

4Deform: Neural Surface Deformation for Robust Shape Interpolation

Lu Sang^{1,2}, Zehranaz Canfes¹, Dongliang Cao³,
 Riccardo Marin^{1,2}, Florian Bernard³, Daniel Cremers^{1,2}

¹Technical University of Munich, ²Munich Center of Machine Learning, ³University of Bonn

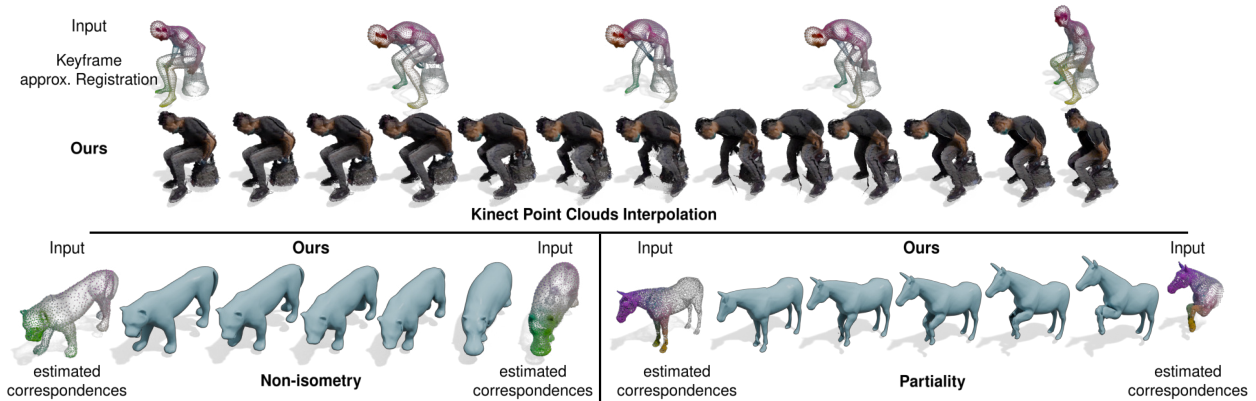


Figure 1. **4Deform** takes a sparse temporal sequence of point clouds as input and generates realistic intermediate shapes. Starting from just pairs of point clouds and estimated sparse, noisy correspondences (indicated using colors in the point clouds), our method obtains realistic long-range interpolations, even for shapes with changing topology (e.g., the human-object interaction in the top row), and can generalize the interpolation results to real-world data (Kinect point clouds in the second row). Meanwhile, our method can handle non-isometrically deformed shapes (bottom left) as well as partial shapes (bottom right).

Abstract

Generating realistic intermediate shapes between non-rigidly deformed shapes is a challenging task in computer vision, especially with unstructured data (e.g., point clouds) where temporal consistency across frames is lacking, and topologies are changing. Most interpolation methods are designed for structured data (i.e., meshes) and do not apply to real-world point clouds. In contrast, our approach, 4Deform, leverages neural implicit representation (NIR) to enable free topology changing shape deformation. Unlike previous mesh-based methods that learn vertex-based deformation fields, our method learns a continuous velocity field in Euclidean space. Thus, it is suitable for less structured data such as point clouds. Additionally, our method does not require intermediate-shape supervision during training; instead, we incorporate physical and geometrical constraints to regularize the velocity field. We reconstruct intermediate surfaces using a modified level-set equation, directly linking our NIR with the velocity field. Experiments show that our method significantly outperforms previous NIR approaches across various scenarios (e.g., noisy, partial, topology-changing, non-isometric shapes) and, for the first time, enables new applications like 4D Kinect sequence up-sampling and real-world high-resolution mesh deformation.

1. Introduction

Inferring the dynamic 3D world from just a sparse set of discrete observations is among the fundamental goals of computer vision. These observations might come, for example, from video sequences [50], Lidar scans [22, 51] or RGB-D cameras [43, 47]. Even more challenging is the recovery of plausible motion in between such observations. Despite the relevance of this problem, just a few works addressed this, likely because the solution requires merging concepts of camera-based reconstruction with techniques of 3D shape analysis and interpolation.

In the computer graphics literature, researchers have developed interpolation approaches for mesh representations [1, 21, 48]. These often require a dense, exact point-to-point correspondence between respective frames [9, 11, 21], which is usually impractical and rare in real-world applications. Also, such methods rely on a predefined topology and do not support changes (e.g., partiality, the interaction of multiple independent components). Moreover, the recovered interpolation is defined only on the mesh surface, which limits the applicability to other data forms.

The recent advent of neural implicit fields [27, 29, 44, 46] opened the door to more flexible solutions. These methods convert the start and final meshes or point clouds into im-

Methods	Corr.Est.	Non-iso.	Part.	Topo.	Seq.	Real.
SMS [9]	✓	✓	✗	✗	✓	✗
Neuromorph [21]	✓	✓	✗	✗	✓	✗
LIMP [13]	✗	✓	✗	✗	✓	✗
NFGP [55]	✗	✗	✗	✓	✗	✗
NISE [38]	-	✗	✗	✓	✗	✗
LipMLP [32]	-	✗	✗	✓	✓	✗
[45]	✗	✓	✓	✓	✗	✓
Ours	✓	✓	✓	✓	✓	✓

Table 1. **Summary of Methods Capability.** We list the capabilities of previous mesh and NIR-based methods. The column Corr. Est. indicates if the method can estimate the correspondences (✓) or needs ground-truth correspondences as input (✗); If the methods can handle non-isometric deformation (Non-iso.), partial shape deformation (Part.), topological changes (Topo.), work for Sequences (Seq.), and for the real-world data (Real.).

implicit representations and recover the intermediate frames by either latent space optimization [32, 38] or deformation modeling [45, 55]. The theoretical advantage comes from the topological flexibility of implicit representations [34]. However, the latent space-based methods generally do not consider the physical properties of the recovered intermediate shapes and therefore fail to produce reasonable interpolations such as rigid movements [32, 38]. Some other methods propose physical constraints during optimization [45, 55] but they either assume ground truth correspondences [45] or user-defined handle points [55], which makes it fail on complicated deformation or non-isometric deformation. Tab. 1 summarizes the strengths and limitations of different methods.

In this work, we address the challenging task of recovering motion between sparse keyframe shapes, relying only on coarse and incomplete correspondences. Our method begins by establishing correspondences through a matching module, followed by representing the shapes with an implicit field and modeling deformation via a velocity field. We introduce two novel loss functions to minimize distortion and stretching, ensuring physically plausible deformations. Our approach encodes shapes in a latent space, enabling both sequence representation and extrapolation of new dynamics.

