Reduced Representation of Deformation Fields for Effective Non-rigid Shape Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work we present a novel approach for computing correspondences between non-rigid objects, by exploiting a reduced representation of deformation fields. Different from existing works that represent deformation fields by training a general-purpose neural network, we advocate for an approximation based on mesh-free methods. By letting the network learn deformation parameters at a sparse set of positions in space (nodes), we reconstruct the continuous deformation field in a closed-form with guaranteed smoothness. With this reduction in degrees of freedom, we show significant improvement in terms of data-efficiency and enable limited supervision. Furthermore, our approximation provides direct access to first-order derivatives of deformation fields, which facilitates enforcing desirable regularization effectively. Our resulting model has a high expressive power and is able to capture complex deformations. We illustrate its effectiveness through state-of-the-art results across multiple deformable shape matching benchmarks.

1 Introduction

Shape correspondence is a central problem in computer vision and computer graphics as it facilitates many downstream tasks, such as tracking [1], texture transfer [2] and statistical modeling [3] to name a few. Due to its ubiquitous applicability, a wide range of techniques have been developed over the past several years [4]. While early approaches relied on axiomatic modeling, recent methods follow data-driven techniques based on different input signals [5, 6, 7, 8] within a shape collection.

A key question in this context is the choice of *representation* used to model the non-rigid shape matching problem. Approaches based on *intrinsic* or pose invariant representations have established a gold standard in the context where surfaces are well-defined [9, 10, 11]. Such methods, however, strongly rely on the presence of clean shapes and struggle when acquisition comes from noisy and non-uniform discretization [12]. In contrast, extrinsic techniques which directly operate on Euclidean space ($\mathbb{R}^3$) show strong resilience to artifacts.

Unfortunately, this robustness of extrinsic methods often comes at the cost of having access to significant amounts of annotated training data [7, 5]. The main limiting factor arises in the representation of the deformation fields. The standard approach is to use general-purpose MLPs to learn deformation fields that can fit an arbitrary shape deformation [13, 14, 5]. However, given the fact that MLPs are often biased towards global patterns (coarser details) in the input signal [15], they require abundant training data to generalize across finer details [5].

To overcome this limitation, we propose to learn a coarse representation of *deformation parameters* at fixed positions in space called "nodes". By learning a reduced representation of deformation fields, intuitively, we restrict the learning process to global patterns of the input signal. Then, to recover finer-details, we reconstruct the continuous deformation field function in a closed-form using

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

a class of mesh-free approximation techniques [16]. This allows us to scale our approach to arbitrary resolution with guaranteed smoothness.

Apart from being theoretically grounded and simple in practice, our reduced representation has two key advantages. First, it is significantly more data-efficient and can learn to capture complex deformations given only a small number of examples. Secondly, it is more amenable to regularization, since it provides explicit access to first order derivatives of the deformation field in closed form. This is especially useful in imposing geometric priors such us local rigidity and volume preservation.

Our contributions can be summarized as follows: (a) We propose to learn a compact representation of deformation parameters, that is data-efficient, resolution agnostic, and facilitates regularization through direct access to deformation gradients (b) We show an efficient way of incorporating desirable regularization to promote a well-structured deformation space. (c) Through extensive experiments, we demonstrate the generalization ability of our method over different down-stream applications such as non-rigid shape matching, registration, unsupervised part segmentation and interpolation. Our code will be publicly released.

2 Related Works

2.1 Non-rigid shape Correspondence and Registration

Shape correspondence is a very well-studied area of computer vision and computer graphics and we refer interested readers to the recent survey [4] for a comprehensive overview. Notable axiomatic approaches in this category are based on the functional maps paradigm [9, 17, 18, 19], that solves an isometric mapping by estimating a linear transformations between functions represented in a reduced basis. This framework has been successfully adapted by learning techniques [6, 20, 21, 7, 10] which demonstrate near-perfect accuracy [11] on several shape correspondence benchmarks. However, these approaches demonstrate weak resilience when exposed to artefacts arising from surface discretization, which are common in acquired data. While registration techniques [22, 23, 24] are relatively a more reliable option, they either use strong human-centeric priors or require significant training data.

2.2 Template-based and Template-free Methods

Deforming a template shape to match a target geometry is a long-standing and well studied problem [25, 26, 27]. Such a template can be a polygonal mesh [28, 29], possibly parameterised [30, 31, 32] or an unordered point-set [33, 34, 35, 36, 37] or implicitly defined through zero-level set of a Neural Field [13, 14, 38]. In the recent years, learning based model-free deformation techniques [33, 34, 24] have emerged successful for registration and correspondence tasks with copious availability of training data [33, 39]. Amongst them, the closest to our approach is 3D-CODED [33] which learns deformation fields through point-wise MLPs. However, since each point in the shape is treated equally likely, this approach fails to generalize across high-frequency features, thereby requiring abundant training data to achieve optimal performance.

2.3 Deformation Field Representation and Shape Interpolation

Deformation between a pair of shapes can be represented as a simple displacement field at every vertex. However, such a representation can be unnecessarily complex, and costly to optimize. As a result, several alternatives have been proposed. The most prominent ways to parameterize the space of deformations include handle-based [40, 41, 42, 43] or cage-based [44, 45, 46, 47] representations (see also [48, 49, 50] for an overview). More recently, a common approach is to construct a reduced representation via a learned latent embedding [51, 52, 53, 54].

Deformation Field regularization Several geometric constraints have been proposed with the aim of preserving desirable properties of the shape by the deformation field, including elasticity [55, 56, 57] and volume preservation[58, 59, 41, 60, 61]. Recently, these constraints have been successfully adapted by data-driven methods [62, 52, 63, 38, 64] and more relevantly through the differential of the map [52, 64, 64]. Distinct from such approaches, our approximation via mesh-free method enables evaluating this map differential at fixed points in a closed-form, which significantly simplifies the deformation field regularization without additional computational overhead.

