4D Recons: Monocular Dynamic Reconstruction with Geometrical and Topological Regularization

Xiaoyan Cong[1], Haitao Yang[2], Liyan Chen[2], Kaifeng Zhang[4], Li Yi[3], Chandrajit Bajaj[2], Qixing Huang[2]

[1]Brown University [2]The University of Texas at Austin [3]Tsinghua University [4]Columbia University

Abstract

This paper presents a novel approach 4DRecons that takes a monocular RGB-D sequence of a dynamic subject as input and outputs a complete textured deforming 3D model over time. 4DRecons encodes the output as a 4D neural implicit surface and presents an optimization procedure that combines a data term and two regularization terms. The data term fits the 4D implicit surface to the input partial observations. We address fundamental challenges in fitting a complete implicit surface to partial observations. The first regularization term enforces that the deformation among adjacent frames is as rigid as possible (ARAP). To this end, we introduce a novel approach to compute correspondences between adjacent textured implicit surfaces, which are used to define the ARAP regularization term. The second regularization term enforces that the topology of the underlying object remains fixed over time. This regularization is critical for avoiding self-intersections that are typical in implicit-based reconstructions. We have evaluated the performance of 4DRecons on a variety of datasets. Experimental results show that 4DRecons can handle large deformations and complex inter-part interactions and outperform state-of-the-art approaches considerably.

***Keywords:** Neural implicit, dynamic reconstruction, topology consistency, shape deformation*

1. Introduction

We seek to reconstruct a deforming object from a single RGB-D sensor. This task is extremely challenging, because of the limited observations presented in each frame. Many recent approaches have relied on data-driven shape priors [9, 12] or pre-trained feature descriptors [3, 46, 21, 24, 7] to mitigate this issue. However, these data-driven solutions require that the observations be within the distribution of the training data. Due to limited 3D data that is available, these approaches typically exhibit limited performance on casual captures. In this paper, we seek to push the limit of non-data-driven solutions, which focus on aggregating information across the input frames. Our goal is to reconstruct the underlying deforming surface, where each point is observed in at least one frame.

Specifically, we formulate dynamic reconstruction as learning a 4D implicit field problem (iso-value of surface and colors) from partial RGB-D scans. This is motivated by the success of implicit neural representations in representing and encoding static [35, 15] and dynamic [38, 36, 47] objects and scenes. Our approach 4DRecons combines a data term and two regularization terms. The data term is a novel loss that fits the implicit field to partial input scans. The first regularization term regularizes the deformation between adjacent frames and the smoothness of the deformations among triplets of frames. This is achieved by a novel approach that computes correspondences between implicit fields, using which we define the regularization terms.

The second regularization term, which is a key contribution of this paper, enforces that the topology of the reconstruction remains fixed over time. This constraint, combined with the implicit field representation, nicely addresses the open problem of obtaining self-intersection-free reconstructions under the explicit representation, e.g., deforming the SMPL model. Our approach is based on recent advances in the optimization of geometry with given topological constraints [14, 37, 30] and is easy to optimize.

We have evaluated 4DRecons on a variety of datasets. The experimental results show that 4DRecons can handle large deformations and complex inter-part interactions and outperforms state-of-the-art dynamic reconstructions. The source code and data will be released upon paper acceptance.

2. Related Works

Dynamic geometry reconstruction. 4DRecons falls into the category of animation reconstruction [49, 48], which has been studied in graphics and 3D vision for more than two decades. Animation reconstruction aims to recover the complete 3D model and the underlying deformations from an RGB-D scan sequence, where each frame captures a deformation object from one view. This problem is generalized from the rigid object reconstruction problem, which has a

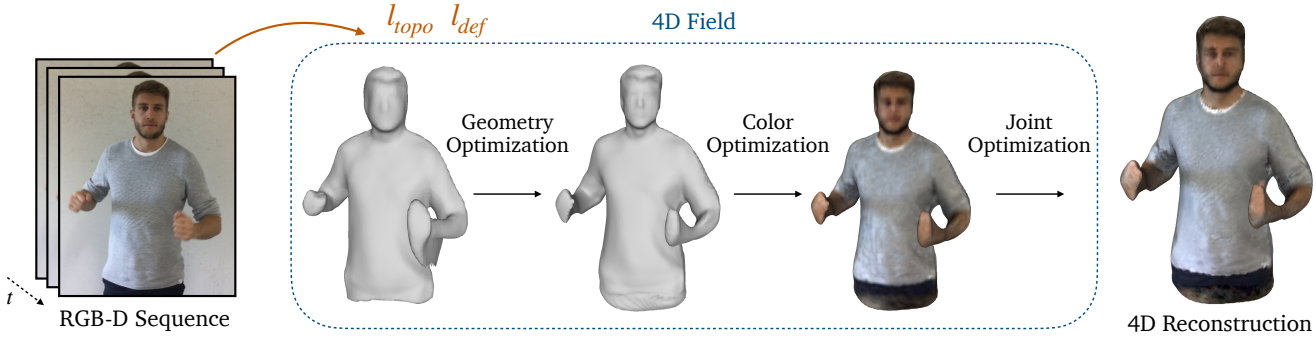


Figure 1: Pipeline Overview of 4DRecons, which performs a joint optimization with an annealing strategy. 4DRecons progressively balance various regularization terms to selectively focus on refining different aspects of the 4D field.

long literature. A fundamental challenge is to recover inter-frame correspondences for aggregating information from multiple frames into a complete model. Many animation reconstruction approaches require a template model. In contrast, we do not assume that we have a template model as an input.

Template-free approaches fall into explicit methods and implicit methods. Early works are explicit methods (e.g., DynamicFusion [33]) that progressively align the next frame to the current reconstruction that aggregates all existing frames. Deformations are modeled using deformation graphs [44, 18, 23, 22, 39] or volumetric deformations [33]. One limitation of explicit methods is that it is very difficult to handle self-collision. This issue is addressed by implicit methods, which seek to reconstruct time-varying implicit surfaces using various regularizations. Sharf et al. [41] and [47] studied how to enforce the incompressibility of a deforming object in the reconstruction procedure. KillingFusion [43] studied how to model the local rigidity constraint by borrowing ideas from Killing vector fields.

4DRecons falls into the category of implicit methods and presents two fundamental contributions. First, deformation is modeled on the underlying surface, in contrast to the volumetric field employed in KillingFusion. This approach places less constraint on the underlying volumetric field and is more flexible. Second, we enforce topological consistency across the input frames, avoiding merging contacting surfaces under the implicit representation.

Neural implicits from point clouds. DeepSDF [35] is pioneering the research area of reconstructing an implicit surface from point-cloud data. The key idea is to generate samples inside and outside the surface, which are used to regress a volumetric neural implicit field. Several approaches improved DeepSDF performance by using better data losses, e.g., SAL [1] and SALD [2]. However, these formulations cannot handle partial observations because the samples generated near the boundary area are not well-defined. 4DRecons uses a different approach which carefully places samples close to the observed surface area and constrains the signed distance function as a loss term on its gradient field. In particular, our sampling strategy, which is based on an analysis of the confidence of the sample, is critical to ensure a high-quality implicit surface.

Another way to address partial observations is to reconstruct an unsigned distance function (UDF) from the observed points [10, 55, 26, 27]. However, a fundamental challenge of UDF is to extract the underlying surface. Moreover, it is applied mainly to objects with boundaries, and the observation in each frame is complete [27]. The reason is that under UDF it is very difficult to aggregate partial observations at different frames to form a complete surface.

Deformation modeling. Embedded deformations [45] are widely used for dynamic reconstruction. The technical challenge in our setting is that embedded deformations require an explicit geometric representation, which is not available in our setting. 4DRecons innovates in computing dense correspondences between adjacent implicit surfaces. Unlike GenCorres [50] where correspondences are completely driven by geometry, 4DRecons computes correspondences by matching geometry and color. Using these correspondences, we introduce regularization terms that penalize deformations and enforce color consistency.

Many dynamic reconstruction approaches penalize the deformation between a template model and the input scans. However, this approach requires either a template model [34, 38, 25], which is not always available, or treats the first frame as a template model [36], which requires the first frame to be complete. Moreover, these approaches cannot handle large deformations, which are difficult to model. In contrast, 4DRecons minimizes deformations between adjacent frames. The deformations between non-adjacent frames can still be large.

