CAMM: Building Category-Agnostic and Animatable 3D Models from Monocular Videos

Tianshu Kuai Akash Karthikeyan Yash Kant Ashkan Mirzaei Igor Gilitschenski
University of Toronto

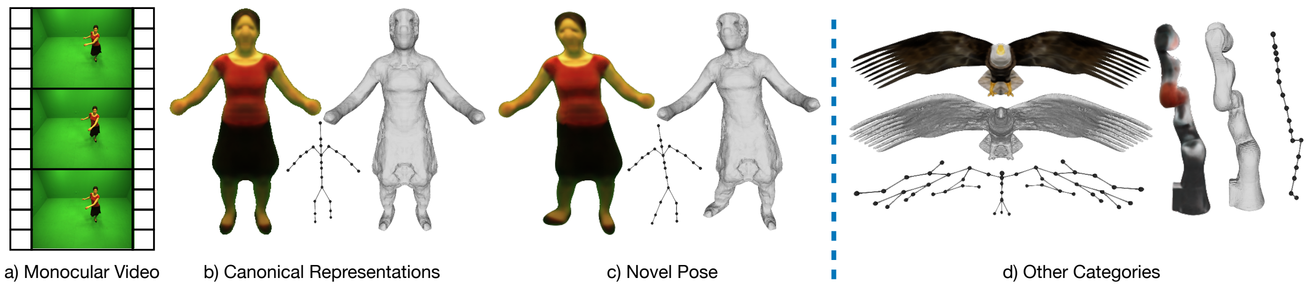

Figure 1. Given monocular videos of an articulated object, our method builds its canonical representation, including 3D shape, appearance, and a corresponding animatable 3D kinematic chain for direct pose manipulations. Our approach does not rely on any information on the object's shape and underlying structure. In c) we show example of re-posing the human in b) to novel pose. We show the learned canonical representations of other object categories in d).

Abstract

Animating an object in 3D often requires an articulated structure, e.g. a kinematic chain or skeleton of the manipulated object with proper skinning weights, to obtain smooth movements and surface deformations. However, existing models that allow direct pose manipulations are either limited to specific object categories or built with specialized equipment. To reduce the work needed for creating animatable 3D models, we propose a novel reconstruction method that learns an animatable kinematic chain for any articulated object. Our method operates on monocular videos without prior knowledge of the object's shape or underlying structure. Our approach is on par with state-of-the-art 3D surface reconstruction methods on various articulated object categories while enabling direct pose manipulations by re-posing the learned kinematic chain. Our project page: https://camm3d.github.io/.

1. Introduction

Building 3D representations of real-world objects from images has long been studied in 3D Computer Vision. Realistic 3D models allow us to synthesize new images of objects from arbitrary viewpoints and to animate these objects for applications such as virtual and augmented reality. Prior works on building 3D representations for static scenes [2, 6, 10, 14, 25, 31–33, 42, 59, 76, 78] assume the objects in the scene to be rigid to obtain accurate correspondences, which cannot be generalized to deformable objects or scenes with drastic motions. Recent works [19, 20, 24, 37, 41, 45, 52, 56, 68–70] tackle the problem of reconstructing dynamic scenes and 3D deformable objects, and have achieved impressive performance on novel view synthesis. However, these methods can only render objects with the body poses that exist in the given training data because they "mimic" the movements and behaviours of the objects. Therefore, their representations of 3D objects are not applicable for direct pose manipulations or animations.

To fulfill such needs, tremendous efforts have been made to build parametric shape models for humans [28, 38, 63, 64] and quadruped animals [80, 81]. These shape models are often built via capturing large amounts of data using specialized scanners and sensors. They serve as base templates for methods that focus on reconstructing and animating 3D humans and animals [7, 21, 26, 36, 39, 40, 55, 58, 60]. Although these template-based models can offer high-fidelity reconstruction and animation results, they can only be applied to limited object categories. Recent efforts [35, 71] aim to build articulated 3D object models without templates or category-specific priors. However, they either require synchronized multi-view videos which are hard to acquire, or a manually created skeleton as initialization.

In this work, we propose a novel approach to build an animatable 3D model for any articulated object using only monocular videos. We do not rely on prior knowledge of the object shape and structure or manual annotations as initialization. Thus, our pipeline significantly reduces the amount of work needed to create animatable 3D models and eliminates the need for synchronized multi-view observations or specialized sensors. Our model can be directly used for pose manipulations and animations in 3D, which we refer to in this paper as *animatable model*.

Specifically, our approach uses Neural Radiance Fields [33] and Signed Distance Functions to represent the appearance and shape of the object. The body pose of the object is represented by a *kinematic chain*. Our kinematic chain is initialized based on the initial estimate of the object's shape, and further optimized jointly with the object's shape, appearance, and deformation parameters. We adopt skinning weights mechanisms from [5, 43, 70] and propose a novel way to use our optimized kinematic chain to drive pose changes and deformations. Due to the under-constrained nature of this problem, we leverage the foreground masks and optical flow predicted by off-the-shelf models as robust visual cues to learn the shape and underlying structure of the object. Inspired by [71], we utilize the 2D image features from DINO-ViT [3] that are trained in a self-supervised manner, to establish long-range correspondences and consistencies at the object parts level [1,53] between different frames. This is achieved by matching canonical features at the object surface to the pre-trained DINO-ViT image features. Examples of rendered objects in canonical representations and novel poses, along with the corresponding 3D meshes and kinematic chains are shown in Figure 1. Our contributions are summarized as follows:

- Our work is the first to build an animatable 3D model for objects of arbitrary categories from monocular videos without any template or prior knowledge of the object's shape and underlying structure.
- We propose a novel optimization technique to iteratively refine the kinematic chain and its associated deformation parameters. Users can directly manipulate the optimized kinematic chain to animate the object.
- We achieve similar surface reconstruction results to state-of-the-art 3D surface reconstruction methods on various articulated and deformable object categories.

