NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go

Marvin Eisenberger*,†, David Novotny*, Gael Kerchenbaum*, Patrick Labatut*,
Natalia Neverova*, Daniel Cremers†, Andrea Vedaldi*
Facebook AI Research*, Technical University of Munich†

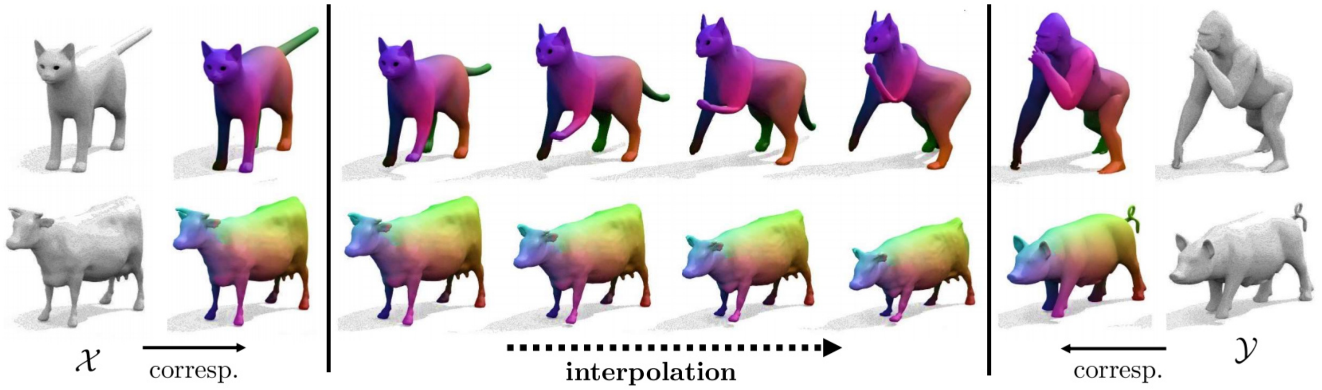

Figure 1: Our **NeuroMorph** neural network takes as input two meshes (left and right) and produces in one go (i.e. in a single feed-forward pass) a continuous **interpolation** and point-to-point **correspondence** between them (color coded). The interpolation, expressed as a displacement field, changes the pose of the source shape while preserving its identity.

Abstract

We present NeuroMorph, a new neural network architecture that takes as input two 3D shapes and produces in one go, i.e. in a single feed forward pass, a smooth interpolation and point-to-point correspondences between them. The interpolation, expressed as a deformation field, changes the pose of the source shape to resemble the target, but leaves the object identity unchanged. NeuroMorph uses an elegant architecture combining graph convolutions with global feature pooling to extract local features. During training, the model is incentivized to create realistic deformations by approximating geodesics on the underlying shape space manifold. This strong geometric prior allows to train our model end-to-end and in a fully unsupervised manner without requiring any manual correspondence annotations. NeuroMorph works well for a large variety of input shapes, including non-isometric pairs from different object categories. It obtains state-of-the-art results for both shape correspondence and interpolation tasks, matching or surpassing the performance of recent unsupervised and supervised methods on multiple benchmarks.

1. Introduction

The ability to relate the 3D shapes of objects is of key importance to fully understand object categories. Objects can change their shape due to articulation, other motions and intra-category variations, but such changes are not arbitrary. Instead, they are strongly constrained by the category of the objects at hand. Seminal works such as [33] express such constraints by learning statistical shape models. In order to do so, they need to put in correspondence large collections of individual 3D scans, which they do by exploiting the fact that individual objects deform continuously in time, and by using some manual inputs to align different object instances. Due to the high complexity of obtaining and preprocessing such 3D data, however, these models remain rare and mostly limited to selected categories such as humans that are of sufficient importance in applications. In this paper, we are thus interested in developing a method that can learn to relate different 3D shapes fully automatically, interpolating a small number of 3D reconstructions, and in a manner which is less specific to a single category (Figure 1).

Due to the complexity of this task, authors have often considered certain sub-problems in isolation. One is to establish *point-to-point correspondences* between shapes, telling which points are either physically identical (for a given articulated object) or at least analogous (for similar objects). A second important sub-problem is *interpolation*, which amounts to continuously deforming a source shape into a target shape. Interpolation must produce a collection of intermediate shapes that are meaningful in their own right, in the sense of being plausible samples from the underlying shape distribution. The interpolation *trajectory* must be also meaningful; for instance, if the deformation between two shapes can be explained by the articulation of an underlying physical object, this solution is preferred.

The correspondence and interpolation problems have been addressed before extensively, by using tools from geometry and, more recently, machine learning. Most of the existing algorithms, however, require at least some manual supervision, for example in the form of an initial set of sparse shape correspondences. Furthermore, correspondence and interpolation are rarely addressed together due to their complexity.

In this paper, we advocate instead for an approach in which the correspondence and interpolation problems are solved *simultaneously*, and in an *unsupervised* manner. To do this, we introduce *NeuroMorph*, a new neural network that solves the two problems in a single feed forward pass. We show that, rather than making learning more difficult, integrating two goals reinforces them, making it possible to obtain excellent empirical results. Most importantly, we show that NeuroMorph can be learned in a *fully unsupervised* manner, given only a collection of 3D shapes as input and certain geometric priors for regularization.

NeuroMorph advances the state of the art in shape matching and interpolation, surpassing by a large margin prior unsupervised methods and often matching the quality of supervised ones. We show that NeuroMorph can establish high-quality point-to-point correspondences without any manual supervision even for difficult cases in which shapes are related by substantial non-isometric deformations (such as between two different types of animals, like a cat and a gorilla, as in Figure 1) which have challenged prior approaches. Furthermore, we also show that NeuroMorph can interpolate effectively between different shapes, acting on the *pose* of a shape while leaving its *identity* largely unchanged. To demonstrate the quality of the interpolation, we use it for *data augmentation*, extending a given dataset of 3D shapes with intermediate ones. Augmenting a dataset in this manner is useful when, as it is often the case, 3D training data is scarce. We show the benefits of this form of data augmentation to supervise other tasks, such as reconstructing continuos surfaces from sparse point clouds.

