. A two-stage approach to align two surfaces of deformable objects. *Graphical Models*, 82:13–28, 2015.
- [7] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, volume 41, pages 559–589. Wiley Online Library, 2022.
- [8] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *Advances in Neural Information Processing Systems*, 32, 2019.
- [9] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2022.
- [10] Omri Efroni, Dvir Ginzburg, and Dan Raviv. Spectral teacher for a spatial student: Spectrum-aware real-time dense shape correspondence. In *2022 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2022.
- [11] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [12] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural information processing systems*, 33:10491–10502, 2020.
- [13] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 869–877, 2018.
- [14] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 36–52. Springer, 2020.
- [15] 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.

- [16] 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.
- [17] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.
- [18] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006.
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [20] Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. Dpc: Unsupervised deep point correspondence via cross and self construction. In *2021 International Conference on 3D Vision (3DV)*, pages 1442–1451. IEEE, 2021.
- [21] Lei Li, Nicolas Donati, and Maks Ovsjanikov. Learning multi-resolution functional maps with spectral attention for robust shape matching. In *Advances in Neural Information Processing Systems*, 2022.
- [22] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017.
- [23] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.
- [24] Mathieu Marsot, Stefanie Wuhrer, Jean-Sébastien Franco, and Stephane Durocher. A structured latent space for human body motion generation. In *Conference on 3D Vision*, 2022.
- [25] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.
- [26] Vincent Nivoliers, Bruno Lévy, and Christophe Geuzaine. Anisotropic and feature sensitive triangular remeshing using normal lifting. *Journal of Computational and Applied Mathematics*, 289:225–240, 2015.
- [27] 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)*, 2012.
- [28] Gabriel Peyré and Laurent D Cohen. Geodesic remeshing using front propagation. *International Journal of Computer Vision*, 69:145–156, 2006.

- [29] Jean-Michel Roufousse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1617–1627, 2019.
- [30] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. *Advances in Neural Information Processing Systems*, 33:19264–19275, 2020.
- [31] Ivan Sipiran and Benjamin Bustos. A fully hierarchical approach for finding correspondences in non-rigid shapes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 817–824, 2013.
- [32] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2598–2606, 2018.
- [33] Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. Isotropic remeshing with fast and exact computation of restricted voronoi diagram. In *Computer graphics forum*, volume 28, pages 1445–1454. Wiley Online Library, 2009.
- [34] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6052–6061, 2021.
- [35] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [36] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017.

This supplementary material provides implementation details of our method in Appendix A, more details on the datasets used in Appendix B, ablation studies in Appendix C, and additional quantitative and qualitative results in Appendix D.

## A Implementation Details

### A.1 Patch Extraction

To compute multi-resolution surface patches, we consider a greedy approach based on the furthest point sampling strategy inspired by [28]. Starting from a randomly selected vertex  $x_1$ , we compute the geodesic distance map  $U_{x_1}$  to all other vertices on the mesh. Then, given that we have a set of vertices  $S_n = \{x_1, \dots, x_n\}$ , and their distance map  $U_n$ , we select the new vertex  $x_{n+1}$  to be the furthest vertex from  $S_n$ . We compute  $U_{x_{n+1}}$ , the distance map from  $x_{n+1}$  and update  $U_{n+1} = \min(U_n, U_{x_{n+1}})$  and add  $x_{n+1}$  to  $S_n$  to get  $S_{n+1}$ . We stop the algorithm when a target number of points is reached. To get multiple patch resolutions we stop the algorithm at increasing numbers of target points.

We select  $L + 1$  patch resolutions where patches at level 0 are restricted to vertices and level  $L$  represents the coarsest patch level. For each hierarchical level, the selected samples are used as patch centers  $\mathcal{C}_l = (c_i^l \in \mathbb{R}^3)_{1 \leq i \leq n_l}$  and their corresponding Voronoi cells on the mesh as patches  $(P_i^l)_{1 \leq i \leq n_l}$  for  $l = 0, \dots, L$ .

In all of our experiments, we extract 4 patch resolutions : all the mesh vertices, 800, 200 and 50 patches. Figure 3 shows an example of these surface patches.



Figure 3: Example of surface patches. From left to right : 800, 200 and 50 patches along with their centers on a mesh with  $\approx 13k$  vertices.