Shape Interpolation Shape interpolation refers to time parameterized deformation, where a source shape is continuously deformed to a target shape. Our work is related to efforts which aim to enforce intermediate shapes to preserve certain intrinsic properties [53, 54, 62, 65, 66, 52]. Among them closest to our approach is LIMP [53] which disentangles the latent space based on style and pose to preserve geodesic distance. In contrast, our approach does not require such a priori information, which can be costly in terms of annotation efforts.

2.4 Reduced representations and Approximations

In this work we use mesh-free function approximation method [67, 68, 16], to approximate deformation fields. Mesh-free methods have been successfully adapted in Smoothed Particle Hydrodynamics (SPH) modeling [69, 70], image processing [40], animation [41, 71, 72] and more recently in a data-driven framework [42]. Differently from [42], instead of learning the weights of the least squares function, we instead learn deformation values at nodes and demonstrate our method to be applicable in wide-range of downstream tasks. Alternatively, Eisenberger *et al.* [61] have proposed to use a compact representation of deformation fields using the first k eigenfunctions of the Laplace Beltrami Operator (LBO). While their approach provides volume preserving deformation, it does not facilitate other regularizations such as as-rigid-as-possible deformation fields without requiring correspondence at inference time [73].

3 Motivation, Background and Notation

3.1 Motivation:

Parametric models such as SMPL [30] have been tremendously useful over the recent years in digitizing and processing human models. This success can largely be attributed to their expressive power, allowing to generate a wide range of styles and poses using a small fixed set of deformation parameters. While this efficacy with such a compact representation is remarkable, it also raises an inspiring question: what is the optimal amount of *learnable* parameters necessary to represent general deformations? Today, general-purpose MLPs form the conventional way of representing deformation fields due to their simplicity and potential of being universal functional approximators [74, 75]. Unfortunately, the generic power of MLPs also comes at a cost of copious training efforts [33, 24]. Furthermore, representing a deformation field using a neural network makes access to certain quantities such Jacobian matrices of deformation fields cumbersome. For these reason, we propose to learn a reduced set of deformation parameters from which we *approximate* the deformation field function using a mesh-free method.

3.2 Mesh-free Approximation

Mesh-free methods are a class of approximation techniques which constructs a continuous function based on independent, potentially sparse and irregular observations. Assume that our domain of interest $\mathbb{R}^3$ is equipped with K fixed points $\mathbf{q}_i \in \mathbb{R}^3$ along with some observations u_i at $\mathbf{q}_i$ and a polynomial basis $p(\cdot)$. We refer to fixed points $\mathbf{q}_i$ as "Nodes". Our main interest is to construct a continuous approximation of some real-valued function $u(.)$ across regions in some subdomain $\Omega \subset \mathbb{R}^3$ of interest. We let $\mathbf{x} \in \Omega$ to be an arbitrary point in our region of interest. The key idea behind this approximation is to use a *local weighted least-squares* fitting (also referred to as Moving Least Squares) approach [16], in the neighbourhood of $\mathbf{q}_i$, using a compactly supported weighting function, similar to [41] given by:

$$w_i(\mathbf{x}) = \begin{cases} \left(1 - \dfrac{||\mathbf{x} - \mathbf{q}_i||_2^2}{r_i}\right)^3, & \text{if } ||\mathbf{x} - \mathbf{q}_i||_2^2 \leq r_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The compactness of this weighting function is useful in preserving the local characteristics of approximation. From this, a *Shape Function* Φ_i associated with each node i, is constructed as:

$$\Phi_i(\mathbf{x}) = p^T(\mathbf{x})[M(\mathbf{x})]^{-1} w_i(\mathbf{x}) p\left(\mathbf{q}_i\right). \tag{2}$$

Here $M(\mathbf{x})$ is the Moment Matrix associated with the approximation defined as,

$$M(\mathbf{x}) = \sum_{i=1}^{K} w_i(\mathbf{x}) p\left(\mathbf{q}_i\right) p^T\left(\mathbf{q}_i\right)$$

The shape function Φ_i is a continuous function that describes how each node $\mathbf{q}_i$ influences the approximation of $u(.)$ across points $\mathbf{x} \in \Omega$. Jointly the the set of Φ_i's enable the reconstruction of arbitrary functions up to n^{th} order consistency [68], where, n is the order of the polynomial $p(\cdot)$. Specifically, a smooth local approximation of $u(\mathbf{x})$ is given as:

$$u(\mathbf{x}) = \sum_{i=1}^{K} \Phi_i(\mathbf{x}) u_i \tag{3}$$

As the construction of Φ involves computing M^{-1} (c.f Equation 2), it is a sufficient condition for each point $\mathbf{x}$ to be compactly supported by 4 non-planar nodes $\mathbf{q}_i$ for M to be non-singular. It is important to note that Equation 3 is approximating and *not interpolating*, i.e $u_i \neq u(\mathbf{q}_i)$. For instance, owing to the compact nature of $w(\mathbf{x})$, it is possible that $u(\mathbf{q}_i)$ is undefined if $\mathbf{q}_i \notin \Omega$. For this reason, we sample the nodes a priori to have a well-supported domain $\Omega \subseteq \mathbb{R}^3$ where $u(\mathbf{x})$ is well-defined.