Our new formulation also gives rise to some interesting applications: Since our method learns a function that produces correspondence and interpolation in a single feed forward pass, it can be used not only to align different shapes, but also for pose transfer, digital puppeteering and other visual effects.

2. Related work

To the best of our knowledge, we are the first to consider the problem of learning a mapping that, given a pair of shapes as input, predicts in a feed-forward manner their correspondences and interpolation. This should be contrasted to other recent approaches to shape understanding such as LIMP [7] that try to learn a shape space. These architectures need to solve the difficult problem of *generating* or *auto-encoding* 3D shapes. Unfortunately, designing good generator networks for 3D shapes remains a challenging problem. In particular, it is difficult for these networks to generalize beyond the particular family of shapes (e.g. humans) experienced during training. By contrast, we do not try to generate shapes outright, but only to *relate* pairs of given input shapes. This replaces the difficult task of shape generation with the easier task of generating a deformation field, working well with a large variety of different shapes.

The rest of the section discusses other relevant work.

Shape correspondences. The problem of establishing correspondences between 3D shapes has been studied extensively (see the recent surveys [55, 52, 49]). Traditional approaches define axiomatic algorithms that focus on a certain subclass of problems like rigid transformations [63, 66], nearly-isometric deformations [39, 1, 57, 44], bounded distortion [35, 13] or partiality [32, 45, 31]. Methods such as functional maps [39] reduce matching to a spectral analysis of 3D shapes.

More recent approaches use machine learning and are often based on developing deep neural networks for non-image data such as point clouds, graphs and geometric surfaces [5]. Charting-based methods define learnable intrinsic patch operators for local feature aggregation [34, 3, 37, 41, 48]. Deep functional maps [30] aim at combining a learnable local feature extractor with a differentiable matching layer based on the axiomatic functional maps framework [39]. Subsequent works [20, 47] extended this idea to the unsupervised setting and combined it with learnable point cloud feature extractors [9, 50]. Moreover, [14] recently proposed to replace the functional maps layer with a multi-scale correspondence refinement layer based on optimal transport. Another related approach is [18] which uses a PointNet [42] encoder to align a human template to point cloud observations to compute correspondences between different human shapes.

Feature extractors for 3D shapes. Several authors have proposed to reduce matching 3D shapes to matching local shape descriptors. A common remedy is learning to refine hand-crafted descriptors such as SHOT [54], e.g. with metric learning [30, 20, 47]. In practice, this approach is highly dependent on the quality of the input features and tends to be unstable due to the noise and the complex variable structures of real 3D data. More recently, authors have thus looked at *learning* such descriptors directly [9, 50] with point cloud feature extractors [53, 43]. Another possibility is to interpret a 3D mesh as a graph and use graph convolutional neural networks [29, 8]. The challenge here is that the specific graph used to represent a 3D shape is partially arbitrary (because we can triangulate a surface in many different ways), and graph convolutions must discount geometrically-irrelevant changes (this is often done empirically by re-meshing as a form of data augmentation).

Shape spaces, manifolds and interpolation. 3D shapes can be interpreted as low-dimensional manifolds in a high-dimensional embedding space [27, 62, 23, 22]. The low-dimensional manifold can, for example, capture the admissible pose of an articulated object [65, 58, 24]. Given a shape manifold, interpolation can then be elegantly formulated as finding geodesic paths between two shapes. However, building shape manifolds may be difficult in practice, especially if the input shapes are not in perfect correspondence. Therefore, also inspired by LIMP [7], for training NeuroMorph we follow approaches such as [12, 11] that avoid building a shape manifold explicitly and instead directly construct geodesic paths that originate at the source shapes and terminate in the vicinity of the target shapes.

Generative shape models. While manifolds provide a geometric characterization of a shape space, generative models provide a statistical one. One particular challenge in this context is designing shape-decoder architectures that can generate 3D surfaces from a latent shape representation. A straightforward solution is predicting occupancy probabilities on a 3D voxel grid [6], but the cost of dense, volumetric representations limits the resolution. Other approaches decode point clouds [15, 64] or 3D meshes [19, 16] directly. A recent trend is encoding an implicit representation of a 3D surface in a neural network [36, 40]. This allows for a compact shape representation and a decoder that can generate shapes of an arbitrary topology. Following the same methodology, [38] predicts a time-dependent displacement field that can be used to interpolate 3D shapes. This approach is related to ours, but it requires 4D supervision during training, whereas our method is trained on a sparse set of poses. ShapeFlow [26] predicts dense velocity fields for template-based reconstruction. Similarly, [60] computes an intrinsic displacement field to align a pair of input shapes, but they do not predict an intermediate sequence.

3. Method

Let $\mathcal{X}$ and $\mathcal{Y}$ be 3D shapes, respectively called the *source* and the *target*, expressed as triangular meshes with vertices $\mathbf{X} = (\mathbf{x}_i)_{1\leq i\leq n} \in \mathbb{R}^{n\times 3}$ and $\mathbf{Y} = (\mathbf{y}_j)_{1\leq j\leq m} \in \mathbb{R}^{m\times 3}$, respectively. Our goal is to learn a function

$$f \;:\; (\mathcal{X},\mathcal{Y}) \longmapsto (\Pi,\Delta),$$

that, given the two shapes as input, predicts 'in one go' a correspondence matrix Π and an interpolation flow Δ between them. The matrix $\Pi \in [0,1]^{n\times m}$ sends probabilistically the vertices $\mathbf{x}_i$ of the source mesh $\mathcal{X}$ to corresponding vertices $\mathbf{y}_j$ in the target mesh $\mathcal{Y}$ and is thus row-stochastic (i.e. $\Pi\mathbf{1} = \mathbf{1}$). The interpolating flow $\Delta(t) \in \mathbb{R}^{n\times 3}$, $t \in [0,1]$ shifts continuously the vertices of the source mesh, forming trajectories:

$$\mathbf{X}(t) := \mathbf{X} + \Delta(t), \tag{1}$$

that take them from their original locations $\mathbf{X}(0) = \mathbf{X}$ to new locations $\mathbf{X}(1) \approx \Pi\mathbf{Y}$ close to the corresponding vertices in the target mesh.