Topological regularization. 4DRecons is motivated by recent work on optimizing a 3D shape with prescribed topology [14, 37] and enforcing that the shape generator outputs connected 3D shapes [30]. The basic idea is to link the ver-

tices of a 3D shape with topological features on the persistent diagram [11]. This allows us to deform a 3D shape to match topological attributes. 4DRecons enforces that the number of topological features on the persistent diagram remains fixed. This approach nicely penalizes inter-penetrations that frequently exhibit in explicit-based and implicit-based dynamic reconstructions. To the best of our knowledge, 4DRecons is the first approach that enforces topological consistency for dynamic reconstruction.

3. Problem Statement and Approach Overview

3.1. Problem Statement

The input is N RGB-D scans $\mathcal{P}_t = \{(\boldsymbol{x}_{ti}, \boldsymbol{n}_{ti}, \boldsymbol{c}_{ti}), 1 \leq i \leq n_i\}, 1 \leq t \leq N\}$ represented in the sensor's local coordinate system. Here, $\boldsymbol{x}_{ti} \in \mathbb{R}^3$ is the sample position; $\boldsymbol{n}_{ti} \in \mathbb{R}^3$ is the sample normal; $\boldsymbol{n}_{ti}$ is estimated using [31] and is oriented using the camera center; $\boldsymbol{c}_{ti} \in \mathbb{R}^3$ is the sample color. Our goal is to reconstruct a 4D implicit field $\boldsymbol{f}^\theta : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R} \times \mathbb{R}^3$, which takes a position $\boldsymbol{x}$ and time t as input and outputs an isovalue $s^\theta(\boldsymbol{x}, t)$ and color $\boldsymbol{c}^\theta(\boldsymbol{x}, t)$. Due to the space constraint, we defer the details of network architecture to the supp. material.

3.2. Approach Overview

We optimize the network parameters θ by combining one data loss and two regularization losses:

$$\min_\theta l_{\text{data}}(\theta, \{\mathcal{P}\}) + \lambda_{\text{def}} l_{\text{def}}(\theta) + \lambda_{\text{topo}} l_{\text{topo}}(\theta) \qquad (1)$$

In the following, we highlight the main ideas of our loss terms. Section 4 explains the technical details.

Data term. The data term measures the distance between the implicit field and the input scans. We fit f^θ to samples close to each input scan $\mathcal{P}_t$. We explicitly model scan boundary regions in which surface normal predictions and perturbed samples along normal directions have high uncertainties. Specifically, we force the samples near boundary regions to be close to the surface and rely on propagating information from other frames that have more complete observations in these regions. We also introduce four regularization terms to stabilize the fitting process.

Deformation regularization term. This term regularizes the deformation between two adjacent frames and the smoothness of the deformations between three consecutive frames. Generalizing GenCorres [50], we introduce a novel approach to compute the correspondences between textured implicit surfaces. Using these correspondences, we then develop the regularization terms for as-rigid-as-possible (ARAP) deformations, deformation smoothness, and color consistency.

Topology regularization term. The second regularization term enforces that the topology of the implicit reconstruction remains fixed during the optimization procedure. We use the persistent diagram (PD) tool and enforce that the PD of the reconstruction in frame t matches the PD of the reconstruction in frame t', where (t, t') is chosen as a dense subset of frame pairs.

4. Approach

4.1. Data Term

We define the total data loss as the sum of the loss associated with each scan, which combines a fitting term l_{fit}, a color-geometry consistent term l_{color}, and a regularization term l_{regu}.

$$l_{\text{data}}(\theta, \{\mathcal{P}_t\}) = \sum_{t=1}^{N} \Big(l_{\text{fit}}(\mathcal{P}_t, \boldsymbol{f}^\theta(\cdot, t)) + \lambda_c l_{\text{color}}(\boldsymbol{f}^\theta(\cdot, t)) + \lambda_r l_{\text{regu}}(\mathcal{P}_t, \boldsymbol{f}^\theta(\cdot, t)) \Big). \qquad (2)$$

The key differences of 4DRecons from the previous work lie in defining the data term l_{fit} and the consistency term l_{color}. The role of the regularization term l_{regu} is to address the issue of overfitting due to the limited observations of each scan. We have explored many previous approaches [35, 29, 42, 17, 52, 53] and find that the formulation in [42], which combines the term of periodic eikonal and two addition terms that regularize the distribution of $f^\theta(\cdot, t)$, offers the best result. In the following, we focus on defining l_{fit} and l_{color}.

Data fitting term. We define the data fitting term by generating $m_t \geq n_t$ samples $\Omega_t = \{(\boldsymbol{p}_{tj}, \boldsymbol{c}_{tj}, \boldsymbol{n}_{tj}, s_{tj}), 1 \leq j \leq m_t\}$ from $\mathcal{P}_t$ where $\boldsymbol{p}_{tj} = \boldsymbol{x}_{tj} + s_{tj}\boldsymbol{n}_{tj}$. In other words, each sample in Ω_t picks one sample in $\mathcal{P}_t$ with a random offset value s_{tj} that satisfies $|s_{tj}| \leq d_{\max} b_{tj}$, where $b_{ti} \in [0, 1]$ is an interior confidence value, and $b_{ti} = 0$ if $\boldsymbol{x}_{tj}$ is on the boundary. Note that two samples in Ω_t may share the same point in $\mathcal{P}_t$ but with different distance values s_{tj}. The motivation of introducing b_{tj} is because on the one hand, placing samples outside the surface is important for distance function fitting, c.f., [8, 35]; while on the other hand, the precision of distance predictions of off-surface samples degrade if their closest points are near surface boundaries. We compute the confidence value b_{ti} by projecting the nearest neighbors of $\boldsymbol{p}_{ti}$ onto the tangent plane at $\boldsymbol{p}_{ti}$. We defer the details to the appendix as this is not our main contribution.

Given these samples, we define the fitting term as

$$l_{\text{fit}}(\mathcal{P}_t, \boldsymbol{f}^\theta(\cdot, t)) := \int_{\Omega_t} \big((s^\theta(\boldsymbol{p}_{tj}, t) - s_{tj})^2 + \mu_c (\boldsymbol{c}^\theta(\boldsymbol{x}_{tj}, t) - \boldsymbol{c}_{tj})^2 + \mu_n (\nabla_{\boldsymbol{x}} s^\theta(\boldsymbol{x}_{tj}, t) - \boldsymbol{n}_{tj})^2 \big) d\boldsymbol{p}_{tj} \qquad (3)$$

Which align the implicit field $\boldsymbol{f}^\theta(\cdot, t)$ with the RGB-D scan. The normal alignment term is inspired from [17, 52]. We set $\mu_c = 0.1, \mu_n = 0.1$ in our experiments.

Figure 2: The qualitative comparisons with and without the Color-geometry consistency term l_{color} defined in Eq. (4).

Color-geometry consistency term. We define $l_{\text{color}}(\boldsymbol{f}^\theta(\cdot,t))$ so that the gradient of the color field is perpendicular to the gradient of the distance field, i.e., color does not change in the normal direction of the underlying surface:

$$l_{\text{color}}(s^\theta(\cdot,t)) := \int_\Omega \|\nabla s^\theta(\boldsymbol{x}_{tj},t)^T \nabla \boldsymbol{c}^\theta(\boldsymbol{x}_{tj},t)\|^2 d\boldsymbol{p}_{tj} \tag{4}$$

Figure 2 shows that the regularization term l_{color} is important in ensuring the quality of the geometry and the color field, particularly near the boundary regions.