Furthermore, an important advantage of using mesh-free approximations comes from an exact analytical expression for the gradient function of $u(\mathbf{x})$. To the scope of our current discussion, considering $u(\mathbf{x})$ to be the approximation of deformation field function, the Jacobian of this deformation field only depends on evaluation point and is independent of observed deformation parameters u_i ,

$$\mathbb{J} = \nabla_{x,y,z} u(\mathbf{x}) = \sum_{i=1}^{K} \left[\frac{\partial \Phi_i(\mathbf{x})}{\partial x}, \frac{\partial \Phi_i(\mathbf{x})}{\partial y}, \frac{\partial \Phi_i(\mathbf{x})}{\partial z} \right]^T u_i \tag{4}$$

This nice property helps us characterize the deformation fields with desired first-order regularization in an efficient manner. We refer interested readers to [68, 67] for a detailed summary.

3.3 Notation:

As our training set, we consider a collection of shapes $\{\mathcal{S}_1 \ldots \mathcal{S}_N\}$ with correspondence $\Pi_{\mathcal{S}_l \mathcal{S}_j}$ between them. They could be a triangular mesh $\mathcal{S}_j \coloneqq \{\mathcal{V}, \mathcal{E}\}$ or simply an unordered set of points (Point Clouds) $\mathcal{S}_j \coloneqq \{\mathcal{V}\}$. From them, we pick one as a template $\mathcal{T}$, and let $[\mathcal{T}]$ be the volume enclosed by the boundary $\partial\mathcal{T}$. We let $\mathcal{D}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be the *deformation mapping*. We refer to U_j as the nodal deformation parameters corresponding to j^{th} shape and analogously define $D_j(\cdot)$. We refer to $\mathcal{Q} \in \mathbb{R}^{K \times 3}$ as *nodal positions*, which are K *fixed* points in space sampled from the template volume $[\mathcal{T}]$. Each node $q_i \in \mathcal{Q}$ has a support radius r_i and associated deformation parameter u_i. We use lower-case notation $u_{i,j}$ to refer to the values of deformation field at node $\mathbf{q}_i$ corresponding to shape j. We denote $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ as points in space which are supported by 4 non-planar nodes. We refer to $U_j(\mathbf{x})$ as the continuous approximation of the deformation field, constructed from deformation parameters U_j using Equation 3. We re-iterate that $U_j \neq U_j(\mathcal{Q}_i)$. The relation between deformation field function and mapping is given by $\mathcal{D}_j(\mathbf{x}) \coloneqq \mathbf{x} + U_j(\mathbf{x})$. For the sake of consistency, we index nodes using i, shape collection using j, l and points within shape using k.

4 Method: Learning Nodal Deformation-Field

Overview. Our network consists of PointNet [76] based Auto-Encoder as shown in Figure 1. Our network $\mathcal{F}_\theta(\cdot)$ predicts nodal deformation parameters for each training shape $\mathcal{S}_j$, i.e $U_j = \mathcal{F}_\theta(\mathcal{S}_j)$ where $U_j \in \mathbb{R}^{K \times 3}$. As previously mentioned, the nodes $\mathcal{Q}$ are fixed a priori. From the predicted U_j, we can reconstruct the shape-specific deformation mapping $\mathcal{D}_j(x)$ and its Jacobian $\mathbb{J}_j$ as

$$\begin{aligned} \mathcal{D}_j(\mathbf{x}) &= \mathbf{x} + \sum_{i=1}^{K} \Phi_i(\mathbf{x}) u_{i,j} \\ \mathbb{J}_j &= \mathbf{I} + \nabla_{x,y,z} U_j(\mathbf{x}) \end{aligned} \tag{5}$$

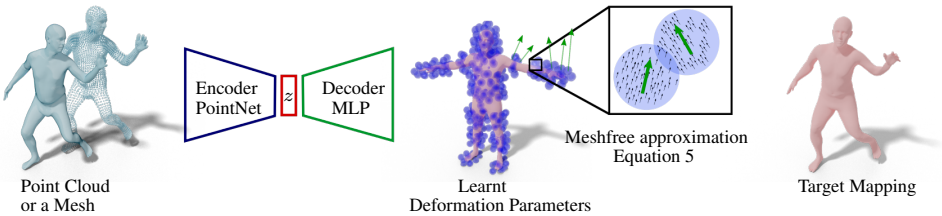


Figure 1: Overview of our approach. First, we learn the deformation parameters at nodes using an Auto-Encoder. Then, we use mesh-free approximation to obtain a continuous deformation mapping.

Where, $\nabla_{x,y,z} U_j(\mathbf{x})$ is given in Equation 4.

4.1 Training

Intuitively, we would like to train a network so that $S_j \approx \{D_j(\mathbf{x}) | \mathbf{x} \in \mathcal{T}\}$, while ensuring $D_j(\mathbf{x})$ is locally rigid and volume preserving. Although $\mathcal{D}_j(\mathbf{x})$ can be approximated at an arbitrary $\mathbf{x}$, which is supported by 4 non-planar nodes, we restrict ourselves to $\mathbf{x} \in \mathcal{T}$ for the ease of learning. As $\mathcal{F}_\theta(\cdot)$ represents an auto-encoder, it can be decomposed as $\mathcal{F}_\theta(\mathcal{S}_j) = Dec(Enc(\mathcal{S}_j)) = Dec(Z_j)$ where Z_j denotes the latent embedding. Leveraging this fact, we provide a novel way to promote plausible latent deformation spaces Z_j by enforcing first-order constraints over the *intermediate shapes* as well. Consequently, the overall optimization objective of our network is given as,

$$\mathcal{L}_{net} = \lambda_1 \mathcal{L}_{cor} + \lambda_2 \mathcal{L}_{vol} + \lambda_3 \mathcal{L}_{arap} + \lambda_4 \mathcal{L}_Z \tag{6}$$

For the unsupervised case, we replace $\mathcal{L}_{cor}$ with $\mathcal{L}_{CD}$ which denotes the Chamfer's distance.

