Neural Surface Reconstruction of Dynamic Scenes with Monocular RGB-D Camera

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose Neural-DynamicReconstruction (NDR), a template-free method to recover high-fidelity geometry and motions of a dynamic scene from a monocular RGB-D camera. In NDR, we adopt the neural implicit function for surface representation and rendering such that the captured color and depth can be fully utilized to jointly optimize the surface and deformations. To represent and constrain the non-rigid deformations, we propose a novel neural invertible deforming network such that the cycle consistency between arbitrary two frames is automatically satisfied. Considering that the surface topology of dynamic scene might change over time, we employ a topology-aware strategy to construct the topology-variant correspondence for the fused frames. NDR also further refines the camera poses in a global optimization manner. Experiments on public datasets and our collected dataset demonstrate that NDR outperforms existing monocular dynamic reconstruction methods.

1 Introduction

Reconstructing 3D geometry shape, texture and motions of the dynamic scene from a monocular video is a classical and challenging problem in computer vision. It has broad applications in many areas like virtual and augmented reality. Although existing methods [60, 62] have demonstrated impressive reconstruction results for dynamic scenes only with 2D images, they are still difficult to recover high-fidelity geometry shapes especially for some casually captured data as abundant potential solutions exist without depth constraints. Only with 2D measurements, dynamic reconstruction methods need motions of interested object hold in a nearby z-plane. Meanwhile, it is difficult to construct reliable correspondences in areas with weak texture, which causes error accumulation in the canonical space.

To solve this under-constrained problem, some methods propose to utilize shape priors for some special object types. For example, category-specific parametric shape models like 3DMM [6], SMPL [39] and SMAL [69] are first constructed and then used to help the reconstruction. However, templated-based methods could not generalize to unknown object types. On the other hand, some methods utilize annotations, like keypoints and optical flow, obtained from manual annotators or off-the-shelf tools [30, 32, 60, 62]. The motion trajectories of sparse or dense 2D points can effectively help recover the exact motion of whole structure. However, it needs human labeling for supervision or highly depends on the quality of learned priors from a large-scale dataset.

One straightforward solution to solve this under-constrained problem is to reconstruct interested object based on observations from RGB-D cameras like Microsoft Kinect [66] and Apple iPhone X. Existing fusion-based methods [42, 26, 51] utilize a dense non-rigid warp field and a canonical truncated signed distance (TSDF) volume to represent motion and shape, respectively. However, these fusion-based methods might fail due to accumulated tracking errors, especially for long sequences. To alleviate this problem, some learning-based methods [9, 8, 58] utilize more accurate correspondences which

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

are annotated or learned from synthesis datasets to guide the dynamic fusion process. However, the captured color and depth information is not represented together within one differentiable framework in these methods. Recently, a neural implicit representation based method [3] has been proposed to reconstruct a room-scale scene from RGB-D inputs, but it is only designed for static scenes and can not be directly applied to dynamic scenes.

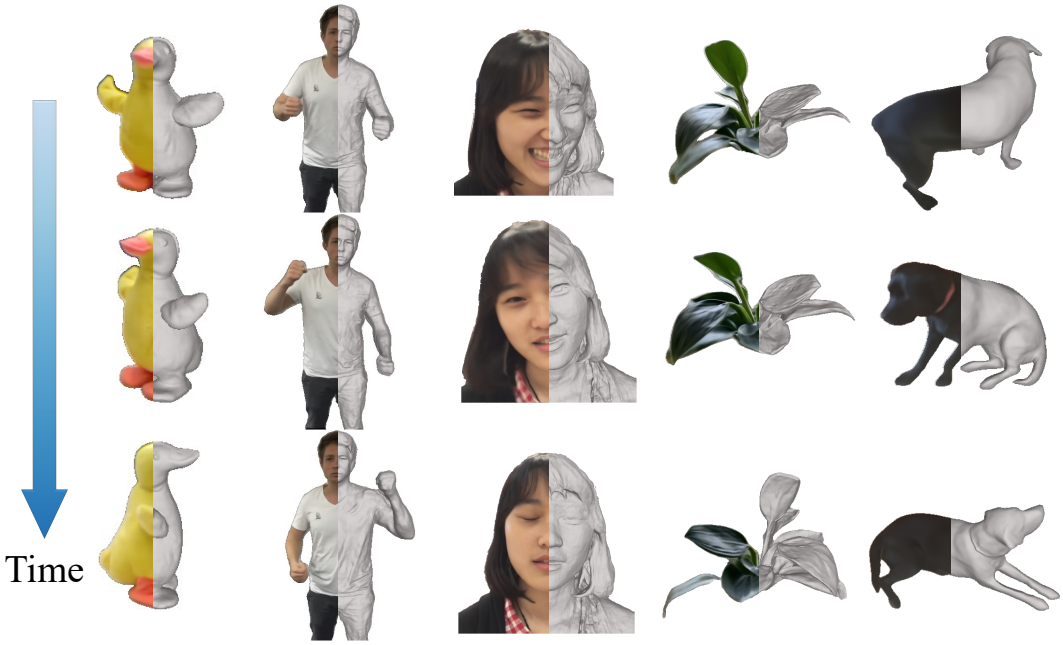

Figure 1: Examples of *reconstructed* (right) and *rendered* (left) results by NDR. Given a monocular RGB-D video sequence, NDR recovers high-fidelity geometry and motions of a dynamic scene.

In this paper, we present Neural-DynamicReconstruction (**NDR**), a neural dynamic reconstruction method from a monocular RGB-D camera (Fig. 1). To represent the high-fidelity geometry and texture of deformable object, NDR maintains a neural implicit field as the canonical space. With extra depth constraint, there still exist multiple potential solutions since the correspondences between different frames are still unknown. In this paper, we propose the following strategies to constrain and regularize the solution space: (1) integrating all RGB-D frames to a high-fidelity textured shape in the canonical space; (2) maintaining cycle consistency between arbitrary two frames; (3) A surface representation which can handle topological changes.

