BANMo: Building Animatable 3D Neural Models from Many Casual Videos

Gengshan Yang[2*] Minh Vo[3] Natalia Neverova[1] Deva Ramanan[2] Andrea Vedaldi[1] Hanbyul Joo[1]
[1]Meta AI [2]Carnegie Mellon University [3]Meta Reality Labs

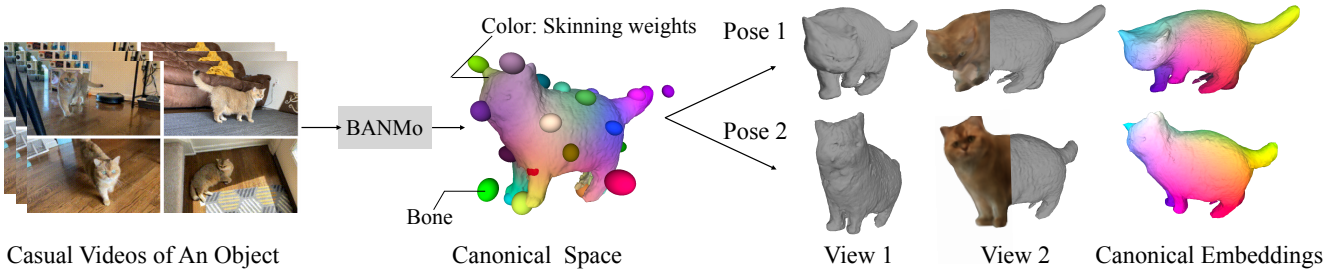

Figure 1. Given multiple casual videos capturing a deformable object, BANMo reconstructs an animatable 3D model, including an implicit canonical 3D shape, appearance, skinning weights, and time-varying articulations, without pre-defined shape templates or registered cameras. **Left**: Input videos; **Middle**: 3D shape, bones, and skinning weights (visualized as surface colors) in the canonical space; **Right**: Posed reconstruction at each time instance with color and canonical embeddings (correspondences are shown as the same colors).

Abstract

Prior work for articulated 3D shape reconstruction often relies on specialized sensors (e.g., multi-camera systems), or pre-built 3D deformable models (e.g., SMPL). Such methods do not scale to diverse sets of objects in the wild. We present a method that requires neither a specialized sensor nor a pre-defined template shape. It builds high-fidelity, articulated 3D models from many monocular casual videos in a differentiable rendering framework. Our key insight is to merge three schools of thought: (1) classic deformable shape models that make use of articulated bones and blend skinning, (2) canonical embeddings that establish correspondences between pixels and a canonical model, and (3) volumetric neural radiance fields (NeRFs) that are amenable to gradient-based optimization. We introduce neural blend skinning models that allow for differentiable and invertible articulated deformations. When combined with canonical embeddings, such models allow us to establish dense correspondences across videos that can be self-supervised with cycle consistency. On real and synthetic datasets, our method shows higher-fidelity 3D reconstructions than prior works for humans and animals, with the ability to render realistic images from novel viewpoints and poses. Project page: https://banmo-www.github.io/.

1. Introduction

We are interested in developing tools that can reconstruct accurate and animatable models of 3D objects from casually collected videos. A representative application is content creation for virtual and augmented reality, where the goal is to 3D-ify images and videos captured by users for consumption in a 3D space or creating animatable assets such as avatars. For rigid scenes, traditional Structure from Motion (SfM) approaches can be used to leverage large collection of uncontrolled images, such as images downloaded form the web, to build accurate 3D models of landmarks and entire cities [1, 45, 46]. However, these approaches do not generalize to deformable objects such as family members, friends or pets, which are often the focus of user content.

We are thus interested in reconstructing 3D deformable objects from *casually collected videos*. However, individual videos may not contain sufficient information to obtain good reconstruction of a given subject. Fortunately, we can expect that users may collect several videos of the same subjects, such as filming a family member over the span of several months or years. In this case, we wish our system to pool information from *all available videos* into a single 3D model, bridging any time discontinuity.

In this paper, we present **BANMo**, a **B**uilder of **A**nimatable 3D **N**eural **Mo**dels from multiple casual RGB videos. By consolidating the 2D cues from thousands of images into a fixed canonical space, BANMo learns a high-

*Work done when interning at Meta AI

fidelity neural implicit model for appearance, 3D shape, and articulations of the target non-rigid object. The articulation of the output model of BANMo is expressed by a neural blend skinning, similar to [5, 62, 63], making the output *animatable* by manipulating bone transformations. As shown in NRSfM [4], reconstructing a freely moving non-rigid object from monocular video is a highly under-constrained task where epipolar constraints are not directly applicable. In our approach, we address three core challenges: (1) how to represent 3D appearance and deformation of the target model in a canonical space; (2) how to find the mapping between canonical space to each individual frame; (3) how to find 2D correspondences across images under view and light changes, and object deformations.

Concretely, we utilize neural implicit functions [29] to represent color and 3D surface in the canonical space. This representation enables higher-fidelity 3D geometry reconstruction compared to approaches based on 3D meshes [62, 63]. The use of neural blending skinning in BANMo provides a way to constrain the deformation space of the target object, allowing better handling of pose pose variations and deformations with unknown camera parameters, compared to dynamic NeRF approaches [5, 22, 33, 38]. We also present a module for fine-grained registration between pixels and the canonical space by matching to an implicit feature volume. To jointly optimize over a large number of video frames with a manageable computational cost, we actively sample pixels locations based on uncertainty. In a nutshell, BANMo presents a way to merge the recent non-rigid object reconstruction approaches [62, 63] in a dynamic NeRF framework [5, 22, 33, 38], to achieve higher-fidelity non-rigid object reconstruction. We show experimentally that BANMo produces higher-fidelity 3D shape details than previous state-of-the art approaches [63], by taking better advantage of the large number of frames in multiple videos.

2. Related work

Human and animal body models. A large body of work in 3D human and animal reconstruction uses parametric shape models [25, 35, 59, 69, 70], which are built from registered 3D scans of real humans or toy animals, and serve to recover 3D shapes given a single image and 2D annotations or predictions (2D keypoints and silhouettes) at test time [2, 3, 15, 15, 68]. Although parametric body models achieve great success in reconstructing categories for which large amounts of ground-truth 3D data are available (mostly in the case of human reconstruction), it is challenging to apply the same methodology to categories with limited 3D data, such as animals and humans in diverse sets of clothing.

Category reconstruction from image/video collections. A number of recent methods build deformable 3D models of object categories from images or videos with weak 2D annotations, such as keypoints, object silhouettes, and optical flow, obtained from human annotators or predicted by off-the-shelf models [7, 12, 16, 20, 21, 58, 66]. Such methods often rely on a coarse shape template [18, 53], and are not able to recover fine-grained details or large articulations. Recently, HDNet [10] leverages social media videos and DensePose human model to learn high-fidelity depth estimators for clothed human.

Category-agnostic video shape reconstruction. Non-rigid structure from motion (NRSfM) methods [4, 8, 17, 19, 43] reconstruct non-rigid 3D shapes from a set of 2D point trajectories in a class-agnostic way. However, due to difficulties in obtaining accurate long-range correspondences [40, 49], they do not work well for videos in the wild. Recent efforts such as LASR and ViSER [62, 63] reconstruct articulated shapes from a monocular video with differentiable rendering. As our results show, they may still produce blurry geometry and unrealistic articulations.

Neural radiance fields. Prior works on NeRF optimize a continuous scene function for novel view synthesis given a set of images, often assuming the scene is rigid and camera poses can be accurately registered to the background [11, 23, 27–29, 57]. To extend NeRF to dynamic scenes, recent works introduce additional functions to deform observed points to a canonical space or over time [22, 33, 34, 38, 52, 55]. However, they heavily rely on background registration, and fail when the motion between objects and background is large. Moreover, the deformations cannot be explicitly controlled by user inputs. Similar to our goal, some recent works [24, 32, 36, 37, 47] produce pose-controllable NeRFs, but they rely on a human body model, or synchronized multi-view video inputs.

3. Method

We model the deformable object in a canonical time-invariant space, i.e. the "rest" body pose space, that can be transformed to the "articulated" pose in the camera space at each time instance with forward mappings, and transform back with backward mappings. We use implicit functions to represent the 3D shape, color, and dense semantic embeddings of the object. Our neural 3D model can be deformed and rendered into images at each time instance via differentiable volume rendering, and optimized to ensure consistency between the rendered images and multiple cues in the observed images, including color, silhouette, optical flow, and 2D pixel feature embeddings. We refer readers to an overview in Fig. 2 and a list of notations in the supplement.

We employ neural blend skinning to express object articulations similarly to [18, 62] but modify it for implicit surface representations rather than meshes. Our self-supervised semantic feature embedding produces dense pixelwise correspondences across frames of different videos, which is critical for optimization on large video collections.