Correspondence Loss Given a set of $\mathcal{C}$ of corresponding points $\{\mathbf{x}_l, \mathbf{x}_k\}$, where $\mathbf{x}_l \in \mathcal{T}$, $\mathbf{x}_k \in \mathcal{S}_j$ our correspondence loss is given as

$$\mathcal{L}_{cor} = \sum_{j=1}^{N} \sum_{(\mathbf{x}_k, \mathbf{x}_l)}^{|\mathcal{C}|} \left\| \mathcal{D}_j(\mathbf{x}_l) - \mathbf{x}_k^j \right\|_2^2 \tag{7}$$

Where x_k^j denotes the kth point in j^{th} shape.

Volume Preserving Field A deformation field is volume preserving iff its Jacobian has unit determinant over the entire shape. Consequently, our volume preservation constraint is given as follows,

$$\mathcal{L}_{vol} = \sum_{j=1}^{N} \sum_{i=1}^{K} |\det(\mathbb{J}_j(\mathbf{q}_i)) - 1|_2^2 \tag{8}$$

We empirically observe poor convergence when this constraint is enforced over the entire shape due to its stringent nature. This is because not all deformation are strictly volume preserving. Thus, we restrict this regularization only at nodes.

As Rigid As Possible (ARAP) Deformation Since rigid motions preserve pair-wise distances, a deformation field associated with such a transformation is characterized by an orthonormal Jacobian matrix. Thus, in order to promote locally rigid deformation field at nodes, we define our ARAP regularization as,

$$\mathcal{L}_{arap} = \sum_{j=1}^{N} \sum_{i=1}^{K} \left\| \mathbb{J}_j^T(\mathbf{q}_i) \, \mathbb{J}_j(\mathbf{q}_i) - \mathbf{I} \right\|_F^2 \tag{9}$$

Structuring Latent Deformation Space A well-known advantage of an auto-encoder architecture is the construction of the latent space, where each shape has an embedding $Z_j \in \mathbb{R}^D$. Then, a parameterized path in this latent space between two shapes $Z_{l,j}(\alpha) = \alpha Z_l + (1-\alpha) Z_j$ continuously deforms $\mathcal{S}_j$ to $\mathcal{S}_l$ with rate of change controlled by α. This enables construction of a shape sequence between a pair of shapes, often referred to as interpolated shapes. Since each $Dec(Z_j) = U_j$, we can further constraint our network $\mathcal{F}_\theta$ to produce a plausible deformation between *each pair of training shapes*. To that end, we introduce our latent smoothness loss as follows,

$$\mathcal{L}_Z = \sum_{l \neq j}^{|\mathcal{S}|} \mathcal{L}_{arap}\left(\text{dec}((1-\alpha)\mathbf{z}_j + \alpha \mathbf{z}_l)\right) + \sum_{l \neq j}^{|\mathcal{S}|} \mathcal{L}_{vol}\left(\text{dec}((1-\alpha)\mathbf{z}_j + \alpha \mathbf{z}_l)\right) \tag{10}$$

Where $\mathcal{L}_{arap}, \mathcal{L}_{vol}$ are defined in Equations 8 9 and $\alpha \in (0,1)$ are sampled randomly.

4.2 Inference

At test-time, given a pair of unseen shapes $(\mathcal{X}, \mathcal{Y})$, we first separately reconstruct $(\mathcal{D}_\mathcal{X}, \mathcal{D}_\mathcal{Y})$ by deforming the template $\mathcal{T}$. Second, we enhance the reconstruction by optimizing the latent vector Z independently for shapes $(\mathcal{X}, \mathcal{Y})$ to minimize the bi-directional Chamfer's Distance [33] while *also* enforcing first-order constraints as follows,

$$Z = \underset{Z}{\text{argmin}}\ \Lambda_1 \mathcal{L}_{\mathcal{CD}} + \Lambda_2 \mathcal{L}_{arap} + \Lambda_3 \mathcal{L}_{vol} \tag{11}$$

As $\mathcal{D}_\mathcal{X}$ is the reconstruction of $\mathcal{X}$, correspondence between them is a nearest neighbor search (analogously for $\mathcal{Y}$). Finally, since $\mathcal{D}_\mathcal{X}, \mathcal{D}_\mathcal{Y}$ are deformed versions of a template (and hence enjoy a natural correspondence), the correspondence between $(\mathcal{X}, \mathcal{Y})$ is a composition of two nearest neighbour searches $\Pi_{\mathcal{XY}} == (\text{NN}(\mathcal{D}_\mathcal{X}, \mathcal{X}), \text{NN}(\mathcal{D}_\mathcal{Y}, \mathcal{Y}))$.

4.3 Extending sparse to dense Correspondence

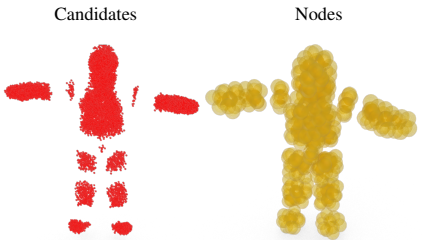
An added advantage of our representation is the ability to retrieve dense shape correspondence between a pair of shape, given a few sparse key-point correspondences $(\mathbf{x}_l, \mathbf{y}_k), \forall \mathbf{x}_l \in \mathcal{X}, \forall \mathbf{y}_k \in \mathcal{Y}$. First, we estimate the deformation parameter u_i at the nodes by solving an optimization,

$$u_i = \underset{u_i}{\text{argmin}}\ \lambda_1 \sum_{\forall(\mathbf{x}_l, \mathbf{y}_k)} \|\mathcal{D}_\mathcal{X}(\mathbf{x}_l) - \mathbf{y}_k\|_2^2 + \lambda_2 \left\|\mathbb{J}_\mathcal{X}^T \mathbb{J}_\mathcal{X} - I\right\|_F^2 + \lambda_3 |\det(\mathbb{J}_\mathcal{X}) - 1|_2^2 \tag{12}$$

Then, a dense mapping can be computed by approximating the deformation field (c.f Equation 5).