Specifically, we adopt the neural SDF and radiance field to respectively represent the high-fidelity geometry and appearance in the canonical space, instead of TSDF volume frequently used in fusion-based methods [27, 42, 51, 9, 8, 58]. In our framework, each RGB-D frame can be integrated into the canonical representation. We propose a novel neural deformation representation that implies a continuous bijective map between observation and canonical space. The designed invertible module applies a cycle consistency constraint through the whole RGB-D video, meanwhile, it fits the natural properties of non-rigid motion well. To support topology changes of dynamic scene, we adopt the topology-aware network in HyperNeRF [45]. Thanks to modeling topology-variant correspondence, our framework can handle topology changes while existing deformation graph based methods [42, 58, 62] can not. NDR also further refines camera intrinsic parameters and poses during training. Extensive experimental results demonstrate that NDR can recover high-fidelity geometry and photorealistic texture for monocular category-agnostic RGB-D videos.

2 Related Works

RGB Based Dynamic Reconstruction. Dynamic reconstruction approach can be divided into template-based and template-free types. Templates [6, 39, 48, 69] are category-specific statistical models constructed from large-scale datasets. With the help of pre-constructed 3D morphable mod-

els [6, 12, 35], some researches [5, 11, 25, 54, 20, 24, 18] reconstruct faces or heads from RGB inputs. Most of them need 2D keypoints as extra supervisory information to guide dynamic tracking [68, 16, 18]. With the aid of human parametric models [1, 39], some works [7, 59, 22, 23, 67, 28] recover digital avatars based on monocular image or video cues. However, it is unpractical to extend templates to general objects with limited 3D scanned priors, such as articulated objects, clothed human and animals. Non-rigid structure from motion (NR-SFM) algorithms [10, 33, 50] is to reconstruct category-agnostic object from 2D observations. Although NR-SFM can reconstruct reasonable result for general dynamic scenes, it heavily depends on reliable point trajectories throughout observed sequences [49, 53]. Recently, some methods [60, 61, 62] obtain promising results from a long monocular video or several short videos of a category. LASR [60] and ViSER [61] recover articulated shapes via a differentiable rendering manner [38], while BANMo [62] models them with the help of Neural Radiance Fields (NeRF) [41]. However, due to the depth ambiguity of input 2D images, the reconstruction might fail for some challenging inputs.

RGB-D Based Dynamic Reconstruction. Recovering 3D deforming shapes from a monocular RGB video is a highly under-constrained problem. On the other hand, The progress in consumer-grade RGB-D sensors has made depth map capture from a single camera more convenient. Therefore, it is quite natural to reconstruct the target objects based on RGB-D sequences. DynamicFusion [42], which is the seminar work of RGB-D camera based dynamic object reconstruction, proposes to estimate a template-free 6D motion field to warp live frames into a TSDF surface. The surface representation strategy has also been used in KinectFusion [27]. VolumeDeform [26] represents motion in a grid and incorporates global sparse SIFT [40] features during alignment. Guo et al. [19] coherents albedo, geometry and motion estimation in an optimization pipeline. KillingFusion [51] and SobolevFusion [52] are proposed to deal with topology changes. During deep learning era, DeepDeform [9] and Bozic et al. [8] aim to learn more accurate correspondences for tracking improvement of faster and more complex motions. OcclusionFusion [58] probes and handles occlusion problem via an LSTM-involved graph neural network but fails when topology changes. Although these methods obtain promising reconstruction results with the additional depth cues, their reconstructed geometry shape mainly depends on the captured depths while the RGB images are not fully utilized to further improve the results.

Dynamic NeRF. Given a range of image cues, prior works on NeRF [41] optimize an underlying continuous scene function for novel view synthesis. Some NeRF-like methods [36, 47, 55, 17, 44, 45] achieve promising result on dynamic scenes without prior templates. Nerfies [44] and AD-NeRF [21] reconstructs free-viewpoint selfies from monocular videos. HyperNeRF [45] models an ambient slicing surface to express topologically varying regions. Recent approaches [56, 3] introduce neural representation for static object/scene reconstruction, but theirs can not be used for non-rigid scenes.

Cycle Consistency Constraint. To maintain cycle consistency between deformed frames is an important regularization in perceiving and modeling dynamic scenes [57]. However, recent methods [61, 36, 62] try to leverage a loss term to constrain estimated surface features or scene flow, which is a weak but not strict property. Therefore, constructing an invertible representation for deformation field is a reasonable design. Several invertible networks are proposed to represent deformation, such as Real-NVP [15], Neural-ODE [13], I-ResNet [4]. Based on these manners, there exist some methods modeling deformation in space [29, 63, 46] or time [43, 34] domain. CaDeX [34] is a novel dynamic surface representation method using a real-valued non-volume preserving module [15]. Different from these strategies, we propose a novel scale-invariant binary map between observation space and 3D canonical space to process RGB-D sequences, which is more suitable for modeling non-rigid motion.