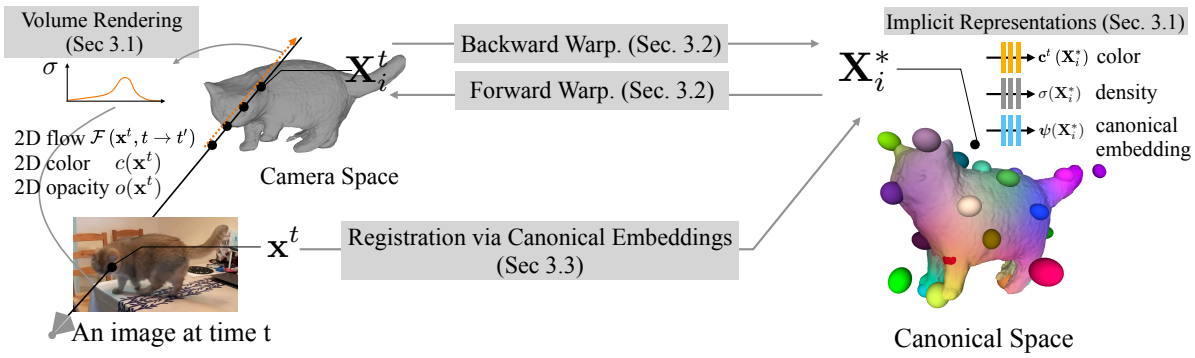

Figure 2. **Method overview.** BANMo optimizes a set of shape and deformation parameters (Sec. 3.1) that describe the video observations in pixel colors, silhouettes, optical flow, and higher-order features descriptors, based on a differentiable volume rendering framework. BANMo uses a neural blend skinning model (Sec. 3.2) to transform 3D points between the camera space and the canonical space, enabling handling large deformations. To register pixels across videos, BANMo jointly optimizes an implicit canonical embedding (CE) (Sec. 3.3).

3.1. Shape and Appearance Model

We first represent shape and appearance of deformable objects in a canonical time-invariant *rest pose* space and then model time-dependent deformations by a neural blend skinning function (Sec. 3.2).

Canonical shape model. In order to model the shape and appearance of an object in a canonical space, we use a method inspired by Neural Radiance Fields (NeRF) [29]. A 3D point $\mathbf{X}^* \in \mathbb{R}^3$ in a canonical space is associated with three properties: color $\mathbf{c} \in \mathbb{R}^3$, density $\sigma \in [0,1]$, and a learned canonical embedding $\boldsymbol{\psi} \in \mathbb{R}^{16}$. These properties are predicted by the Multilayer Perceptron (MLP) networks:

$$\mathbf{c}^t = \mathbf{MLP}_{\mathbf{c}}(\mathbf{X}^*, \mathbf{v}^t, \boldsymbol{\omega}_e^t), \quad (1)$$

$$\sigma = \Gamma_\beta(\mathbf{MLP}_{\text{SDF}}(\mathbf{X}^*)), \quad (2)$$

$$\boldsymbol{\psi} = \mathbf{MLP}_{\boldsymbol{\psi}}(\mathbf{X}^*). \quad (3)$$

As in NeRF, color $\mathbf{c}^t$ depends on a time-varying view direction $\mathbf{v}^t \in \mathbb{R}^2$ and a learnable environment code $\boldsymbol{\omega}_e^t \in \mathbb{R}^{64}$ that is designed to capture environment illumination conditions [27], shared across frames in the same video.

The shape is given by $\mathbf{MLP}_{\text{SDF}}$, computing the Signed-Distance Function (SDF) of a point to the surface. To convert SDF to density $\sigma \in [0,1]$ for volume rendering, we use $\Gamma_\beta(\cdot)$, the cumulative of a unimodal distribution with zero mean and β scale, such as Sigmoid(βx). β is a single learnable parameter that controls the solidness of the object, approaching zero for solid objects [56, 64]. In practice, we use the cumulative of Laplace distribution. Compared with ReLU of Softplus, it provides a principled way of extracting surface as the zero level-set of the SDF.

Finally, the network $\boldsymbol{\psi}$ maps points to a feature descriptor (or canonical embedding) that can be matched by pixels from different viewpoints, enabling long-range correspondence across frames and videos. This feature can be interpreted as a variant of Continuous Surface Embeddings (CSE) [30] but defined volumetrically and fine-tuned in a self-supervised manner (described in Sec 3.3).

Space-time warping model. We consider a pair of time-dependent warping functions: *forward warping function* $\mathcal{W}^{t,\rightarrow} : \mathbf{X}^* \rightarrow \mathbf{X}^t$ mapping canonical location $\mathbf{X}^*$ to camera space location $\mathbf{X}^t$ at current time and the *backward warping function* $\mathcal{W}^{t,\leftarrow} : \mathbf{X}^t \rightarrow \mathbf{X}^*$ for inverse mapping.

Prior work such as Nerfies [33] and Neural Scene Flow Fields (NSFF) [22] learn deformations assuming that (1) camera transformations are given, and (2) the residual object deformation is small. As detailed in Sec. 3.2 and Sec. 3.4, we do not make such assumptions; instead, we adopt a neural blend-skinning model that can handle large deformations, but without assuming a pre-defined skeleton.

Volume rendering. To render images, we use the volume rendering in NeRF [29], but modified to warp the 3D ray to account for the deformation [33]. Specifically, let $\mathbf{x}^t \in \mathbb{R}^2$ be the pixel location at time t, and $\mathbf{X}_i^t \in \mathbb{R}^3$ be the i-th 3D point sampled along the ray emanates from $\mathbf{x}^t$. The color $\mathbf{c}$ and the opacity $o \in [0,1]$ of the pixel are given by:

$$\mathbf{c}(\mathbf{x}^t) = \sum_{i=1}^{N} \tau_i \mathbf{c}^t \left(\mathcal{W}^{t,\leftarrow}\left(\mathbf{X}_i^t\right)\right), \quad o(\mathbf{x}^t) = \sum_{i=1}^{N} \tau_i,$$

where N is the number of samples, τ_i is the probability $\mathbf{X}_i^t$ visible to the camera and is given by $\tau_i = \prod_{j=1}^{i-1} p_j(1-p_i)$. Here $p_i = \exp\left(-\sigma_i \delta_i\right)$ is the probability that the photon is transmitted through the interval δ_i between the i-th $\mathbf{X}_i^t$ sample and the next, and $\sigma_i = \sigma\left(\mathcal{W}^{t,\leftarrow}\left(\mathbf{X}_i^t\right)\right)$ is the density from Eq. 2. Note we *pull back* the ray points in the camera space to the canonical space using the warping $\mathcal{W}^{t,\leftarrow}$, as the color and density are defined in the canonical space.

Besides color and opacity, we compute the expected ray-surface intersection in the canonical space:

$$\mathbf{X}^*(\mathbf{x}^t) = \sum_{i=1}^{N} \tau_i \left(\mathcal{W}^{t,\leftarrow}(\mathbf{X}_i^t)\right). \quad (4)$$

To render 2D flow, we *push forward* the backward warped ray points to another time t' via forward warping $\mathcal{W}^{t',\rightarrow}$to find its expected 2D re-projection:

$$\mathbf{x}^{t'} = \sum_{i=1}^{N} \tau_i \Pi^{t'} \left(\mathcal{W}^{t',\rightarrow} \left(\mathcal{W}^{t,\leftarrow}(\mathbf{X}_i^t) \right) \right), \tag{5}$$

where $\Pi^{t'}$ is the projection matrix of a pinhole camera. We optimize video-specific $\Pi^{t'}$ given a rough initialization. With this, we compute a 2D flow rendering as:

$$\mathcal{F}\left(\mathbf{x}^t, t \rightarrow t'\right) = \mathbf{x}^{t'} - \mathbf{x}^t. \tag{6}$$

3.2. Deformation Model via Neural Blend Skinning

We define mappings $\mathcal{W}^{t,\rightarrow}$ and $\mathcal{W}^{t,\leftarrow}$ based on a neural blend skinning model approximating articulated body motion. Defining invertible warps for neural deformation representations is difficult [5]. Our formulation represents 3D warps as compositions of weighted *rigid-body transformations*, each of which is differentiable and invertible.