For the first time, we present results that begin with imprecise correspondences obtained from standard shape matching and registration pipelines and even not always spatially aligned with the input points *e.g.*, real-world data. Our method not only outperforms state-of-the-art approaches, even when they are designed to overfit specific frame pairs, but also demonstrates versatility in real-world applications, including Kinect point cloud interpolation for human-motion interaction sequences, upsampling of real captures with partial supervision, and sequence extrapolation, as shown in Fig. 1. In summary, our contributions are:

- A data-driven framework for 3D shape interpolation that merely requires an estimated (noisy and incomplete) corre-

spondence, and for the first time demonstrates applicability to real data such as noisy and partial point clouds.

- The derivation of two losses to prevent distortion and stretching in implicit representation, promoting desirable physical properties on the interpolated sequence.
- Experimental results confirming state-of-the-art performance on shape interpolation and the applicability to challenging downstream tasks like temporal super-resolution and action extrapolation.

Our code is available on our project page <https://4deform.github.io/>.

2. Related Work

4D reconstruction from discrete observations involves recovering a continuous deformation space that not only aligns with the input data at specific time steps but also provides plausible intermediate reconstructions. The reconstructed sequences should preserve the input information while filling in any missing details absent in the original data, ultimately creating a complete and coherent representation of the observed sequence. This task involves shape deformation and interpolation.

2.1. Surface Deformation

Modeling the 3D dynamic world involves surface deformation, essential in fields like gaming, simulation, and reconstruction. Physically plausible deformations are often needed. However, the task relies heavily on the representation of the data.

Mesh Deformation. Mesh deformation typically involves directly adjusting vertex positions within a mesh pair, taking advantage of the inherent neighboring information. This allows mesh-based methods to incorporate physical constraints, such as As-Rigid-As-Possible (ARAP) [7, 48], area-preserving or elasticity loss [2] to constrain the deformation. While mesh deformation is well-studied [10, 11, 19–21], it requires predefined spatial discretization and fixed vertex connectivity to maintain topology. This leads to challenges with topology changes [9] and resolution limitations, as methods must process all vertices together. Consequently, techniques like LIMP [13] downsample meshes to 2,500 vertices, and others [9, 21, 41] re-mesh inputs to 5,000 vertices, with output resolution tied to these constraints.

Implicit Field Deformation. Unlike mesh representations, implicit surface representations offer several advantages. First, neural implicit fields allow flexible spatial resolution during inference, as discretization isn’t pre-fixed. Second, they can represent arbitrary topologies, making them well-suited for complex cases that mesh-based methods struggle with, such as noisy or partial observations. However, directly deforming an implicitly represented surface is challenging because surface properties aren’t explicitly stored, limiting direct manipulation of surface points. This area remains

underexplored, with only a few approaches addressing it. For instance, NFGP [55] introduces a deformation field on the top of an implicit field, constraining it physically using user-defined handle points to match target shapes. This pioneering work directly deforms the implicit field; however, its practical applications are limited as it only provides the starting and ending shapes, with processing times for a single shape pair extending over several hours. Other methods, such as those in [37, 38], focus on shape smoothing or morphing without targeting specific shapes. The work [45] introduced a fast, flexible framework for directly deforming the implicit field, capable of generating physically plausible intermediate shapes. However, as an optimization-based approach, it requires training separately for each shape pair and struggles with handling large deformations.

2.2. Shape Interpolation

Shape interpolation is a subset of shape deformation that involves generating intermediate shapes between a start and target shape. The interpolated sequence should reflect a physically meaningful progression, making it essential that the interpolated shapes are not only geometrically accurate but also serve to complete the narrative implied by the initial and final shapes. Therefore, we emphasize the concept of **physically plausible** shape interpolation, which should be a guiding principle for tasks in this area. There are two main approaches: generative models and physics-based methods. Generative models [25, 28] rely on extensive datasets to produce shapes, but their outputs can lack realism due to data dependency. Physics-based methods, like [32, 38, 45], optimize over a shape pair to generate intermediates but may have limited applications. Another way is to learn a deformation space of shapes, such as a latent space [26], and hope that generating intermediate shapes is equivalent to interpolating the latent shape [13]. However, such methods suffer from the same problem as generative models: the latent space interpolation may be far from physically plausible. To address these limitations, we propose a lightweight solution that can be trained on datasets of any size. Our model adopts an AutoDecoder architecture to maintain generative capability and combines physical and geometric constraints to ensure the generated results are physically plausible.

3. From Implicit Surface Deformation ...

Implicit representation of the moving surface. We represent the moving surface $\mathcal{S}_t$ in the volume $\Omega \subset \mathbb{R}^3$ implicitly as the zero-crossing of a time-evolving signed distance function $\phi : \Omega \times [0, T] \rightarrow \mathbb{R}$:

$$\mathcal{S}_t = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}, t) = 0\}. \quad (1)$$

This implies that

$$\phi(\mathcal{S}_t, t) = 0 \quad \forall t. \quad (2)$$

It follows that the total time derivative is zero, i.e.

$$\frac{d}{dt}\phi(\mathbf{x}, t) = \partial_t \phi + \mathcal{V}^\top \nabla \phi = 0, \quad (3)$$

where $\mathcal{V} = \frac{d}{dt}\mathbf{x}$ denotes the velocity of the moving surface. Eq. (3) is known as the *level-set equation* [14, 15]. It tells us how to move the implicitly represented surface $\mathcal{S}_t$ along the vector field $\mathcal{V}$ by deforming the level-set function ϕ . Since over time, the level-set function will generally lose its property of being a signed distance function, we add an Eikonal regularizer with a weight λ_l to obtain the modified level-set equation [45], i.e.

$$\partial_t \phi + \mathcal{V}^\top \nabla \phi = -\lambda_l \phi \mathcal{R}(\mathbf{x}, t) \quad \text{in } \Omega \times \mathcal{I}, \quad (4)$$

where $\mathcal{R}(\mathbf{x}, t) = -\langle (\nabla \mathcal{V}) \frac{\nabla \phi}{\|\nabla \phi\|}, \frac{\nabla \phi}{\|\nabla \phi\|} \rangle$. This equation combines the level-set equation and Eikonal constraint, freeing the level-set approach from the reinitialization process [6, 23, 45].