4.2. Deformation Regularization Term

The deformation regularization term $l_{\text{def}}(\theta) = l^1_{\text{def}}(\theta) + l^2_{\text{def}}(\theta)$ has two components. The first component $l^1_{\text{def}}(\theta)$ enforces that the underlying deformation between adjacent frames $t \sim \boldsymbol{f}^\theta(\cdot,t)$ and $t+1 \sim \boldsymbol{f}^\theta(\cdot,t+1)$ is as rigid as possible (ARAP) [19, 51]. In $l^1_{\text{def}}(\theta)$, we also want to enforce that the color field is consistent between adjacent frames. The second component $l^2_{\text{def}}(\theta)$ ensures that the deformations across $\boldsymbol{f}^\theta(\cdot,t-1)$, $\boldsymbol{f}^\theta(\cdot,t)$, and $\boldsymbol{f}^\theta(\cdot,t+1)$ are smooth. To formulate the deformation prior and enforce the consistency of the color field, we need to solve the fundamental problem of computing correspondences between the implicit fields. 4DRecons builds on the approach in GenCorres [50] for computing correspondences between adjacent implicit surfaces. While the approach of GenCorres focuses on geometric shapes, 4DRecons non-trivially extends it to include color information for correspondence computation. Specifically, we first apply the Marching Cube algorithm [28] to obtain a discrete mesh $\mathcal{M}_t = (\mathcal{V}_t, \mathcal{E}_t)$ from the implicit surface $s^\theta(\boldsymbol{x},t) = 0$ (See Figure 3 (Left)). Each vertex $v_{ti} \in \mathcal{V}_t$ has a position $\boldsymbol{v}_{ti}$ and a color $\boldsymbol{c}_{ti}$, and they depend on the network parameters θ. For each $\boldsymbol{v}_{ti}$, our goal is to compute its correspondence $\boldsymbol{v}^+_{ti}$ and $\boldsymbol{v}^-_{ti}$ on the surface of frame $t+1$ and frame $t-1$, respectively. For simplicity, we describe the procedure for $\boldsymbol{v}^+_{ti}$ and that of $\boldsymbol{v}^-_{ti}$ to be identical. In the following, we use $\boldsymbol{v}_t(\boldsymbol{v}^+_t)$ to stack the vertex positions $\boldsymbol{v}_{ti}(\boldsymbol{v}^+_{ti})$ into vectors.

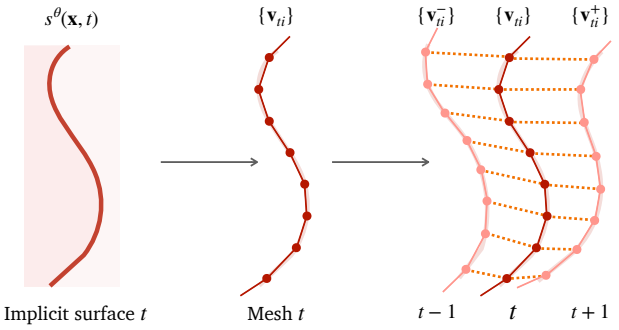


Figure 3: We first discretize an implicit surface in frame t into a triangular mesh. Then we solve an optimization problem to determine the correspondences of the vertices of this mesh in the next frame $t+1$ and the previous frame $t-1$. These correspondences are then used to define the deformation regularization term.

The derivative of the implicit surface constraint at $\boldsymbol{v}_{ti}$ with respect to t provides one constraint on $\boldsymbol{v}^+_{ti}$:

$$\frac{\partial s^\theta(\boldsymbol{v}_{ti},t)}{\partial t} + \frac{\partial s^\theta(\boldsymbol{v}_{ti},t)}{\partial \boldsymbol{x}}^T (\boldsymbol{v}^+_{ti} - \boldsymbol{v}_{ti}) = 0. \tag{5}$$

To uniquely determine $\boldsymbol{v}^+_{ti}$, we solve a global optimization problem to compute $\boldsymbol{d}_t := \boldsymbol{v}^+_t - \boldsymbol{v}_t$ (See Figure 3 (Right)). The objective function consists of two terms. The first term minimizes an as-rigid-as-possible energy between $\boldsymbol{v}^+_t$ and $\boldsymbol{v}_t$ and is defined as $e_{\text{arap}}(\boldsymbol{d}_t) =$

$$\sum_{v_{ti}\in\mathcal{V}_t} \min_{\boldsymbol{a}^+_{ti}} \sum_{v_{tj}\in\mathcal{N}_{ti}} \|(I + \boldsymbol{a}^+_{ti}\times)(\boldsymbol{v}_{ti} - \boldsymbol{v}_{tj}) - (\boldsymbol{v}^+_{ti} - \boldsymbol{v}^+_{tj})\|^2 \tag{6}$$

where $\mathcal{N}_{ti}$ collects adjacent vertices of v_{ti} on $\mathcal{M}_t$; $I + \boldsymbol{a}_{ti}\times$ is a linear approximation of the local rotation from $\boldsymbol{v}_{ti}$ to $\boldsymbol{v}^+_{ti}$. Based on [19], we can express

$$e_{\text{arap}}(\boldsymbol{d}_t) = \boldsymbol{d}_t{}^T L^{\text{arap}}_t \boldsymbol{d}_t \tag{7}$$

where the expression of L^{arap}_t is in the appendix.

Similarly, the second term minimizes the color differences between adjacent frames t and $t+1$. To this end, we use a linear approximation

$$\boldsymbol{c}^\theta(\boldsymbol{v}^+_{ti}, t+1) \approx \boldsymbol{c}^\theta(\boldsymbol{v}_{ti}, t+1) + \frac{\partial \boldsymbol{c}^\theta(\boldsymbol{v}_{ti}, t+1)}{\partial \boldsymbol{x}} \boldsymbol{d}_{ti}$$

where $\boldsymbol{d}_{ti}$ is the i-th element of $\boldsymbol{d}_t$. We then define this term

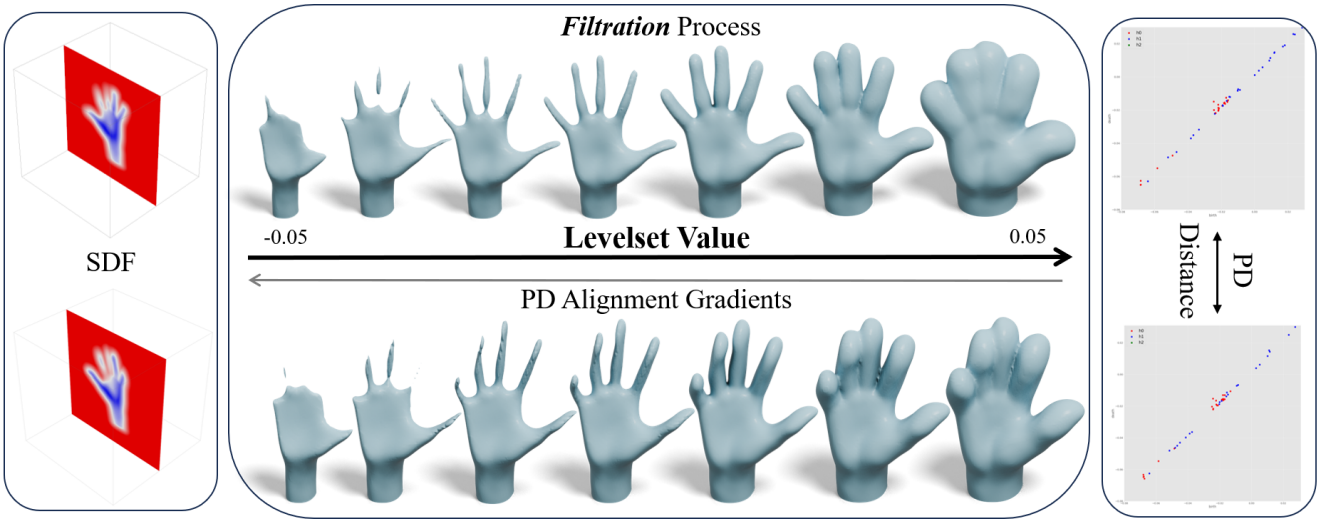


Figure 4: The *filtration* processes of two SDF volumes, resulting in two comparable PD plots that we enforce alignment. Our method allows the alignment gradients to manipulate the underlining SDF volumes to synchronize their topological signatures. In practice, PD loss is an elegant holistic one-pass calculation without the need to extract explicit meshes, which treats the whole volume at once with linear complexity [11].

as

$$\begin{aligned} e_{\text{color}}(\boldsymbol{d}_t) := \sum_{v_{ti}\in\mathcal{V}_t} & \|\boldsymbol{c}^\theta(\boldsymbol{v}_{ti}, t+1) - \boldsymbol{c}^\theta(\boldsymbol{v}_{ti}, t) \\ & + \frac{\partial \boldsymbol{c}^\theta(\boldsymbol{v}_{ti}, t+1)}{\partial \boldsymbol{x}} \cdot \boldsymbol{d}_{ti}\|^2 \\ & = \boldsymbol{d}_t^T L_t^{\text{color}} \boldsymbol{d}_t - 2\boldsymbol{b}_t^{\text{color}} \boldsymbol{d}_t + r_t^{\text{color}}. \end{aligned} \tag{8}$$