Blend skinning deformation. Given a 3D point $\mathbf{X}^t$ at time t, we wish to find its corresponding 3D point $\mathbf{X}^*$ in the canonical space. Conceptually, $\mathbf{X}^*$ can be considered as points in the "rest" pose at a fixed camera view point. Our formulation finds mappings between $\mathbf{X}^t$ and $\mathbf{X}^*$ by blending the rigid transformations of B bones (3D coordinate systems). Let $\mathbf{G}^t \in SE(3)$ be a root body transformation of the object from canonical space to time t, and $\mathbf{J}_b^t \in SE(3)$ be a rigid transformation that moves the b-th bone from its "zero-configuration" to deformed state t, then we have

$$\mathbf{X}^t = \mathcal{W}^{t,\rightarrow}(\mathbf{X}^*) = \mathbf{G}^t\big(\mathbf{J}^{t,\rightarrow}\mathbf{X}^*\big), \tag{7}$$

$$\mathbf{X}^* = \mathcal{W}^{t,\leftarrow}(\mathbf{X}^t) = \mathbf{J}^{t,\leftarrow}\big((\mathbf{G}^t)^{-1}\mathbf{X}^t\big), \tag{8}$$

where $\mathbf{J}^{t,\rightarrow}$ and $\mathbf{J}^{t,\leftarrow}$ are weighted averages of B rigid transformations $\{\Delta\mathbf{J}_b^t\}_{b\in B}$ that move the bones between rest configurations (denoted as $\mathbf{J}_b^*$) and time t configurations $\mathbf{J}_b^t$, following linear blend skinning deformation [9]:

$$\mathbf{J}^{t,\rightarrow} = \sum_{b=1}^{B} \mathbf{W}_b^{t,\rightarrow} \Delta\mathbf{J}_b^t, \quad \mathbf{J}^{t,\leftarrow} = \sum_{b=1}^{B} \mathbf{W}_b^{t,\leftarrow} (\Delta\mathbf{J}_b^t)^{-1}. \tag{9}$$

$\mathbf{W}_b^{t,\rightarrow}$ and $\mathbf{W}_b^{t,\leftarrow}$ represent blend skinning weights for point $\mathbf{X}^*$ and $\mathbf{X}^t$ relative to the b-th bone (described further below), and $\Delta\mathbf{J}_b^t = \mathbf{J}_b^t \mathbf{J}_b^{*-1}$.

Latent pose code. We represent root pose $\mathbf{G}^t$ and body pose $\mathbf{J}_b^*, \mathbf{J}_b^t$ with angle-axis rotations and 3D translations, regressed from MLPs:

$$\mathbf{G}^t = \mathbf{MLP}_{\mathbf{G}}(\boldsymbol{\omega}_r^t), \quad \mathbf{J}_b^t = \mathbf{MLP}_{\mathbf{J}}(\boldsymbol{\omega}_b^t) \tag{10}$$

where $\boldsymbol{\omega}_r^t$ and $\boldsymbol{\omega}_b^t$ are 128-dimensional latent codes for root pose and body pose at frame t respectively. Similarly, we have $\mathbf{J}_b^* = \mathbf{MLP}_{\mathbf{J}}(\boldsymbol{\omega}_b^*)$ and $\boldsymbol{\omega}_b^*$ is the 128-dimensional latent code for the rest body pose. To stabilize the optimization of poses, we leverage temporal smoothness by representing body pose code with a Fourier embedding [29] $\boldsymbol{\omega}_b^t = \mathcal{F}(\tilde{t})$, where the $\tilde{t} = \frac{t}{\max_{i=1}^{M} |t_i|}$ is normalized across videos, and the maximum frequency of Fourier basis is determined by the sampling rate after normalization. Compared with directly optimizing SE(3) poses, we find such over-parameterized representations converges faster with stochastic first-order gradient methods.

Skinning weights. Similar to SCANimate [39], we define a skinning weight function $\mathcal{S} : (\mathbf{X}, \boldsymbol{\omega}_b) \rightarrow \mathbf{W} \in \mathbb{R}^B$ that assigns $\mathbf{X}$ to bones given body pose code $\boldsymbol{\omega}_b$. To compute the forward and backward skinning weights in Eq. 9, we apply $\mathcal{S}$ separately at rest pose as well as time t pose, and we have $\mathbf{W}^{t,\rightarrow} = \mathcal{S}(\mathbf{X}^*, \boldsymbol{\omega}_b^*)$, $\mathbf{W}^{t,\leftarrow} = \mathcal{S}(\mathbf{X}^t, \boldsymbol{\omega}_b^t)$.

Directly representing $\mathcal{S}$ as neural networks can be difficult to optimize. Therefore, we condition neural skinning weights on explicit 3D Gaussian ellipsoids that move along with the bone coordinates. Following LASR [62], the Gaussian skinning weights are determined by the Mahalanobis distance between $\mathbf{X}$ and the Gaussian ellipsoids:

$$\mathbf{W}_\sigma = (\mathbf{X} - \mathbf{C}_b)^T \mathbf{Q}_b (\mathbf{X} - \mathbf{C}_b), \tag{11}$$

where $\mathbf{C}_b$ is the bone center and $\mathbf{Q}_b = \mathbf{V}_b^T \mathbf{\Lambda}_b^0 \mathbf{V}_b$ is the precision matrix composed by bone orientation matrix $\mathbf{V}_b$ and scale $\mathbf{\Lambda}_b^0$. Bone centers and orientations are transformed from the "zero-configurations" as $\left(\mathbf{V}_b|\mathbf{C}_b\right) = \mathbf{J}_b \left(\mathbf{V}_b^0|\mathbf{C}_b^0\right)$. $\mathbf{\Lambda}_b^0$, $\mathbf{V}_b^0$ and $\mathbf{C}_b^0$ are learnable parameters and $\mathbf{J}_b$ are time-variant body poses parameters determined by body pose code $\boldsymbol{\omega}_b^*$ and $\boldsymbol{\omega}_b^t$.

To model the skinning weights for fine geometry, we find it helpful to add delta skinning weights after the coarse component is well-optimized. Delta skinning weights are represented as a coordinated-MLP $\mathbf{W}_\Delta = \mathbf{MLP}_\Delta(\mathbf{X}, \boldsymbol{\omega}_b)$. The final skinning function is the softmax-normalized sum of the coarse and fine components,

$$\mathbf{W} = \mathcal{S}(\mathbf{X}, \omega_b) = \sigma_{\text{softmax}}\big(\mathbf{W}_\sigma + \mathbf{W}_\Delta\big). \tag{12}$$

The Gaussian component regularizes the skinning weights to be spatially smooth and temporally consistent, and handles large deformations better than purely implicitly-defined ones. Furthermore, our formulation of the skinning weights are dependent on only pose status by construction, and therefore regularizes the space of skinning weights.

3.3. Registration via Canonical Embeddings

To register pixel observations at different time instances, BANMo maintains a canonical feature embedding that encodes semantic information of 3D points in the canonical space, which can be uniquely matched by the pixel features, and provide strong cues for registration via a joint optimization of shape, articulations, and embeddings (Sec. 3.4).

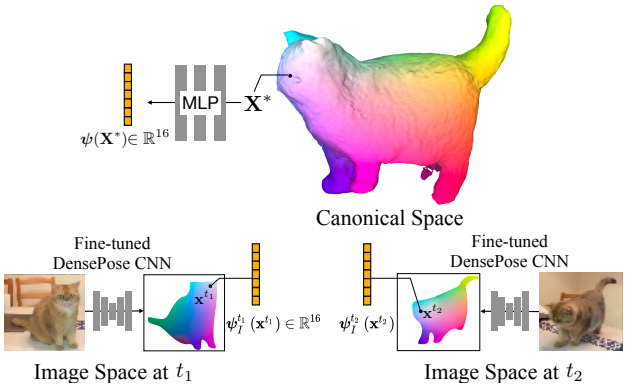

Figure 3. **Canonical Embeddings.** We jointly optimize an implicit function to produce canonical embeddings from 3D canonical points that match to the 2D DensePose CSE embeddings [30].

Canonical embeddings matching. Given a pixel at $\mathbf{x}^t$ of frame t, our goal is to find a point $\mathbf{X}^*$ in the canonical space whose feature embedding $\psi(\mathbf{X}^*) \in \mathbb{R}^{16}$ best matches the pixel feature embedding $\psi_I^t(\mathbf{x}^t) \in \mathbb{R}^{16}$. The pixel embeddings ψ_I^t (of frame t) are computed by a CNN. Different from ViSER [63] that learns embeddings from scratch, we initialize pixel embeddings with CSE [30, 31] that produces consistent features for semantically corresponding pixels, and optimize pixel and canonical embeddings jointly. Recall that the embedding of a canonical 3D point is computed as $\psi(\mathbf{X}^*) = \mathbf{MLP}_\psi(\mathbf{X}^*)$ in Eq. 3. Intuitively, $\mathbf{MLP}_\psi$ is optimized to ensure the output 3D descriptor matches 2D descriptors of corresponding pixels across multiple views. To compute the 3D surface point corresponding to a 2D point $\mathbf{x}^t$, we apply soft argmax descriptor matching [13, 26]:

$$\hat{\mathbf{X}}^*(\mathbf{x}^t) = \sum_{\mathbf{X} \in \mathbf{V}^*} \tilde{\mathbf{s}}^t(\mathbf{x}^t)\mathbf{X}, \tag{13}$$

where $\mathbf{V}^*$ are sampled points in a canonical 3D grid, whose size is dynamically determined during optimization (see supplement), and $\tilde{\mathbf{s}}$ is a normalized feature matching distribution over the 3D grid: $\tilde{\mathbf{s}}^t(\mathbf{x}^t) = \sigma_{\text{softmax}}\Big(\alpha_s \big\langle \psi_I^t(\mathbf{x}^t), \psi(\mathbf{X}) \big\rangle\Big)$, where α_s is a learnable scaling to control the peakness of the softmax function and $\langle ., . \rangle$ is the cosine similarity score.