Spatial Smoothness and Volume Preservation. To make sure the velocity field models physical movement, we can impose respective regularizers on the velocity field. Two popular regularizers are spatial smoothness $\mathcal{L}_s$ [18, 45] and volume preservation $\mathcal{L}_v$ [13, 20]:

$$\begin{aligned} \mathcal{L}_s &= \int_{\Omega} \|(-\alpha \Delta + \gamma \mathbf{I})\mathcal{V}(\mathbf{x}, t)\|_{l_2} d\mathbf{x}, \\ \mathcal{L}_v &= \int_{\Omega} |\nabla \cdot \mathcal{V}(\mathbf{x}, t)| d\mathbf{x}. \end{aligned} \quad (5)$$

4. ... to Neural Implicit Surface Deformation

Previous implicit or point cloud deformation methods either rely on ground truth correspondences, struggle with large deformations [45], or are limited to shape pairs [32, 38, 45], restricting their applicability. To overcome these limitations, we propose a method that first incorporates a corresponding block to obtain sparse correspondences, then handles larger deformations by establishing stronger physical constraints, and extends functionality to *temporal sequence* of point clouds via an AutoDecoder architecture [26]. Given point clouds sequence $\{\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_n, \dots\}$ with an initialized latent vector to each point cloud $\{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_n, \dots\}$, where $\mathcal{P}_k = \{\mathbf{x}_i^k\}_i \subset \mathbb{R}^3$ and $\mathbf{z}_i \in \mathbb{R}^m$ is a trainable latent vector that is assigned to each input point cloud. We pair the inputs as a source point cloud and a target point cloud (for convenience we label them as $\mathcal{P}_0$ and $\mathcal{P}_1$). We aim to generate physically plausible intermediate stages of all training pairs accordingly. To this end, we propose to model the 4D movements using a time-varying implicit neural field of the form:

$$\mathcal{S}_t = \{\mathbf{x} \mid \phi_{\mathbf{z}}(\mathbf{x}, t) = 0\}, \text{ for } \mathbf{z} := \mathbf{z}_0 \oplus \mathbf{z}_1. \quad (6)$$

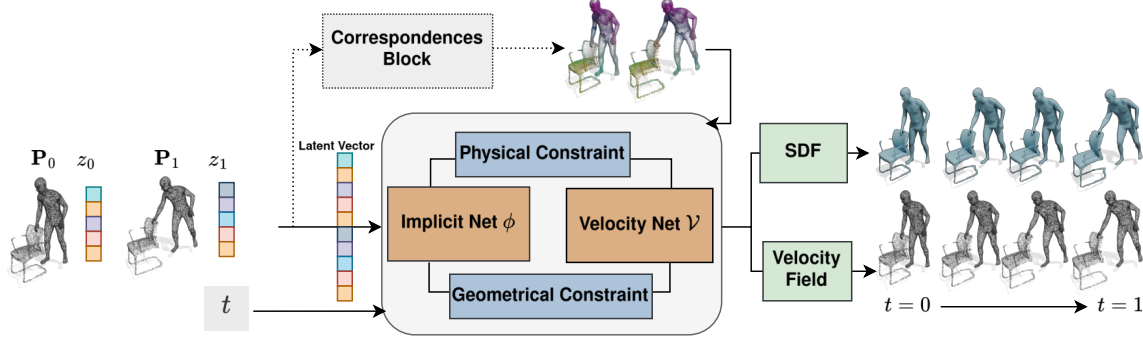


Figure 2. **Pipeline of 4Deform:** Given a temporal sequence of inputs, we initialize a latent vector to each point cloud. Then the network takes pairs of point clouds $\mathcal{P}_0$ and $\mathcal{P}_1$ (with sparse correspondences), together with the concatenated latent vector $\mathbf{z}_0$ and $\mathbf{z}_1$ as input. At training time, we jointly optimize two neural fields: a time-varying implicit representation (Implicit Net ϕ) and a velocity field (Velocity Net $\mathcal{V}$) with proposed geometric and physical constraints losses. Conditioning on a time stamp t , we instantaneously obtain a continuous time-varying signed distance function (SDF), an offset of the input toward the target (velocity field).

Each shape $\mathcal{S}_t$ in time t is encoded by the zero-crossing of the implicit function ϕ . Particularly, $\mathcal{S}_0$ and $\mathcal{S}_1$ should coincide with $\mathcal{P}_0$ and $\mathcal{P}_1$.

4.1. Correspondence Block

Instead of relying on ground-truth correspondences, which are difficult to obtain in real-world settings, our method obtains the correspondences based on the state-of-the-art unsupervised non-rigid 3D shape matching method [8]. This method is based on the functional map framework [39] and follow-up learning-based approaches [17, 42]. The key ingredient of the functional map framework is that, instead of directly finding correspondences in the spatial domain, it encodes the correspondences in the compact spectral domain and thus is robust to large deformation [8]. More details about functional maps can be found in this lecture note [40]. It is worth noting that our method is agnostic to the choice of shape-matching methods. For instance, for specific types of input shapes (e.g. humans), specialized registration methods can also be utilized to obtain more accurate correspondences [3, 35, 36].

4.2. Implicit and Velocity Fields

To ensure the physically plausible intermediate stage, we model the deformation from the source point cloud to the target point cloud by tracking the point cloud path using the velocity field $\mathcal{V}_{\mathbf{z}}(\mathbf{x}, t) \in \mathbb{R}^3$. We adopt the velocity field for two reasons, (i) The velocity field allows us to control the generated deformation. It is easy to add physical-based constraints directly on velocity to force the intermediate movement to follow certain physical laws. (ii) There is a link from the velocity field to the implicit field as the implicit field can be seen as a macroscopic field. We can perform the material deviation to derive the natural relationship between velocity field $\mathcal{V}_{\mathbf{z}}(\mathbf{x}, t)$ and implicit field $\phi_{\mathbf{z}}(\mathbf{x}, t)$. For simplicity, we omit the latent vector $\mathbf{z}$ in the following equations.