We solve $\boldsymbol{d}_t$ by minimizing $e_{\text{arap}}(\boldsymbol{d}_t) + \lambda_{\text{color}} e_{\text{color}}(\boldsymbol{d}_t)$ with linear constraints (5). Introduce $L_t = L_t^{\text{arap}} + \mu_{\text{color}} L_t^{\text{color}}$, where $\mu_{\text{color}} = 0.001$. Let $C_t \boldsymbol{d}_t = -F_t$ be the matrix representation of (5). We arrive at the following quadratic program with linear constraints (details are in the appendix) to solve $\boldsymbol{d}_t$:

$$\boldsymbol{d}_t = \underset{\boldsymbol{d}}{\text{argmin}} \quad \boldsymbol{d}^T L_t \boldsymbol{d} - 2\boldsymbol{b}_t^T \boldsymbol{d} \quad s.t. \quad C_t \boldsymbol{d} = -F_t$$

which leads to a closed-form expression of

$$\boldsymbol{d}_t = L_t^{-1}\Big(\boldsymbol{b}_t - C_t^T \left(C_t L_t^{-1} C_t^T\right)^{-1} \left(C_t L_t^{-1} \boldsymbol{b}_t + F_t\right)\Big). \tag{9}$$

Using (9), we define the first component of the deformation regularization term

$$l_{\text{def}}^1(\theta) := \sum_{t=1}^{N-1} \left(e_{\text{arap}}(\boldsymbol{d}_t) + \lambda_{\text{color}} e_{\text{color}}(\boldsymbol{d}_t)\right) \tag{10}$$

where $\lambda_{\text{color}} = 1.0$. The second component $l_{\text{def}}^2(\theta)$ enforces that the deformations are smooth between triplets of adjacent frames. Let $\boldsymbol{c}_t^+$ collect the optimal solution $\boldsymbol{c}_{ti}^+$ in (6). Let $\boldsymbol{c}_t^-$ be defined accordingly. Note that $\boldsymbol{c}_t^+(\boldsymbol{c}_t^-)$ are linear in $\boldsymbol{v}_t^+(\boldsymbol{v}_t^-)$ and $\boldsymbol{v}_t$. We define

$$l_{\text{def}}^2(\theta) = \sum_{t=2}^{N-1} \left(\mu_r \|\boldsymbol{v}_t^+ + \boldsymbol{v}_t^-\|^2 + \mu_p \|\boldsymbol{v}_t^- + \boldsymbol{v}_t^+ - 2\boldsymbol{v}_t\|^2\right) \tag{11}$$

4.3. Topology Regularization Term

The topology regularization term employs the persistent diagram (PD) [11], a widely used topological signature, to align implicit fields defined in different time frames t. In essence, a PD is constructed based on a *filtration* of a topological space, specifically a cubic lattice evaluated on a 3D SDF volume in this context. Following the approach outlined in [14] we use the super-levelsets $\{\boldsymbol{x} | s^\theta(\boldsymbol{x}, t) \geq \alpha\}$ with varying α to build the *filtration*. This *filtration* process identifies **critical levelset values** that influence changes in the surface topology of the level sets. These **critical values** signify the formation of voids or holes and their convergence into fewer solid entities (see Figure 4). Our topology regularization is designed to directly govern these **critical levelset values** and their presence in a differentiable manner. Consequently, our regularization aligns the topology of all SDF-induced super-levelsets along with their zero-levelset surfaces.

A PD consists of a set of 2D points, each of which corresponds to the birth and death times of a topological feature of the *filtration*. In our setting, the PD $PD(s^\theta(\cdot, t)) = \{(b_i, d_i), i \in \mathcal{I}_t\}$ of $s^\theta(\cdot, t)$ is given by pairs of **local minima/maxima** of $s^\theta(\cdot, t)$. Therefore, we can backpropagate

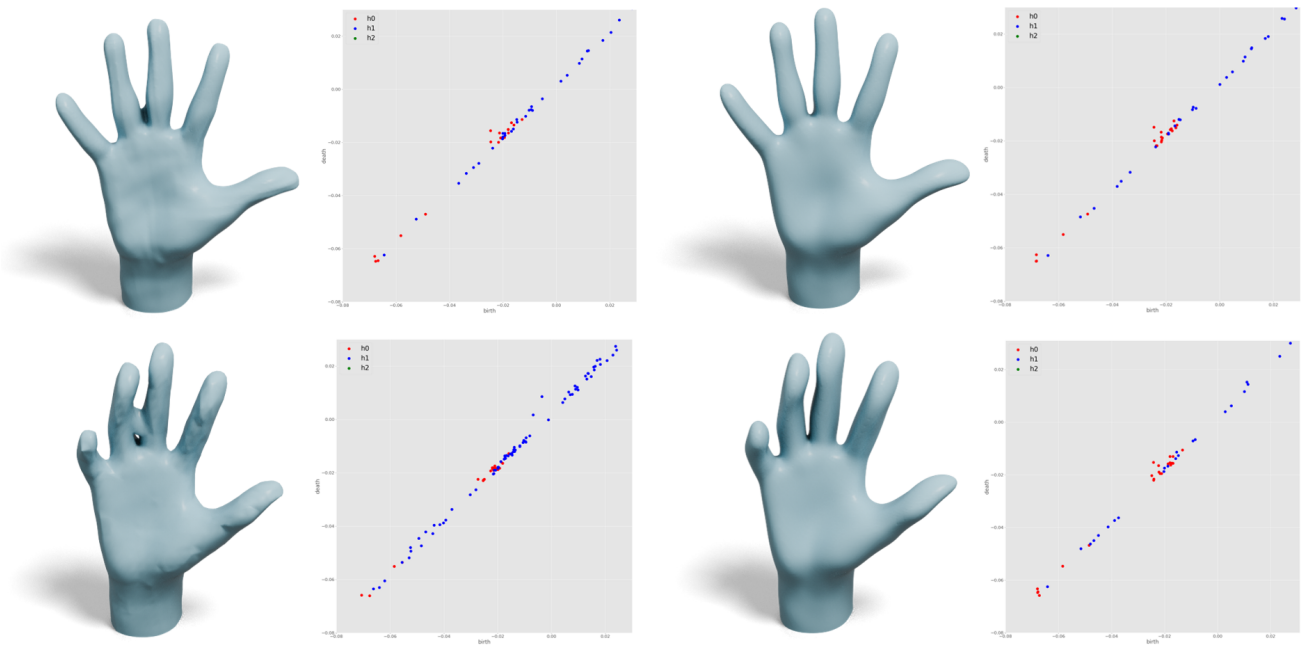


Figure 5: Illustration of aligning PDs of two frames.

gradients w.r.t. PDs further to $s^\theta(\cdot, t)$, as is done in [14]. In the following, we use PD_t^i to denote $\{b_i, d_i\}$ given by $PD_t = PD(s^\theta, t)$.

For two timestamps t, t', we minimize the p-Wasserstein distance between the corresponding PDs PD_t and $PD_{t'}$:

$$d(PD_t, PD_{t'}) = \left(\inf_{\substack{\sigma: \mathcal{I}_t \to \mathcal{I}_{t'} \\ \sigma \in S_{|\mathcal{I}_t|}}} \sum_{i \in \mathcal{I}_t} |PD_t^i - PD_{t'}^{\sigma(i)}|_p^p \right)^{1/p} \tag{12}$$

where σ is a permutation of indices $\mathcal{I}_t$. We defer the illustration of aligning PDs to the supp. material.

Having established the distance of PD, we define the PD loss as a summation of the PD distances in pairs between timestamps $[1, N]$. Here we sum PD distances over pairs on evenly spaced cycles along timestamps. The final PD loss is defined as

$$l_{\text{topo}}(\theta) = \sum_{k=1}^{N-1} \frac{1}{|C(k)|} \sum_{i,j \in C(k)} d(PD_i, PD_j) \tag{13}$$

where $C(k)$ is the collection of edges starting from 1 with $k-1$ timestamps skipped, e.g., $C(2) = \{(1,3), (3,5), \cdots\}$. Similar to training point cloud generators [13], minimizing (12) combines alternating optimization of permutation σ and θ.