Self-supervised canonical embedding learning. As describe later in Eq. 15-16, the canonical embedding is self-supervised by enforcing the consistency between feature matching and geometric warping. By jointly optimizing the shape and articulation parameters via consistency losses, canonical embeddings provide strong cues to register pixels from different time instance to the canonical 3D space, and enforce a coherent reconstruction given observations from multiple videos, as validated in ablation studies (Sec. 4.3).

3.4. Optimization

Given multiple videos, we optimize all parameters described above, including MLPs, $\{\mathbf{MLP_c}, \mathbf{MLP}_{\text{SDF}}, \mathbf{MLP}_\psi, \mathbf{MLP_G}, \mathbf{MLP_J}, \mathbf{MLP}_\Delta\}$, learnable codes $\{\boldsymbol{\omega}_e^t, \boldsymbol{\omega}_r^t, \boldsymbol{\omega}_b^t, \boldsymbol{\omega}_b^*\}$ and pixel embeddings ψ_I.

Losses. The model is learned by minimizing three types of losses: reconstruction losses, feature registration losses, and a 3D cycle-consistency regularization loss:

$$\mathcal{L} = \underbrace{\big(\mathcal{L}_{\text{sil}} + \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{OF}}\big)}_{\text{reconstruction losses}} + \underbrace{\big(\mathcal{L}_{\text{match}} + \mathcal{L}_{\text{2D-cyc}}\big)}_{\text{feature registration losses}} + \mathcal{L}_{\text{3D-cyc}}.$$

Reconstruction losses are similar to those in existing differentiable rendering pipelines [29, 65]. Besides color reconstruction loss $\mathcal{L}_{\text{rgb}}$ and silhouette reconstruction loss $\mathcal{L}_{\text{sil}}$, we further compute flow reconstruction losses $\mathcal{L}_{\text{OF}}$ by comparing the rendered $\mathcal{F}$ defined in Eq. 6 with the observed 2D optical flow $\hat{\mathcal{F}}$ computed by an off-the-shelf flow network:

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{x}^t} \big\|\mathbf{c}(\mathbf{x}^t) - \hat{\mathbf{c}}(\mathbf{x}^t)\big\|^2, \ \mathcal{L}_{\text{sil}} = \sum_{\mathbf{x}^t} \big\|\mathbf{o}(\mathbf{x}^t) - \hat{\mathbf{s}}(\mathbf{x}^t)\big\|^2,$$

$$\mathcal{L}_{\text{OF}} = \sum_{\mathbf{x}^t, (t,t')} \Big\|\mathcal{F}\left(\mathbf{x}^t, t \to t'\right) - \hat{\mathcal{F}}\left(\mathbf{x}^t, t \to t'\right)\Big\|^2, \tag{14}$$

where $\hat{\mathbf{c}}$ and $\hat{\mathbf{s}}$ are observed color and silhouette. Additionally, we define feature matching losses to enforce 3D points predicted via canonical embedding $\hat{\mathbf{X}}^*(\mathbf{x}^t)$ (Eq. 13) to match the prediction from backward warping (Eq. 4):

$$\mathcal{L}_{\text{match}} = \sum_{\mathbf{x}^t} \Big\|\hat{\mathbf{X}}^*(\mathbf{x}^t) - \mathbf{X}^*(\mathbf{x}^t)\Big\|_2^2, \tag{15}$$

and a 2D cycle consistency loss [18, 63] that forces the image projection after forward warping of $\hat{\mathbf{X}}^*(\mathbf{x}^t)$ to land back on its original 2D coordinates:

$$\mathcal{L}_{\text{2D-cyc}} = \sum_{\mathbf{x}^t} \Big\|\Pi^t\Big(\mathcal{W}^{t,\rightarrow}(\hat{\mathbf{X}}^*(\mathbf{x}^t))\Big) - \mathbf{x}^t\Big\|_2^2. \tag{16}$$

Similar to NSFF [22], we regularize the deformation function $\mathcal{W}^{t,\rightarrow}(\cdot)$ and $\mathcal{W}^{t,\leftarrow}(\cdot)$ by a 3D cycle consistency loss, which encourages a sampled 3D point in the camera coordinates to be backward deformed to the canonical space and forward deformed to its original location:

$$\mathcal{L}_{\text{3D-cyc}} = \sum_i \tau_i \Big\|\mathcal{W}^{t,\rightarrow}\Big(\mathcal{W}^{t,\leftarrow}(\mathbf{X}_i^t)\Big) - \mathbf{X}_i^t\Big\|_2^2, \tag{17}$$

where τ_i is the opacity that weighs the sampled points so that a point near the surface receives heavier regularization.

Our optimization is highly non-linear with local minima, and we consider two strategies for robust optimization.

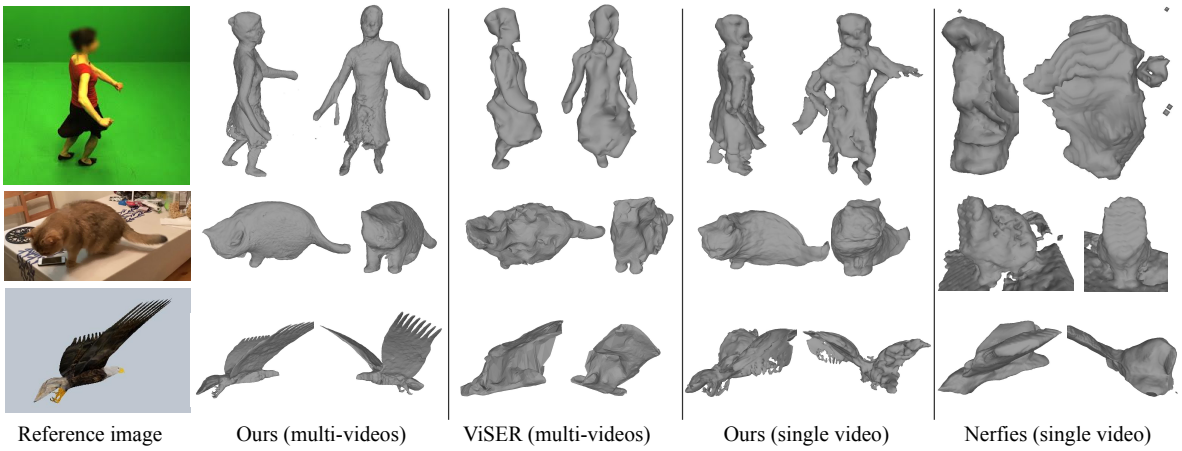

Figure 4. **Qualitative comparison of our method with prior art [33, 63].** From top to bottom: AMA's `samba`, `casual-cat`, `eagle`.

Root pose initialization. Due to ambiguities between object geometry and root poses, we find it helpful to provide a rough per-frame initialization of root poses ($\mathbf{G}^t$ in Eq. 7), similar to NeRS [67]. Specifically, we train a separate network PoseNet , which is applied to every test video frame. Similar to DenseRaC [60], PoseNet takes a DensePose CSE [30] feature image as input and predicts the root pose $\mathbf{G}_0^t = \text{PoseNet}(\boldsymbol{\psi}_I^t)$, where $\boldsymbol{\psi}_I^t \in \mathbb{R}^{112\times112\times16}$ is the embedding output of DensePose CSE [30] from an RGB image I_t. We train PoseNet by a synthetic dataset produced offline. See supplement for details on training. Given the pre-computed $\mathbf{G}_0^t$, BANMo only needs to compute a delta root pose via MLP:

$$\mathbf{G}^t = \mathbf{MLP}_{\mathbf{G}}(\boldsymbol{\omega}_r^t)\mathbf{G}_0^t. \tag{18}$$

Active sampling over (x, y, t). Inspired by iMAP [48], our sampling strategy follows an easy-to-hard curriculum. At the early iterations, we randomly sample a batch of N^p pixels for volume rendering and compute reconstruction losses. At the same time, we optimize a compact 5-layer MLP function to represent the uncertainty over the image coordinate and frame index: $\hat{\mathbf{U}}(x, y, t) = \mathbf{MLP}_{\mathbf{U}}(x, y, t)$. The uncertainty MLP is optimized by comparing against the color reconstruction errors in the current forward step:

$$\mathcal{L}_{\mathbf{U}} = \sum_{\mathbf{x},t} \left\| \mathcal{L}_{\text{rgb}}(\mathbf{x}^t) - \hat{\mathbf{U}}(\mathbf{x}^t) \right\|. \tag{19}$$

Note that the gradient from $\mathcal{L}_{\mathbf{U}}$ to $\mathcal{L}_{\text{rgb}}(\mathbf{x}^t)$ is stopped such that $\mathcal{L}_{\mathbf{U}}$ does not generate gradients to parameters besides $\mathbf{MLP}_{\mathbf{U}}$. After 40% of the optimization steps, we replace half of the samples with *active* samples from pixels with high uncertainties. To do so, we randomly sample $N^{a'} = 24576$ pixels, and evaluate their uncertainties by passing their coordinates and frame indices to $\mathbf{MLP}_{\mathbf{U}}$. Active samples dramatically improves reconstruction fidelity, as shown in Fig. 8.