We follow [45] to use the Modified Level Set equation to link the velocity field and implicit field via level-set loss

$$\mathcal{L}_i = \int_{\Omega} \|\partial_t \phi + \mathcal{V} \cdot \nabla \phi + \lambda_l \phi \mathcal{R}(\mathbf{x}, t)\|_{l^2} d\mathbf{x}. \quad (7)$$

Since the surface normal $\mathbf{n}$ coincides as implicit function gradient $\nabla \phi$, this loss offers *geometry constraint* to Velocity Net, and as the point is moved by Velocity Net, Eq. (7) also works as a *physical constraint* from Velocity Net to Implicit Net, as shown in Fig. 2. We force the Velocity Net to move the point with known correspondences to the target input position by

$$\mathcal{L}_m = \int_{\Omega^*} \left\| \mathbf{x}^0 + \int_0^1 \mathcal{V}(\mathbf{x}, \tau) d\tau - \mathbf{x}^1 \right\|_{l^2} d\mathbf{x}, \quad (8)$$

where Ω^* only contains points with known correspondences.

4.3. Physical Deformation Loss

A key challenge in deforming implicit neural fields with external velocity fields is ensuring physically realistic movement, as commonly used constraints like as-rigid-as-possible (ARAP) [48] are difficult to enforce without a connectivity structure. To address this, we introduce new physical deformation losses on both the implicit and velocity fields to better control movement. These losses do not require connectivity information, thus it can be enforced on unordered input and allow arbitrary resolution of input.

Distortion Loss. In the Eulerian perspective of continuum mechanics, the rate of deformation tensor provides a measure of how the fluid or solid material deforms over time from a fixed reference point in space, excluding rigid body rotations. It measures the rate of stretching, compression, and shear that a material element undergoes as it moves through the flow field [49]. The rate of deformation tensor is defined by

$$\mathbf{D} = \frac{1}{2}(\nabla \mathcal{V} + (\nabla \mathcal{V})^T). \quad (9)$$

The distortion of the particle moved under the velocity $\mathcal{V}$ can be described by the deviatoric form

$$\mathcal{L}_d = \int_{\Omega} \left\| \frac{1}{6} \text{Tr}(\mathbf{D})^2 - \frac{1}{2} \text{Tr}(\mathbf{D} \cdot \mathbf{D})^2 \right\|_F dx. \quad (10)$$

Eq. (10) is the complement of the volumetric changes. This term removes the volumetric part, leaving behind the deviatoric (distortional) component. It offers *physical constraint* to both networks during training.

Stretching Loss. By tracking point cloud movement with a velocity field in our approach, we can establish constraints to control surface stretching along the deformation. We follow the idea of strain tensor from continuum mechanics [49]. Consider deformation happens in infinitesimal time Δt , the displacement of point $\mathbf{x}$ is moved to $\mathbf{x}'$ such that

$$\mathbf{x}' = \mathbf{x} + \mathcal{V}(\mathbf{x}, t) \Delta t. \quad (11)$$

Consider a infinitesimal neighbourhood of $\mathbf{x}'$, denote as $d\mathbf{x}'$, the length of it is given by $(ds')^2 = d\mathbf{x}'^\top d\mathbf{x}'$. Similarly, the length of infinitesimal neighborhood of $\mathbf{x}$ is given by $(ds)^2 = d\mathbf{x}^\top d\mathbf{x}$. Together with Eq. (11), the stretched length after deformation is given by

$$(ds')^2 - (ds)^2 = d\mathbf{x}^\top (\mathbf{F}^\top \mathbf{F} - \mathbf{I}) d\mathbf{x}, \quad (12)$$

where $\mathbf{F} = \partial \mathbf{x}' / \partial \mathbf{x} = \mathbf{I} + \nabla \mathcal{V}$. However, instead of considering the stretch on the neighborhood patch of one surface point, preventing stretching on the tangent plane of a point is what makes a deformation physically realistic [55]. We project $d\mathbf{x}$ in Eq. (12) to its tangent space using projection operator $\mathbf{P}(\mathbf{x}) = \mathbf{I} - \mathbf{n}(\mathbf{x}) \mathbf{n}(\mathbf{x})^\top$ to compute the stretching on the tangent plane, where $\mathbf{n}(\mathbf{x})$ is the normal vector on point $\mathbf{x}$. Thus, the stretching on the tangent plane is

$$(dl)^2 = d\mathbf{x}^\top \mathbf{P}^\top(\mathbf{x}) (\mathbf{F}^\top \mathbf{F} - \mathbf{I}) \mathbf{P}(\mathbf{x}) d\mathbf{x}. \quad (13)$$

Finally, thanks to the nice properties of the implicit field, we have $\mathbf{n}(\mathbf{x}) = \frac{\nabla \phi(\mathbf{x}, t)}{\|\nabla \phi(\mathbf{x}, t)\|}$. Therefore, for any $d\mathbf{x}$, we constraint the matrix Frobenius norm as

$$\mathcal{L}_{st} = \int_{\Omega} \left\| \mathbf{P}^\top (\nabla \mathcal{V}^\top \nabla \mathcal{V} + \nabla \mathcal{V} + \nabla \mathcal{V}^\top) \mathbf{P} \right\|_F dx \quad (14)$$

where $\mathbf{P} = \mathbf{I} - \nabla \phi \nabla \phi^\top$.

4.4. Training and Inference

Training. Given a temporal sequence of inputs $\{\mathcal{P}_k\}_k$, which may be point clouds or meshes, we start by using our correspondence blocks to obtain the correspondences of each training pair. Importantly, our method does not require full correspondence for every training point; it only requires a subsample of points. During training, we initialize a trainable latent vector for each shape. We concatenate the

latent vectors of each training pair and optimize them using our Implicit Net ϕ and Velocity Net $\mathcal{V}$ jointly. We sample $T+1$ discrete time steps uniformly for $t \in \{0, 1/T, \dots, 1\}$ to compute the loss at each time step. The total loss is

$$\mathcal{L} = \lambda_i \mathcal{L}_i + \lambda_s \mathcal{L}_s + \lambda_v \mathcal{L}_v + \lambda_{st} \mathcal{L}_{st} + \lambda_m \mathcal{L}_m, \quad (15)$$

where $\lambda_i, \lambda_s, \lambda_v, \lambda_{st}$ and λ_m are weights for each loss term. For further details about implementation and training, we refer to supplementary materials.

Inference. During inference, we give the optimized latent vector for each trained pair into the Implicit Net ϕ to generate intermediate shapes at different discrete time steps t . Given an initial point cloud, we pass it with the optimized latent vector to the Velocity Net $\mathcal{V}$, producing a sequence of deformed points at each time step.