### A.2 Architecture Detail

**Pooling and Unpooling** The extracted surface patches are not strictly hierarchical in the sense that vertices of a patch at level  $l$  can belong to multiple patches at coarser levels  $l + 1$ . We use these surface patches for pooling and unpooling operations in our architectures.

To pool features from patches  $(P_i^l)_{1 \leq i \leq n_l}$  of level  $l$  to coarser patches  $(P_i^{l+1})_{1 \leq i \leq n_{l+1}}$  of level  $l + 1$  we proceed in two step:



1. We first unpool to the vertex level (level 0) such that each vertex is associated with the features of the patch it belongs to in level  $l$ .
2. We then employ max-pooling to go to patch level  $l + 1$ .

To unpool features from patches  $(P_i^{l+1})_{1 \leq i \leq n_{l+1}}$  of level  $l + 1$  to finer patches  $(P_i^l)_{1 \leq i \leq n_l}$  of level  $l$  we proceed similarly. We first unpool to the vertex level (level 0) such that each vertex is associated with the features of the patch it belongs to in level  $l + 1$ . We then employ max-pooling to go to patch level  $l$ .

**Feature Extractor** In both the association and the deformation networks, we use identical feature extractors based on hierarchical graph convolutional network FeaStConv operators [32]. In all cases, we fixed the number of attention heads of this operator to 9. Figure 4 illustrates the feature extractor architecture.

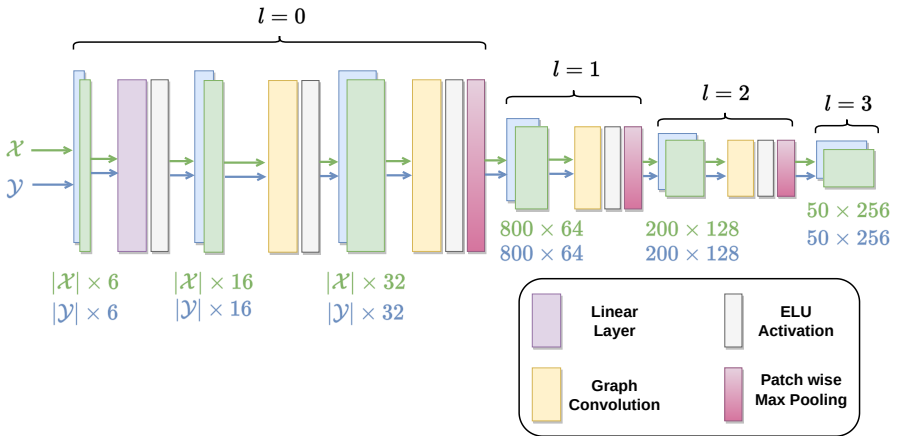


Figure 4: Feature extractor architecture. Inputs  $\mathcal{X}$  and  $\mathcal{Y}$  are decomposed into a hierarchy of 800, 200 and 50 patches and fine-to-coarse features are extracted. Input features at the vertex level are 3D coordinates and normals.

**Association Network** Figure 5 details the architecture of the association network. We fix the temperature parameter of the softmax operator to  $10^{-2}$ .

**Deformation Network** Figure 6 details the architecture of the deformation network. It uses a deformation decoder module that outputs per-patch rotation and translation parameters detailed in Figure 7.

### A.3 Training and Inference

**Loss Weights** The final loss per level  $l$  is the weighted combination of five self-supervised criteria, as given in Equation 5 of the main paper. Table 3 shows the weights used to train the network. The geodesic criterion was not used at the vertex level (weight fixed to 0) as it involves matrix multiplications that are computationally prohibitive. The matching