The function f is given by two deep neural networks. The first, discussed in Section 3.1, establishes the correspondence matrix Π and the second, discussed in Section 3.2, outputs the shifts $\Delta(t)$ for arbitrary values of $t \in [0,1]$. Both networks are trained end-to-end in an unsupervised manner, as described in Section 3.3.

3.1. Correspondences and vertex features

The correspondence matrix Π between meshes $\mathcal{X}$ and $\mathcal{Y}$ is obtained by extracting and then matching features of the mesh vertices. The features are computed by a deep neural network: $\tilde{\mathbf{X}} = \Phi(\mathcal{X}) \in \mathbb{R}^{n\times d}$ that takes the shape $\mathcal{X}$ as input and outputs a matrix $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_i)_{1\leq i\leq n}$ with a feature vector $\tilde{\mathbf{x}}_i$ for each vertex i of the mesh. Given analogous features $\tilde{\mathbf{Y}} = (\tilde{\mathbf{y}}_i)_{1\leq i\leq m} \in \mathbb{R}^{m\times d}$ for the target shape, the correspondence matrix is obtained by comparing features via the cosine similarity and normalizing the rows using the softmax operator:

$$\Pi_{ij} := \frac{\exp(\sigma s_{ij})}{\sum_{k=1}^{m}\exp(\sigma s_{ik})} \;\; \text{s.t.} \;\; s_{ij} := \frac{\langle\tilde{\mathbf{x}}_i,\tilde{\mathbf{y}}_j\rangle_2}{\|\tilde{\mathbf{x}}_i\|_2\|\tilde{\mathbf{y}}_j\|_2}, \tag{2}$$

with temperature $\sigma \in \mathbb{R}^+$. In this way, $\Pi\mathbf{1} = \mathbf{1}$ and Π can be interpreted as a soft assignment of source vertices $\mathbf{x}_i$ in $\mathcal{X}$ to target vertices $\mathbf{y}_j$ in $\mathcal{Y}$.

Feature extractor network. Next, we describe the neural network $\Phi(\mathcal{X})$ that extracts the feature vectors that appear in Equation (2) (this network is also illustrated in Figure 2). While different designs are conceivable, we propose here one based on successive local feature aggregation and global feature pooling. The architecture makes use of the mesh vertices $\mathbf{X}$ as well as the mesh topology. The

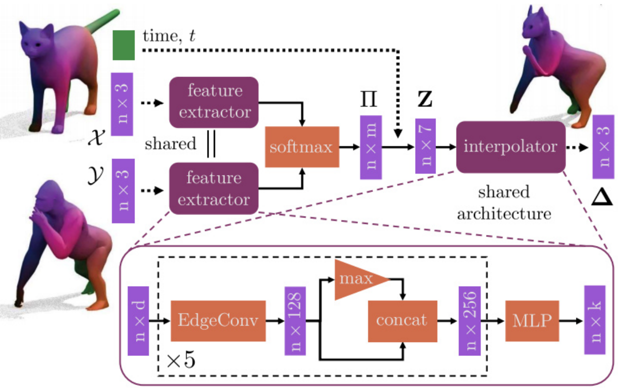

Figure 2: **NeuroMorph.** An overview of our model.

latter is specified by the neighborhood structure $\mathcal{E}$, where $(i,j) \in \mathcal{E} \subset \{1,\dots,n\}^2$ means that vertex $\mathbf{x}_j$ is connected to vertex $\mathbf{x}_i$ by a triangle edge. Thus, the mesh is fully specified by the pair $\mathcal{X} = (\mathbf{X},\, \mathcal{E})$.

The layers of the network Φ are given by EdgeConv [61] graph convolution operators implemented via residual subnetworks [21]. In more detail, each EdgeConv layer takes as input vertex features $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_i)_{1\leq i\leq n}$ on $\mathcal{X}$ and computes an improved set of features $\tilde{\mathbf{X}}' = (\tilde{\mathbf{x}}_i')_{1\leq i\leq n}$ via the expression:

$$\tilde{\mathbf{x}}_i' := \max_{j:(i,j)\in\mathcal{E}} h_\phi(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i). \tag{3}$$

Here, a small residual network h_ϕ is used to combine the feature $\tilde{\mathbf{x}}_i$ of the i-th vertex with the feature $\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$ of one of the edges adjacent to it. This is repeated for all edges incident on the i-th vertex and the results are aggregated via component-wise max-pooling over the mesh neighborhood $\big\{j : (i,j) \in \mathcal{E}\big\}$, resulting in an updated vertex feature $\tilde{\mathbf{x}}_i'$.

The EdgeConv layer can effectively learn the local geometric structures in the vicinity of a point. However, that alone is not sufficient to resolve dependencies in terms of the global geometry, since the message passing only allows for a local information flow. Therefore, we append a global feature vector to the point features after each EdgeConv refinement, by applying the max pooling operator globally:

$$\tilde{\mathbf{x}}_i'' := \big(\tilde{\mathbf{x}}_i',\, \max_{1\leq i\leq n} \tilde{\mathbf{x}}_i'\big). \tag{4}$$

The network Φ is given by a succession of these layers, forming a chain $\tilde{\mathbf{X}} \to \tilde{\mathbf{X}}' \to \tilde{\mathbf{X}}'' \to \dots$ alternating global (3) and local (4) update steps. The input features $\tilde{\mathbf{X}} = (\mathbf{X}, \mathbf{N})$ are given by the concatenation of the absolute position of the mesh vertices $\mathbf{X}$ with the outer normals at the vertices $\mathbf{N} = (\mathbf{n}_i)_{1\leq i\leq n}$ (the normal vectors $\mathbf{n}_i$ are computed by averaging over face normals adjacent to $\mathbf{x}_i$).