4.4 Implementation details

Analytical Gradients and Timing advantages We leverage the advantage of inexpensive access to Jacobians as mentioned in Equation 4. Because our evaluation points are known a priori, due to the use of a fixed template $\mathcal{T}$, the matrix $\mathbb{J}$ can be pre-computed and re-used at training and evaluation. In practice, we observe a $\mathbf{10}\times$ speed-up at training time when enforcing our first-order constraints and a $\mathbf{350}\times$ speed-up incorporating the latent constraints (c.f. Eqn 10). We provide more timing details in the supplementary.



Node Sampling: Since the deformation field is determined by its nodal neighbourhood, it is important to prevent a node from asserting its influence on semantically different regions. Bearing this in mind, our node sampling strategy is divided into three main steps. First, we construct a dense sampling of points in the volume and around the boundary of the template $\partial\mathcal{T}$. Second, we use rejection sampling to *exclude* a node that exerts its influence in semantically different regions [30]. Finally, we perform Farthest Point Sampling (FPS) until each surface point is covered by 4 non-planar nodes. Second and third steps are shown in the inset Figure.

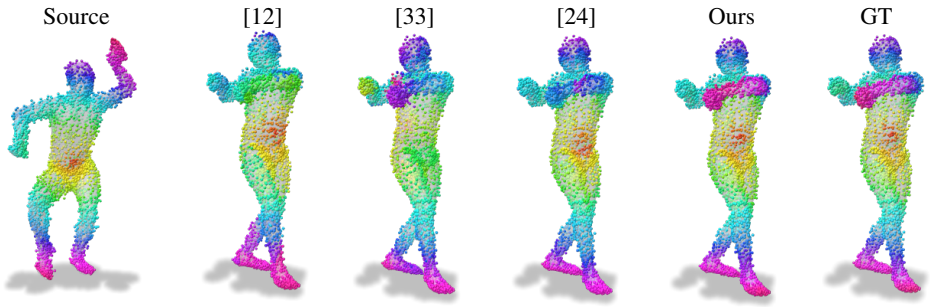


Figure 2: Color-coded correspondences on the SCAPE (PC+N) dataset. "Twist" is a challenging articulation due to proximity of points in localized regions (e.g hands). Ours is the only method which computes accurate correspondence even in the presence of noise. Underlying surfaces are shown for visualization purpose only.

Method			Correspondence Error		
Type	Name	#Tr data	SHREC'19	FAUST(NI)	SCAPE(PC+N)
Spectral	GeoFMap [7]	1.7	11.2	20.1	27.7
Pair-wise	Diff-FMap [12]	1.0	15.1	5.4	26.0
	CorrNet3D [8]	15.0	9.6	25.9	38.0
Template based	3D-CODED [5]	23.0	10.3	7.0	18.7
	TransMatch [24]	1.0	6.1	6.5	17.1
	Ours	**0.1**	**4.8**	**5.3**	**6.6**

Table 1: We report correspondence error as geodesic distortion (in cm) scaled by square root of shape area. #Tr data denotes number of training shapes scaled by 10^{-4}.

5 Experiments

We consider four main tasks : Non-rigid 3D shape correspondence, Shape registration, Unsupervised part segmentation and Shape interpolation.

5.1 Shape Correspondence

We consider three challenging benchmarks, namely, SHREC'19, FAUST (PC), SCAPE (PC+N). SHREC'19 [77] is a standardised benchmark consisting of 430 evaluation pairs with significant variations in mesh resolution and connectivity. FAUST (PC) denotes a more recent Non-Isometric Point Cloud variant [12] of the FAUST dataset consisting of 1000 points with large variance in point sampling density. Third, we evaluate on a variant of the recent SCAPE-Remesh dataset [78] consisting of 20 shapes of the same human in 20 distinct poses. We further augment the challenge by adding random Gaussian noise and refer to as SCAPE (PC+N). We evaluate correspondence error following the Princeton benchmark protocol [79]. We train our method on a subset of 1000 SURREAL shapes [80] for 1000 epochs with data-augmentation along Y-axis.

Baselines We compare our method against data-driven correspondence methods broadly classified into Spectral, Pair-wise and Template based. We use GeoFMap [7] with the more robust feature extractor Diffusion-Net [11] as our spectral baseline, Diff-FMaps [12] and CorrNet3D [8] as our pair-wise baseline. For our template based baselines, we use 3D-CODED [33] and TransMatch [33]. For evaluation of baselines on our proposed SCAPE (PC+N), we use the author provided pre-trained models upon desired pre-processing.

Discussion Our approach consistently outperforms baselines as summarised in Table 1. While our quantitative correspondence results are persuasive, it is remarkable to note that our method requires an order of magnitude less training data in comparison to competing methods. This supports our premise that characterizing articulated motions requires far fewer parameters than what is leveraged by existing data-driven methods. We show qualitative correspondence results through color transfer for a challenging pair with "twisted" motion in Figure 2.