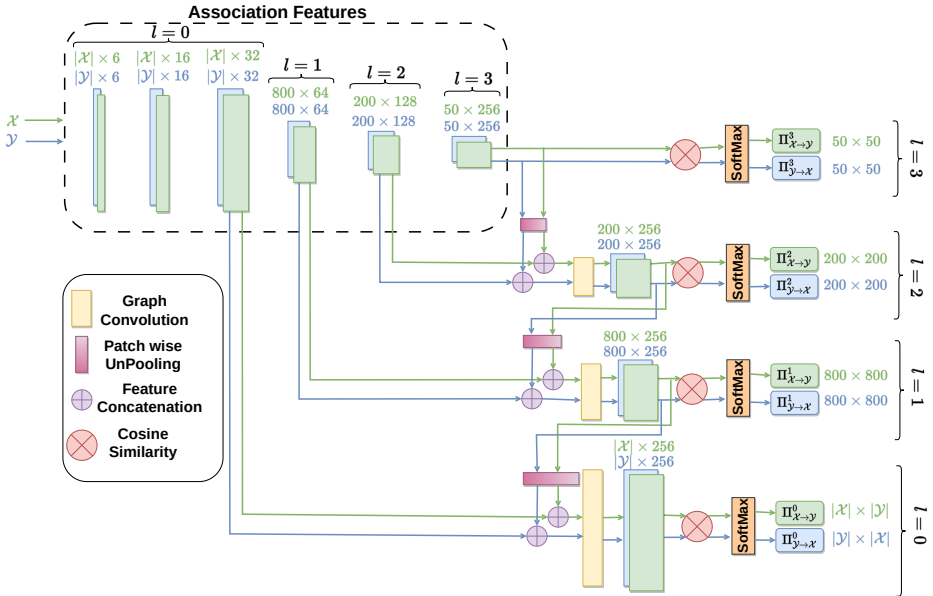


Figure 5: Architecture of the association network. Given meshes  $\mathcal{X}$  and  $\mathcal{Y}$  it outputs coarse-to-fine association maps at every level of the patch hierarchy, i.e.  $\Pi^l_{\mathcal{X} \rightarrow \mathcal{Y}}$  and  $\Pi^l_{\mathcal{Y} \rightarrow \mathcal{X}}$  for every level  $l$ .

Weight \ Level	$l = 0$	$l = 1$	$l = 2$	$l = 3$
$\lambda_g^l$	0.0	1.0	1.0	1.0
$\lambda_c^l$	10.0	10.0	10.0	10.0
$\lambda_r^l$	10.0	10.0	10.0	10.0
$\lambda_m^l$	2.0	2.0	2.0	2.0
$\lambda_{ri}^l$	200.0	100.0	50.0	10.0

Table 3: Loss weights for every level.

loss and the rigidity loss are somehow in opposition: The rigidity loss promotes isometric deformations whereas the matching loss promotes deformations that reflect the association matrix. Their weights are fixed so as to satisfy the association matrices while preserving the spatial continuity at the patch borders.

**Training** Our network is trained using Adam [19] with gradient clipping. The learning rate is fixed to  $1 \times 10^{-3}$  for the first epoch,  $5 \times 10^{-4}$  between the 2<sup>nd</sup> and 10<sup>th</sup> epochs and  $2.5 \times 10^{-4}$  after the 10<sup>th</sup> epoch. Our model takes 372 epochs to train on the Extended FAUST training set. We select the model that achieves the smallest loss on a validation set.

**Inference** At test time we allow the network to specialise for each new shape pair to improve the matching. This is achieved by resuming the training of the selected model on a training set restricted to the two input shapes, with the same fixed architecture, optimization technique and hyper-parameters. We refer to this specialisation as fine tuning in the main