3.2. Interpolator

We are now ready to describe the interpolator component of our model. Recall that the goal is to predict a displacement operator Δ such that the trajectory $\mathbf{X}(t) = \mathbf{X} + \Delta(t)$ smoothly shifts the point of the first mesh to points in the second. Notice that $\Delta(t) \in \mathbb{R}^{n\times 3}$ is just a collection of 3D vectors associated to each mesh vertex, just like the vertex positions, normals and feature vectors in the previous section. Thus, we offload the calculation of the displacements to a similar convolutional neural network and write: $\Delta(t) = \Psi(\mathcal{X}, \mathcal{Y}, t)$. The difference is the input to the network Ψ, which is now given by the 7-dimensional feature vectors $\mathbf{Z} \in \mathbb{R}^{n\times 7}$:

$$\mathbf{Z} := \big(\mathbf{X},\, \Pi\mathbf{Y} - \mathbf{X},\, \mathbf{1}t\big). \tag{5}$$

These feature vectors consist of the vertices $\mathbf{X}$ of the source shape $\mathcal{X}$, the offset vectors $\Pi\mathbf{Y} - \mathbf{X}$ predicted by the correspondence module of Section 3.1[1], and the time variable t ('broadcast' to all vertices by multiplication with a vector of all ones). Just like the network Φ in Section 3.1, the network Ψ alternates global (3) and local (4) update steps to compute a sequence of updated features $\mathbf{Z} \to \mathbf{Z}' \to \mathbf{Z}'' \cdots \to \mathbf{V}$ terminating in a matrix $\mathbf{V} \in \mathbb{R}^{n\times 3}$. The final displacements are then given by a scaled version of $\mathbf{V}$, and are set to $\Delta(t) = t\mathbf{V}(t)$.

In this manner, the network can immediately obtain a trivial (degenerate) solution to the interpolation problem by setting $\mathbf{V}(t) = \Pi\mathbf{Y} - \mathbf{X}$, which amounts to copying verbatim part of the input features $\mathbf{Z}$. This result is a simple linear interpolation of the mesh vertices, trivially satisfying the boundary conditions of the interpolation:

$$\mathbf{X}(0) = \mathbf{X} + 0 \cdot \mathbf{V}(0) = \mathbf{X}, \tag{6}$$

$$\mathbf{X}(1) = \mathbf{X} + 1 \cdot (\Pi\mathbf{Y} - \mathbf{X}) = \Pi\mathbf{Y}. \tag{7}$$

Linear interpolation provides a sensible initialization, but is in itself a degenerate solution as we wish to obtain 'geometrically plausible' deformations of the mesh. To prevent the network from defaulting to this case, we thus need to incentivize geometrically meaningful deformations during training, which we do in the next section.

3.3. Learning

In this section, we show how we can train the model in an *unsupervised*[2] *manner*. That is, given only a collection of example meshes with no manual annotations, our method simultaneously learns to interpolate and establish

[1] Note that $\Delta(1) = \Pi\mathbf{Y} - \mathbf{X}$.

[2] In practice, the only assumption we make about the input objects is that they are in an approximately canonical rigid pose in terms of the up-down and front-back orientation. For most existing benchmarks this holds trivially without any further preprocessing. The recent paper by [50] calls this setup weakly supervised.

point-to-point correspondences between them. This sets it apart from prior work on shape interpolation which either require dense correspondences during training or, in the case of classical axiomatic interpolation methods, even at test time. Learning comprises three signals, encoded by three corresponding losses:

$$\ell := \lambda_{\text{reg}}\ell_{\text{reg}} + \lambda_{\text{arap}}\ell_{\text{arap}} + \lambda_{\text{geo}}\ell_{\text{geo}}. \tag{8}$$

The loss ℓ_{reg} ensures that correspondences and interpolation correctly map the source mesh on the target mesh, and the other two ensure that this is done in a geometrically meaningful way. The latter is done by constraining the trajectory $(\mathbf{X}(t))_{t\in[0,1]}$ generated by the model. Recall that the model can be queried for an arbitrary value $t \in [0,1]$, and it is thus able to produce interpolations that are truly continuous in time. During training, in order to compute our losses, we sample predictions $\mathbf{X}_0,\dots,\mathbf{X}_T$ for an equidistant set of discrete time steps $\mathbf{X}_k := \mathbf{X}(k/T)$ where $k = 0,\dots,T$.

Registration loss. Requirement Equation (6) holds trivially as $\Delta(0) = 0$ is built into our model definition (see Equation (1)). For Equation (7), we introduce the registration loss: $\ell_{\text{reg}}\big(\mathbf{X}_T,\mathbf{Y},\Pi\big) := \|\Pi\mathbf{Y} - \mathbf{X}_T\|_2^2$. Since our goal is to compute shape interpolations without any supervision, we use the soft correspondences Π estimated by our model instead of ground-truth annotations.