4. Experiments

Implementation details. Our implementation of implicit shape and appearance models follows NeRF [29], except that our shape model outputs SDF, which is transformed to density for volume rendering. To extract the rest surface, we find the zero-level set of SDF by running marching cubes on a 256^3 grid. To obtain articulated shapes at each time instance, we articulate points on the rest surface with forward deformation $\mathcal{W}^{t,\rightarrow}$.

Optimization details. We initialize $\mathbf{MLP}_{\text{SDF}}$ such that it approximates a unit sphere [65]. We use $B = 25$ rest bones, which are initialized with unit scale, identity orientation, and centers uniformly spaced on the initial rest surface. During optimization, we reinitialize the rest bones at $\{20\%, 67\%\}$ of total iterations and further encourage them to stay close to the surface with a sinkhorn divergence loss [6]. In a batch, we sample $N^I = 512$ image pairs and sub-sample $N^p = 6144$ pixels for rendering. The interval between image pairs is randomly chosen $\Delta T \in \{1, 2, 4, 8, 16, 32\}$. To stabilize optimization, we find N_I needs to roughly match the number of input frames. The reconstruction quality improves with more iterations and we find 36k iterations (15 hours on a V100 GPU) already produces high-fidelity details. Please find a list of hyperparameters in supplement.

4.1. Dataset and Metrics

Qualitative: Casual videos dataset. We demonstrate BANMo's ability to reconstruct 3D models from casual videos of animals and humans. Object silhouette and op-

Table 1. **Difference between Nerfies, ViSER, and BANMo.**

Method	shape	deformation	registration
Nerfies	implicit	dense SE(3)	photometric
ViSER	mesh	control points	self-supervised feature
BANMo	implicit	control points	CSE feature

tical flow (for computing reconstruction losses Eq. 14) are extracted by off-the-shelf models, PointRend and VCN-robust [14,61]. Two special challenges arise from the casual nature of the video captures. First, each video collection contains around 1k images, an order of magnitudes larger those used in prior work [22, 29, 33, 63], which requires the method to handle reconstructions at a larger scale. Second, the dataset makes no control over camera movement or object movement. In particular, objects freely moves in a video and background changes across videos, posing challenges to standard SfM camera registration pipelines. We show results on 11 videos (totaling 900 images) of a British shorthair cat denoted as `casual-cat` below. Please find other results in the project webpage.

Quantitative: AMA human dataset. Articulated Mesh Animation (AMA) dataset [54] contains multi-view videos captured by 8 synchronized cameras. It provides high-fidelity ground-truth meshes with clothing. We use 2 sets of videos of the same actor (`swing` and `samba`), totaling 2600 frames, as the input to optimization. We use the ground-truth object silhouettes. Time synchronization and camera extrinsics are *not* used.

Quantitative: Animated Objects dataset. We download free animated 3D models from TurboSquid, including an `eagle` model and a model for human `hands`. We render them from different camera trajectories with partially overlapping motions. Each animated object is rendered as 5 videos with 150 frames per video. We provide ground-truth root poses and object silhouettes to BANMo and baselines.

Metrics. We quantify the results using both Chamfer distances and F-scores. Chamfer distance computes the average distance between the ground-truth and the estimated surface points by finding the nearest neighbour matches, but it is sensitive to outliers. Therefore, we further report the F-score at distance thresholds $d = 2\%$ of the longest edge of the axis-aligned object bounding box [50]. To account for the unknown scale and global rigid motion, we pre-align the estimated shape to the ground-truth via Iterative Closest Point (ICP) up to a 3D similarity transformation.

4.2. Reconstruction Results

We compare with Nerfies and ViSER and summarize the differences in Tab. 1. We show qualitative comparison in Fig. 4 and quantitative comparison in Tab. 2.

Baseline setup. Nerfies [33] is designed for a single continuously captured video, assuming object root body pose can be compensated by background-SfM. In our setup, object moves and background SfM does not provide root poses for the object. When focused on the deformable object, SfM (such as COLMAP) failed to converge due to violation of rigidity, leading to very few successful registrations (18 over 900 images registered on `casual-cat`). To make a fair comparison, we provide Nerfies with rough initial root poses (obtained from our PoseNet, Sec. 3.4). After optimization, meshes are extracted by running marching cubes on a 256^3 grid. Another baseline, ViSER [63], directly optimizes object shape and poses using optical flow, silhouette, and color reconstruction losses. It does not assume category-level priors such as CSE features, and therefore applicable to generic object categories. However, ViSER's root pose estimation is sensitive to large deformation and a large number of input frames (more than 20). Since it produces worse results than our PoseNet, we provide ViSER the same root poses from our initialization pipeline.

Table 2. **Quantitative results on AMA and Animated Objects.** 3D Chamfer distance (cm, ↓) and F-score (%, ↑) averaged over all frames. The 3D models for `eagle` and `hands` are resized such that the longest edge of the axis-aligned object bounding box is 2m. * *with ground-truth root poses.* S: single-video results. All methods are assigned with the same initial root pose.

Method	`AMA-swing`		`Eagle`*		`Hands`*	
	CD	F@2%	CD	F@2%	CD	F@2%
Ours	**9.1**	**57.0**	**8.1**	**56.7**	**7.5**	**49.6**
ViSER	15.7	52.2	23.0	20.6	16.8	21.3
Ours^S	9.4	56.8	10.8	48.6	10.5	35.2
Nerfies^S	22.6	13.2	18.4	18.0	24.4	14.9

Comparison with Nerfies. Nerfies optimizes SE(3) fields with photometric error, which fails at large motion and fails to register pixels across videos. In contrast, BANMo optimizes an articulated bones model using "featuremetric" consistency wrt a pre-trained CSE feature embedding. As shown in Fig. 4, although single-video Nerfies reconstructs reasonable 3D shapes of moving objects given rough initial root pose, it fails to reconstruct large articulations, such as the fast motion of the cat's head (2nd row). Furthermore, as shown in Fig. 10, Nerfies is not able to leverage more videos to improve the reconstruction quality, while the reconstruction of BANMo improves given more videos. The results in Tab. 2 suggests BANMo produces more accurate geometry than Nerfies for all sequences.

Comparison with ViSER. As shown in Fig. 4, ViSER produces reasonable articulated shapes. However, detailed geometry, such as ears, eyes, nose and rear limbs of the cat are blurred out. Furthermore, detailed articulation, such as head rotation and leg switching are not recovered. In contrast, BANMo faithfully recovers these high-fidelity geometry and motion. We observed that the neural implicit volume

representation is compliant to topology changes during gradient updates (see Fig. 5), and is therefore able to recover from bad local optima. In contrast, sub-optimal topology that happens during optimization, such as inverted faces, prevents ViSER to improve given more iterations. Compared to meshes with finite number of vertices, implicit shape representation maintains a continuous geometry, enabling us to recover detailed shape without additional cost in rendering high-res meshes.

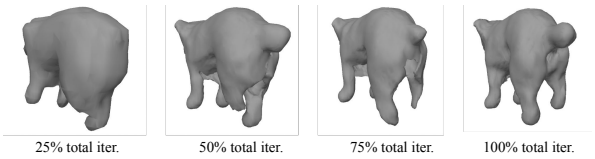

Figure 5. **Compliance to topology changes in optimization.** BANMo incorrectly reconstructs a single rear leg of the dog, but automatically corrects the topology with gradient updates.

4.3. Diagnostics

We ablate the importance of each component, by using a subset of videos. To also ablate root pose initialization and registration, we test on AMA's `samba` and `swing` (325 frames in total). We include exhaustive ablations in supplement, and only highlight crucial aspects of BANMo below.

Root pose initialization. We show the effect of PoseNet for root pose initialization (Sec 3.4) in Fig. 6: without it, the root poses (or equivalently camera poses) collapsed to a degenerate solution after optimization.

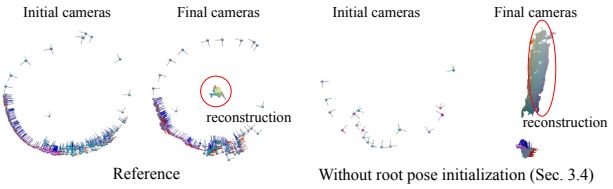

Figure 6. **Diagnostics of root pose initialization (Sec.3.4).** With randomly initialized root poses, the estimated poses (on the right) collapsed to a degenerate solution, causing reconstruction to fail.