3 Method

The input of NDR is an RGB-D sequence $\{(\mathcal{I}_i, \mathcal{D}_i), i = 1, \cdots, N\}$ captured by a monocular RGB-D camera (e.g., Kinect and iPhone X), where $\mathcal{I}_i \in \mathbb{R}^{H \times W \times 3}$ is the i-th RGB frame and $\mathcal{D}_i \in \mathbb{R}^{H \times W \times 1}$ is the corresponding aligned depth map. To optimize a canonical textured shape and motion through the sequence, we leverage whole N color frames $\mathcal{I}_i$ as well as corresponding depth frames $\mathcal{D}_i$. Specifically, we first adopt video segmentation methods [14, 37] to obtain the mask $\mathcal{M}_i$ of interested object. Then, we integrate RGB-D video sequence into a canonical hyper-space composed of a 3D canonical space and a topology space. We propose a continuous bijective representation between

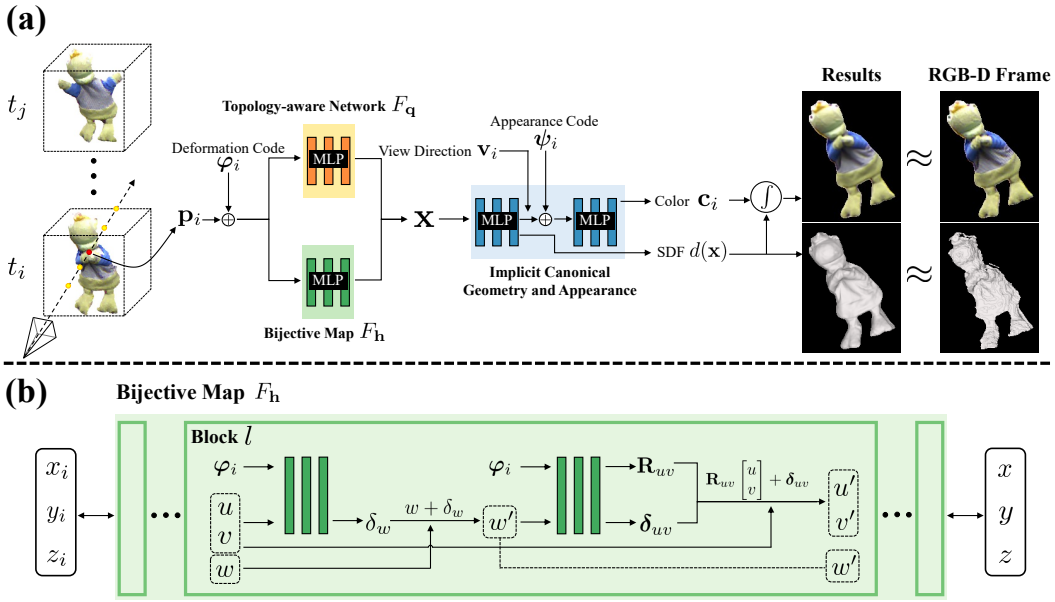

Figure 2: (a) Pipeline of NDR. (b) The structure of bijective map $F_{\mathbf{h}}$ in deformation field.

the 3D canonical and observation space such that the cycle consistency can be strictly satisfied. The implicit surface is represented by a neural SDF and volume rendering field, as a function of input hyper-coordinate and camera view. The geometry, appearance, and motions of dynamic object are optimized without any template or structured priors, like optical flow [62], 2D annotations [9] and estimated normal map [28]. The pipeline of NDR is shown in Fig. 2(a).

3.1 Bijective Map in Space-time Synthesis

Invertible Representation. Given a 3D point sampled in the space of i-th frame, recent methods [42, 62, 45] model its motion as a 6D transformation in $\mathbf{SE}(3)$ space. Nerfies [44] and HyperNeRF [45] construct a continuous dense field to estimate the motion. To reduce the complexity, DynamicFusion [42] and BANMo [62] define warp functions based on several control points. The latter designs both 2D and 3D cycle consistency loss terms to apply bijective constraint to deformation representation, but it is just a guidance for learning instead of a rigorous inference module. Similar to the previous works, we also construct the deformation between each current frame and the 3D canonical space. Further, we employ a strictly invertible bijective mapping, which is naturally compatible with the cycle consistency strategy. Specifically, we decompose the non-rigid deformation into several reversible bijective blocks, where each block represents the transformation along and around a certain axis. In this manner, our deformation representation is strictly invertible and fits the natural properties of non-rigid motion well, which is helpful for the reconstruction effect.

We denote $\mathbf{p}_i = [x_i, y_i, z_i] \in \mathbb{R}^3$ as a position of the observation space at time t_i, in which a deformed surface U_i is embedded. It is noticeable that $\mathbf{p}_i$ represents any position, both surface and free-space points. A continuous homeomorphic mapping $\mathcal{H}_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ maps $\mathbf{p}_i$ back to the 3D canonical position $\mathbf{p} = [x, y, z]$. Supposes that there exists a canonical shape U of the interested object which is independent of time and is shared across the video sequence. Note that map $\mathcal{H}_i$ is invertible, and thus we can directly obtain the deformed surface at time t_i:

$$U_i = \{\mathcal{H}_i^{-1}([x, y, z]) | \forall [x, y, z] \in U\}. \tag{1}$$

Then, the correspondence of $\mathbf{p}_i$ can be expressed by the bijective map, factorized as:

$$[x_j, y_j, z_j] = \mathcal{G}_{ij}([x_i, y_i, z_i]) = \mathcal{H}_j^{-1} \circ \mathcal{H}_i([x_i, y_i, z_i]). \tag{2}$$

The deformation representation $\mathcal{G}$ is cycle consistent strictly, since it is invariant on deforming path ($\mathcal{G}_{jk} \circ \mathcal{G}_{ij} = \mathcal{G}_{ik}$). As a composite function of two bijective maps (Eq. 2), it is a topology-invariant function between arbitrary double time stamps.

Implementation. Based on these observations, we implement the bijective map $\mathcal{H}$ by a novel invertible network h. While Real-NVP [15] seems a suitable network structure, its scale-variant property limits its usage in our object reconstruction task. Inspired by the idea of Real-NVP to split the coordinates, we decompose our scale-invariant deformation into several blocks. In each block, we set an axis and represent the motion steps as simple axis-related rotations and translations, which are totally shared by the forward and backward deformations. In this manner, the inverse deformation $\mathcal{H}^{-1}$ can be viewed as the composite of the inverse of these simple rotations and translations in $\mathcal{H}$.