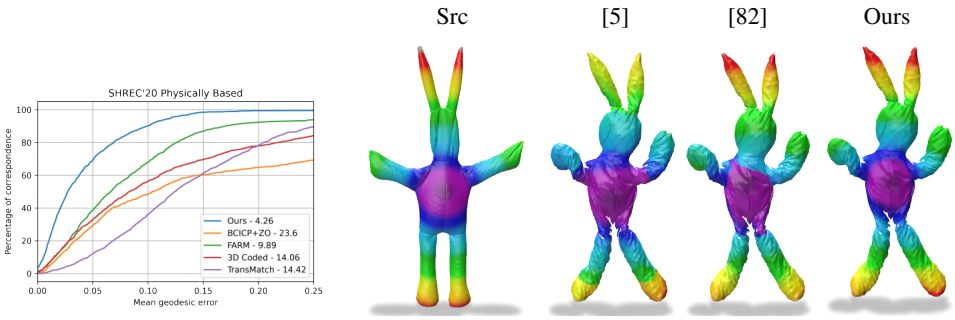


Figure 3: Quantitative and qualitative results on SHREC'20. Our approach predicts smooth correspondences across highly-granular surface-level deformation.

5.2 Shape Registration

Shape registration is a special case of correspondence, where, our goal is to find an optimal deformation between the scan and a fixed template. For this, we consider the recent SHREC'20 benchmark [81], consisting of 11 partial scans of stuffed toy rabbits to be registered to a single scan. This benchmark is particularly challenging due to granulated surface deformation, scanning aretefacts, and limited data and supervision.

Experiment We split this dataset into 7 training shapes and 4 shapes for evaluation. The 4 shapes in our test set is made of "chickpea" material, which exhibits the largest magnitude of protrusions and indentations on the surface. We stress that this category of deformation is unseen during training time. We compare our method against 4 baseline methods namely, FARM [22], BCICP [78]+ZoomOut [83], 3D-CODED [33] and TransMatch [24]. We summarize our quantitative and qualitative results in Figure 3. It is remarkable that our approach outperforms axiomatic and competing data-driven baselines by at least a two-fold margin. Importantly, despite the fact that our approach shares the same encoder [76] as 3D-CODED [33], there is a striking difference in performance. We attribute this to our well-regularized deformation space.

5.3 Unsupervised Segmentation Transfer

In this section, we demonstrate the generalization ability of our approach to model deformation fields between shapes with considerable topological differences. To that end, we consider the task of part level segmentation over point clouds consisting of table and plane categories from ShapeNet [84] dataset. Apart from topological differences and large structural variance, absence of ground truth annotations exacerbates the challenge. In this setting, we compare our method with two Deep Implicit networks, namely DIF-Net [13] and DIT [14], which model a volumetric deformation fields between a learnt template and training shapes. Our choice of baseline endows us with a fair ground of comparison between the two representations of deformation field - MLP based and ours.

Experiment and Discussion We train our approach using the unsupervised loss mentioned in Section 4.1 over 1000 objects sampled at random from each category. We consider 190 evaluation pairs per-category from the prescribed validation set and measure the segmentation accuracy by the

	Plane		Table	
	CD (x1e4)	IoU (%)	CD (x1e4)	IoU (%)
DIT [14]	24.6	69.1	26.7	68.9
DIF [13]	15.0	78.0	11.2	79.3
Ours (w/o Con)	**0.5**	88.8	**2.6**	88.9
Ours	0.6	**89.3**	3.0	**90.0**

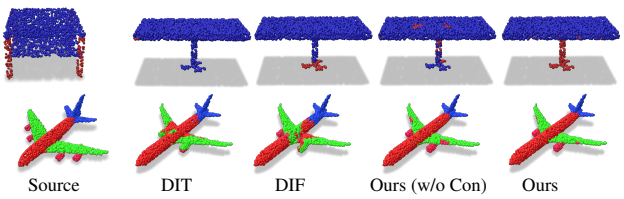


Figure 4: Ours (w/o Con) is a variant of our approach without any deformation constraints. Both of our variants show a significant improvement over baseline that models volumetric deformation fields. Qualitative result demonstrates color coded segmentation transfer across significant shape variability.

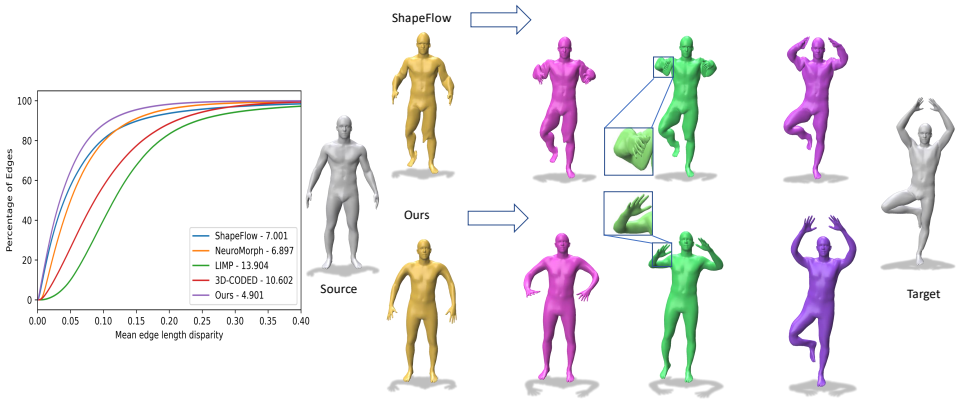


Figure 5: Quantitative and qualitative comparison of interpolation. While ShapeFlow [52] enforces volume preservation prior, its latent deformation space are not distortion-free. By better structuring the latent-space (c.f Eqn 10), our sampled intermediate shapes are near distortion-free.