2. Related Work

3D deformable shape reconstruction from images. Inspired by NeRF [33], recent works [19,24,37,41,52,56] are able to obtain robust 3D reconstructions on dynamic scenes or deformable objects from a collection of multi-view images. Some works [15, 18, 23, 74] use additional supervisions from human annotated data or shape templates to recover 3D shapes. These methods suffer from large performance degradation when the input has large deformations and self-occlusions. Recent works [20,45,77] explore structure from motion approaches to reconstruct non-rigid 3D scenes using video sequences. Similar to us, LASR [68], ViSER [69], and BANMo [70] utilize monocular videos for reconstructing deformable and articulated 3D shapes and achieve great surface reconstruction results. However, these approaches do not suffice the needs of direct pose manipulations of the reconstructed 3D objects as they reconstruct by memorizing the poses and movements in training data.

Category-specific animatable models. A group of works [4,27,30,75,79] tackle the problem of animating humans in 2D using annotated data with human poses. To enable animations in 3D space, significant efforts from the community have been put into creating 3D human and animal shape models [28,38,63,64]. Many works [7,12,13,21, 26,36,39,40,55,58,60–62,65] utilize these template shapes or pose priors from shape models to recover 3D shapes and perform animations. Recent methods [22, 46, 47] do not use shape templates, but rely on pre-defined skeletons or poses predicted by models [17, 18] trained on human annotated data, and these methods require synchronized multi-view inputs. These pre-defined skeletons or poses are used as priors for learning the shape and articulation for animations. However, the annotations are usually expensive to get and category-specific, which cannot be applied to other object categories in the real world and do not work for out-of-distribution articulation modes or poses.

Category-agnostic animatable models. Several methods have recently emerged for building 3D animatable model for any articulated object. Watch it Move [35] builds a 3D articulated structure by identifying physically meaningful joints in an unsupervised manner from calibrated multi-view videos and corresponding foreground masks. Recent work [50] learns deformation behaviours from a given 3D mesh of the object to allow users to define desired deformations at regions of interest. However, synchronized multi-videos and 3D meshes are not easily accessible. Similar to our goal, LASSIE [71] recovers a 3D shape and skeleton from image ensembles without pre-defined shape priors, but it requires a manually annotated 3D skeleton as input. In contrast, we aim to simplify the process of building animatable 3D models for any articulated object using only collections of monocular videos that can be easily collected.

3. Method

Given a collection of monocular videos, our goal is to build an animatable 3D model for the articulated object in the scene. Our method optimizes the canonical representation (Section 3.1) of the object, where its shape and appearance are modeled implicitly using Multi-layer Per-

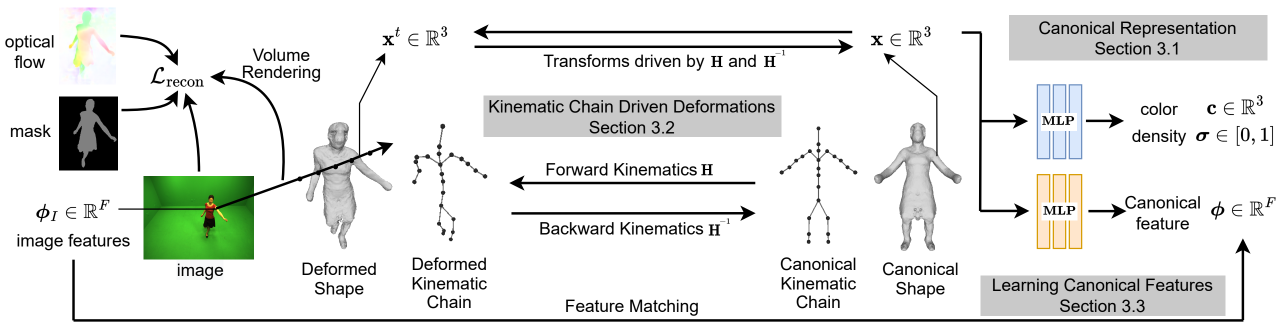

Figure 2. **Method overview**. Our method optimizes the canonical representation, including the object's shape, appearance, and kinematic chain (Section 3.1), to enable direct pose manipulations. It uses the kinematic chain to transform 3D points between the canonical and deformed space (Section 3.2), and render 2D observation predictions of colors, foreground masks and optical flows to match the actual 2D observations. In addition, canonical feature embeddings (Section 3.3) are learned by matching the corresponding 2D image features to establish the object parts level correspondences across different frames within a collection of monocular videos.

ceptrons (MLPs). Its canonical body pose is represented by a kinematic chain. Our approach consists of two optimization stages: the first stage (Section 3.4.1) optimizes the canonical shape and deformation parameters in each frame without a kinematic chain to get an initial estimate of the 3D shape. We then apply RigNet [66], an off-the-shelf category-agnostic skeleton estimator, on the initial estimate of the 3D mesh to obtain an initial kinematic chain. In the second stage (Section 3.4.2), the kinematic chain adapts to object shape to enable kinematic chain driven deformations (Section 3.2). We supervise the learning by matching the rendered images, foreground masks, and optical flows to the actual 2D observations in both stages. In addition, we learn canonical feature embeddings (Section 3.3) to obtain long-range correspondences between video frames, supervised by pre-trained 2D image features from [3].