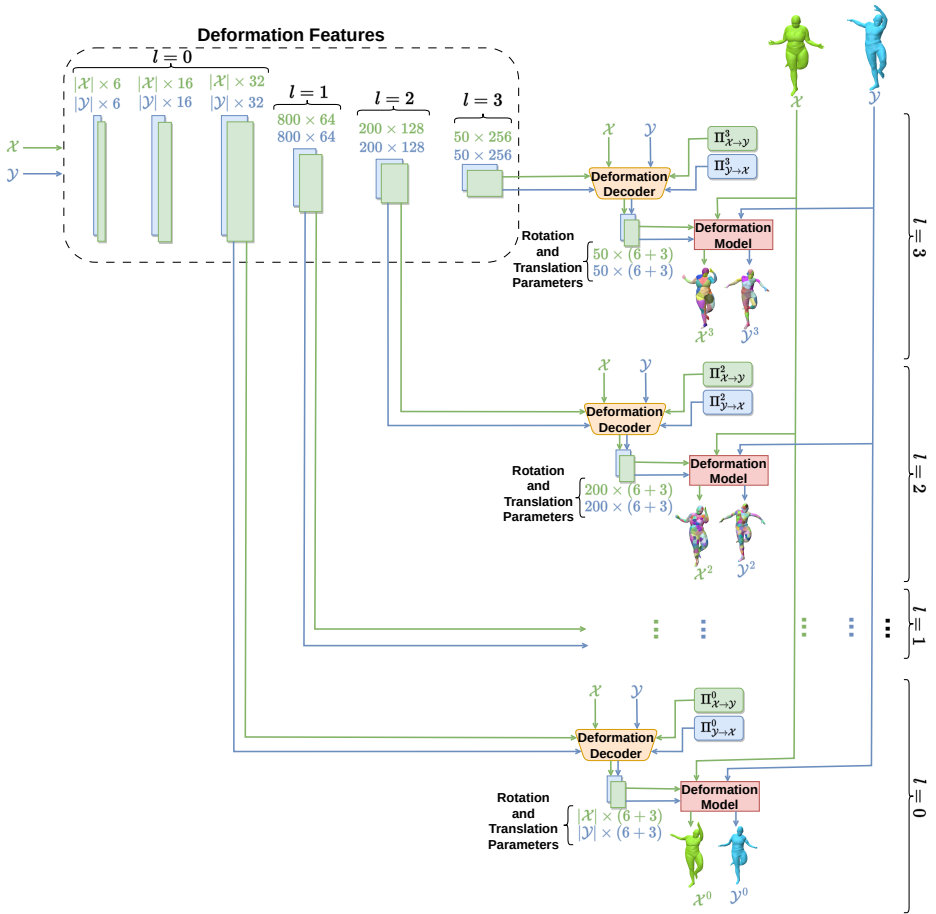


Figure 6: Architecture of the deformation network. Given  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $\Pi_{\mathcal{X} \rightarrow \mathcal{Y}}^l$ , and  $\Pi_{\mathcal{Y} \rightarrow \mathcal{X}}^l$  it estimates the 3D deformations  $\mathcal{X}^l$  and  $\mathcal{Y}^l$  of  $\mathcal{X}$  and  $\mathcal{Y}$  for every level  $l$ .

paper. In all our experiments we fix the number of epochs for fine tuning to 50.

## B Data

We detail below how ground truth correspondences were obtained to evaluate our approach.

**Pre-processed Data** For our experiments on pre-processed data, we used the extended FAUST dataset [1] which is composed of meshes with the same template connectivity. In order to remove the connectivity consistency, which can strongly bias the matching, we remeshed all shapes individually and created the 3 test sets mentioned in the main paper (sec 4.1). To get the ground truth correspondences for vertices on the remeshed shapes, we revert to the connectivity consistent meshes by searching, for each vertex on a remeshed shape,

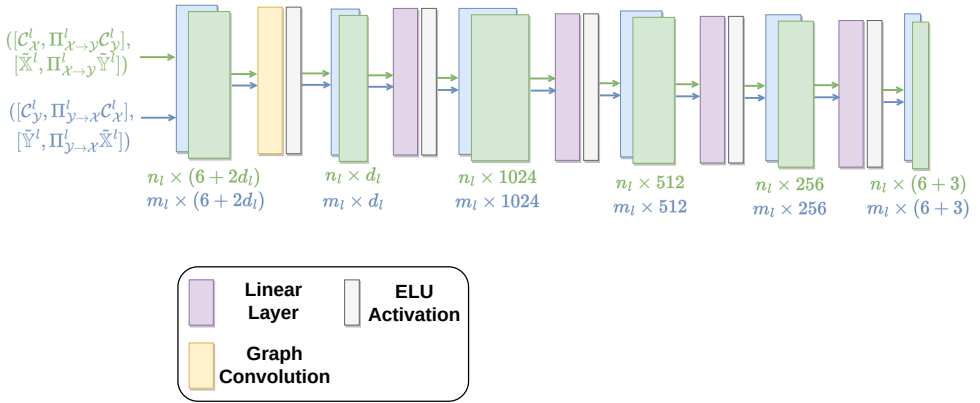


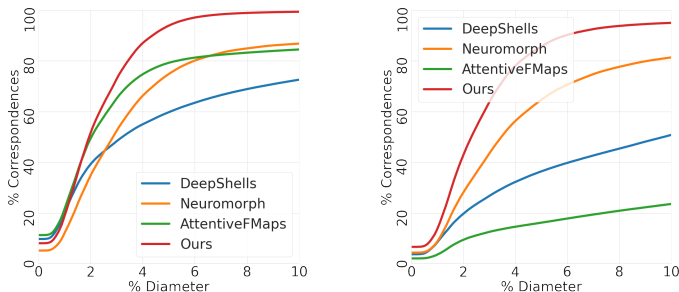
Figure 7: Architecture of the deformation decoder. It is composed of a graph convolution followed by an MLP.