IoU metric [84]. In addition, we also measure the bi-directional Chamfer's distance of reconstructed geometries. We summarize our quantitative observation along with a qualitative example in Figure 4. We remark that while deformation fields between aforementioned categories are not strictly volume preserving, we still observe a noticeable improvement over the baseline. This is because our deformation priors help in *structuring* the space of deformations, which explicitly avoids degeneracy such as collapsing shape parts. This remark is corroborated by a lower (preferred) Chamfer's distance while there is *a decline* in the accuracy when no regularization is applied.

5.4 Shape Interpolation

A notable characteristic of a well-structured latent space is the ability to produce plausible intermediate shapes given a source and a target. This task is commonly referred to as shape interpolation. Since there exists no canonical path, interpolation sequences are gauged by the extent to which intrinsic metrics are preserved, in particular isometric distortion [85]. For this setting, we consider the FAUST [3] dataset, where, we train our method on the first 80 shapes and evaluate over the last 20 shape pairs. We compare our method against four baselines namely 3D-CODED [5], NeuroMorph [62], LIMP [53] and ShapeFlow [52]. Since LIMP employs a fixed-size decoder and NeuroMorph uses a separate interpolation module involving an explicit computation of correspondence matrix, both of these approaches are limited by shape resolution. On the other hand, our approach is resolution agnostic and outperforms the baselines by a discernible margin as summarized in Figure 5. This improvement over the baseline is due to the incorporation of our latent deformation priors in a computationally feasible manner, which we will be justified through an extensive ablation study in the supplementary.

6 Conclusion, Limitations and Future Work

We presented an effective representation of deformation fields by learning a reduced set of shape-specific deformation parameters while constructing the continuous deformation field function using mesh-free approximation technique. This elegant representation endowed us with access to first-order derivatives in closed-form, thereby facilitating the use of strong first-order regularization. A key observation behind our method is that in many settings, the space of realistic deformations is well-constrained and expressed with a small set of parameters. To that end, we demonstrated that our approach can achieve significant improvement upon existing baselines across challenging downstream applications and remarkably reduce the dependence on training data. Our approach still has some limitations and leads to possible exciting future work. Firstly, while our approach produces smooth deformation field in principle, there is no guarantee on bijectivity or invertibleness. Second, instead of fixed nodal positions, optimizing with respect to our approximation function is also an interesting idea to ponder.

References

[1] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379. Springer, 2016.

[2] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3794–3801, 2014.

[4] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020.

[5] 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.

[6] 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.

[7] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020.

[8] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[9] 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):1–11, 2012.

[10] 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.

[11] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics*, 41(3):1–16, June 2022.

[12] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *ArXiv*, abs/2010.13136, 2020.

[13] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10296, 2021.

[14] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1429–1439, 2021.

[15] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019.

[16] Peter Lancaster and Kestutis Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37:141–158, 1981.

[17] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448. Wiley Online Library, 2013.

[18] Yonathan Aflalo and Ron Kimmel. Spectral multidimensional scaling. *Proceedings of the National Academy of Sciences*, 110(45):18052–18057, 2013.

[19] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017.

[20] 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.

[21] Jean-Michel Roufosse, 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.

[22] R. Marin, S. Melzi, E. Rodolà, and U. Castellani. Farm: Functional automatic registration method for 3d human bodies. *Computer Graphics Forum*, 39(1):160–173, 2020.

[23] D. Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *European Conf. on Computer Vision (ECCV)*, LNCS 7577, Part IV, pages 242–255. Springer-Verlag, October 2012.

[24] Giovanni Trappolini, Luca Cosmo, Luca Moschella, Riccardo Marin, Simone Melzi, and Emanuele Rodolà. Shape registration in the time of transformers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[25] Brett Allen, Brian Curless, and Zoran Popović. Articulated body deformation from range scan data. *ACM Trans. Graph.*, 21(3):612–619, jul 2002.

[26] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parameterization from range scans. 22(3):587–594, jul 2003.

[27] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[28] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[29] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 173–182, New York, NY, USA, 1995. Association for Computing Machinery.

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

[31] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[32] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 28(2):337–346, April 2009.

[33] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *ECCV*, 2018.

[34] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *NeurIPS*, 2019.

[35] Thibault Groueix, Matthew Fisher, Vova Kim, Bryan Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. In *Symposium on Geometry Processing (SGP)*, 2019.

[36] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[38] Matan Atzmon, David Novotny, Andrea Vedaldi, and Yaron Lipman. Augmenting implicit neural shape representations with explicit deformation fields. *arXiv preprint arXiv:2108.08931*, 2021.

[39] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018.

[40] Scott Schaefer, Travis McPhail, and Joe D. Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25:533–540, 2006.

[41] Bart Adams, Maks Ovsjanikov, Michael Wand, Hans-Peter Seidel, and Leonidas J. Guibas. Meshless modeling of deformable shapes and their motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, page 77–86, Goslar, DEU, 2008. Eurographics Association.

[42] Meitar Shechter, Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. DeepMLS: geometry-aware control point deformation. *Proc. Eurographics Short Papers*, 2022.

[43] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4), jul 2011.

[44] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Trans. Graph.*, 27(3):1–10, aug 2008.

[45] Michael S. Floater, Géza Kós, and Martin Reimers. Mean value coordinates in 3d. *Computer Aided Geometric Design*, 22(7):623–631, October 2005.

[46] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26(3):71–es, jul 2007.

[47] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020.

[48] Dominique Bechmann. Space deformation models survey. *Computers & Graphics*, 18(4):571–586, 1994.

[49] James Gain and Dominique Bechmann. A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics (TOG)*, 27(4):1–21, 2008.