3.1. Canonical Representation

Canonical shape. Similar to [33, 70], we represent the shape and appearance of an object implicitly in the canonical space. A 3D point $\mathbf{x} \in \mathbb{R}^3$ has two properties: its color $\mathbf{c} \in \mathbb{R}^3$ and density $\boldsymbol{\sigma} \in [0, 1]$. The color $\mathbf{c}$ is given by a Multilayer Perceptron (MLP) network, and the density $\boldsymbol{\sigma}$ is given by the cumulative of a Laplacian distribution with zero mean and learnable scale on the learned Signed Distance Function (SDF) [57, 72] in the canonical space. The value of SDF at any 3D point is given by a separate MLP network, and the canonical mesh can be extracted by finding the zero-level set of the SDF [72] with the marching cubes algorithm. Please see our supplementary material for additional details about the MLPs.

Canonical kinematic chain. We use a kinematic chain to represent the body pose of any articulated object. The canonical kinematic chain is defined as a set of connected 3D joints $\mathbf{P} = \{\mathbf{p}_i \mid i = 1, \ldots, n_p\}$, where $\mathbf{p}_i \in \mathbb{R}^3$ denotes joint positions in the canonical space, with a set of $n_j - 1$ links $\mathbf{L} = \{\boldsymbol{\ell}_{jk} = (\mathbf{p}_j, \mathbf{p}_k)\}^{(n_j-1)}$. The joints are connected in a hierarchical manner with a pre-defined root joint to form a tree structure. As a result, there is no cycle, and a unique path exists between every two joints.

3.2. Kinematic Chain Driven Deformations

3.2.1 Neural Blend Skinning Deformations

Inspired by [68–70], we define a set of learnable *deformation anchors* $\mathbf{A} = \{\mathbf{a}_i \mid i = 1, \ldots, n_a\}$, where $\mathbf{a}_i \in \mathbb{R}^3$ denotes the anchor positions in the canonical space. We compute surface deformations with respect to anchors via linear blend skinning with learned weights. At time t, let $\mathbf{C}^t \in SE(3)$ be the object's root pose with respect to the canonical root pose, $\mathbf{T}_i^t \in SE(3)$ be the transformation of anchor $\mathbf{a}_i$ from canonical space to the deformed space, and $\mathbf{w}_{a_i}^t$ be the skinning weight of $\mathbf{x}$ with respect to anchor $\mathbf{a}_i$. Given any 3D point $\mathbf{x} \in \mathbb{R}^3$ in the canonical space, its corresponding point in the deformed space $\mathbf{x}^t$ at time t is given by the weighted average of anchor's transformation $\mathbf{T}_i^t$ by:

$$\mathbf{x}^t = \mathbf{C}^t \sum_{i=1}^{n_a} (\mathbf{w}_{a_i} \mathbf{T}_i^t) \mathbf{x} \tag{1}$$

where we define $\mathbf{W} = \left[\mathbf{w}_{a_1}, \ldots, \mathbf{w}_{a_{n_a}}\right] \in \mathbb{R}^{n_a}$ to be the forward skinning weights for canonical space point $\mathbf{x}$ at time t, and the weights are determined by the Euclidean distances between point $\mathbf{x}$ and all anchors $\mathbf{A}$ in the canonical space as:

$$\mathbf{W} = \sigma\Big(- \|\mathbf{x} - \mathbf{A}\|_2^2 \Big) \tag{2}$$

where σ is softmax to normalize the skinning weights and its temperature τ is a learnable parameter.

3.2.2 Kinematic Chain Driven Deformations

As the kinematic chain represents the body pose of the object, the deformation anchors' transformations must be driven by the kinematic chain to obtain properly deformed surfaces corresponding to the actual body pose. To achieve this, we associate each anchor $\mathbf{a}_i$ to its closest kinematic chain link $\boldsymbol{\ell}_{jk}$ based on Euclidean distances in the canonical space. Example associations are shown as dashed lines in Figure 3. We let the intersection between the dashed line of anchor $\mathbf{a}_i$ and the associated link $\boldsymbol{\ell}_{jk}$ be $\mathbf{m}_{ijk}$, where the parent joint and child joint of link $\boldsymbol{\ell}_{jk}$ are $\mathbf{p}_j$ and $\mathbf{p}_k$. Then this association between anchor $\mathbf{a}_i$ and link $\boldsymbol{\ell}_{jk}$ can be represented by the following terms:

$$\begin{aligned} \boldsymbol{\alpha}_i &= \left\|\mathbf{m}_{ijk} - \mathbf{p}_j\right\|_2, \boldsymbol{\beta}_i = \left\|\mathbf{a}_i - \mathbf{m}_{ijk}\right\|_2, \\ \mathbf{G}_i &= \mathbf{Rot}(\mathbf{m}_{ijk} - \mathbf{p}_j, \mathbf{a}_i - \mathbf{m}_{ijk}), \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}_i$ is the Euclidean distance between the parent joint $\mathbf{p}_j$ and the intersection $\mathbf{m}_{ijk}$, and $\boldsymbol{\beta}_i$ is the Euclidean distance between the intersection $\mathbf{m}_{ijk}$ and associated anclhor $\mathbf{a}_i$. The rotation matrix $\mathbf{G}_i \in SO(3)$ aligns vectors $\mathbf{m}_{ijk} - \mathbf{p}_j$ and $\mathbf{a}_i - \mathbf{m}_{ijk}$ about the intersection $\mathbf{m}_{ijk}$. We refer to these three terms as the *association parameters* for anchor $\mathbf{a}_i$, and they are kept constant in *any* kinematic chain configuration to ensure stable surface deformations.