5. Experimental Results

Datasets. We validate our method on a number of data from different shape categories. We considered registered human shapes from **FAUST** [5], non-isometric four-legged animals from **SMAL** [56], and partial shapes from **SHREC16** [12]. The input shapes are roughly aligned and we train our correspondence block on each one of them individually [8]. These datasets do not include temporal sequences, so we train on all possible pair combinations. We also evaluate our method on real-world scans, using motion sequences scans of clothed humans from **4D-DRESS** [54], and noisy Kinect acquisitions of human-object interactions from **BeHave** [4]. For both cases, correspondences are obtained by template shape registration. Notably, the obtained correspondences are rough estimations, often imprecise, and thus do not guarantee continuous bijective maps between shapes, *e.g.*, due to garments, occlusions, or noise in the acquisitions.

Baseline methods. We compare our method against recent NIR-based methods that solve similar problems. LipMLP [32] encourages smoothness in the pair-wise interpolation by relying on Lipschitz constraints; NISE [38], similar to us, relies on the level-set equation and uses pre-defined paths to interpolate between neural implicit surfaces. NFGP [55] relies on a user-defined set of points as handles, together with rotation and translation parameters. We also consider [45] as the most relevant baseline method. Nevertheless, all these methods are tailored for shape pairs. To this end, to evaluate the performance of our method in the context of shape sequences, we compare our method to LIMP [13], a mesh-based approach that constructs a latent space and preserves geometric metrics during interpolations.

Training time. We train all methods on a commodity GeForce GTX TITAN X GPU. Our method needs approximately 8 to 10 minutes per pair, *i.e.*, for a sequence with 10 pairs, our method takes roughly 1.5 hours. The work [45] and LipMLP [32] require similar runtimes as our method

when training one pair. NISE [38] takes around 2 hours for each pair. LIMP [13] first needs to downsample mesh to 2,500 vertices and training time takes around 30 minutes per pair. NFGP [55] requires training separately for each step and each step takes 8 to 10 hours, which makes the training time go up to 40 hours for recovering 5 intermediate shapes.

Evaluation Protocol. We compare three main settings. First, we train on a single registered shape pair (S) and test the interpolation quality (**Pairs S/S**). Second, we consider the case of training on registered shape sequences and test the interpolation quality (**Seq. S/S**), for which we rely on similar metrics. Finally, we consider training on registered shape pairs but *testing on Real point clouds (R)* (**Pairs S/R**). As metrics, we consider the standard deviation of surface area ($SA\sigma = \sqrt{\sum_{t=0}^N (A_t - \bar{A})^2 / N}$, where A_t is the surface area for mesh at time t and $\bar{A}$ is the average surface area over the interpolated meshes), which is expected to be close to 0 for the isometric cases. When we have access to ground truth for the intermediate frames, we also report the Chamfer Distance (CD) and the Hausdorff Distance (HD) errors of the predictions. For the Pairs S/R setting, we also report the pointwise root-mean-square error (P-RMSE), which indicates the Euclidean distance of deformed mesh vertices to the ground truth mesh vertices. In the case a method is not applicable to a certain setting, we note that with a cross (X).

5.1. Isometric Shape Interpolation

Quantitative comparison. We show a quantitative comparison of isometric human shapes from 4D-Dress for the three settings in Table 2. We chose this dataset since the frequency of the scans lets us have ground truth for the intermediate frames, as well as access to real scans. Despite competitors being tailored for the Pairs S/S setting, our method performs on par, with better area preservation. Our method also supports sequences, contrary to the majority of previous methods. LIMP fails to generate intermediate shapes that are faithful to the ground truth. We believe that LIMP’s poor performance is caused by its strong data demand that is not fulfilled here. To prove our generalization, we also show results on the interpolation of SMAL isometric animals in Table 3. The ground-truth evaluation frames are obtained by interpolation of SMAL pose parameters. Our method outperforms the competitors on all the metrics. We report all qualitative results in supplementary materials.

Large Deformations. A major advantage of our method is that we support large deformations. We show an example of this between two FAUST shapes in Figure 3. Although isometric, their drastic change of limbs constitutes a challenge for purely extrinsic methods, causing evident artifacts. Our method preserves areas an order of magnitude better than mesh-based methods (LIMP).

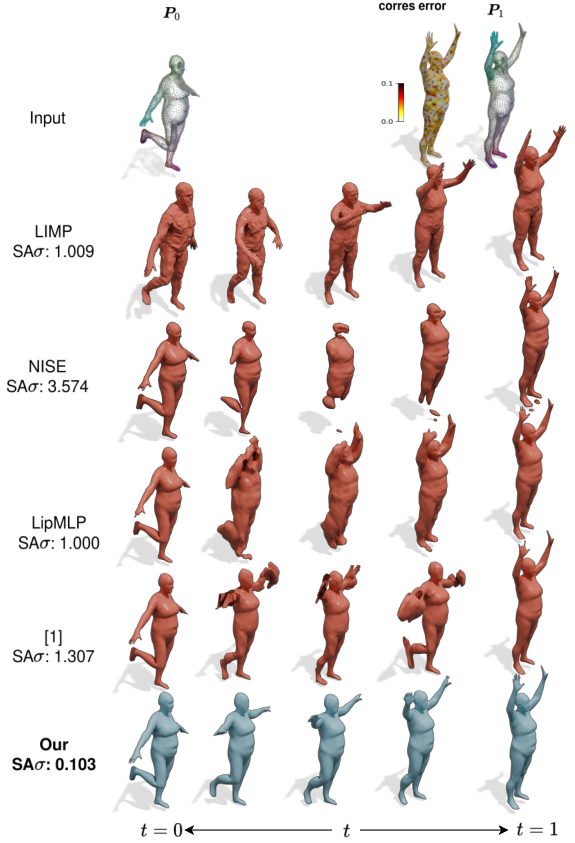


Figure 3. **Large Deformations.** 4Ddeform handles large deformations better than previous works, providing one order of magnitude less area distortion, even compared to mesh-based ones (LIMP [13]). In the top row, we visualize the error in the estimated input correspondence.

5.2. Non-isometry and Partiality

Non-isometry. A significant challenge is modeling interpolation when the shape metric is drastically changing between frames. This significantly hampers the chances of obtaining reliable correspondence and control over the full-shape geometry. In Fig. 4, we show an interpolation between a cougar and a cow from SMAL. As can be seen on top of the image, the correspondence error is quite noisy. As a consequence, our method is the only one that shows consistency also in the thin geometry (e.g., legs). We argue this is a direct contribution of our losses Eq. (10) and Eq. (14). Due to the lack of ground truth intermediate shapes, we only report $SA\sigma$ for each method’s results.