Fig. 2(b) shows the detailed structure of each block. Given a latent deformation code $\boldsymbol{\varphi}$ binding with time, we firstly consider the forward deformation, where the 3D positions $[u, v, w] \in \mathbb{R}^3$ of observation space is input, and the positions $[u', v', w'] \in \mathbb{R}^3$ of 3D canonical space is output. Without loss of generality, let the w-axis to be the chosen axis. With $[u, v]$ fixed, we compute a displacement $\boldsymbol{\delta}_w$ and update w' as $w + \boldsymbol{\delta}_w$. With $[w']$ fixed, we then compute the rotation $\mathbf{R}_{uv}$ and translation $\boldsymbol{\delta}_{uv}$ for $[u, v]$ and updated them as $[u', v']$. Oppositely, for the backward deformation, we apply $-\boldsymbol{\delta}_{uv}$, $\mathbf{R}_{uv}^{-1}$, and $-\boldsymbol{\delta}_w$ in turn to recover $[u', v', w']$ back to $[u, v, w]$. Therefore, if the network $\mathbf{h}$ consists of these invertible blocks, it can represent a bijective map as well. At time t_i, $\mathbf{h}(\cdot|\boldsymbol{\varphi}_i) : \mathbb{R}^3 \to \mathbb{R}^3$ maps 3D positions $\mathbf{p}_i$ of observation space back to 3D canonical correspondences $\mathbf{p}$, where $\boldsymbol{\varphi}_i$ denotes the deformation code of i-th frame. In our experiment, we use a Multi-Layer Perceptron (MLP) as the implementation of $\mathbf{h}$, so we design a continuous bijective map $F_{\mathbf{h}}$ for space-time synthesis.

3.2 Deformation Field

Although the proposed deformation representation is a continuous homeomorphic mapping that satisfies the cycle consistency between different frames, it also preserves the surface topology. However, some dynamic scenes (e.g., varying body motion and facial expression) may undergo topology changes. Therefore, we combine a topology-aware design [45] into our deformation field. 3D positions $\mathbf{p}_i$ observed at time t_i are mapped to topology coordinates $\mathbf{q}(\mathbf{p}_i)$ through a network $\mathbf{q} : \mathbb{R}^3 \to \mathbb{R}^m$. We regress topology coordinate from a MLP $F_{\mathbf{q}}$. Then the corresponding coordinate of $\mathbf{p}_i$ in the canonical hyper-space is represented as:

$$\mathbf{x} = [\mathbf{p}, \mathbf{q}(\mathbf{p}_i)] = [F_{\mathbf{h}}(\mathbf{p}_i, \boldsymbol{\varphi}_i), F_{\mathbf{q}}(\mathbf{p}_i, \boldsymbol{\varphi}_i)] \in \mathbb{R}^{3+m}, \tag{3}$$

conditioned on time-varying deformation $\boldsymbol{\varphi}_i$.

3.3 Implicit Canonical Geometry and Appearance

Inspired by NeRF [41], we consider that a sample point $\mathbf{x} \in \mathbb{R}^{3+m}$ in the canonical hyper-space is associated with two properties: density σ and color $\mathbf{c} \in \mathbb{R}^3$.

Neural SDF. Notes that the object embeds in the $(3 + m)$-D canonical hyper-space. In this work, we represent its geometry as the zero-level set of an SDF:

$$S = \{\mathbf{x} \in \mathbb{R}^{3+m} | d(\mathbf{x}) = 0\}. \tag{4}$$

Following NeuS [56], we utilize a probability function to calculate the density value $\sigma(\mathbf{x})$ based on the estimated signed distance value, which is an unbiased and occlusion-aware approximation. We refer the reader to their paper for more details.

Implicit Rendering Network. We utilize a neural renderer $F_{\mathbf{c}}$ as the implicit appearance network. At time t_i, it takes in a 3D canonical coordinate $\mathbf{p}$, its corresponding normal, a view direction $\mathbf{v}_i$ as well as a geometry feature vector, then outputs the color of the point, conditioned on a time-varying appearance code $\boldsymbol{\psi}_i$. Specifically, we first compute its normal $\mathbf{n}_{\mathbf{p}} = \nabla_{\mathbf{p}} d(\mathbf{x})$ by gradient calculation. Then, the view direction $\mathbf{v}_{\mathbf{p}}$ in the canonical hyper-space can be obtained by transforming $\mathbf{v}_i$ with the Jacobian matrix $J_{\mathbf{p}}(\mathbf{p}_i) = \partial \mathbf{p} / \partial \mathbf{p}_i$ of the 3D canonical map $\mathbf{p}$ w.r.t $\mathbf{p}_i$. Except the SDF value, we adopt a larger MLP $F_d(\mathbf{x}) = (d(\mathbf{x}), \mathbf{z}(\mathbf{x}))$ to compute the embedded geometry feature $\mathbf{z}_{\mathbf{x}} = \mathbf{z}(\mathbf{x})$ to help the prediction of global shadow [64]. Finally, noticing $\mathbf{p}_i$ is the correspondence of $\mathbf{x}$ at time t_i, we can formulate its color $\mathbf{c}_i$ as:

$$\mathbf{c}_i = F_{\mathbf{c}}(\mathbf{p}, \mathbf{n}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}}, \mathbf{z}_{\mathbf{x}}, \boldsymbol{\psi}_i) = F_{\mathbf{c}}(\mathbf{p}, \nabla_{\mathbf{p}} d(\mathbf{x}), J_{\mathbf{p}}(\mathbf{p}_i)\mathbf{v}_i, \mathbf{z}(\mathbf{x}), \boldsymbol{\psi}_i). \tag{5}$$

It can be seen that the color of point $\mathbf{p}_i$ viewed from direction $\mathbf{v}_i$ depends on the deformation field, canonical representation, a deformation code as well an appearance code combined with time.