As shown in Figure 5, without $l_{\text{topo}}(\theta)$, the top and bottom PDs of the left column are misaligned, indicating a lack of topological consistency. With $l_{\text{topo}}(\theta)$, the top and bottom PDs of the right column match, showing topological consistency.

4.4. Network Architecture

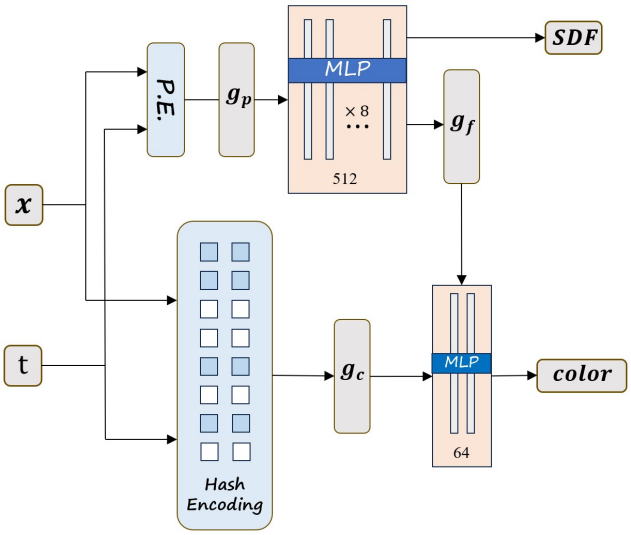


Figure 6: Illustration of the network architecture.

As shown in Figure 6, we implement our 4D implicit field $f^\theta : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R} \times \mathbb{R}^3$ by combining a geometry branch $s^\theta : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^{1+f_g}$ and a color branch $c^\theta : \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^{f_g} \to \mathbb{R}^3$. Both branches employ an encoder and multilayer perceptrons (MLP). Specifically, the geometry branch s^θ applies the positional encoding to the

3D coordinate $\boldsymbol{x}$ with 8 frequencies to derive a positional feature $\boldsymbol{g}_p$. We then feed the concatenation of $\boldsymbol{g}_p$ and the frame index t into eight fully connected layers (with softplus activations and 512 channels per layer) to decode the implicit value $s^\theta(\boldsymbol{x}, t)$ and a latent feature vector $\boldsymbol{g}_f \in \mathbb{R}^{f_g}$ ($f_g = 256$). We use a skip connection [35] to connect the input with the output of the fourth MLP layer and weight normalization to stabilize the optimization of the parameters. The color field utilizes the multiresolution hash grid [32] to $\boldsymbol{x}$ and t to obtain a color feature $\boldsymbol{g}_c$. The number of feature dimensions per entry is 2, the number of levels is 16, the base resolution is 16 and the scale factor per level is 1.3819. We then feed $\boldsymbol{g}_f$ and $\boldsymbol{g}_c$ into three fully connected layers (using ReLU activations) to generate color $\boldsymbol{c}^\theta(\boldsymbol{x}, t)$. Note that $\boldsymbol{g}_p$ encodes the correlation between the geometry branch and the color branch.

5. Optimization

The total objective function in Eq. (1) consists of multiple objective terms with very different energy landscapes. Direct end-to-end optimization can easily fall into local minimums. As shown in Figure 1, we employ a joint optimization with an annealing strategy to learn $\boldsymbol{f}^\theta$ from the input partial scans. The annealing strategy can be divided into four steps.

Step I: Initialization. The first step initializes the 4D implicit geometry and color field from the input partial observations without using the regularization terms. In other words, we set $\lambda_{\text{def}} = 0$ and $\lambda_{\text{topo}} = 0$ in Eq. (1).

Step II: Geometry field regularization. The second step optimizes the geometry field while ignoring the color field. In this step, we involve only the regularization terms associated with the geometry field. Specifically, in this step we set $\lambda_c = 0$ in Eq. (2) and $\lambda_{\text{color}} = 0$ in Eq. (10).

Step III: Color field regularization. The third step optimizes the color field while fixing the geometry field. In other words, we freeze the parameters of the geometry field by stopping the gradient backpropagation and focus on optimizing the color field with related loss terms. This step essentially uses the inter-frame correspondences derived from the geometry field and minimizes Eq. (8) to propagate color information across invisible regions at each time step.

Step IV: Joint refinement. In the fourth step, we jointly refine the geometry field and the color field. This allows us to use color information to obtain improved inter-frame correspondences which lead to an improved geometry field. Similarly, the improved geometry field can better propagate color information among invisible regions, resulting in more consistent and sharper texture reconstructions.

We provide a quantitative comparison in Table 1 to validate the effectiveness of our optimization strategy. We observe that the optimization strategy is an inevitable component and essential for the model's convergence.

Table 1: Quantitative ablation study about our optimization strategy on the geometry and color field reconstruction in three datasets: DeepDeform (D_D), KillingFusion (K_F), Our collected Data (O_D). O-S denotes 'Optimization Strategy' introduced in Section 5.

Model	Geometry (in mm.) ↓			Color (PSNR in dB.) ↑		
	D_D	K_F	O_D	D_D	K_F	O_D
w/o O-P	4.827	9.762	9.162	17.48	14.93	14.05
w/ O-P	**0.884**	**1.249**	**1.823**	**32.06**	**31.01**	**27.72**

6. Experimental Evaluation

This section presents an experimental evaluation of 4DRecons. We begin with the experimental setup in Section 6.1. Section 6.2 and Section 6.3 compare 4DRecons with state-of-the-art approaches in the geometry reconstruction and texture reconstruction. Section 6.4 presents an ablation study. Please refer to the supp. material for more results.

6.1. Experimental Setup

Datasets. To evaluate 4DRecons and baselines, we use a wide range of sequences from (1) DeepDeform [4] dataset, (2)KillingFusion [43] dataset and (3) a new dataset captured by us using an iPhone 14 Pro with a resolution of 1920×1080 at 30 FPS. In particular, this new dataset highlights sequences where implicit representations are likely to struggle with maintaining topological consistency. Specifically, it consists of ten sequences with an average duration of 30 seconds, performed by five individuals. Each person recorded two types of sequences: moving their arms close to their body and then away, and moving their fingers close and then apart. Both types present significant challenges for implicit methods in maintaining topological consistency. For every sequence, we use the RGB/depth images and the camera intrinsics.

Baseline approaches. We compare 4DRecons with five current top-performing baseline approaches to evaluate the geometry and texture reconstruction: a non-deep learning-based approach DynamicFusion [33] using our own implementation; state-of-the-art neural-based monocular dynamic reconstruction approaches NDR [5], D-NeRF [38], Hexplane [6] and Tensor4D [40].

In Table 2, We summarize the input modalities supported by different approaches and whether each approach is used for geometry or color comparisons. Since the official implementations of D-NeRF [38], Hexplane [6], and Tensor4D [40] are designed to take only RGB images as input, we adapt these methods by incorporating an additional loss term. This term compares the rendered depth to the input depth, ensuring a fair comparison.