As-rigid-as-possible loss. In general, there are infinitely many conceivable paths between a pair of shapes. In order to restrict our method to plausible sequences, we regularize the path using the theory of shape spaces [27, 62, 23]. As we work with discrete time, we approximate the 'distance' between shapes in the shape space manifold by means of the local distortion metric between two consecutive states $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$. To that end, we choose the as-rigid-as-possible [51] metric:

$$E_{\text{arap}}\big(\mathbf{X}_k,\mathbf{X}_{k+1}\big) :=$$
$$\frac{1}{2}\min_{\substack{\mathbf{R}_i\in SO(3)\\ i=1,\dots,n}} \sum_{(i,j)\in\mathcal{E}} \big\|\mathbf{R}_i(\mathbf{X}_{k,j}-\mathbf{X}_{k,i}) - (\mathbf{X}_{k+1,j}-\mathbf{X}_{k+1,i})\big\|_2^2.$$

Intuitively, this functional rotates the local coordinate frame of each point in $\mathbf{X}_k$ to the corresponding deformed state $\mathbf{X}_{k+1}$ and penalizes deviations from locally rigid transformations. Moreover, the rotation matrices $\mathbf{R}_i$ can be computed in closed form which allows for an efficient optimization of E_{arap} (see [51] for more details). Finally, we can use this functional to construct the first component of our loss function for the whole sequence $(\mathbf{X}_k)_k$:

$$\ell_{\text{arap}}\big(\mathbf{X}_0,\dots,\mathbf{X}_T\big) :=$$
$$\sum_{k=0}^{T-1} E_{\text{arap}}(\mathbf{X}_k,\mathbf{X}_{k+1}) + E_{\text{arap}}(\mathbf{X}_{k+1},\mathbf{X}_k). \tag{9}$$

		err.	p.p.	w/o p.p.
Axiom.	BCICP [44]	6.4	—	—
	ZoomOut [35]	6.1	—	—
	Smooth Shells [13]	2.5	—	—
Sup.	3D-CODED [18]	2.5	—	—
	FMNet [30]	5.9	PMF	11
	GeoFMNet [9]	1.9	ZO	3.1
Unsup.	SurFMNet [47]	7.4	ICP	15
	Unsup. FMNet [20]	5.7	PMF	10
	Weakly sup. FMNet [50]	1.9	ZO	3.3
	Deep shells [14]	1.7	—	—
	NeuroMorph (Ours)	**1.5**	SL	**2.3**

Table 1: **Unsupervised correspondences on FAUST [2] remeshed.** Mean geodesic error in % of the diameter on the test set. For methods that use an axiomatic technique for refinement (PMF [57], ZO [35], ICP [39] or SL [13]), we also show the result without.

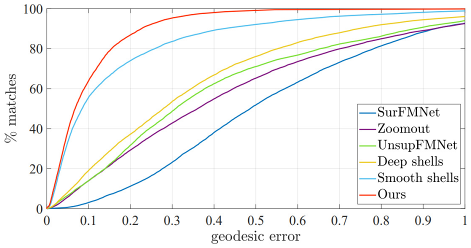

Figure 3: **Unsupervised correspondences on SHREC20 [10].** We only compare our method to other unsupervised methods here, since there are no dense ground-truth correspondences for this benchmark which is a requirement for most supervised approaches.

Geodesic distance preservation loss. The final component of our loss function in Equation (8) aims at preserving the pairwise geodesic distance matrices $\mathbf{D}_{\mathcal{X}}$ and $\mathbf{D}_{\mathcal{Y}}$ under the estimated mapping Π, this is given by $\ell_{\text{geo}}\big(\Pi\big) := \|\Pi\mathbf{D}_{\mathcal{Y}}\Pi^\top - \mathbf{D}_{\mathcal{X}}\|_2^2$. Note that this energy only regularizes the estimation of the correspondences Π as $\mathcal{X}$ and $\mathcal{Y}$ are the (fixed) source and target shapes.

Intuitively, this objective promotes correspondences Π with bounded geodesic distortion. Variants of this objective are commonly used in classical shape matching [4, 59, 57] and have been also successfully integrated in a learning pipeline [20] in combination with functional maps [39].

3.4. Implementation details

During training, we sample a pair of input shapes from our training set, predict an interpolation and a set of dense point-to-point correspondences and optimize the model parameters according to our composite loss (8). All hyper-

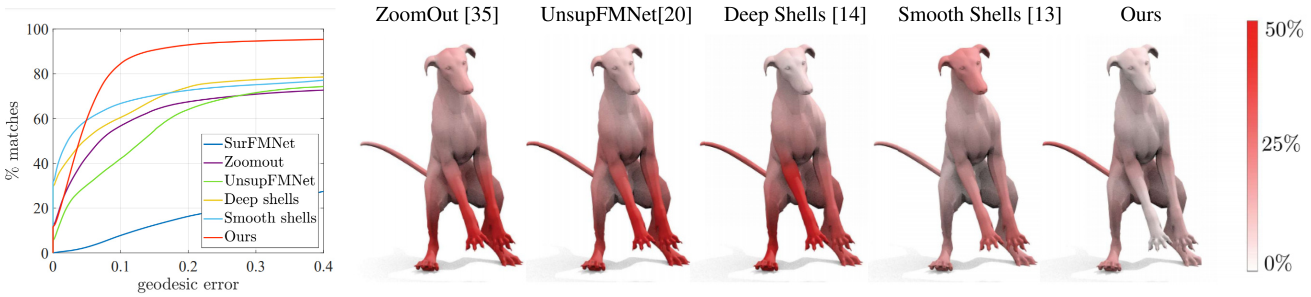

Figure 4: **Unsupervised correspondences on G-S-H.** We provide the cumulative geodesic error curves (in % of the diameter) of different approaches (left). For a detailed comparison, we display heat maps on one pose of the 'Galgo' shape from our dataset (right). We color code the mean geodesic error for each point of the surface, averaged over all 1024 pairs from the test set. Our method is particularly good at discovering structural correspondences, i.e. matching extremities correctly.