the triangle on the original mesh, with minimal distance along the vertex normal direction. As a result of the connectivity changes, such distance can be large and we discard in the evaluation vertices for which this distance is higher than  $2/10$  of the remeshed shape’s mean edge length. This corresponds to 4.87% of the vertices for uniform remeshings to  $5k$  vertices, 1.2% for uniform remeshing to  $15k$  vertices and 1.01% for curvature adapted remeshings to  $5k$  vertices. A similar strategy is used for meshes altered with topological noise. In this case, 7.69% of the vertices are discarded in the evaluation.

**Raw 3D Acquisition** We also experimented our matching approach with raw 3D scans, e.g. [24]. In this case, ground truth matchings were obtained by fitting the SMPL model [23] to the scans and considering closest vertices on the template in the normal direction. Again here, vertices with distances to the template higher than 2 times the scan’s mean edge length are discarded in the evaluation. This corresponds in practice to 8.04% for the naked raw acquisition dataset and 17.42% for the clothed one.

## C Ablation Studies

We present ablation studies that evaluate the respective benefits of the main components of our approach, i.e. the hierarchical modeling and the deformation model constraint. The impact of fine tuning is also given. To assess the hierarchical modeling, we use only two levels of hierarchy: the mesh vertices and 800 patches. To assess the deformation model constraint, the network is restricted to the association network, losses involving the deformations, i.e. matching and rigidity loss, are discarded. We trained the models on extended FAUST and tested on both extended FAUST and the raw 3D acquisition data with everyday clothing. Tab. 4 shows the results and the number of epochs required to train each model. The complete model achieves the best results on both pre-processed and raw acquisition data. Note that the dimensionality reduction using the hierarchical modeling of associations is essential to the unsupervised learning and allows for much faster training. Note also that the



(a) Naked raw acquisition data. (b) Clothed raw acquisition data.

Figure 8: Comparison to state-of-the-art on raw acquisition data. Percentage of correct correspondences within a certain geodesic error tolerance radius.

deformation model improves the quality of the matching and that the fine tuning improves the results across all models.

Hierarchical Feature Space	Deformation Model	Fine Tuning	Number of Epochs to Train	Pre-processed data		Raw 3D acquisitions	
				Cycle Geodesic Error	Mean Geodesic Error	Cycle Geodesic Error	Mean Geodesic Error
✓	✓	✓	372	<b>0.0259</b>	<b>4.09</b>	<b>0.0463</b>	<b>6.02</b>
✓	✓	✗	372	0.0352	4.99	0.0980	9.13
✗	✓	✓	803	0.0653	14.01	0.0556	9.41
✗	✓	✗	803	0.0855	14.49	0.0890	11.26
✓	✗	✓	311	0.0260	4.24	0.0519	6.44
✓	✗	✗	311	0.0336	4.88	0.1247	10.77

Table 4: Ablation tests for the hierarchy in feature space, the deformation model and the fine tuning.

To prove the added benefit of the Self-Reconstruction Criterion that ensures that each patch, on the shape itself, is identified to avoid many-to-one matches, we present an ablation where we both train and test the models on extended FAUST. Tab. 5 shows the results.

Self Reconstruction Criterion	Fine Tuning	Cycle Geodesic Error	Mean Geodesic Error
✓	✗	<b>0.0352</b>	<b>4.99</b>
✗	✗	0.0390	6.90

Table 5: Ablation test of the Self-Reconstruction Criterion.

## D Additional Results

### D.1 Additional Quantitative Evaluation

Fig. 8 shows cumulative error plots giving the percentage of correct correspondences within a certain tolerance radius of geodesic error for the raw acquisitions in the naked regime in (a) and with everyday clothing in (b). On naked raw acquisitions, our method achieves 99% of exact matches when we tolerate errors smaller than 8.1% of the geodesic diameter, where the second best i.e. Neuromorph achieves 85.1%. On clothed raw acquisitions, which is the

most challenging test dataset, our method is more accurate both when considering details (i.e. small errors) and global alignments (i.e. large errors).