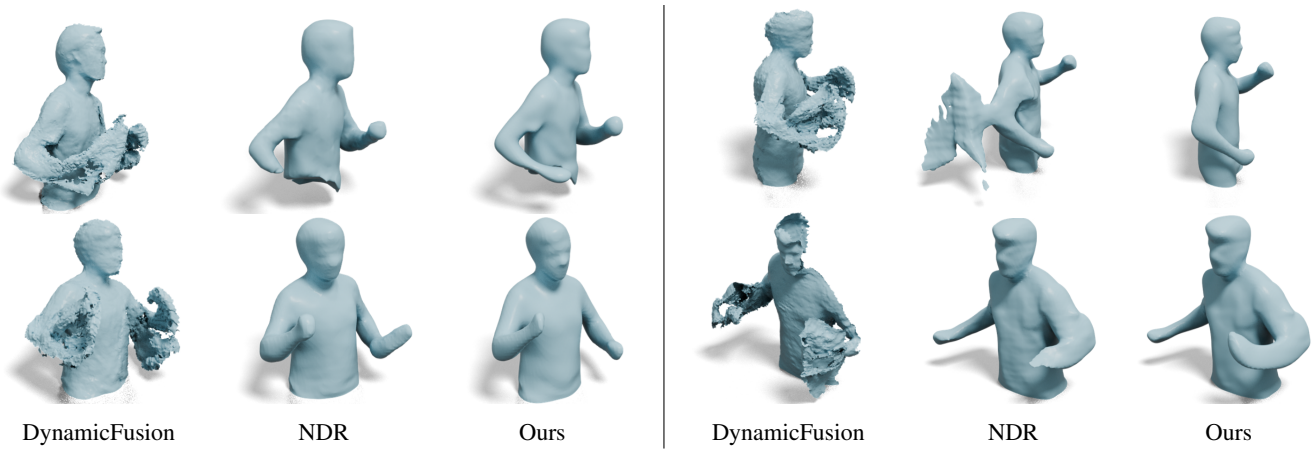


Figure 7: Qualitative comparisons of geometry field reconstruction on sequences without topological changes.

Table 2: Quantitative comparisons on the geometry and color field reconstruction in three datasets: DeepDeform (D_D), KillingFusion (K_F), Our collected Data (O_D).

Method	Input Modality		Geometry (in mm.) ↓			Color (PSNR in dB.) ↑		
	RGB	Depth	D_D	K_F	O_D	D_D	K_F	O_D
D-NeRF (w/o $\mathbf{D}$)	√	×	4.007	4.179	7.191	27.91	27.05	22.42
D-NeRF (w/ $\mathbf{D}$)	√	⊙	3.118	3.936	5.016	28.78	27.73	22.86
Hexplane (w/o $\mathbf{D}$)	√	×	2.891	3.139	4.912	31.89	30.63	26.80
Hexplane (w/ $\mathbf{D}$)	√	⊙	2.319	2.968	4.628	**32.08**	**31.11**	27.11
Tensor4D (w/o $\mathbf{D}$)	√	×	3.109	3.891	5.280	30.53	30.09	24.18
Tensor4D (w/ $\mathbf{D}$)	√	⊙	2.891	3.616	5.157	30.92	30.41	25.20
DynamicFusion	×	√	5.428	4.129	14.19	×	×	×
NDR	√	√	0.923	1.323	1.899	31.08	30.92	25.09
Ours	√	√	**0.884**	**1.249**	**1.823**	**32.06**	**31.01**	**27.72**

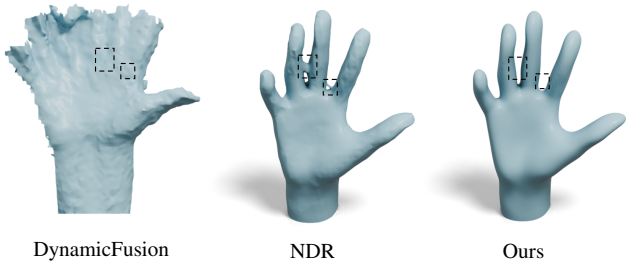


Figure 8: Qualitative evaluation on sequences with topology changes.

Evaluation protocol. We conduct both qualitative evaluations, where we visually compare the reconstruction results of 4DRecons and the baseline approaches, and quantitative evaluations. Quantitative evaluations report reconstruction errors in geometry by the difference between the reconstructed mesh and depth values inside the mask, as well as color by PSNR between rendering results and masked input RGB images.

Implementation Details The deformation regularization term $l_{\text{def}}(\theta)$ is based on the mesh extracted from the zero-level set of the geometry field $s^{\theta}(\boldsymbol{x}, t)$. We use the Marching Cube algorithm for discretization by a voxel grid of size $50 \times 50 \times 50$. The output mesh typically contains more than 5000 vertices. Following Gencorres [50], we simplify the output mesh into 2000 faces [16] to reduce the computation complexity. The number of vertices n is around 1000.

We initialize the geometry field $s^{\theta}(\boldsymbol{x}, t)$ as an approximate unit sphere [1] at the beginning of training. We train our neural networks using the ADAM optimizer [20]. Empirically, we set the weights in Eq. 1 as: $\lambda_{\text{def}} = 0.001, \lambda_{\text{topo}} = 0.001$. We use Fast-Robust-ICP [54] as a preliminary step to initialize the camera extrinsic for each frame. We use autograd in PyTorch to compute F_t and C_t and use finite differences to approximate other derivative computations. we use marching cubes with 128 grid resolution to extract the zero-level set of implicit surfaces.

All the experiments are conducted on a single NVIDIA RTX A6000.

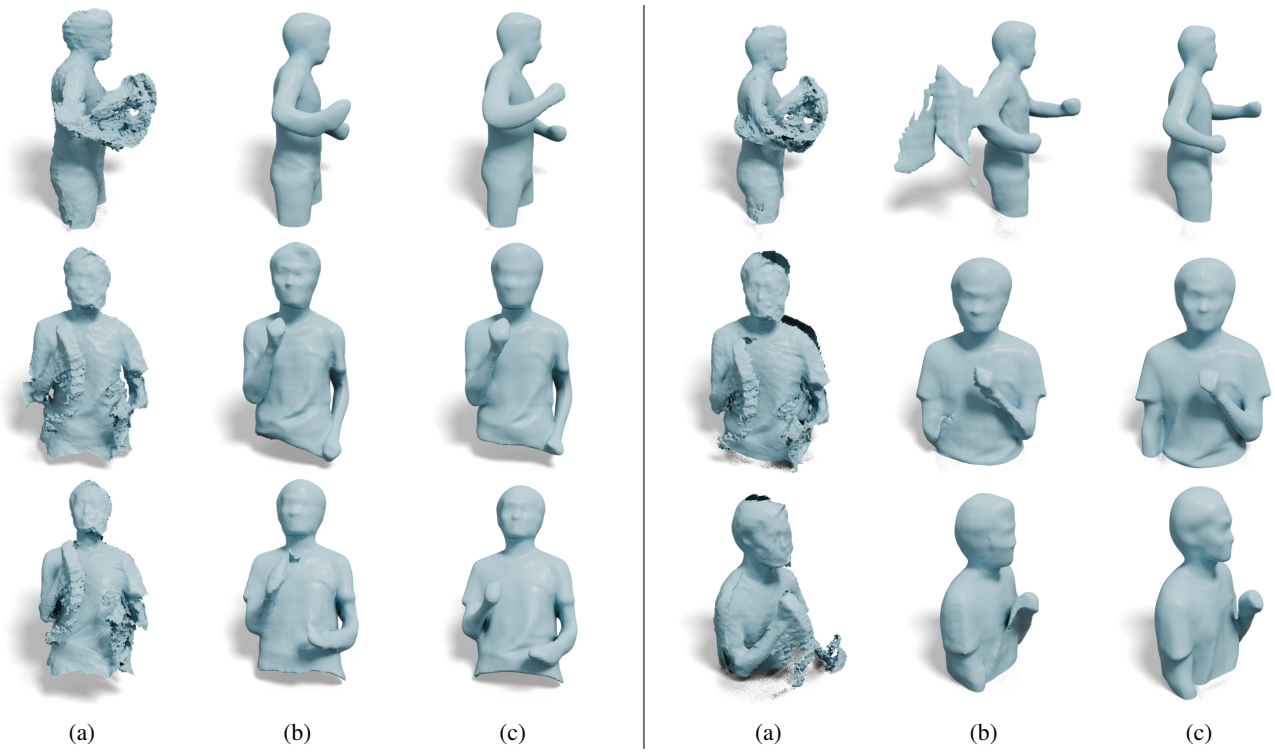


Figure 9: Qualitative comparisons of geometry field reconstruction among (a) DynamicFusion [33], (b) NDR [5] and (c) 4DRecons.

6.2. Evaluation on Geometry Reconstruction

Figure 7 and Figure 8 present qualitative results. As shown in Figure 7, 4DRecons can reconstruct detailed geometries. The non-deep learning-based method DynamicFusion [33] exhibits artifacts around the surface. Moreover, the neural-based method NDR [5] fails to recover the underlying approximate articulated deformation, especially in cases with large deformation and fast motion. As shown in Figure 8, our topology regularization term 4.3 allows 4DRecons to ensure that the reconstruction topology remains fixed and consistent over time. In contrast, baselines produce an inconsistent and unfixed topology, as different parts of fingers randomly merge when they are close to each other.