Partiality. Finally, an extremely challenging case are partially observed shapes. Here, an ideal interpolation would provide a smooth interpolation of the overlapping part, while keeping the non-overlapping part as consistent and rigid as possible. We provide an example of a cat from SHREC16 in Fig. 5. Despite the highly imprecise correspondence, we see that our method is the one with the best preservation of

	Pairs S/S			Seq. S/S			Pairs S/R		
	CD ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	SA σ ($\times 10$) ↓	CD ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	SA σ ($\times 10$) ↓	CD ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	P-RMSE ($\times 10$) ↓
LIMP [13]	21.980	3.175	0.507	136.787	15.974	0.155	$\times$	$\times$	$\times$
NFGP [55]	0.272	0.025	0.075	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
LipMLP [32]	14.99	2.125	1.252	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
NISE [38]	6.588	2.167	0.321	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
[45]	0.279	0.047	0.023	$\times$	$\times$	$\times$	0.548	0.083	0.024
Ours	0.269	0.046	0.018	0.327	0.038	0.063	0.390	0.101	0.014

Table 2. **Human dataset isometric deformation metrics.** We evaluate our method on both pairs input (Pairs S/S) and sequence input (Seq. S/S) and compare against previous methods that either only work for temporal sequences or pairs. Additionally, as our method can directly operate on real-world data that is not trained on, we also report the error w.r.t. to real-world results (see Sec. 5.4) on (Seq. S/R). For the methods that cannot directly deform real-world data, thus the Pairs S/R columns are marked as $\times$.

	Pairs S/S			
	CD ($\times 10^3$) ↓	HD ($\times 10^2$) ↓	SA σ ($\times 10$) ↓	P-RMSE ($\times 10$) ↓
NFGP [55]	0.770	0.906	0.217	$\times$
LipMLP [32]	68.452	43.327	1.192	$\times$
NISE [38]	7.223	1.237	0.771	$\times$
[45]	0.173	0.626	0.064	0.081
Ours	0.137	0.221	0.062	0.061

Table 3. **Animal dataset isometric deformation metrics.** We show that our method achieves the best results on the SMAL dataset.

the absent area and, overall, the smallest area distortion. For both non-isometric and partial shapes, our method provides more realistic deformations than [45].

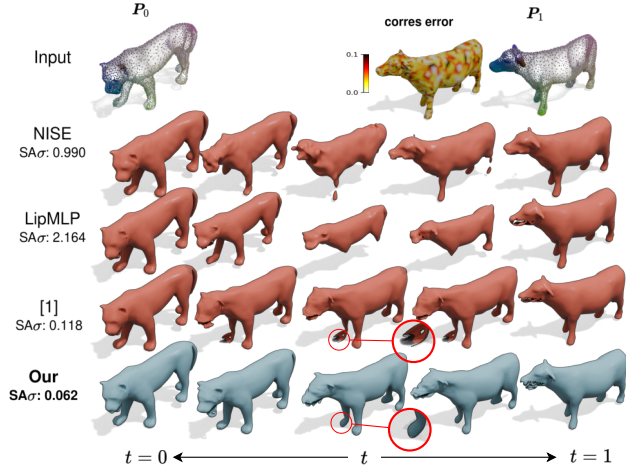


Figure 4. **Non-isometric deformation.** We deform two different animals from SMAL, relying on a noisy correspondence (top row). Compared to the previous methods, our method results in plausible deformations, while preserving thin geometric details (e.g., legs).

5.3. Ablation Study

To demonstrate the effectiveness of our distortion loss $\mathcal{L}_d$ and stretching loss $\mathcal{L}_{st}$, we perform a quantitative comparison on the 4D-Dress dataset. We report quantitative evaluation in Tab. 4. We highlight that although the losses have a minor influence in the Pair S/S case, they are useful in providing

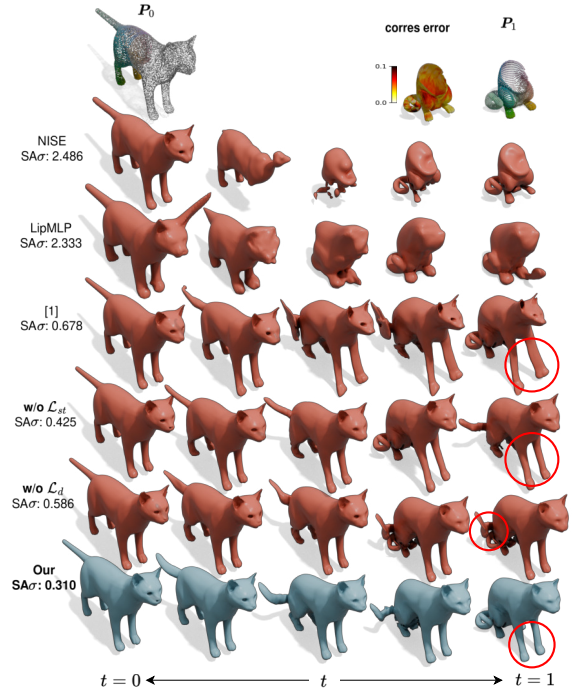


Figure 5. **Partial shape deformation.** We consider the case in which one of the input shapes is only partially available while having noisy correspondences (correspondence error visualized in the top row). Other methods often collapse the unseen part or create unreasonable stretches. Similar effects are observed when we remove some of our novel losses. Our method provides plausible interpolations, both for the visible and missing parts.

consistency when the network has to capture relations on a wider set of shapes (Seq. S/S); further, they show robustness in the presence of real noise (Pairs S/R). This follows our intuition that such losses serve as regularization, especially in the more challenging cases. This is further highlighted by the qualitative results of partial shapes of Fig. 5.

5.4. Applications

Our method enables a series of new applications, such as the upsampling of real captures and the handling of noisy point clouds. We also show that the learned networks are capable

	Pairs S/S			Seq. S/S			Pairs S/R		
	CD ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	SA σ ($\times 10$) ↓	CD ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	SA σ ($\times 10$) ↓	CD ($\times 10^4$) ↓	HD ($\times 10^2$) ↓	P-RMSE ($\times 10$) ↓
w/o $\mathcal{L}_d$	0.265	0.041	0.115	0.348	0.101	0.065	0.490	0.233	0.023
w/o $\mathcal{L}_{st}$	0.284	0.045	0.018	0.363	0.091	0.062	0.620	0.126	0.023
w/o both	0.279	0.047	0.023	0.390	0.084	0.065	0.548	0.083	0.024
Ours	0.269	0.046	0.018	0.327	0.038	0.063	0.390	0.101	0.014

Table 4. **Loss ablation.** We ablate our method on pair setting (Paris S/S) and temporal sequences setting (Seq. S/S) and report the quantitative measurements. We also report the error on the real-world mesh CD and HD in Pairs (S/R) columns.