3.4 Optimization

Given an RGB-D sequence with the masks of interested object $\{(\mathcal{I}_i, \mathcal{D}_i, \mathcal{M}_i), i = 1, 2, \cdots, N\}$, the optimizable parameters include MLPs $\{F_{\mathbf{h}}, F_{\mathbf{q}}, F_d, F_{\mathbf{c}}\}$, learnable codes $\{\boldsymbol{\varphi}_i, \boldsymbol{\psi}_i\}$, RGB and depth camera intrinsics $\{\mathcal{K}_{\text{rgb}}, \mathcal{K}_{\text{depth}}\}$, as well as $\mathbf{SE}(3)$ camera pose $\mathcal{T}_i$ at each time t_i. Our target is to design the loss terms to match input masks, color images and depth images. Since we leverage neural implicit functions for representing the geometry, appearance and motion of dynamic object, we divide all constraints into two parts, on free-space points and on surface points:

$$\mathcal{L} = \underbrace{\Big(\lambda_1 \mathcal{L}_{\text{mask}} + \lambda_2 \mathcal{L}_{\text{color}} + \lambda_3 \mathcal{L}_{\text{depth}} + \lambda_4 \mathcal{L}_{\text{reg}}\Big)}_{\text{free-space}} + \underbrace{\Big(\lambda_5 \mathcal{L}_{\text{sdf}} + \lambda_6 \mathcal{L}_{\text{visible}}\Big)}_{\text{surface}}, \tag{6}$$

where $\lambda_j (j = 1, 2, \cdots, 6)$ are balancing weights.

Constraints on Free-space. Given a ray parameterized as $\mathbf{r}(s) = \mathbf{o} + s\mathbf{v}$ (pass through a pixel), we sample the implicit radiance field at points lying along this ray to approximate its color and depth:

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\sigma(s)\mathbf{c}(s)\,\mathrm{d}s, \quad \hat{\mathbf{D}}(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\sigma(s)s\,\mathrm{d}s, \tag{7}$$

where s_n and s_f represent near and far bounds, and $T(s) = \exp(-\int_{s_n}^{s} \sigma(u)\,\mathrm{d}u)$ denotes the accumulated transmittance along the ray. The density and color calculation are described in Sec. 3.3. Then the color and depth reconstruction loss are defined as:

$$\mathcal{L}_{\text{color}} = \sum_{\mathbf{r} \in \mathcal{R}(\mathcal{K}_{\text{rgb}}, \mathcal{T}_i)} \|M(\mathbf{r})(\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}))\|_1, \tag{8}$$

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r} \in \mathcal{R}(\mathcal{K}_{\text{depth}}, \mathcal{T}_i)} \|M(\mathbf{r})(\hat{\mathbf{D}}(\mathbf{r}) - \mathbf{D}(\mathbf{r}))\|_1, \tag{9}$$

where $\mathcal{R}(\mathcal{K}_{\text{rgb}}, \mathcal{T}_i)$ and $\mathcal{R}(\mathcal{K}_{\text{depth}}, \mathcal{T}_i)$ represent the set of rays to RGB and depth camera, respectively. $M(\mathbf{r}) \in \{0, 1\}$ is the object mask value, while $\mathbf{C}(\mathbf{r})$ and $\mathbf{D}(\mathbf{r})$ are the observed color and depth value. To focus on dynamic object reconstruction, we also defines a mask loss as

$$\mathcal{L}_{\text{mask}} = \text{BCE}(\hat{M}(\mathbf{r}), M(\mathbf{r})), \tag{10}$$

where $\hat{M}(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\sigma(s)\,\mathrm{d}s$ is the density accumulation along the ray, and BCE is the binary cross entropy loss.

An Eikonal loss is introduced to regularize $d(\mathbf{x})$ to be a signed distance function of $\mathbf{p}$, and it has the following form:

$$\mathcal{L}_{\text{reg}} = \sum_{\mathbf{x} \in \mathcal{X}} (\|\nabla_{\mathbf{p}} d(\mathbf{x})\|_2 - 1)^2, \tag{11}$$

where $\mathbf{x}$ are points sampled in the canonical hyper-space $\mathcal{X}$. In our implementation, to obtain $\mathbf{x}$, we first sample some points $\mathbf{p}_i$ on the observed free-space, and then deform sampled points back to $\mathcal{X}$ using Eq. 3. We constrain points sampled by a uniform and importance sampling strategy.

Constraints on Surface. Except for the losses on the free-space, we also constrain the property of points lying on the depth images $\mathcal{D}_i$. We add an SDF loss term:

$$\mathcal{L}_{\text{sdf}} = \sum_{\mathbf{p}_i \in \mathcal{D}_i} \|d(\mathbf{x})\|_1. \tag{12}$$

To avoid the deformed surface at each time fuses into the canonical space which causes multi-surfaces phenomenon, we design a visible loss term to constrain surface:

$$\mathcal{L}_{\text{visible}} = \sum_{\mathbf{p}_i \in \mathcal{D}_i} \max(\langle \frac{\mathbf{n}_{\mathbf{p}}}{\|\mathbf{n}_{\mathbf{p}}\|_2}, \frac{\mathbf{v}_{\mathbf{p}}}{\|\mathbf{v}_{\mathbf{p}}\|_2} \rangle, 0), \tag{13}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. This loss term tends to force the points sampled on depth images still visible from RGB-D camera.