Registration. In Fig. 7, we show the benefit of using canonical embeddings (Sec 3.3), and measured 2D flow (Eq. 14) to register observations across videos and within a video. Without the canonical embeddings and corresponding losses (Eq. 15-16), each video will be reconstructed separately. With no flow reconstruction loss, multiple ghosting structures are reconstructed due to failed registration.

Active sampling. We show the effect of active sampling (Sec 3.4) on a `casual-cat` video (Fig. 8): removing it results in slower convergence and inaccurate geometry.

Deformation modeling. We demonstrate the benefit of using our neural blend skinning model (Sec 3.2) on an `eagle` sequence, which is challenging due to its large wing articulations. If we swap neural blend skinning for MLP-SE(3) [33], the reconstruction is less regular. If we swap for MLP-translation [22, 38], we observe ghosting wings due to wrong geometric registration (caused by large motion). Our method can model large articulations thanks to the regularization from the Gaussian component, and also handle complex deformation such as close contact of hands.

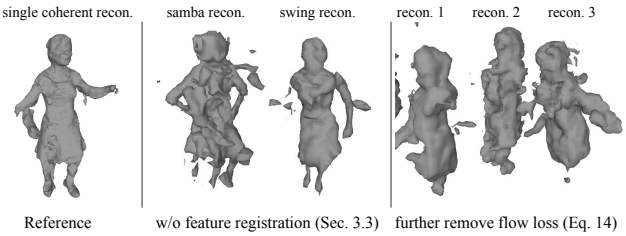

Figure 7. **Diagnostics of registration (Sec. 3.3).** Without canonical embeddings (middle) or flow loss (right), our method fails to register frames to a canonical model, creating ghosting effects.

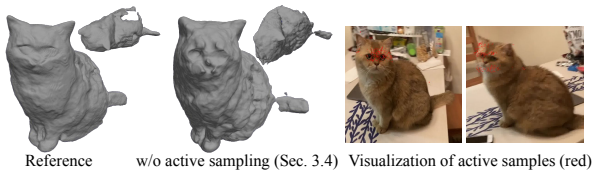

Figure 8. **Diagnostics of active sampling over (x, y) (Sec. 3.4).** With no active sampling, our method converges slower and misses details (such as ears and eyes). Active samples focus on face and boundaries pixels where the color reconstruction errors are higher.

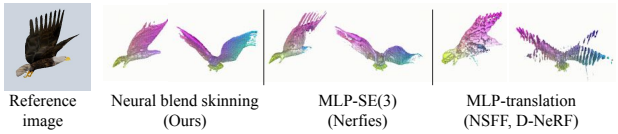

Figure 9. **Diagnostics of deformation modeling (Sec.3.2).** Replacing our neural blend skinning with MLP-SE(3) results in less regular deformation in the non-visible region. Replacing our neural blend skinning with MLP-translation as in NSFF and D-Nerf results in reconstructing ghosting wings due to significant motion.

Ability to leverage more videos. We compare BANMo to Nerfies in terms of the ability to leverage more available video observations. To demonstrate this, we compare the reconstruction quality of optimizing 1 video vs. 8 videos from the AMA `samba` sequences. Results are shown in Fig. 10. Given more videos, our method can register them to the same canonical space, improving the reconstruction completeness and reducing shape ambiguities. In contrast, Nerfies does not produce better results given more video observations.

Motion re-targeting. As a distinctive application, we

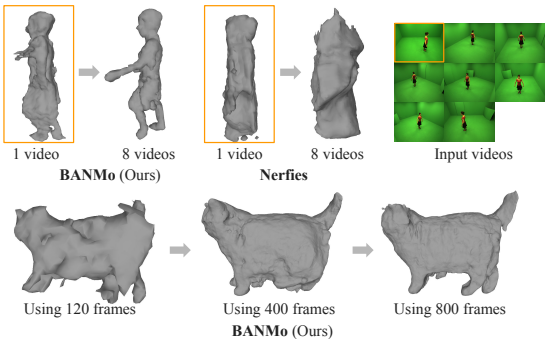

Figure 10. **Reconstruction completeness vs number of input videos and video frames.** BANMo is capable of registering more input videos if they are available, improving the reconstruction.

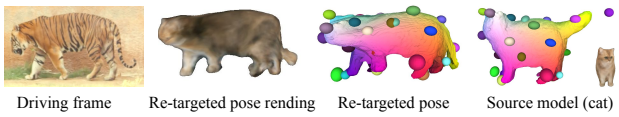

Figure 11. **Motion re-targeting from a pre-optimized cat model to a tiger.** Color coded by point locations in the canonical space.

demonstrate BANMo's ability of to re-target the articulations of a driving video to our 3D model by optimizing the frame-specific root and body pose codes $\boldsymbol{\omega}_r^t$, $\boldsymbol{\omega}_b^t$, as shown in Fig. 11. To do so, we first optimize all parameters over a set of *training* videos from our `casual-cat` dataset. Given a driving video of a tiger, we freeze the shared model parameters (including shape, skinning, and canonical features) of the cat model, and only optimize the video-specific and frame-specific codes, i.e. root and body pose codes $\boldsymbol{\omega}_r^t$, $\boldsymbol{\omega}_b^t$, as well as the environment lighting code $\boldsymbol{\omega}_e^t$.

5. Discussion

We have presented BANMo, a method to reconstruct high-fidelity animatable 3D models from a collection of casual videos, without requiring a pre-defined shape template or pre-registered cameras. BANMo registers thousands of *unsynchronized* video frames to the same canonical space by fine-tuning a generic DensePose prior to specific instances. We obtain fine-grained registration and reconstruction by leveraging neural implicit representation for shape, appearance, canonical features, and skinning weights. On real and synthetic datasets, BANMo shows strong empirical performance for reconstructing clothed human and quadruped animals, and demonstrates the ability to recover large articulations, reconstruct fine-geometry, and render realistic images from novel viewpoints and poses.

Limitations. BANMo uses a pre-trained DensePose-CSE (with 2D keypoint annotaionts [31]) to provide rough root body pose registration, and therefore not currently applicable to categories beyond humans and quadruped animals. To build generic pipelines of deformable 3D model reconstruction, either a relative root pose estimator or a generic feature embedding is needed for future work. Further investigation is needed to extend our method to arbitrary object categories. Similar to other works in differentiable rendering, BANMo requires a lot of compute, which increases linearly with the number of input images. We leave speeding up the optimization as future work.

A. Notations

We refer readers to a list of notations in Tab. 6 and a list of learnable parameters in Tab. 7.

B. Method details

B.1. Root Pose Initialization

As discussed in Sec. 3.4, to make optimization robust, we train a image CNN (denoted as PoseNet) to initialize root body transforms $\mathbf{G}^t$ that aligns the camera space of time t to the canonical space of CSE, as shown in Fig. 12.

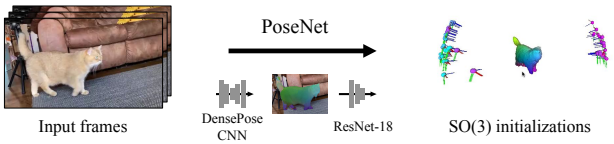

Figure 12. **Inference pipeline of PoseNet.** To initialize the optimization, we train a CNN PoseNet to predict root poses given a single image. PoseNet uses a DensePose-CNN to extract pixel features and decodes the pixel features into root pose predictions with a ResNet-18. We visualize the initial root poses on the right. Cyan color represents earlier time stamps and magenta color represent later timestamps.

Preliminary DensePose CSE [30, 31] trains pixel embeddings ψ_I and surface feature embeddings ψ for humans and quadruped animals using 2D keypoing annotations. It represents surface embeddings by a canonical surface with N vertices and vertex features $\psi \in \mathbb{R}^{N\times 16}$. A SMPL mesh is used for humans, and a sheep mesh is used for quadraped animals. The embeddings are trained such that given an pixel feature, a 3D point on the canonical surface can be uniquely located via feature matching.

Naive PnP solution Given 2D-3D correspondences provided by CSE, one way to solve for $\mathbf{G}^t$ is to use perspective-n-points (PnP) algorithm assuming objects are rigid. However, the PnP solution suffers from catastrophic failures due to the non-rigidity of the object, which motivates our PoseNet solution. By training a feed-forward network with data augmentations, our PoseNet solution produces fewer gross errors than the naive PnP solution.