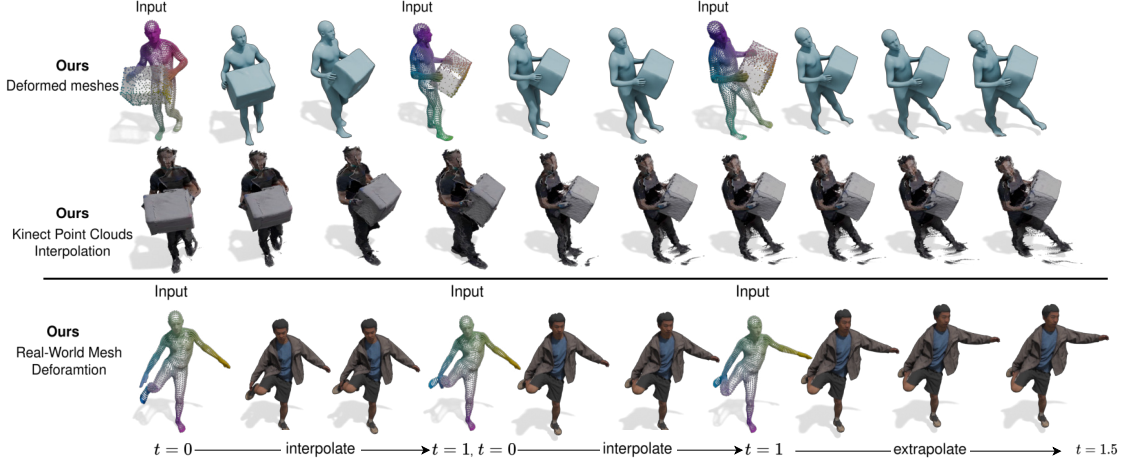


Figure 6. **Upsampling and extrapolation.** The top shows an example of the BeHave sequence. Starting from a sparse set of keyframes (1fps, colored point clouds), our method lets us interpolate the registration (first row), as well as the real Kinect point clouds (second row) between keyframes at an arbitrary continuous resolution. On the bottom, we show extrapolation on a 4D-Dress sequence. With just a few key frames, we can deform the real point cloud even beyond the final frame, obtaining an estimation of the plausible continuation of the action.

of generalizing and extrapolating in the time domain.

Temporal Upsampling. In many real-world datasets, human movements are recorded by sensors such as RGBD cameras, which provide real-world point clouds of humans and possible object interactions. However, due to technical constraints or device setup differences, human motion datasets have different frame rates for recording the movement [4, 52, 54]. This is the case for BeHave [4], where the input Kinect sequences are carefully annotated with significant manual intervention to align SMPL and an object template to the input, resulting in a frame rate of only 1 FPS. We show that our interpolation method can efficiently up-sample not only the annotated SMPL data, but also the noisy real-world Kinect point clouds without additional effort. We highlight that our Velocity Net can easily be generalized to untrained real-world point clouds to obtain upsampling sequences. We show the upsampling results in Fig. 6

Real-World Data Deformation. Here, we show another application scenario in high-quality meshes. High-quality data with tens of vertices, defining a dense, precise point-to-point correspondence is demanding and often unfeasible to be processed. Therefore, it is a common practice to fit a template to such input high-quality data and use it as rough guidance. In Fig. 6, we show how, from just a few frames equipped with a rough SMPL alignment, we can manipulate

a 40k vertices real scan, maintaining its structure along all the sequences. We refer to Tab. 2 as an evaluation of our method’s performance both on the real-world and SMPL interpolation.

Extrapolation for Movement Generation. Our method lets us obtain dense intermediate frames at arbitrary temporal resolution and allows us to deform the data that is a bit far from the sparse input correspondence. Moreover, the learned physical deformation allows us to generate movements even beyond the considered sequence. Our velocity field can extrapolate outside of the training time domain (0 to 1), while remaining physically plausible, as shown in Fig. 6.

6. Discussion and Future Work

While our method integrates physical plausibility, certain types of deformations, such as mechanical joints and fluid dynamics, may not yet be fully captured by our model. Future work could incorporate additional physical constraints to address these complexities. Additionally, some applications require separate deformation estimation, as in the case of a human with loose clothing, where the deformations of the body and clothing do not align. We plan to extend our work to handle these cases in future developments.

Acknowledgements

This project is supported by ERC Advanced Grant “SIMULACRON” (agreement #884679), GNI Project “AI4Twinning”, and DFG project CR 250/26-1 “4D YouTube”. We also would like to thank Weirong Chen for helpful insights and proof reading.