4 Experiments

4.1 Experimental settings

Implementation details. We initialize $d(\mathbf{x})$ such that it approximates a unit sphere [2]. We train our neural networks using the ADAM optimizer [31] with a learning rate 5×10^{-4}. We run all of our experiments with 6×10^4 iterations for 12 hours on a single NVIDIA A100 40GB GPU. On free-space, we sample $2,048$ rays per batch (128 points along each ray). Following NeuS [56], we first uniformly sample 64 points, and then adopt importance sampling iteratively for 4 times (16 points each iteration). On depth map, we uniformly sample $2,048$ points per batch. For coarse-to-fine training, we utilize an incremental positional encoding strategy on sampled points, similar with Nerfies [44]. The weights in Eq. 6 are set as: $\lambda_1 = 0.1, \lambda_2 = 1.0, \lambda_3 = 0.5, \lambda_4 = 0.1, \lambda_5 = 0.5, \lambda_6 = 0.1$.

For non-rigid object segmentation, we leverage off-the-shelf methods, RVM [37] for human and MiVOS [14] for other objects. Since we assume the region of object is inside a unit sphere, we normalize the points back-projected from depth maps first. If the collected sequence implies larger global rotation, we leverage Robust ICP method [65] for per-frame initialization of poses $\mathcal{T}_i$.

Datasets. To evaluate our NDR and baseline approaches, we use 6 scenes from the DeepDeform [9] dataset, 7 scenes from the KillingFusion [51] dataset, and 11 scenes captured by ourselves. The evaluation data contains 6 classes: human faces, human bodies, domestic animals, plants, toys, and clothes. It includes challenging cases, such as rapid movement, self-rotation motion, topology change and complex shape. The DeepDeform [9] dataset is captured by an iPad. Its RGB-D streams are recorded and aligned at a resolution of 640×480 and 30 frames per second. Since our NDR does not need any annotated or estimated correspondences, we only leverage RGB-D sequences and camera intrinsics as initialization when evaluating NDR, without scene flow or optical flow data. We choose 6 scenes from the whole dataset, including human bodies, dogs, and clothes. All sequences in KillingFusion [51] dataset were recorded with a Kinect v1, also aligned to 640×480 resolution. We choose all scenes from it, which contain toys and human motions. To increase the data diversity, especially for adding more challenging but routine conditions (e.g., topology change and complex details), we capture some human head and plant videos with iPhone X (resolution 480×640 at 30 fps). When capturing head data, we ask the person to rotate the face while freely varying expressions. When capturing plant data, we record the states of leaf swings.

Comparison Methods. (1) A widely-used classical fusion-based method, DynamicFusion [42]: It is the pioneering work that estimates and utilizes the motion of hierarchical node graph for deforming guidance, and it assumes the shape inside a canonical TSDF volume. (2) Two recent fusion-based methods, DeepDeform [9] and Bozic et al. [8]: These methods utilize the learning-based correspondences to help handle challenging motions. (3) A state-of-the-art fusion-based method, OcclusionFusion [58]: It computes occlusion-aware 3D motion through a neural network for modeling guidance. (4) A state-of-the-art RGB reconstruction method from monocular video, BANMo [62]: It models articulated 3D shapes in a neural blend skinning and differentiable rendering framework. For comparison with RGB-D based methods, we use our re-implementation of DynamicFusion [42] and the results provided by the authors of OcclusionFusion [58].

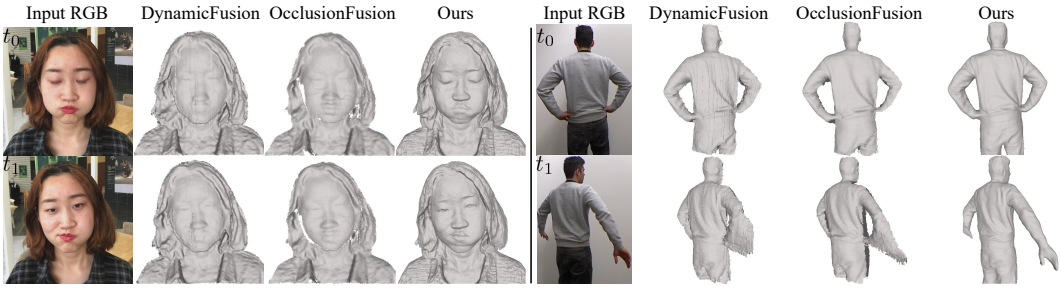

Figure 3: Qualitative comparisons with DynamicFusion [42] and OcclusionFusion [58] on our collected dataset and the KillingFusion [51] dataset.

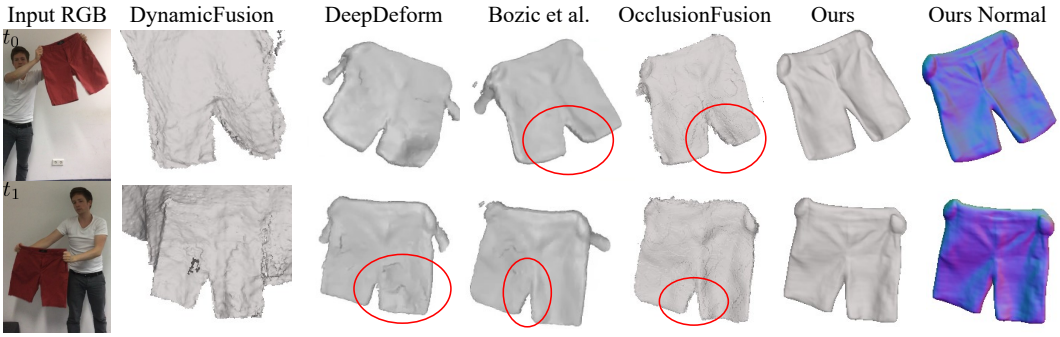

Figure 4: Qualitative comparisons with DynamicFusion [42], DeepDeform [9], Bozic et al. [8] and OcclusionFusion [58] on the DeepDeform [9] dataset. The results of DeepDeform [9] and Bozic et al. [8] are copied from the video of Bozic et al. [8].