With defined association parameters, we can express an anchor's position using its associated link and association parameters. As anchor $\mathbf{a}_i$ is associated to link ℓ_{jk} in the canonical space, its universal position $\tilde{\mathbf{a}}_i$ in *any* kinematic chain configuration can be computed as:

$$\tilde{\mathbf{a}}_i = \tilde{\mathbf{p}}_j + \underbrace{\boldsymbol{\alpha}_i \frac{\tilde{\mathbf{p}}_k - \tilde{\mathbf{p}}_j}{\left\|\mathbf{p}_k - \mathbf{p}_j\right\|_2}}_{\text{parent joint to intersection}} + \underbrace{\boldsymbol{\beta}_i \mathbf{G}_i \frac{\tilde{\mathbf{p}}_k - \tilde{\mathbf{p}}_j}{\left\|\mathbf{p}_k - \mathbf{p}_j\right\|_2}}_{\text{intersection to anchor}}, \tag{4}$$

where $\tilde{\mathbf{p}}_j$ and $\tilde{\mathbf{p}}_k$ are given by:

$$\tilde{\mathbf{p}}_{j,k} = \begin{cases} \mathbf{p}_{j,k} & \text{in canonical space} \\ \mathbf{H}^t_{j,k}\mathbf{p}_{j,k} & \text{in deformed space at time } t \end{cases} \tag{5}$$

Note that $\tilde{\mathbf{a}}_i = \mathbf{a}_i$ in the canonical space. $\mathbf{H}^t_j \in SE(3)$ and $\mathbf{H}^t_k \in SE(3)$ are the rigid transformations of joint $\mathbf{p}_j$ and joint $\mathbf{p}_k$ from the canonical space to the deformed space without breaking the kinematic chain, which we refer to as *forward kinematics* on the kinematic chain.

Then the rigid transformation $\mathbf{T}^t_i$ of anchor $\mathbf{a}_i$ at time t can be directly inferred based on its new position in the deformed space given by Eq. (4):

$$\mathbf{T}^t_i = \left[\mathbb{I} \mid \tilde{\mathbf{a}}_i - \mathbf{a}_i\right], \tag{6}$$

where $\mathbb{I} \in \mathbb{R}^{3\times3}$ is the identity matrix.

Given optimized kinematic chain and anchors, we can perform explicit re-posing of the object by directly applying user defined forward kinematics $\mathbf{H}$ to the kinematic chain.

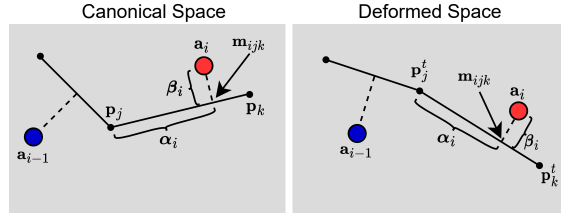

Figure 3. Simplified 2D example on how anchors (red and blue dots) move along with associated kinematic chain links. Associations can be visualized at dashed lines.

3.3. Learning Semantically Consistent Features

Large deformations and self-occlusions often cause 3D reconstruction methods to fail or perform poorly. This is usually due to the lack of strong correspondence cues or the lack of long-range correspondences between different frames in long videos [44, 49]. Recent works [1, 53] show that the pre-trained image features from DINO-ViT [3, 8] can provide robust correspondences at the object parts level between different 2D images even under large appearance variations and viewing angle changes.

To better optimize the canonical shape and deformations, we follow [69,70] to learn surface feature embeddings $\phi(\mathbf{x}) \in \mathbb{R}^F$ for points on the canonical shape's surface. The feature is given by a separate MLP network that takes in any 3D point in the canonical space and it is trained in a self-supervised manner such that the surface feature embeddings match the pre-trained 2D image features at its corresponding pixel given by the camera projection model. Compared to BANMo [70] where pre-trained DensePose CSE features [34] specifically designed for humans and animals are used for learning surface embedding, DINO-ViT [3] features have shown to be meaningful and rich in downstream tasks for a wide variety of objects [1, 3]. This is because DINO-ViT is pre-trained in a self-supervised manner on large-scale datasets with diverse object categories.

3.4. Two-Stage Optimizations

3.4.1 Initial Optimization

The initial optimization stage aims to build a reasonable object shape with unconstrained anchors to capture deformations in each frame of the training videos. We initialize the MLP network that predicts the canonical shape of the object to approximate the object as a 3D ellipsoid, and initialize all the anchors $\mathbf{A}$ at the global origin in the canonical space. During the initial optimization stage, the anchors are updated without any constraints in terms of their positions. Inspired by [70], the *unconstrained transformations* of the anchors at time t are given by a separate MLP, $\mathcal{F}_{\mathbf{A}}$ as:

$$\hat{\mathbf{T}}^t = \mathcal{F}_{\mathbf{A}}(\boldsymbol{\psi}^t_a), \tag{7}$$

where $\boldsymbol{\psi}^t_a \in \mathbb{R}^{128}$ is a learnable latent code for time t.

As we define the transformation of any 3D point $\mathbf{x}$ from the canonical space to the deformed space as weighted rigid transformation of anchors in Eq. (1), we can directly infer the backward transformation for any 3D point $\mathbf{x}^t$ in the deformed space by taking the inverse of the rigid transformations as:

$$\mathbf{x} = \Big(\sum_{i=1}^{n_a}\big(\mathbf{w}^t_{a_i}(\hat{\mathbf{T}}^t_i)^{-1}\big)\Big)(\mathbf{C}^t)^{-1}\mathbf{x}^t$$
$$\mathbf{W^t} = \sigma\Big(-\left\|\mathbf{x}^t - \mathbf{A}^t\right\|_2^2\Big), \tag{8}$$

where the backward skinning weights $\mathbf{W^t} = \left[\mathbf{w}^t_{a_1}, \ldots, \mathbf{w}^t_{a_{n_a}}\right] \in \mathbb{R}^{n_a}$ are computed in the same way as in forward skinning in Eq. (2) but based on deformed anchors and the 3D point of interest in the deformed space instead. This allows us to directly transform any 3D point back and forth between the canonical space and each frame's deformed space.