References

- [1] Seung-Yeob Baek, Jeonghun Lim, and Kunwoo Lee. Isometric shape interpolation. *Computers & Graphics*, 46:257–263, 2015. [1](#)
- [2] Lennart Bastian, Yizheng Xie, Nassir Navab, and Zorah Löhner. Hybrid functional maps for crease-aware non-isometric shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3313–3323, 2024. [2](#)
- [3] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. *Advances in Neural Information Processing Systems*, 33:12909–12922, 2020. [4](#)
- [4] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. [5](#), [8](#), [2](#), [4](#)
- [5] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, 2014. [5](#)
- [6] Dieter Bothe, Mathis Fricke, and Kohei Soga. Mathematical analysis of modified level-set equations. *Mathematische Annalen*, pages 1–41, 2024. [3](#)
- [7] Mario Botsch, Mark Pauly, Markus H Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, 2006. [2](#)
- [8] Dongliang Cao, Paul Roetzer, and Florian Bernard. Unsupervised learning of robust spectral shape matching. *ACM Transactions on Graphics*, 42(4):1–15, 2023. [4](#), [5](#)
- [9] Dongliang Cao, Marvin Eisenberger, Nafie El Amrani, Daniel Cremers, and Florian Bernard. Spectral meets spatial: Harmonising 3d shape matching and interpolation. In *CVPR*, 2024. [1](#), [2](#)
- [10] Dongliang Cao, Paul Roetzer, and Florian Bernard. Revisiting map relations for unsupervised non-rigid shape matching. In *International Conference on 3D Vision (3DV)*, 2024. [2](#)
- [11] Wei Cao, Chang Luo, Biao Zhang, Matthias Nießner, and Jiapeng Tang. Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking. In *CVPR*, 2024. [1](#), [2](#)
- [12] Luca Cosmo, Emanuele Rodola, Michael M Bronstein, Andrea Torsello, Daniel Cremers, Y Sahillioğlu, et al. Shrec’16: Partial matching of deformable shapes. In *Eurographics Workshop on 3D Object Retrieval, EG 3DOR*, 2016. [5](#)
- [13] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. *LIMP: Learning Latent Shape Representations with Metric Preservation Priors*, page 19–35. Springer International Publishing, 2020. [2](#), [3](#), [5](#), [6](#), [7](#), [1](#)
- [14] A. Dervieux and F. Thomasset. A finite element method for the simulation of Raleigh-Taylor instability. *Springer Lect. Notes in Math.*, 771:145–158, 1979. [3](#)
- [15] A. Dervieux and F. Thomasset. Multifluid incompressible flows by a finite element method. *Lecture Notes in Physics*, 11:158–163, 1981. [3](#)
- [16] George Ellwood Dieter and David Bacon. *Mechanical metallurgy*. McGraw-hill New York, 1976. [1](#)
- [17] 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. [4](#)
- [18] Paul Dupuis, Ulf Grenander, and Michael I Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of applied mathematics*, pages 587–600, 1998. [3](#)
- [19] M. Eisenberger and D. Cremers. Hamiltonian dynamics for real-world shape interpolation. In *ECCV*, 2020. [2](#)
- [20] Marvin Eisenberger, Zorah Löhner, and Daniel Cremers. Divergence-free shape interpolation and correspondence. *arXiv preprint arXiv:1806.10417*, 2018. [3](#)
- [21] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *CVPR*, 2021. [1](#), [2](#)
- [22] Gianfranco Forlani, Carla Nardinocchi, Marco Scaioni, and Primo Zingaretti. C omplete classification of raw lidar data and 3d reconstruction of buildings. *Pattern analysis and applications*, 8:357–374, 2006. [1](#)
- [23] Mathis Fricke, Tomislav Marić, Aleksandar Vučković, Ilia Roisman, and Dieter Bothe. A locally signed-distance preserving level set method (sdpls) for moving interfaces. *arXiv preprint arXiv:2208.01269*, 2022. [3](#)
- [24] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. [1](#)
- [25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. [3](#)
- [26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. [3](#)
- [27] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020. [1](#)
- [28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. [3](#)
- [29] L Härenstam-Nielsen, L Sang, A Saroha, N Araslanov, and D Cremers. Diffcd: A symmetric differentiable chamfer distance for neural implicit surface fitting. In *European Conference on Computer Vision (ECCV)*, 2024. [1](#)

- [30] Fridtjov Irgens. *Continuum Mechanics in Curvilinear Coordinates*, pages 599–624. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. 1
- [31] A Kaye, RFT Stepto, WJ Work, JV Aleman, and A Ya Malkin. Definition of terms relating to the non-ultimate mechanical properties of polymers (recommendations 1998). *Pure and applied chemistry*, 70(3):701–754, 1998. 1
- [32] Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–13, 2022. 2, 3, 5, 7, 1
- [33] 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, 2015. 6
- [34] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2): 158–175, 1995. 2
- [35] Riccardo Marin, Simone Melzi, Emanuele Rodola, and Umberto Castellani. Farm: Functional automatic registration method for 3d human bodies. In *Computer Graphics Forum*, pages 160–173. Wiley Online Library, 2020. 4
- [36] Riccardo Marin, Enric Corona, and Gerard Pons-Moll. Nicp: Neural icp for 3d human registration at scale. In *European Conference on Computer Vision*, 2024. 4
- [37] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A level set theory for neural implicit evolution under explicit flows. In *ECCV*, 2022. 3
- [38] Tiago Novello, Vinícius da Silva, Guilherme Schardong, Luiz Schirmer, Hélio Lopes, and Luiz Velho. Neural implicit surface evolution. In *ICCV*, 2023. 2, 3, 5, 6, 7
- [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):1–11, 2012. 4
- [40] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, pages 1–60. 2016. 4
- [41] Paul Roetzer, Ahmed Abbas, Dongliang Cao, Florian Bernard, and Paul Swoboda. Discomatch: Fast discrete optimisation for geometrically consistent 3d shape matching. In *Proceedings of the European conference on computer vision (ECCV)*, 2024. 2
- [42] 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. 4
- [43] L Sang, B Haefner, X Zuo, and D Cremers. High-quality rgb-d reconstruction via multi-view uncalibrated photometric stereo and gradient-sdf. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Hawaii, USA, 2023. 1
- [44] L Sang, A Saroha, M Gao, and D Cremers. Weight-aware implicit geometry reconstruction with curvature-guided sampling. *arXiv preprint arXiv:2306.02099*, 2023. 1
- [45] L Sang, Z Canfes, D Cao, F Bernard, and D Cremers. Implicit neural surface deformation with explicit velocity fields. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 3, 4, 5, 7, 1
- [46] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 1
- [47] C Sommer, L Sang, D Schubert, and D Cremers. Gradient-SDF: A semi-implicit surface representation for 3d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1
- [48] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, pages 109–116. Citeseer, 2007. 1, 2, 4
- [49] Anthony James Merrill Spencer. *Continuum mechanics*. Courier Corporation, 2004. 4, 5
- [50] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. *CVPR*, 2021. 1
- [51] Julian Tachella, Yoann Altmann, Ximing Ren, Aongus McCarthy, Gerald S Buller, Stephen McLaughlin, and Jean-Yves Tourneret. Bayesian 3d reconstruction of complex scenes from single-photon lidar data. *SIAM Journal on Imaging Sciences*, 12(1):521–550, 2019. 1
- [52] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. 8
- [53] Stephen P Timoshenko. *Strength of Materials: Part II Advanced Theory and Problems*. D. Van Nostrand, 1956. 1
- [54] Wenbo Wang, Hsuan-I Ho, Chen Guo, Boxiang Rong, Artur Grigorev, Jie Song, Juan Jose Zarate, and Otmar Hilliges. 4d-dress: A 4d dataset of real-world human clothing with semantic annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5, 8, 2, 6
- [55] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. In *NeurIPS*, 2021. 2, 3, 5, 6, 7, 1
- [56] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017. 5, 2, 3