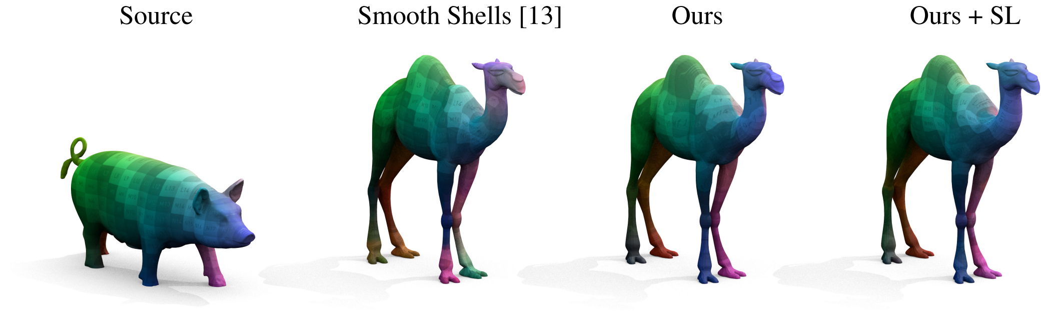

Figure 5: **Unsupervised correspondences on SHREC20.** A qualitative comparison on non-isometric pairs from SHREC20 [10]. While the correspondences predicted by our method are generally very accurate, the postprocessing still helps to remove local noise. The baseline [13] naturally produces smooth matches, but global parts of the geometry are sometimes mismatched (compare for instance the front legs and and head of the camel shown here).

parameters were selected on a validation set and the same configuration is used in all of our experiments.

Two parameters are varied during training: In the beginning, we set the number of discrete time steps to $T = 1$ and then increase it on a logarithmic scale. This multi-scale optimization strategy, which is motivated by classical non-learning interpolation algorithms [27, 23], leads to an overall faster and more robust convergence. The geodesic loss ℓ_{geo} initially helps to guide the optimization such that it converges to meaningful local minima. On the other hand, we found that it can actually be detrimental in the case of extremely non-isometric pairs (e.g. two different classes of animals). Therefore, we decay the weight $\lambda_{\text{geo}} = 0$ of this loss as a fine-tuning step during training after a fixed number of epochs.

As a form of data augmentation, we randomly subsample the triangulation of both input meshes separately and rotate the input pair along the azimuth axis in each iteration. This prevents our method from relying on pairs with compatible connectivity, since we ideally want our predictions to be independent from the discretization.

At test time, we simply query our model to obtain an interpolation of an input pair of shapes. The soft correspondences Π obtained with our method are generally very accurate, but the conversion to hard correspondences (i.e. point-to-point matches) via thresholding leads to a certain degree of local noise. To create more smooth correspondences, we additionally post-process our results with the multi-scale matching method smooth shells [13]. Post-processing is standard in unsupervised correspondence learning.

4. Experiments

We now evaluate the performance of NeuroMorph in terms of shape correspondence and interpolation (Sec. 4.1, 4.2), as well as for data augmentation in Sec. 4.3.

4.1. Shape correspondence

Datasets. We evaluate the matching accuracy of our method on two benchmarks. The first is FAUST [2], which contains 10 humans with 10 different poses each. We split it in a training and test set of 80 and 20 shapes respectively. Instead of the standard meshes, we use the more recent version of the benchmark [44] where each shape was re-meshed individually. This makes it challenging but also more realistic, since for real-world scans the sampling

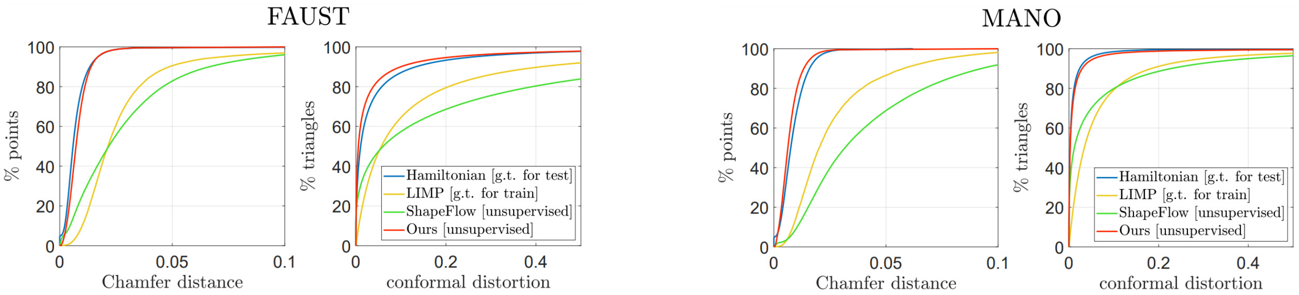

Figure 6: **Interpolation on FAUST [2] and MANO [46].** We show a quantitative comparison of interpolations obtained with Hamiltonian shape interpolation [11], LIMP [7], ShapeFlow [26] and our method. ShapeFlow [26] computes an extrinsic flow to interpolate a pair of objects in an unsupervised manner, but they do not model shape correspondences explicitly which is suboptimal for the large pose variations of deformable object categories. On both benchmarks, our method outperforms LIMP [7], despite the fact that the it uses g.t. correspondences for training. It is also on par with the axiomatic baseline Hamiltonian interpolation [11], which is remarkable since [11] requires dense correspondences even at test time.

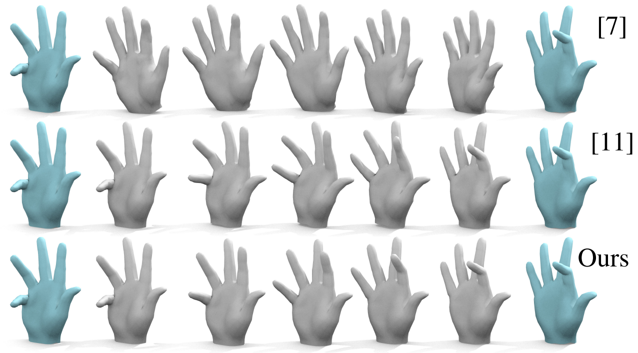

Figure 7: **Interpolation on MANO [46].** We show the interpolation sequence (gray) for a pair (blue) from the test set. LIMP [7], which requires ground-truth correspondences for training, explicitly reconstructs the geometry of intermediate shapes in a variational autoencoder architecture which limits the generalization to unseen poses. Hamiltonian shape interpolation [11] yields high quality results that are comparable to ours, but it is an axiomatic method that requires ground-truth correspondences at test time and multiple minutes of optimization per pair.

of surfaces is generally incompatible. The second benchmark we consider is the recent SHREC20 challenge [10] which focuses on non-isometric deformations. It contains 14 shapes of different animals, some of which are real scans with holes, topological changes and partial geometries. The ground truth for this dataset consists of sparse annotated keypoints which we use for evaluation. Since there are no dense annotated point-to-point correspondences, most existing supervised methods do not apply here. The final benchmark we show is G-S-H (Galgo, Sphynx, Human), for which we created our own dataset, see **??** for more details. It contains non-isometric pairs from three object categories (a dog 'Galgo', a cat 'Sphynx' and a human) with multiple challenging poses each, as well as dense ground truth matches.