4.2 Comparisons

RGB-D based methods. For qualitative evaluation, we exhibit some comparisons with DynamicFusion [42] and OcclusionFusion [58] in Fig. 3, also with DeepDeform [9] and Bozic et al. [8] in Fig. 4. Specifically, results of detailed modeling verify that bijective deformation mapping help match photometric correspondences between observed frames. As Fig. 3 shown, our NDR models geometry details while fusion-based methods [42, 58] are easy to form artifacts on the reconstructed surfaces. NDR also achieves considerable reconstruction accuracy on handling rapid movement (Fig. 4).

For quantitative evaluation, we calculate geometry errors on some testing sequences, following previous works [9, 8, 58]. The geometry metric is to compare depth values inside the object mask to the reconstructed geometry. The sequences are on behalf of various class objects and cases, including domestic animal (seq. *Dog* from DeepDeform [9]), rotated body, human-object interaction, general object (seq. *Alex, Hat, Frog* from KillingFusion [51], separately), and human heads (seq. *Human1, Human2* from our collected dataset). The quantitative results are shown in Tab. 1. We can see that our NDR outperforms previous works [42, 58], owing to joint optimizing geometry, appearance and motion on a total video. On seq. *Alex*, the geometry error of OccluionFusion [58] is lower than that of ours. However, NDR can handle topology varying well, as shown in the corresponding qualitative results on the right of Fig. 3.

Metric	Method	Dataset					
		DeepDeform	KillingFusion			Human Head	
		Dog	Alex	Hat	Frog	Human1	Human2
Mean	DynamicFusion	92.53	5.39	9.92	2.75	1.87	1.63
	OcclusionFusion	3.64	**3.75**	6.77	1.61	1.30	1.25
	Ours	**3.42**	4.24	**4.93**	**1.34**	**1.26**	**1.08**
Median	DynamicFusion	12.25	5.19	6.72	2.74	1.67	1.59
	OcclusionFusion	3.48	**3.41**	6.47	1.59	1.30	1.25
	Ours	**3.39**	4.11	**4.66**	**1.33**	**1.25**	**1.07**

Table 1: Quantitative results on 6 sequences. The geometry error represents the difference between reconstructed shape and depth values inside the mask. All values are in *mm*.

RGB based method. Fig. 5 exhibits several comparisons with a recent RGB based method - BANMo [62]. BANMo takes the RGB sequence as input and optimizes the geometry, appearance and motion based on the precomputed annotations, including the camera pose and optical flow. For a fair comparison, we also compare BANMo [62] with our NDR with only RGB supervision, where we provide them with the same camera initialization and frame-wise mask. For the RGB-only situation, both our method and BANMo may make some structural mistakes, such as the human arm in ours and the Snoopy's ears in BANMo. Moreover, compared to our RGB-only results, BANMo suffers more from the local geometry noise, which should be due to the error caused by incorrect precomputed annotations. Meanwhile, our method does not rely on any precomputed annotations and achieves flat results. With the RGB-D sequence as input, our NDR full model performs robust and well in modeling geometry details and rapid motions.

4.3 Ablation Studies.

We evaluate 3 components of our NDR regarding their effects on the final reconstruction result.

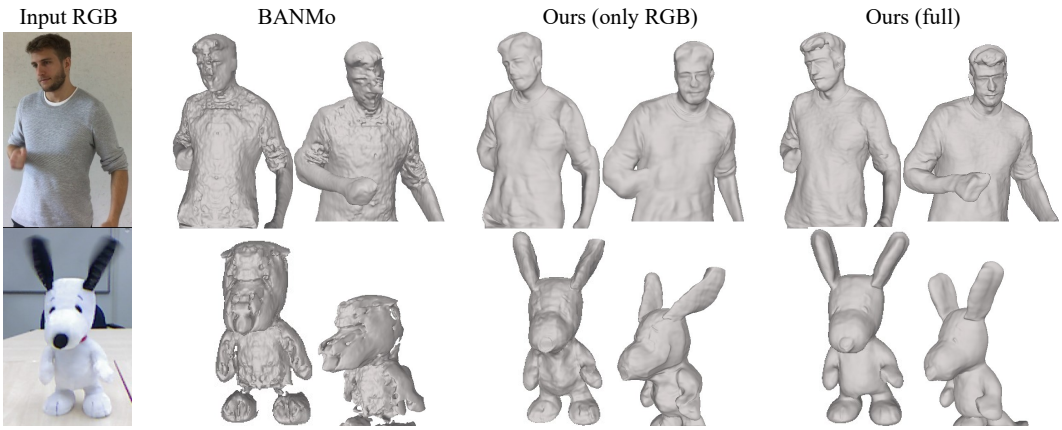

Figure 5: Qualitative comparison with BANMo [62] and ablation study on optimization without depth images on the DeepDeform [9] dataset and the KillingFusion [51] dataset. Each sample is shown in two view directions (the camera view and a novel view).