Synthetic dataset genetarion. We train separate PoseNet, one for human, and one for quadruped animals. The training pipeline is shown in Fig. 13. Specifically, we render surface features as feature images $\psi_{\text{rnd}} \in \mathbb{R}^{112\times 112\times 16}$ given viewpoints $\mathbf{G}^* = (\mathbf{R}^*, \mathbf{T}^*)$ randomly generated on a unit sphere facing the origin. We apply occlusion augmentations [44] that randomly mask out a rectangular region in the rendered feature image and replace with mean values of the corresponding feature channels. The random occlusion augmentation forces the network to be robust to outlier inputs, and empirically helps network to make robust predictions in presence of occlusions and in case of out-of-distribution appearance.

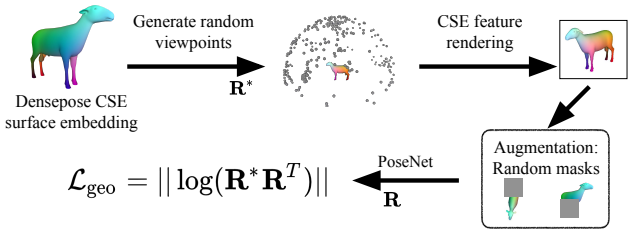

Figure 13. **Training pipeline of PoseNet.** To train PoseNet, we use DesePose CSE surface embeddings, which is pertained on 2D annotations of human and quadruped animals. We first generate random viewpoints on a sphere that faces the origin. Then we render surface embeddings as 16-channel images. We further augment the feature images with random adversarial masks to improve the robustness to occlusions. Finally, the rotations predicted by PoseNet are compared against the ground-truth rotations with geodesic distance.

Loss and inference. We use the geodesic distance between the ground-truth and predicted rotations as a loss to update PoseNet,

$$\mathcal{L}_{\text{geo}} = ||\log(\mathbf{R}^*\mathbf{R}^T)||, \quad \mathbf{R} = \text{PoseNet}(\psi_{\text{rnd}}), \tag{20}$$

where we find learning to predict rotation is sufficient for initializing the root body pose. In practice, we set the initial object-to-camera translation to be a constant $\mathbf{T} = (0, 0, 3)^T$. We run pose CNN on each test video frame to obtain the initial root poses $\mathbf{G}_0^t = (\mathbf{R}, \mathbf{T})$, and compute a delta root pose with the root pose MLP:

$$\mathbf{G}^t = \mathbf{MLP}_{\mathbf{G}}(\omega_r^t)\mathbf{G}_0^t. \tag{21}$$

B.2. Optimization details

Canonical 3D grid. As mentioned in Sec 3.3, we define a canonical 3D grid $\mathbf{V}^* \in \mathbb{R}^{20\times 20\times 20}$ to compute the matching costs between pixels and canonical space locations. The canonical grid is centered at the origin and axis-aligned with bounds $[x_{\text{min}}, x_{\text{max}}]$, $[y_{\text{min}}, y_{\text{max}}]$, and $[z_{\text{min}}, z_{\text{max}}]$. The bounds are initialized as loose bounds and are refined during optimization. For every 200 iterations, we update the bounds of the canonical volume as an approximate bound of the object surface. To do so, we run marching cubes on a 64^3 grid to extract a surface mesh and then set L as the axis-aligned (x, y, z) bounds of the extracted surface.

Near-far planes. To generate samples for volume rendering, we dynamically compute the depth of near-far planes (d_n^t, d_f^t) of frame t at each iterations of the optimization. To do so, we compute the projected depth of the canonical surface points $d_i^t = (\Pi^t\mathbf{G}^t\mathbf{X}_i^*)_2$. The near plane is set as $d_n^t = \min(d_i) - \epsilon_L$ and the far plane is set as

Table 3. Table of hyper-parameters.

Name	Value	Description
B	25	Number of bones
N	128	Sampled points per ray
N^p	6144	Sampled rays per batch
(H, W)	(512,512)	Resolution of observed images

$d_f^t = \max(d_i) + \epsilon_L$, where $\epsilon_L = 0.2\big(\max(d_i) - \min(d_i)\big)$. To avoid the compute overhead, we approximate the surface with an axis-aligned bounding box with 8 points.

Hyper-parameters. We use 1cycle learning rate scheduler, which warms-up with a low learning rate to the maximum, and anneals the learning rate to a final learning rate. We apply $lr_{init} = 2e-5$, $lr_{max} = 5e-4$, $lr_{final} = 1e-4$. We refer readers to a complete list of hyper-parameters in Tab. 3.

Experiment details When running Nerfies on AMA and animated objects, we found using RGB reconstruction loss does not produce meaningful results possibly due to the homogeneous background color. To improve Nerfies results, we provide it with ground-truth object silhouettes, and optimize a carefully balanced RGB+silhouette loss [65].

C. Additional results

C.1. SFM root pose initialization

COLMAP [41, 42] failed to converge when focused on the deformable object due to violation of rigidity, leading to very few successful registrations (18 over 811 images registered on `casual-cat`). A recent end-to-end method, DROID-SLAM [51], registered all the images but the accuracy is low compared to PoseNet, as shown in Tab. 4. We also tried SFM to estimate and compensate for the camera motion (using background as rigid anchor), but this did not help to recover the pose of the object due to its global movement w.r.t. to the background.

Table 4. **Evaluation on root pose prediction.** Mean and standard deviation of the rotation error (°) over all frames (↓). We use BANMo-optimized poses as ground-truth. Rotations are aligned to the ground-truth by a global rotation under chordal L2 distance.

Method	`c-cat`	`c-human`	`ama-human`
CSE-PoseNet	**18.6**±16.2	**12.8**±8.9	**11.8**±17.4
DROID-SLAM	65.5 ± 44.5	55.8 ± 39.2	83.6 ± 50.5

C.2. More ablation study

In Sec. 4.3, we presented qualitative results of diagnostics experiments. In Tab. 5, we report the results of other ablations followed by analysis.

Table 5. **Results on AMA swing and samba.** 3D Chamfer distance (cm, ↓) and F-score (%, ↑) averaged over all frames.

Method	`CD`	`F@1%`	`F@2%`	`F@5%`
number-bone=4	9.88	28.1	52.4	84.1
number-bone=9	9.08	31.2	56.4	86.8
number-bone=16	**9.02**	**31.8**	**57.2**	87.2
number-bone=25	9.08	**31.8**	57.0	87.1
–w/o in-surface loss	9.14	29.9	54.8	86.7
–quad. embedding	9.70	29.8	54.2	85.4
number-bone=64	9.18	31.1	56.6	**87.5**
number-bone=100	9.11	31.4	56.7	87.3
pose error ϵ=20°	**8.75**	**30.9**	**57.0**	**88.1**
pose error ϵ=50°	8.91	29.8	56.1	**88.1**
pose error ϵ=90°	9.91	28.4	54.8	85.7
coverage=90° (2 vids)	10.61	29.3	54.3	84.1
coverage=180° (4 vids)	**8.94**	**33.0**	**59.8**	**87.9**
coverage=270° (6 vids)	9.09	29.8	56.1	87.6
active-sample=0%	9.63	29.1	53.7	85.8
active-sample=25%	**8.60**	**32.3**	**57.9**	**88.0**
active-sample=50%	9.14	29.9	54.8	86.7

Number and location of bones As shown in the first group of Tab. 5 and Fig. 14, using too few bones fails to recover all body parts due to over-regularization. Using more than 16 bones produces good reconstructions, but consumes more memory when computing skinning weights. Enforcing them to stay close to the surface with a sinkhorn divergence loss improves the results (Tab. 5, L16-17).

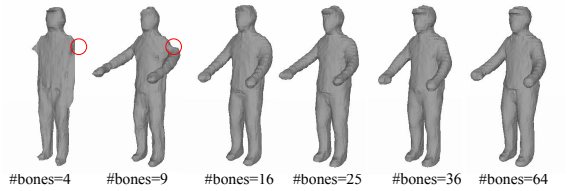

Figure 14. **Sensitivity to number of bones.**

Sensitivity to incorrect initial pose We inject different levels of Gaussian noise into the initial poses, leading to average rotation errors $\epsilon \in \{20, 50, 90\}°$. As shown in the second group of Tab. 5, BANMo is stable up to 50° rotation error.

Pre-trained embeddings Pre-trained embeddings help BANMo outperform Nerfies, but it is not crucial given good initial root poses ($\epsilon = 12.8 \pm 8.9°$). As shown in Tab. 5, using embeddings pre-trained for quadruped animals for human optimization produces slightly worse results.

How much data are needed? To reconstruct a complete shape, BANMo requires all object surface to be visible from at least one frame. Beside completeness, more videos allows to estimate better skinning weights and a more regular

motion. We evaluate view coverage in the third group of Tab. 5.

Importance sampling We use active sampling to avoid sampling from uninformative frames and pixels. It consistently improves reconstruction results as shown in the last group of Tab. 5.

Bone re-initialization We qualitatively evaluate the effect of rest bone re-initialization, which re-initializes bone parameters according to the current estimation of shape. As shown in Fig. 15, without re-initializing the bones, the optimization may stuck at bad local optima and the final reconstruction may become less accurate.