Evaluation metrics. Following the Princeton benchmark protocol [28], the accuracy of a set of point-to-point correspondences is defined as the geodesic distance of the predicted and the ground-truth matches, normalized by the square root area of the mesh. For FAUST remeshed, we compute the distance for all points, whereas for SHREC20 this is done for all available sparse annotations.

Discussion. As shown in Table 1, Figure 3 and Figure 4, NeuroMorph obtains state-of-the-art results on FAUST remeshed, SHREC20 and our own benchmark G-S-H, respectively. The overall suboptimal performance of existing methods on the latter two benchmarks can be attributed to the fact that, expect for [13], most of them implicitly assume near-isometry or at least compatible local features. This, however, does not hold for most examples in SHREC20 (see Figure 5 for a qualitative comparison) and G-S-H (for instance, on SHREC20, NeuroMorph matches 92% of the vertices within 0.25 geodesic error vs. 79% of the second best, smooth shells). NeuroMorph is particularly good at discovering structural correspondences, which can then be further refined in post-processing.

4.2. Shape interpolation

Datasets. For shape interpolation, we report results on the FAUST [2] (see Section 4.1) and MANO [46] datasets. The latter consists of synthetic hands in various poses — we use 100 shapes for training and 20 different samples for testing.

Evaluation metrics. We use two metrics to quantify the precision of an interpolation. The conformal distortion metric signifies how much individual triangles of a mesh distort throughout an interpolation sequence, in comparison to the reference pose $\mathcal{X}$, see [25, Eq. (3)] for a definition. Less

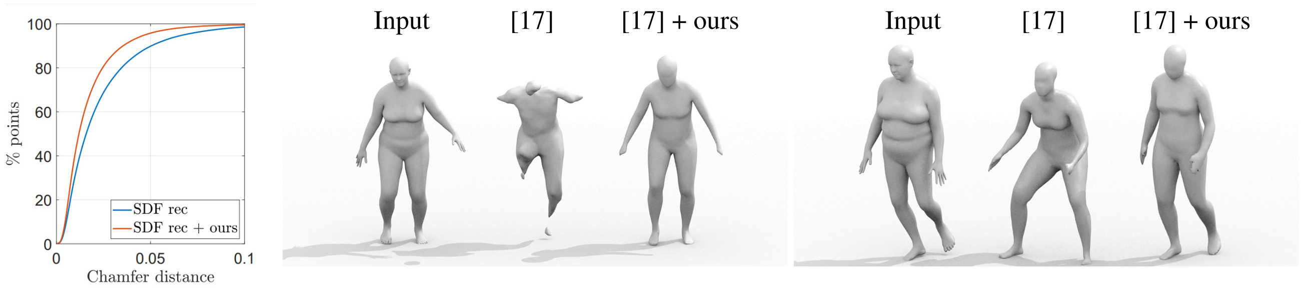

Figure 8: **Data augmentation for implicit surface reconstruction.** We show, as a proof of concept, that our method can be used to augment a small training set with additional poses. We verify this by comparing the reconstruction error of an implicit surface reconstruction method [17] with and without data augmentation. The general idea behind this is to construct a shape space that encodes arbitrary poses in a latent representation. As we show here, supplementing collections of sparse observations with intermediate poses constitutes a natural extension which helps to learn a meaningful shape distribution.

distortion corresponds to more realistic shapes. The other metric we consider measures the reconstruction error of the target shape $\mathcal{Y}$, defined as the Chamfer distance between $\mathcal{Y}$ and the deformed shape $\mathcal{X}(1)$. A good overlap at $t = 1$ is an important quality criterion because, while our interpolations exactly coincide with the first shape $\mathcal{X} = \mathcal{X}(0)$, they only approximately align with $\mathcal{Y} \approx \mathcal{X}(1)$. The same holds true for the three baselines [7, 26, 11] that we compare against.

Discussion. Results are shown in Figure 6. On both of these benchmarks, our method significantly outperforms the supervised baseline LIMP [7] which requires ground-truth correspondences for training. Similar to our approach, the unsupervised method ShapeFlow [26] continuously deforms a given input shape to obtain an interpolation. However, they do not estimate correspondences explicitly which limits the performance on deformable object categories like humans, animals or hands[3]. More surprisingly, our approach is even on par with the axiomatic, non-learning interpolation baseline [11] which requires to know dense correspondences at *test* time. See Figure 7 for a qualitative comparison on MANO.

4.3. Application: data augmentation

Our method is, to the best of our knowledge, the first one that jointly predicts correspondences and an interpolation of deformable objects in a single learning framework. As an application of unsupervised interpolation, we show how our method can be used to create additional training samples as a form of data augmentation. To that end, we train an implicit surface reconstruction method [17] on a small set of 20 SMPL shapes from the SURREAL dataset [56] and evaluate the obtained reconstructions on a separate test set of 100 shapes. Additionally, we use our method to create 3 additional, interpolated training poses for each pair in the training set and compare the results with the vanilla training. To measure the quality of the obtained reconstructions, we report the reconstruction error on the test set, defined as the Chamfer distance of the test shapes to the reconstructed surface, see Figure 8.