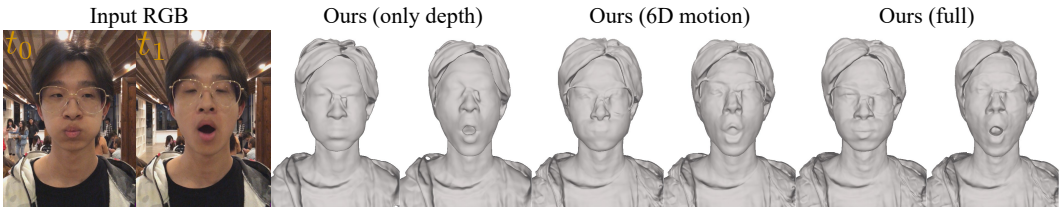

Figure 6: Ablation studies on optimization without RGB images and using 6D motion instead of the bijective map in the deformation field.

Depth cues. We evaluate the reconstruction results with only RGB supervision, i.e. removing depth images and only supervised with loss terms $\mathcal{L}_{\text{mask}}, \mathcal{L}_{\text{color}}, \mathcal{L}_{\text{reg}}$. As shown in Fig. 5, the reconstruction results with only RGB information are not correct (especially seen from a novel view) since monocular camera scenes exist the ambiguity of depth.

RGB cues. We also evaluate the reconstruction results with only depth supervision, i.e. removing RGB images and color loss term $\mathcal{L}_{\text{color}}$. As shown in Fig. 6, the reconstructed shapes lack geometry details as color messages are not used.

Bijective map. To verify the effect of our proposed bijective map $F_{\mathbf{h}}$ (Sec. 3.1), we change it to 6D deformation representation in $\mathbf{SE}(3)$ space. As shown in Fig. 6, since $F_{\mathbf{h}}$ can satisfy the cycle consistency strictly, it is less prone to accumulate artifacts and thus performs better in local geometry. In comparison, the irreversible transformation is easy to fail in preserving high-quality surfaces.

5 Conclusion

We have presented NDR, a new approach for reconstructing the high-fidelity geometry and motions of a dynamic scene from a monocular RGB-D video without any template priors. Other than previous works, NDR integrates observed color and depth into a canonical SDF and radiance field for joint optimization of surface and deformation. For maintaining cycle consistency throughout the whole video, we propose an invertible bijective mapping between observation space and canonical space, which fits perfectly with non-rigid motions. To handle topology change, we employ a topology-aware network to model topology-variant correspondence. On public datasets and our collected dataset, NDR shows a strong empirical performance in modeling different class objects and handling various challenging cases. Negative societal impact and limitation: like many other works with neural implicit representation, our method needs plenty of computational resources and optimization time, which can be a concern for energy resource consumption. We will explore some strategies to alleviate these problems in future work.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2565–2574, 2020.

[3] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[4] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning (ICML)*, pages 573–582. PMLR, 2019.

[5] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *Computer graphics forum (CGF)*, volume 22, pages 641–650. Wiley Online Library, 2003.

[6] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 187–194, 1999.

[7] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision (ECCV)*, pages 561–578. Springer, 2016.

[8] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18727–18737, 2020.

[9] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7002–7012, 2020.

[10] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 690–696. IEEE, 2000.

[11] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):1–10, 2014.

[12] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(3):413–425, 2013.

[13] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems (NeurIPS)*, 31, 2018.

[14] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5559–5568, 2021.

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR (Poster)*. OpenReview.net, 2017.

[16] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

[17] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5712–5721, 2021.

[18] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2017.

[20] Yudong Guo, Jianfei Cai, Boyi Jiang, Jianmin Zheng, et al. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 41(6):1294–1307, 2018.

[21] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5764–5774. IEEE, 2021.

[22] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2):1–17, 2019.

[23] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5052–5063, 2020.

[24] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[25] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (TOG)*, 34(4):1–14, 2015.

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

[27] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[28] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[30] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[32] Filippos Kokkinos and Iasonas Kokkinos. To the point: Correspondence-driven monocular 3d category reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[33] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1558–1567, 2019.

[34] Jiahui Lei and Kostas Daniilidis. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[35] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph. (TOG)*, 36(6):194–1, 2017.

[36] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6508, 2021.

[37] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 238–247, 2022.

[38] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7708–7717, 2019.

[39] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[40] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision (IJCV)*, 60(2):91–110, 2004.

[41] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision (ECCV)*, pages 405–421. Springer, 2020.

[42] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 343–352, 2015.

[43] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 5379–5389, 2019.

[44] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, 2021.

[45] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph. (TOG)*, 40(6), 2021.

[46] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3204–3215, 2021.

[47] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10318–10327, 2021.

[48] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Graph. (TOG)*, 36(6):245:1–245:17, 2017.

[49] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision (IJCV)*, 80(1):72–91, 2008.

[50] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. Neural dense non-rigid structure from motion with latent space constraints. In *European Conference on Computer Vision (ECCV)*, pages 204–222. Springer, 2020.

[51] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1395, 2017.

[52] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2646–2655, 2018.

[53] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *European conference on computer vision (ECCV)*, pages 438–451. Springer, 2010.

[54] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2387–2395, 2016.

[55] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12959–12970, 2021.

[56] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021.

[57] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2566–2576, 2019.

[58] Jun-Hai Yong Feng Xu Wenbin Lin, Chengwei Zheng. Occlusionfusion: Occlusion-aware motion estimation for real-time dynamic 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[59] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (TOG)*, 37(2):1–15, 2018.

[60] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15980–15989, 2021.

[61] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[62] Gengshan Yang, Minh Vo, Neverova Natalia, Deva Ramanan, Vedaldi Andrea, and Joo Hanbyul. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[63] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[64] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2492–2502, 2020.

[65] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[66] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[67] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2021.

[68] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum (CGF)*, volume 37, pages 523–550. Wiley Online Library, 2018.

[69] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6365–6373, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss the limitations in the end of Sec. 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We also discuss them in the end of Sec. 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We upload them as a part of the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We discuss them in Sec. 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We discuss them in Sec. 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]