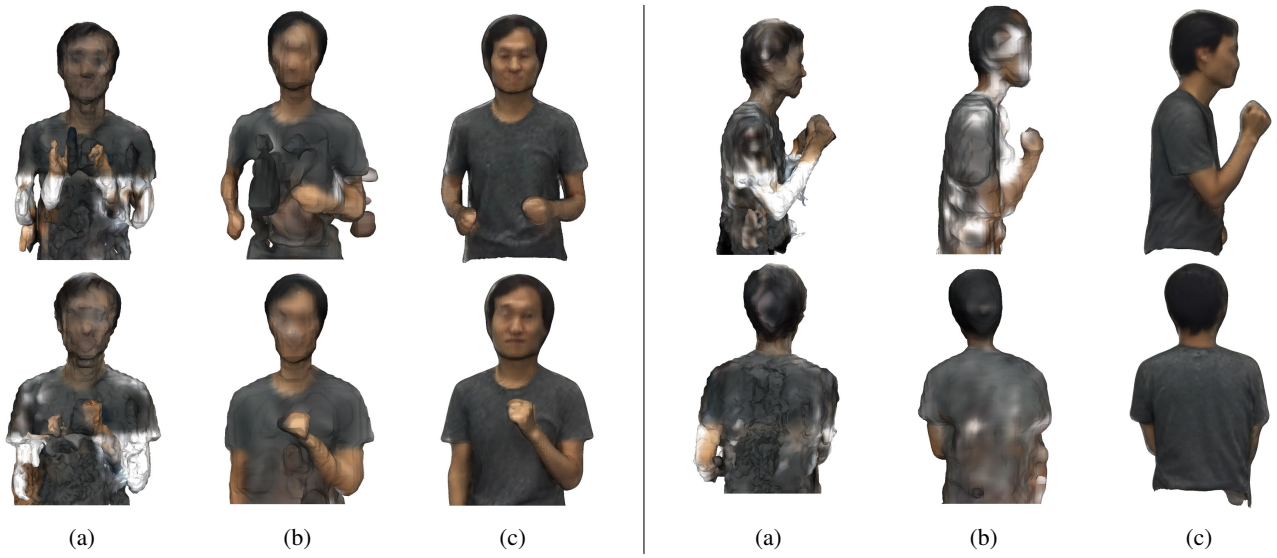
Quantitatively, as shown in Table 2, our approach reduces the mean reconstruction error of D-NeRF [38], Hexplane [6], DynamicFusion [33], Tensor4D [40], and NDR [5] by an average of 67.86%, 60.14%, 80.19%, 767.28%, and 4.45% on all the datasets, respectively. Quantitative improvements are consistent with qualitative results. These numbers again show the effectiveness of 4DRecons in integrating observations from different frames by modeling suitable deformation and topology regularization losses.

In Figure 9, we show more visual comparisons of the geometry reconstruction with two important baselines DynamicFusion [33] and NDR [5]. With the help of our deformation regularization term, 4DRecons can reconstruct a more complete, smooth, and detailed geometry.

6.3. Evaluation on Color Field Reconstruction

Figure 11 shows visual comparisons of the texture quality. 4DRecons can achieve rendering results that are on par with state-of-the-art baselines trained via volume rendering and image-based optimization. Quantitative comparisons are consistent with qualitative results, as shown in Table 2. However, when comparing the extracted textured mesh in Figure 12, 4DRecons yields significantly more detailed results than the baselines. This is particularly encouraging because NeRF-based techniques tend to overfit training data and show artifacts under novel viewpoints and poses. The benefit of reconstructing a textured mesh is that it enables fast rendering and many downstream applications.

In Figure 10, we present more qualitative comparisons of the colored mesh reconstruction with two important baselines, D-NeRF [38] and Hexplane [6]. 4DRecons can achieve rendering results that are on par with, but reconstruct much more detailed textured meshes than baselines trained via an imaged-based optimization procedure.



(a) (b) (c) (a) (b) (c)

Figure 10: Qualitative comparisons of color field reconstruction among (a) D-NeRF [38], (b) Hexplane [6] and (c) 4DRecons.

D-NeRF [38] Hexplane [6] Ours

Figure 11: Qualitative evaluation on the color field. The center is the rendering results, and the lower right corner is the colored mesh.

D-NeRF Hexplane Ours

Figure 12: Qualitative evaluation on the colored mesh.

6.4. Ablation Study

Quantitatively, we validate the effectiveness of different components of 4DRecons with an extensive ablation study in Table 3. Row 4 shows our complete model as a reference. In rows 1–3 we remove the three key components one at a time from the complete model, observing that the deformation regularization term (row 1) defined in Section 4.2 provides the largest quantitative benefit, which is essential for the model's convergence. It is followed by the color consistency term (row 3) defined in Eq. (8) and topology regularization term (row 2) defined in Section 4.3. Each of them

Table 3: Quantitative ablation study on the geometry and color field reconstruction in three datasets: DeepDeform (D_D), KillingFusion (K_F), Our collected Data (O_D).

Model	Geometry (mm) ↓			Color (PSNR) ↑		
	D_D	K_F	O_D	D_D	K_F	O_D
w/o l_{def}	1.371	3.227	3.903	27.19	29.84	19.61
w/o l_{topo}	0.920	1.383	2.249	31.79	30.08	25.97
w/o e_{color}	0.947	1.480	2.421	30.17	29.22	22.82
Complete	**0.884**	**1.249**	**1.823**	**32.06**	**31.01**	**27.72**

plays its own role in enhancing the geometry and color field reconstruction. Qualitatively, Figure 13 illustrates that the deformation regularization term can significantly improve the reconstruction results. Omitting this term leads to incomplete reconstructions with deformations that are neither smooth nor locally rigid. This is expected, as we rely on the deformation term to propagate partial observations across the entire sequence. Without this term, propagation is done by the smoothness of the network, which does not understand the underlying approximate articulated motions. Figure 14 indicates the topology regularization term is crucial for enforcing the topology to remain fixed by aligning the persistent diagram (PD) throughout the sequence. Dropping this term compromises the topological consistency, thus diminishing the performance of 4DRecons. Finally, as shown in Figure 15, enforcing color consistency improves the accuracy of correspondences and therefore enhances the quality of the reconstruction, which is the key difference between our approach and the formulation in GenCorres [50]. Without this term, both the geometry reconstruction quality and the texture reconstruction quality drop.

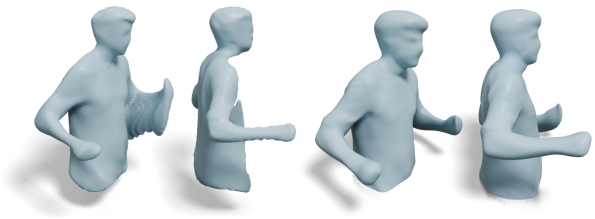
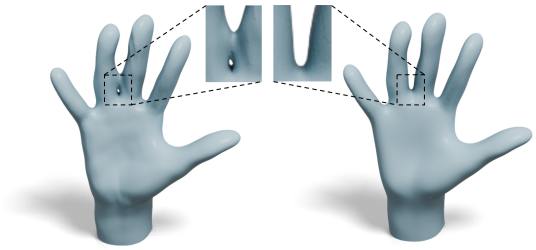


Figure 13: Comparisons of the geometry field reconstruction with (right) and without (left) the deformation regularization term $l_{\text{def}}(\theta)$ defined in 4.2.



w/o $l_{\text{topo}}(\theta)$ w/ $l_{\text{topo}}(\theta)$

Figure 14: Comparisons of geometry field reconstruction with (right) and without (left) $l_{\text{topo}}(\theta)$.

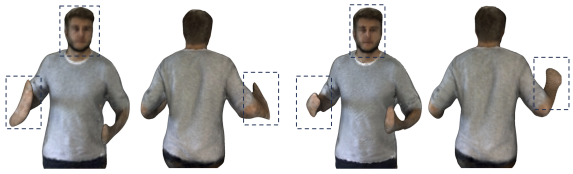


Figure 15: The qualitative comparisons with (right) and without (left) the color consistency term e_{color} defined in Eq. (8).