Similar to [69, 70], the canonical shape and appearance are learned by minimizing the reconstruction losses on 2D observations: RGB images, foreground masks, and optical flow. The reconstruction losses are formulated in the same way as in [33, 68–70, 73], where we minimize the differences between the rendered and the actual observations:

$$\mathcal{L}_{\text{recon}} = \sum_{\mathbf{x}_I}\left\|\mathbf{o}_r - \mathbf{o}_{gt}\right\|^2, \tag{9}$$

where $\mathbf{o}_r$ and $\mathbf{o}_{gt}$ are the pairs of rendered and ground-truth 2D observations (2D images, foreground masks, optical flow) at pixels of interest $\mathbf{x}^t_I \in \mathbb{R}^2$ at time t. To perform volumetric rendering at time t, we first sample a camera ray starting at the pixel of interest on the 2D image at time t, and use Eq. (8) to transform the sampled points along the camera ray back to the canonical space. We then follow the same rendering process in NeRF [33] on the camera ray in the canonical space to get the colour $\mathbf{c}^t$ at the pixel of interest. To render optical flow, we follow [70] to transform canonical space points to consecutive frames' deformed spaces, and use the camera projection model to find their corresponding positions on the 2D image. We compute the difference in the pixel locations as the optical flow.

The anchors and their transformations are learned from scratch with additional losses to enforce cycle consistency. The consistency losses enforce any 3D point in the deformed space after a backward transformation and a forward transformation to end up at the same position. To supervise canonical feature embeddings, we apply soft argmax descriptor matching [16, 29, 69, 70] on the 2D pixel of interest to determine the most probable corresponding surface point in the canonical space. The matching is based on cosine similarities between the canonical features and the pre-trained 2D image features. A 2D consistency loss [70] is used to minimize the difference between matched position and the location given by the camera projection model. The details of loss functions are in the supplementary material.

3.4.2 Kinematic chain aware optimization

After the initial optimization stage, we obtain an estimate of the canonical shape and a set of unconstrained anchors. However, the unconstrained anchors introduce unrealistic artifacts in deformations, such as undesired shrinks and stretches on the canonical shape. And the anchors are only optimized to "mimic" object shape in each frame of the videos. A kinematic chain is needed to remove these undesired effects and enable explicit re-posing of the object.

We initialize our canonical kinematic chain with pre-trained RigNet [66] that takes in our canonical mesh from the initial optimization stage and outputs a set of joints and connections. The pre-trained RigNet is able to provide a reasonable estimate of the kinematic chain. However, it requires further optimization such that our kinematic chain can better adapt to the object's movements in the training videos, and therefore align itself with the object's actual underlying structure in each frame.

As any point in 3D can be transformed back and forth between the canonical space and the deformed space using the transformations defined by anchors, we utilize the learned unconstrained anchor transformations $\hat{\mathbf{T}}^t$ from the initial optimization stage to transform the canonical kinematic chain joints $\mathbf{P}$ following Eq. (1):

$$\hat{\mathbf{P}}^t = \mathbf{C}^t\sum_{i=1}^{n_a}(\mathbf{w}_{a_i}\hat{\mathbf{T}}^t_i)\mathbf{P}, \tag{10}$$

where $\hat{\mathbf{P}}^t = \{\hat{\mathbf{p}}_i \mid i = 1, \ldots, n_p\}$ is the set of *unconstrained kinematic chain joints* in the deformed space at time t. The transformations of anchors are learned without any regularization on how they move or any knowledge on the existence of the kinematic chain. Therefore the transformed joints $\hat{\mathbf{P}}^t$ computed using unconstrained anchors will break the kinematic chain. We can recover the properly deformed kinematic chain as a set of *revised joints* $\tilde{\mathbf{P}}^t = \{\tilde{\mathbf{p}}_i \mid i = 1, \ldots, n_p\}$ in the deformed space. It is done by updating each unconstrained joint's position in a hierarchical order to enforce the length of each kinematic chain link to be constant. Note that the order of joint connections is preserved at all times after initialization to maintain the hierarchical structure of the kinematic chain and stability during optimization. The detailed steps to recover the kinematic chain are explained in the supplementary material.

As we have the fixed associations between anchors and the initial kinematic chain built in the canonical space as $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$, and $\mathbf{G}_i$ by Eq. (3). We then use Eq. (4) to recover each anchor's revised position $\tilde{\mathbf{a}}_i$ and thus apply Eq. (6) to infer a revised anchor transformation $\tilde{\mathbf{T}}_i^t$ that satisfies the association constraints between the anchors and the kinematic chain, while not breaking the kinematic chain at the same time. During the optimization, we introduce a novel loss that minimizes the differences between the unconstrained anchors and the revised anchors:

$$\mathcal{L}_{\text{anchors}} = \sum_{i=1}^{n_a} \left\| \tilde{\mathbf{a}}_i - \hat{\mathbf{T}}_i^t \mathbf{a}_i \right\|_2^2. \tag{11}$$

This loss aims to make the optimized deformation anchors and the MLP network $\mathcal{F}_{\mathbf{A}}$ kinematic chain aware.

Additionally, we optimize the kinematic chain by introducing a learnable *additive residual* term for each kinematic chain link. The length of each link in the kinematic chain is updated using the learnable residual during optimization such that the kinematic chain can better adapt to the object's shape and the learned deformation anchors. The details on how the kinematic chain is updated without breaking itself are included in the supplementary material.

In the kinematic chain aware optimization stage, all the MLP networks are also optimized jointly to finetune the object's shape and appearance with the loss functions in the initial optimization stage. At the end of this stage, a fully controllable articulated model for the object is built, and animations of the object can be easily done by directly manipulating the final kinematic chain.

4. Experiments

4.1. Datasets

Synthetic Datasets. We validate our approach on two synthetic datasets. The Eagle [70] dataset consists of 5 single-view videos (900 frames in total) of an animated eagle model from Turbosquid. We also created a new dataset (iiwa dataset) that contains 4 single-view videos (1200 frames in total) of an animated iiwa robot arm model from BlenderKit. Please find the details and examples of our iiwa dataset in the supplementary material.