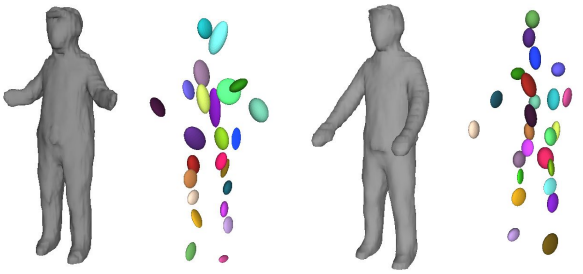

Figure 15. **Effect of bone re-initialization.** We find it important to re-initialize rest bone parameters after finding a better approximation of object geometry.

Delta skinning weights We qualitatively evaluate the effect of delta skinning weights. As shown in Fig. 16, without learning a delta skinning weights specific to each 3D point, the reconstructed shape and motion may be over-regularized by the 3D Gaussians.

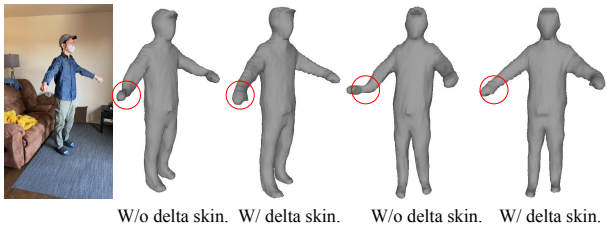

Figure 16. **Effect of delta skinning weights.** We find it important to learn a point-specific delta skinning weight function to reconstruction motions in high-quality.

C.3. Qualitative results

We refer readers to our supplementary webpage for complete qualitative results.

Table 6. Table of notations.

Symbol	Description
	Index
t	Frame index, $t \in \{1, \ldots, T\}$
b	Bone index $b \in \{1, \ldots, B\}$ in neural blend skinning
i	Point index $b \in \{1, \ldots, N\}$ in volume rendering
	Points
$\mathbf{x}$	Pixel coordinate **x** = (x,y)
$\mathbf{X}^t$	3D point locations in the frame t camera coordinate
$\mathbf{X}^*$	3D point locations in the canonical coordinate
$\hat{\mathbf{X}}^*$	Matched canonical 3D point locations via canonical embedding
	Property of 3D points
$\mathbf{c} \in \mathbb{R}^3$	Color of a 3D point
$\sigma \in \mathbb{R}$	Density of a 3D point
$\boldsymbol{\psi} \in \mathbb{R}^{16}$	Canonical embedding of a 3D point
$\mathbf{W} \in \mathbb{R}^B$	Skinning weights of assigning a 3D point to B bones
	Functions on 3D points
$\mathcal{W}^{t,\leftarrow}(\mathbf{X}^t)$	Backward warping function from $\mathbf{X}^t$ to $\mathbf{X}^*$
$\mathcal{W}^{t,\rightarrow}(\mathbf{X}^*)$	Forward warping function from $\mathbf{X}^*$ to $\mathbf{X}^t$
$\mathcal{S}(\mathbf{X}, \omega_b)$	Skinning function that computes skinning weights of $\mathbf{X}$ under body pose ω_b
	Rendered and Observed Images
$\mathbf{c}/\hat{\mathbf{c}}$	Rendered/observed RGB image
$\mathbf{o}/\hat{\mathbf{s}}$	Rendered/observed object silhouette image
$\mathcal{F}/\hat{\mathcal{F}}$	Rendered/observed optical flow image

Table 7. Table of learnable parameters.

Symbol	Description
	Canonical Model Parameters
$\mathbf{MLP_c}$	Color MLP
$\mathbf{MLP}_{\text{SDF}}$	Shape MLP
$\mathbf{MLP}_{\boldsymbol{\psi}}$	Canonical embedding MLP
	Deformation Model Parameters
$\mathbf{\Lambda}^0 \in \mathbb{R}^{3\times3}$	Scale of the bones in the "zero-configuration" (diagonal matrix).
$\mathbf{V}^0 \in \mathbb{R}^{3\times3}$	Orientation of the bones in the "zero-configuration".
$\mathbf{C}^0 \in \mathbb{R}^3$	Center of the bones in the "zero-configuration".
$\mathbf{MLP}_{\Delta}$	Delta skinning weight MLP
$\mathbf{MLP_G}$	Root pose MLP
$\mathbf{MLP_J}$	Body pose MLP
	Learnable Codes
$\boldsymbol{\omega}_b^* \in \mathbb{R}^{128}$	Body pose code for the rest pose
$\boldsymbol{\omega}_b^t \in \mathbb{R}^{128}$	Body pose code for frame t
$\boldsymbol{\omega}_r^t \in \mathbb{R}^{128}$	Root pose code for frame t
$\boldsymbol{\omega}_e^t \in \mathbb{R}^{64}$	Environment lighting code for frame t, shared across frames of the same video
	Other Learnable Parameters
$\boldsymbol{\psi}_I$	CNN pixel embedding initialized from DensePose CSE
α_s	Temperature scalar for canonical feature matching
β	Scale parameter that controls the solidness of the object surface
$\Pi \in \mathbb{R}^{3\times3}$	Intrinsic matrix of the pinhole camera model

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 2011. 1

[2] Marc Badger, Yufu Wang, Adarsh Modh, Ammon Perkes, Nikos Kolotouros, Bernd Pfrommer, Marc Schmidt, and Kostas Daniilidis. 3D bird reconstruction: a dataset, model, and shape recovery from a single view. In *ECCV*, 2020. 2

[3] Benjamin Biggs, Ollie Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. Who left the dogs out: 3D animal reconstruction with expectation maximization in the loop. In *ECCV*, 2020. 2

[4] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000. 2

[5] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. 2021. 2, 4

[6] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019. 6

[7] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoints without keypoints. In *ECCV*, 2020. 2

[8] Paulo FU Gotardo and Aleix M Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR*, 2011. 2

[9] Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, 2014. 4

[10] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *CVPR*, pages 12753–12762, June 2021. 2

[11] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 2

[12] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 2

[13] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 5

[14] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 7

[15] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, June 2020. 2

[16] Filippos Kokkinos and Iasonas Kokkinos. To the point: Correspondence-driven monocular 3d category reconstruction. In *NeurIPS*, 2021. 2

[17] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *ICCV*, 2019. 2

[18] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, pages 452–461, 2020. 2, 5

[19] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *WACV*, 2020. 2

[20] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. In *NeurIPS*, 2020. 2

[21] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. *ECCV*, 2020. 2

[22] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2, 3, 5, 7, 8

[23] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2

[24] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *SIGGRAPH Asia*, 2021. 2

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 2015. 2

[26] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 2019. 5

[27] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 2, 3

[28] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *ICCV*, 2021. 2

[29] 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 *ECCV*, 2020. 2, 3, 4, 5, 6, 7

[30] Natalia Neverova, David Novotny, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. In *NeurIPS*, 2020. 3, 5, 6, 10

[31] Natalia Neverova, Artsiom Sanakoyeu, Patrick Labatut, David Novotny, and Andrea Vedaldi. Discovering relationships between object categories via universal canonical maps. In *CVPR*, 2021. 5, 9, 10

[32] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021. 2

[33] 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 *ICCV*, 2021. 2, 3, 6, 7, 8

[34] 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. *arXiv*, 2021. 2

[35] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 2

[36] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021. 2

[37] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 2

[38] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2020. 2, 8

[39] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *CVPR*, 2021. 4

[40] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. In *IJCV*, 2008. 2

[41] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 11

[42] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 11

[43] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. Neural dense non-rigid structure from motion with latent space constraints. In *ECCV*, 2020. 2

[44] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 10

[45] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*. 2006. 1

[46] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 2008. 1

[47] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. In *NeurIPS*, 2021. 2

[48] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*, 2021. 6

[49] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 2

[50] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, pages 3405–3414, 2019. 7

[51] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems*, 34, 2021. 11

[52] 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 *ICCV*, 2021. 2

[53] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. In *arXiv*, 2020. 2

[54] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *TOG*, 2008. 7

[55] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 2

[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. 3

[57] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. In *arXiv preprint arXiv:2102.07064*, 2021. 2

[58] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Dove: Learning deformable 3d objects by watching videos. *arXiv preprint arXiv:2107.10844*, 2021. 2

[59] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019. 2

[60] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019. 6

[61] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 7

[62] 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 *CVPR*, 2021. 2, 4

[63] 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. In *NeurIPS*, 2021. 2, 5, 6, 7

[64] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv preprint arXiv:2106.12052*, 2021. 3

[65] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 5, 6, 11

[66] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *CVPR*, 2021. 2

[67] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In *NeurIPS*, 2021. 6

[68] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild". In *ICCV*, 2019. 2

[69] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In *CVPR*, 2018. 2

[70] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017. 2