[50] Jesús R Nieto and Antonio Susín. Cage based deformations: a survey. In *Deformation models*, pages 75–99. Springer, 2013.

[51] Mikaela Angelina Uy, Vladimir G. Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas Guibas. Joint learning of 3d shape retrieval and deformation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[52] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, and Leonidas Guibas. Shapeflow: Learnable deformations among 3d shapes. In *Advances in Neural Information Processing Systems*, 2020.

[53] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. LIMP: Learning latent shape representations with metric preservation priors. In *Computer Vision – ECCV 2020*, pages 19–35. Springer International Publishing, 2020.

[54] Marie-Julie Rakotosaona and Maks Ovsjanikov. Intrinsic point cloud interpolation via dual latent space navigation. In *Computer Vision – ECCV 2020*, pages 655–672. Springer International Publishing, 2020.

[55] Olga Sorkine and Marc Alexa. As-Rigid-As-Possible Surface Modeling. In Alexander Belyaev and Michael Garland, editors, *Geometry Processing*. The Eurographics Association, 2007.

[56] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, page 62–67, Goslar, DEU, 2003. Eurographics Association.

[57] José A. Iglesias, Martin Rumpf, and Otmar Scherzer. Shape-aware matching of implicit surfaces based on thin shell energies. *Foundations of Computational Mathematics*, 18(4):891–927, June 2017.

[58] A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):19–27, 1996.

[59] Kun Zhou, Jin Huang, John Snyder, Xinguo Liu, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph.*, 24(3):496–503, jul 2005.

[60] G. Hirota, R. Maheshwari, and M.C. Lin. Fast volume-preserving free-form deformation using multi-level optimization. *Computer-Aided Design*, 32(8-9):499–512, August 2000.

[61] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. *Computer Graphics Forum*, 38, 2019.

[62] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go, 2021.

[63] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[64] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[65] B. Heeren, C. Zhang, M. Rumpf, and W. Smith. Principal geodesic analysis in the space of discrete shells. *Computer Graphics Forum*, 37(5):173–184, August 2018.

[66] Qingyang Tan, Zherong Pan, Lin Gao, and Dinesh Manocha. Realtime simulation of thin-shell deformable materials using CNN-based mesh embedding. *IEEE Robotics and Automation Letters*, 5(2):2325–2332, April 2020.

[67] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47, December 1996.

[68] T. P. Fries and Hermann G. Matthies. Classification and overview of meshfree methods. 2004.

[69] I. Hammani, S. Marrone, A. Colagrossi, G. Oger, and D. Le Touzé. Detailed study on the extension of the SPH model to multi-phase flow. *Computer Methods in Applied Mechanics and Engineering*, 368:113189, August 2020.

[70] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani. Delta SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1526–1542, March 2011.

[71] M. Muller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04*. ACM Press, 2004.

[72] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM Transactions on Graphics*, 24(3):471–478, July 2005.

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

[74] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989.

[75] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, January 1999.

[76] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[77] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Matching humans with different connectivity. *Eurographics Workshop on 3D Object Retrieval*, 2019.

[78] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics*, 37(6):1–16, December 2018.

[79] Vladimir Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4), July 2011.

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

[81] Roberto M. Dyke, Feng Zhou, Yu-Kun Lai, Paul L. Rosin, Daoliang Guo, Kun Li, Riccardo Marin, and Jingyu Yang. SHREC 2020 Track: Non-rigid shape correspondence of physically-based deformations. In Tobias Schreck, Theoharis Theoharis, Ioannis Pratikakis, Michela Spagnuolo, and Remco C. Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2020.

[82] Riccardo Marin, Simone Melzi, Emanuele Rodola, and Umberto Castellani. Farm: Functional automatic registration method for 3d human bodies. In *Computer Graphics Forum*, volume 39, pages 160–173. Wiley Online Library, 2020.

[83] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. ZoomOut. *ACM Transactions on Graphics*, 38(6):1–14, December 2019.

[84] L. Yi, Lin Shao, Manolis Savva, Haibin Huang, Yang Zhou, Qirui Wang, Benjamin Graham, Martin Engelcke, Roman Klokov, Victor S. Lempitsky, Yuan-Zhu Gan, Pengyu Wang, Kun Liu, Fenggen Yu, Panpan Shui, Bingyang Hu, Yan Zhang, Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Minki Jeong, Jaehoon Choi, Changick Kim, Angom Geetchandra, Narasimha Murthy, Bhargava Ramu, Bharadwaj Manda, M. Ramanathan, Gautam Kumar, P. Preetham, Siddharth Srivastava, Swati Bhugra, Brejesh Lall, Christian Häne, Shubham Tulsiani, Jitendra Malik, Jared Lafer, Ramsey Jones, Siyuan Li, Jie-Yan Lu, Shi Jin, Jingyi Yu, Qixing Huang, Evangelos Kalogerakis, Silvio Savarese, Pat Hanrahan, Thomas A. Funkhouser, Hao Su, and Leonidas J. Guibas. Large-scale 3d shape reconstruction and segmentation from shapenet core55. *ArXiv*, abs/1710.06104, 2017.

[85] Martin Kilian, Niloy J. Mitra, and Helmut Pottmann. Geometric modeling in shape space. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):#64, 1–8, 2007.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not see any negative societal impact of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] We do not include theoretical results; all assumptions are reported in Section 3
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.1
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details reported in the supplementary
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.1
   (b) Did you mention the license of the assets? [N/A] The involved assets are publicly available for research purpose; check the related references for further details.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The involved data are publicly available for research purpose; check the related references for further details.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The involved data are publicly available for research purpose; check the related references for further details.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]