AMA Dataset. We also evaluate our approach on the Articulated Mesh Animation (AMA) Dataset [54]. It contains 10 sets of multi-view videos of 3 different clothed humans using synchronized cameras. We use 2 challenging sets of videos and treat the multi-view videos as a collection of monocular videos by discarding the time synchronization information, which yields 2600 monocular frames in total.

In-the-wild Datasets. In addition, we validate our approach on in-the-wild monocular captures of various deformable and articulated objects datasets. In the main paper, we focus on experimental results on the synthetic and AMA datasets. Please see the supplementary material for implementation details and results on in-the-wild datasets.

4.2. Implementation Details

For synthetic and AMA datasets, we use ground-truth foreground masks during the optimization. For root pose initializations, we use ground-truth root poses for synthetic datasets, while a pre-trained PoseNet [70] is used to initialize the root poses for AMA Dataset. The root poses are also updated during optimization with learned residual terms for AMA Dataset. The optical flows for all datasets are computed by VCN [67] at frame intervals of $\{1, 2, 4, 8, 16, 32\}$ in both forward and backward time directions. We use the *keys* of the last transformer layer from a pre-trained ViT-S/8 model [3] as supervisions for the canonical features. We perform PCA on the 2D image features to reduce the dimension from 384 to 16, to match our canonical feature's dimension. To generate mesh from our implicit representations of the object shape, we run marching cubes on $256 \times 256 \times 256$ grids to extract the zero-level set of the learned SDF. We use 10, 25, and 36 anchors for iiwa, Eagle, and AMA Dataset, respectively. Our code and iiwa dataset will be made publicly available.

4.3. Pose Manipulation Results

The advantage of having a kinematic chain is that users can manipulate the 3D object to poses that had never occurred in training videos. Although some surface reconstruction methods [69, 70] allow users to move their learned "floating" control points around for custom animations, these control points are not intuitive to manipulate because it's unclear how each "floating" control point contributes to the overall deformation. Moreover, these "floating" control points are optimized for only memorizing the poses and movements in training videos. In our approach, the pose manipulations are done by directly applying rotations at kinematic chain joints to rotate and twist corresponding kinematic chain links, which is much more straightforward and intuitive for animation purposes. We do not impose any constraint on the kinematic chain configuration as long as the order of joint connections and the length of each link are preserved. Users are expected to apply valid transformations to the kinematic chain to obtain feasible shapes without mesh degeneration. We show the learned canonical shapes, optimized kinematic chains, and the re-posed objects in novel poses that do not occur in training videos in Figure 4. Please see our supplementary material for additional re-posing results.

4.4. Reconstruction Results

There is no prior work on building category-agnostic and animatable 3D models from monocular videos, and reconstruction quality is an important factor for realistic novel view synthesis and animations. Therefore we evaluate the performance of our method on 3D surface reconstruction using ground-truth 3D meshes provided by each dataset.

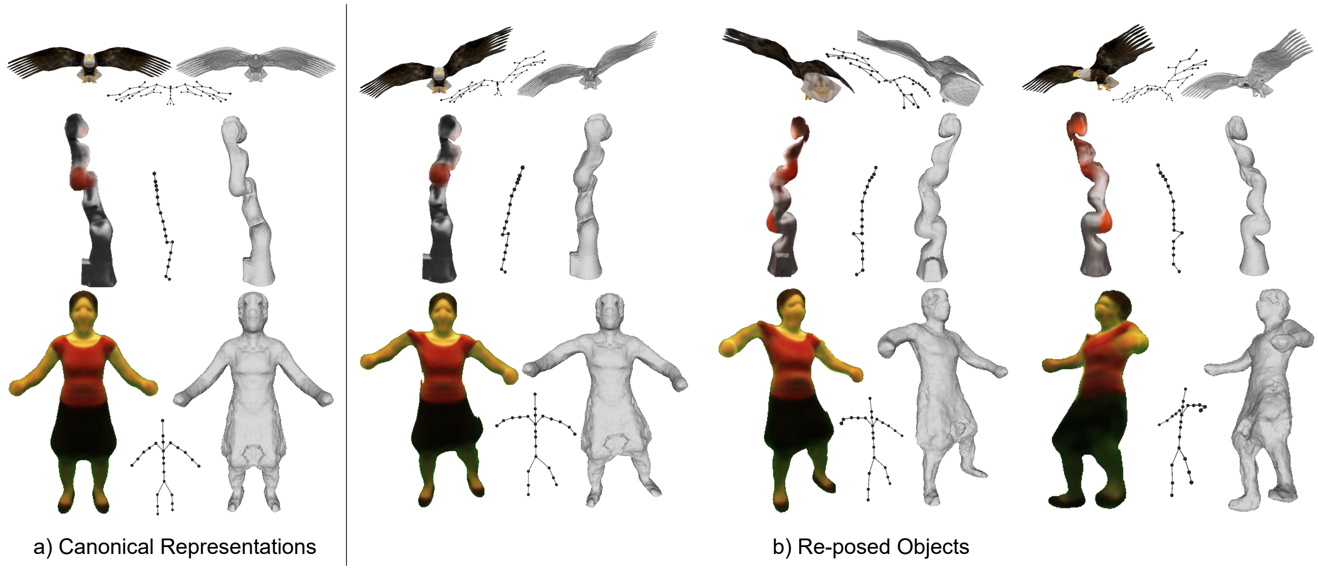

Figure 4. **Pose manipulation examples**. In a) we show the learned canonical representations of the object for each dataset. We perform pose manipulations using optimized kinematic chain and show the transformed kinematic chain, re-posed mesh as well as the rendered object in three different views in b).