## D.2 Additional Qualitative Comparisons

Fig. 9 shows qualitative comparisons on pre-processed meshes in the first row, pre-processed meshes with topological noise in the second row (the left heel is glued to the right calf and the left arm is glued to the head on the target shape), naked raw 3D scans in the third row and clothed raw 3D scans in the fourth row. The target is color coded and the colors are transferred using the correspondences as estimated by the different methods. Ours is both locally and globally accurate in all four cases and is robust to the presence of hairs, clothes and severe topological noises (see the last row). DeepShells makes global alignment errors (e.g. the arms are flipped in the first row, the belly area also in the other rows). Neuromorph suffers from local distortions and makes local errors (e.g. see the right hand in all rows). AttentiveFMaps is globally and locally accurate on the pre-processed meshes in the first row but fails in the presence of topological noise which is ubiquitous in raw scans (e.g. see the full body in the second and fourth rows and the left arm area in the third row).

## D.3 Qualitative Deformation Results

Fig. 10 shows an example of deformed shapes output by our network at every hierarchical level. The top row shows the deformation in the  $\mathcal{X} \rightarrow \mathcal{Y}$  direction and the bottom row shows the deformation in the  $\mathcal{Y} \rightarrow \mathcal{X}$  direction. The deformed shapes at the finest level, i.e. the vertex level are close to the target deformation shapes ( $\mathcal{X}^0 \approx \mathcal{Y}$  and  $\mathcal{Y}^0 \approx \mathcal{X}$ ). In coarser levels, the deformation approximates the pose (global alignment) while in finer levels, where the rigidity constraint is weaker, it approximates the body shape (local alignment). This deformation is the induced alignment in 3D that guides the matching output of our method as shown in the top row of Fig. 9.



Figure 9: Comparisons with Deep Shells [12], Neuromorph [13] and AttentiveFMaps [14] on pre-processed human meshes (first and second rows) and on raw human 3D scans (third and fourth rows). Each point on the target mesh (left) is assigned a color, which is transferred to the source mesh (right) using the correspondences estimated by the different methods.



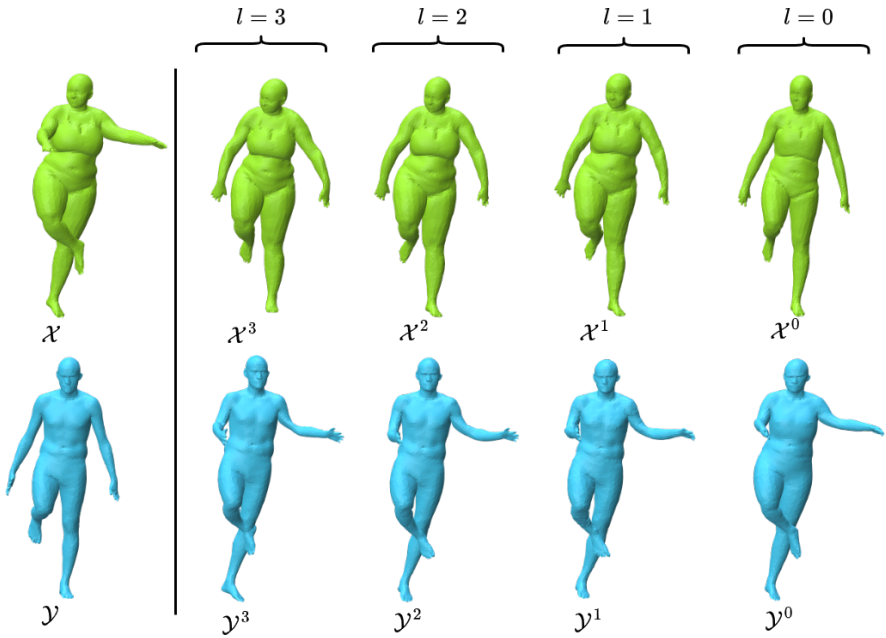


Figure 10: Deformation examples output by our network for every level in the hierarchy. The top row shows the deformation results in the  $\mathcal{X} \rightarrow \mathcal{Y}$  direction and the bottom row shows the deformation results in the  $\mathcal{Y} \rightarrow \mathcal{X}$  direction.