7. Limitations

One limitation is that our approach assumes that the final reconstruction is a closed deforming surface, and it requires that each point of the underlying object be observed from at least one frame. In the future, we want to address this issue by using unsigned distance fields that can model open surfaces. Another limitation is that the topology regularization term, which improves the topological consistency, cannot guarantee that the topology of the reconstruction remains fixed, as it is enforced in the least-square sense. In the future, we plan to address this issue by employing an explicit mesh representation and using the implicit field to ensure that the mesh reconstruction is self-collision-free.

8. Conclusions

In this paper, we introduced 4DRecons, a monocular dynamic reconstruction of a deforming subject from partial scans of a single RGB-D camera. It combines a data term, a deformation regularization term, and a topology regularization term. The geometry regularization term computes inter-frame correspondences to propagate observed geometry and color signals. The topology regularization term promotes the topological consistency of reconstructions across all frames. The experimental results show that 4DRecons outperforms baseline approaches both qualitatively and quantitatively.

There are ample opportunities for future research. First, 4DRecons focuses on a single sequence; it would be interesting to study how to learn a 4D representation from multiple sequences that is generalizable to new sequences. Another direction is to build a multi-resolution deformation model to capture detailed deformations introduced by cloth. Finally, it is interesting to combine the strength of implicit representations and explicit representations for dynamic reconstruction. For example, one approach is to use the implicit field to guide the fusion of point clouds acquired at different frames. Potential negative societal impact: our approach requires extensive computational resources and optimization time, which may raise concerns about energy consumption. We will continue working on optimizing that in future work.

Acknowledgement

This work was supported by NSF-2047677, NSF-2413161, NSF-2504906, NSF-2515626, and GIFTs from Adobe and Google. This work was supported by computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at UT Austin.

References

[1] M. Atzmon and Y. Lipman. SAL: sign agnostic learning of shapes from raw data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2562–2571. Computer Vision Foundation / IEEE, 2020. 2, 8

[2] M. Atzmon and Y. Lipman. SALD: sign agnostic learning with derivatives. In *ICLR*. OpenReview.net, 2021. 2

[3] A. Bozic, P. R. Palafox, M. Zollhöfer, A. Dai, J. Thies, and M. Nießner. Neural non-rigid tracking. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1

[4] A. Božič, M. Zollhöfer, C. Theobalt, and M. Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. 2020. 7

[5] H. Cai, W. Feng, X. Feng, Y. Wang, and J. Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 7, 9

[6] A. Cao and J. Johnson. Hexplane: A fast representation for dynamic scenes. *CVPR*, 2023. 7, 9, 10

[7] W. Cao, C. Luo, B. Zhang, M. Nießner, and J. Tang. Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20496–20506. IEEE, 2024. 1

[8] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 67–76, New York, NY, USA, 2001. Association for Computing Machinery. 3

[9] J. Chibane, T. Alldieck, and G. Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6968–6979. Computer Vision Foundation / IEEE, 2020. 1

[10] J. Chibane, A. Mir, and G. Pons-Moll. Neural unsigned distance fields for implicit function learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[11] H. Edelsbrunner and J. Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010. 3, 5

[12] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer. Points2surf learning implicit surfaces from point clouds. In A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2020. 1

[13] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2463–2471. IEEE Computer Society, 2017. 6

[14] R. B. Gabrielsson, V. Ganapathi-Subramanian, P. Skraba, and L. J. Guibas. Topology-aware surface reconstruction for point clouds. *Comput. Graph. Forum*, 39(5):197–207, 2020. 1, 2, 5, 6

[15] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li. Nerf: Neural radiance field in 3d vision, a comprehensive review, 2022. 1

[16] Garland, Michael, Heckbert, and P. S. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 8

[17] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 3

[18] Q. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. *Comput. Graph. Forum*, 27(5):1449–1457, 2008. 2

[19] Q. Huang, X. Huang, B. Sun, Z. Zhang, J. Jiang, and C. Bajaj. Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators, 2021. 4

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 8

[21] J. Lei and K. Daniilidis. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6614–6624. IEEE, 2022. 1

[22] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*, 28(5):175, 2009. 2

[23] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Comput. Graph. Forum*, 27(5):1421–1430, 2008. 2

[24] W. Lin, C. Zheng, J. Yong, and F. Xu. Occlusionfusion: Occlusion-aware motion estimation for real-time dynamic 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 1726–1735. IEEE, 2022. 1

[25] L. Liu, M. Habermann, V. Rudnev, K. Sarkar, J. Gu, and C. Theobalt. Neural actor: neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.*, 40(6):219:1–219:16, 2021. 2

[26] Y.-T. Liu, L. Wang, J. Yang, W. Chen, X. Meng, B. Yang, and L. Gao. Neudf: Leaning neural unsigned distance fields with volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 237–247, June 2023. 2

[27] X. Long, C. Lin, L. Liu, Y. Liu, P. Wang, C. Theobalt, T. Komura, and W. Wang. Neuraludf: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20834–20843, June 2023. 2

[28] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, Aug. 1987. 4

[29] L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4460–4470. Computer Vision Foundation / IEEE, 2019. 3

[30] M. Mezghanni, M. Boulkenafed, A. Lieutier, and M. Ovsjanikov. Physically-aware generative network for 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9330–9341. Computer Vision Foundation / IEEE, 2021. 1, 2

[31] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, SCG '03, page 322–328, New York, NY, USA, 2003. Association for Computing Machinery. 3

[32] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 7

[33] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 343–352. IEEE Computer Society, 2015. 2, 7, 9

[34] A. Noguchi, X. Sun, S. Lin, and T. Harada. Neural articulated radiance field. In *ICCV*, pages 5742–5752. IEEE, 2021. 2

[35] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, June 2019. 1, 2, 3, 7

[36] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5845–5854. IEEE, 2021. 1, 2

[37] A. Poulenard, P. Skraba, and M. Ovsjanikov. Topological function optimization for continuous shape matching. *Comput. Graph. Forum*, 37(5):13–25, 2018. 1, 2

[38] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10318–10327. Computer Vision Foundation / IEEE, 2021. 1, 2, 7, 9, 10

[39] A. Schulz, A. Shamir, I. Baran, D. I. W. Levin, P. Sitthi-Amorn, and W. Matusik. Retrieval on parametric shape collections. *ACM Trans. Graph.*, 36(1):11:1–11:14, Jan. 2017. 2

[40] R. Shao, Z. Zheng, H. Tu, B. Liu, H. Zhang, and Y. Liu. Tensor4d : Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering, 2023. 7, 9

[41] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph.*, 27(5):110, 2008. 2

[42] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 3

[43] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5474–5483. IEEE Computer Society, 2017. 2, 7

[44] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 2

[45] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3):80–es, July 2007. 2

[46] J. Tang, D. Xu, K. Jia, and L. Zhang. Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 6022–6031. Computer Vision Foundation / IEEE, 2021. 1

[47] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12939–12950. IEEE, 2021. 1, 2

[48] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. J. Guibas, H. Seidel, and A. Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.*, 28(2):15:1–15:15, 2009. 1

[49] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. J. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In A. G. Belyaev and M. Garland, editors, *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Barcelona, Spain, July 4-6, 2007*, volume 257 of *ACM International Conference Proceeding Series*, pages 49–58. Eurographics Association, 2007. 1

[50] H. Yang, X. Huang, B. Sun, C. Bajaj, and Q. Huang. Gencorres: Consistent shape matching via coupled implicit-explicit shape generative models, 2024. 2, 3, 4, 8, 10

[51] H. Yang, B. Sun, L. Chen, A. Pavel, and Q. Huang. Geolatent: A geometric approach to latent space design for deformable shape generators. *ACM Trans. Graph.*, 42(6), dec 2023. 4

[52] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4805–4815, 2021. 3

[53] L. Yariv, P. Hedman, C. Reiser, D. Verbin, P. P. Srinivasan, R. Szeliski, J. T. Barron, and B. Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In E. Brunvand, A. Sheffer, and M. Wimmer, editors, *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, pages 46:1–46:9. ACM, 2023. 3

[54] J. Zhang, Y. Yao, and B. Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, 2022. 8

[55] J. Zhou, B. Ma, Y. Liu, Y. Fang, and Z. Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *NeurIPS*, 2022. 2