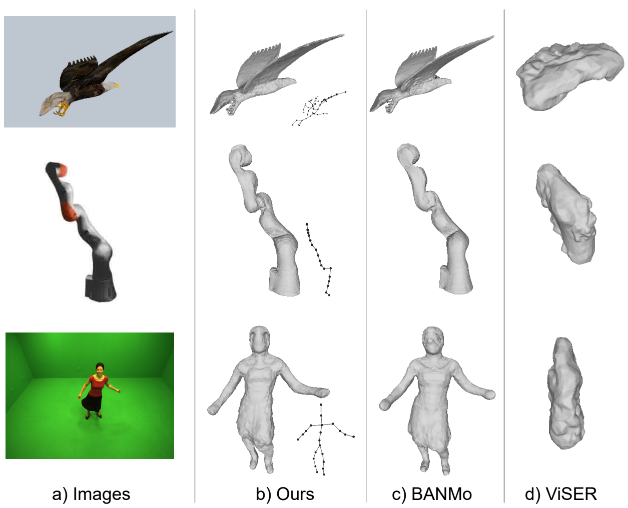

Figure 5. **Qualitative comparision of our method with BANMo [70] and ViSER [69].**

Metrics. We align the reconstructed 3D mesh to the ground-truth mesh using Iterative Closest Point (ICP) if they are in different scales and orientations before evaluation. Following [9, 11, 48, 70], we report 3D Chamfer Distances between reconstructed mesh vertices and the ground-truth mesh vertices measured in centimeters. In addition, we follow [51] to report the F-scores at a distance threshold of 2%. All the reported numbers are averaged across all frames for each dataset. In general, these metrics measure the following two aspects of the reconstruction quality for each frame in the training videos: the overall quality of the reconstructed 3D mesh surface, and whether the model is able to predict an accurate shape corresponding to the correct object body pose.

Table 1. 3D surface reconstruction results evaluated in 3D Chamfer Distances (↓) and F-scores (↑) at a distance threshold of 2% averaged across all frames.

Method	Eagle		iiwa		AMA-swing	
	CD	F@2%	CD	F@2%	CD	F@2%
ViSER	32.6	9.4	20.6	18.1	17.9	39.2
BANMo	4.44	82.72	5.55	54.58	**9.28**	**56.24**
Ours	**4.21**	**83.38**	**5.52**	**56.53**	9.69	53.29

Baselines. We compare our approach with two baselines in 3D surface reconstruction for deformable objects: BANMo [70] and ViSER [69]. We provide the same root pose initializations to all the methods for fair comparison on each dataset. Note that BANMo and ViSER *cannot* perform direct pose manipulations, and BANMo utilizes DensePose CSE feature embeddings [34] designed for humans and quadruped animals, which makes it a category-specific method. We show qualitative comparision of our method with BANMo [70] and ViSER [69] in Figure 5. Our approach can achieve similar 3D surface reconstruction quality using a category-agnostic approach while enabling direct pose manipulations. We also show the quantitative comparison with the same baselines: BANMo [70] and ViSER [69] on each dataset in Table 1.

4.5. Ablations

We run ablation studies on important components of our method to evaluate their effects. Please refer to our supplementary materials for more comprehensive results.

Table 2. Ablation on kinematic chain aware optimization evaluated in 3D Chamfer Distances (↓) and F-scores (↑) at a distance threshold of 2% averaged across all frames.

Method	Eagle		iiwa		AMA-swing	
	CD	F@2%	CD	F@2%	CD	F@2%
w/o opt.	7.44	57.87	5.69	52.44	11.13	48.88
with opt.	4.21	83.38	5.52	56.53	9.69	53.29
Improv.	-3.23	+25.51	-0.17	+4.09	-1.44	+4.41

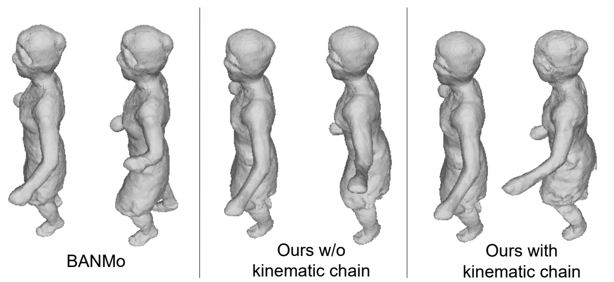

Figure 6. **Ablation on kinematic chain**. We show canonical shapes (left) and the predicted shapes (right) in the same time instance for each method. The length of the left arm is preserved with the help of kinematic chain.

Kinematic chain. A kinematic chain enables explicit pose manipulations and effectively reduces the unrealistic surface deformations that often occur in 3D surface reconstruction methods. We show the improvements of having a kinematic chain in Figure 6. The kinematic chain correctly re-poses the person's arm without volume loss, while BANMo [70] and our method with unconstrained anchors fail to model the correct deformations. We also compare the reconstruction results of directly using kinematic chain initialization from [66] and using optimized kinematic chain in Table 2. We are able to gain considerable improvements with the kinematic chain aware optimization stage (Section 3.4.2) because the optimized kinematic chain aligns better with the underlying object body pose in each frame.

Canonical features. DensePose CSE feature embeddings [34] provide pre-trained 2D-3D correspondences for humans and are used by BANMo [70] on the AMA dataset. We also test our approach by replacing the DINO-ViT [3] features with the same DensePose CSE feature embeddings to supervise our canonical features. As shown in Figure 7, the arm is merged with the human body if we disable the canonical feature learning in our method. This is because the model cannot differentiate between the human body and the arm at this pose due to the lack of the object parts level visual cues. With DINO-ViT features, our method can separate the arm from the human body. However, it does not accurately predict the pose of the arm. The small gap between our approach with DINO-ViT and BANMo in Table 1 mainly comes from these inaccurate pose predictions because the quality of the reconstructed mesh surface is similar, as shown in Figure 5.