Overall, these results indicate that using our method to enlarge a training set of 3D shapes can be useful for downstream tasks, especially when training data is limited.

5. Conclusions

We presented a new framework for 3D shape understanding that simultaneously addresses the problems of shape correspondence and interpolation. The key insight we want to advocate is that these two goals mutually reinforce each other: Better correspondences yield more accurate interpolations and, vice versa, meaningful deformations of 3D surfaces act as a strong geometric prior for finding correspondences. In comparison to related existing approaches, our model can be trained in a fully unsupervised manner and generates correspondence and interpolation in a single pass. We show that our method produces stable results for a variety of correspondence and interpolation tasks, including challenging inter-class pairs with high degrees of non-isometric deformations. We expect that NeuroMorph will facilitate 3D shape analysis on large real-world datasets where obtaining exact ground-truth matches is prohibitively expensive.

Acknowledgements

We would like to thank Matan Atzmon and Aysim Toker for useful discussions and Roberto Dyke for quick help with the SHREC20 ground truth. This work was supported by the Munich Center for Machine Learning and by the ERC Advanced Grant SIMULACRON.

[3] These types of objects typically have a high pose variation with large degrees of non-rigid deformations, whereas ShapeFlow mainly specializes on man-made objects like chairs or cars. The few deformable examples that they show in [26, fig. 5] are intended as a proof of concept since they use ground-truth correspondences and overfit on a single pair of shapes.

References

[1] Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *IJCV*, 118(3):380–392, 2016.

[2] 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.

[3] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in neural information processing systems*, pages 3189–3197, 2016.

[4] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006.

[5] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016.

[7] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. Limp: Learning latent shape representations with metric preservation priors. *arXiv preprint arXiv:2003.12283*, 2020.

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[9] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. *arXiv preprint arXiv:2003.14286*, 2020.

[10] Roberto M Dyke, Yu-Kun Lai, Paul L Rosin, Stefano Zappalà, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. Shrec'20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020.

[11] Marvin Eisenberger and Daniel Cremers. Hamiltonian dynamics for real-world shape interpolation. In *ECCV*, 2020.

[12] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38, pages 1–12. Wiley Online Library, 2019.

[13] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020.

[14] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixe, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *arXiv preprint*, 2020.

[15] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[16] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019.

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*, pages 3569–3579. 2020.

[18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mache approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.

[20] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[22] Behrend Heeren, Martin Rumpf, Peter Schröder, Max Wardetzky, and Benedikt Wirth. Splines in the space of shells. *Computer Graphics Forum*, 35(5):111–120, 2016.

[23] Behrend Heeren, Martin Rumpf, Max Wardetzky, and Benedikt Wirth. Time-discrete geodesics in the space of shells. In *Computer Graphics Forum*, volume 31, pages 1755–1764. Wiley Online Library, 2012.

[24] Behrend Heeren, Chao Zhang, Martin Rumpf, and William Smith. Principal geodesic analysis in the space of discrete shells. In *Computer Graphics Forum*, volume 37, pages 173–184. Wiley Online Library, 2018.

[25] Kai Hormann and Günther Greiner. Mips: An efficient global parametrization method. Technical report, Erlangen-Nuernberg University (Germany) Computer Graphics Group, 2000.

[26] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, Leonidas Guibas, et al. Shapeflow: Learnable deformations among 3d shapes. *arXiv preprint arXiv:2006.07982*, 2020.

[27] Martin Kilian, Niloy J Mitra, and Helmut Pottmann. Geometric modeling in shape space. In *ACM Transactions on Graphics (TOG)*, volume 26, page 64. ACM, 2007.

[28] Vladimir G Kim, Yaron Lipman, and Thomas A Funkhouser. Blended intrinsic maps. *Transactions on Graphics (TOG)*, 30(4), 2011.

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] Or Litany, Tal Remez, Emanuele Rodolà, 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.

[31] Or Litany, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):1681–1707, 2017.

[32] Or Litany, Emanuele Rodolà, Alex M Bronstein, Michael M Bronstein, and Daniel Cremers. Non-rigid puzzles. *Computer Graphics Forum (CGF), Proceedings of Symposium on Geometry Processing (SGP)*, 35(5), 2016.

[33] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. on Graphics (TOG)*, 2015.

[34] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[35] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155, 2019.

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[37] 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.

[38] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019.

[39] 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.

[40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[41] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

[42] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[44] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, Dec. 2018.

[45] Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum (CGF)*, 2016.

[46] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[47] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1627, 2019.

[48] Klaus Hildebrandt Ruben Wiersma, Elmar Eisemann. Cnns on surfaces using rotation-equivariant features. *Transactions on Graphics*, 39(4), July 2020.

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

[50] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional map for shape matching. *arXiv preprint arXiv:2009.13339*, 2020.

[51] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[52] Gary K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, July 2013.

[53] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019.

[54] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010.

[55] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[56] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017.

[57] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex M Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] Philipp von Radziewsky, Elmar Eisemann, Hans-Peter Seidel, and Klaus Hildebrandt. Optimized subspaces for deformation-based modeling and shape interpolation. *Computers & Graphics*, 58:128–138, 2016.

[59] Chaohui Wang, Michael M Bronstein, Alexander M Bronstein, and Nikos Paragios. Discrete minimum distortion correspondence problems for non-rigid shape matching. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 580–591. Springer, 2011.

[60] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019.

[61] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[62] Benedikt Wirth, Leah Bar, Martin Rumpf, and Guillermo Sapiro. A continuum mechanical approach to geodesics in shape space. *International Journal of Computer Vision*, 93(3):293–318, Jul 2011.

[63] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[64] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

[65] Chao Zhang, Behrend Heeren, Martin Rumpf, and William AP Smith. Shell pca: Statistical shape modelling in shell space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1671–1679, 2015.

[66] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.