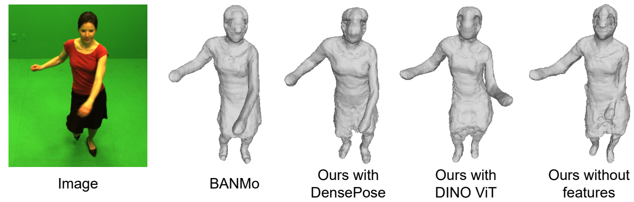

Figure 7. **Ablation on canonical features.** All methods recover high-fidelity surface, but only BANMo and our approach with DensePose supervision correctly predict the left arm's pose.

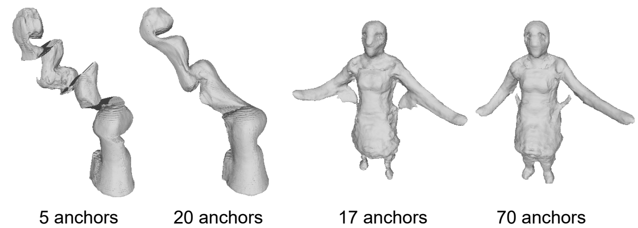

Figure 8. **Ablation on number of anchors.** We show the results of using half and double amounts of anchors on iiwa and AMA.

Number of deformation anchors. The number of deformation anchors is an object-specific hyper-parameter that plays a crucial role in modeling surface deformations. We show the results of using half and double amounts of anchors on AMA and iiwa datasets in Figure 8. Having a small number of anchors limits our method's ability to model deformations at the robot arm's joints and the human's elbows, while having too many anchors makes the optimization process harder due to the over-parameterization of the deformations at the object's surface.

5. Conclusion

In this paper, we present a novel reconstruction pipeline that builds an animatable 3D model for any articulated objects from a collection of monocular videos. We leverage the foreground masks, optical flow, and pre-trained image features to iteratively optimize the kinematic chain along with the object's shape, appearance and deformation parameters. Our method can be generalized to any articulated object as it does not rely on category-specific priors. It allows users to render the object in arbitrary viewpoints and to perform pose manipulations in 3D directly while achieving state-of-the-art 3D surface reconstruction results. However, our kinematic chain does not check for unreachable states, such as chain collisions and foldings. We leave the task of learning physically plausible kinematic chains to future work. In addition, our deformation anchors are optimized based on the observed deformations from training videos. Therefore, our model does not generalize well to the novel poses that involve unseen object surface deformations.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021. 2, 4

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 3, 4, 6, 8

[4] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, 2019. 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. In *ICCV*, 2021. 2

[6] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *ICCV*, 2019. 1

[7] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *ECCV*, 2020. 1, 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4

[9] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 7

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 1

[11] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018. 7

[12] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *ICCV*, 2021. 2

[13] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, 2020. 2

[14] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 1

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

[16] 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

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

[18] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 2

[19] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *ICCV*, 2019. 1, 2

[20] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *WACV*, 2020. 1, 2

[21] Jogendra Nath Kundu, Mugalodi Rakesh, Varun Jampani, Rahul Mysore Venkatesh, and R Venkatesh Babu. Appearance consensus driven self-supervised human mesh recovery. In *ECCV*, 2020. 1, 2

[22] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *ECCV*, 2022. 2

[23] 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

[24] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 1, 2

[25] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 1

[26] 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. In *SIGGRAPH*, 2021. 1, 2

[27] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019. 2

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *SIGGRAPH*, 2015. 1, 2

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

[30] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017. 2

[31] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 1

[32] 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. 1

[33] 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. 1, 2, 3, 5

[34] Natalia Neverova, David Novotny, Marc Szafraniec, Vasil Khalidov, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. In *NeurIPS*, 2020. 4, 7, 8

[35] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects. In *CVPR*, 2022. 1, 2

[36] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *CVPR*, 2021. 1, 2

[37] 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. 1, 2

[38] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 1, 2

[39] 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. 1, 2

[40] 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. 1, 2

[41] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 1, 2

[42] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. In *SIGGRAPH*, 2022. 1

[43] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *CVPR*, 2021. 2

[44] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. In *IJCV*, 2008. 4

[45] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. Neural dense nonrigid structure from motion with latent space constraints. In *ECCV*, 2020. 1, 2

[46] Shih-Yang Su, Timur Bagautdinov, and Helge Rhodin. Danbo: Disentangled articulated neural body representations via graph neural networks. In *ECCV*, 2022. 2

[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] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 7

[49] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 4

[50] Jiapeng Tang, Lev Markhasin, Bi Wang, Justus Thies, and Matthias Nießner. Neural shape deformation priors. In *NeurIPS*, 2022. 2

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

[52] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Nonrigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021. 1, 2

[53] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *CVPR*, 2022. 2, 4

[54] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH*, 2008. 6

[55] Minh Phuoc Vo, Yaser A Sheikh, and Srinivasa G Narasimhan. Spatiotemporal bundle adjustment for dynamic 3d human reconstruction in the wild. In *TPAMI*, 2020. 1, 2

[56] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 1, 2

[57] 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. In *NeurIPS*, 2021. 3

[58] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: animatable volume rendering of articulated human sdfs. In *ECCV*, 2022. 1, 2

[59] Fangyin Wei, Rohan Chabra, Lingni Ma, Christoph Lassner, Michael Zollhöfer, Szymon Rusinkiewicz, Chris Sweeney, Richard Newcombe, and Mira Slavcheva. Self-supervised neural articulated shape and appearance models. In *CVPR*, 2022. 1

[60] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022. 1, 2

[61] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Magicpony: Learning articulated 3d animals in the wild. *arXiv preprint arXiv:2211.12497*, 2022. 2

[62] Yuefan Wu, Zeyuan Chen, Shaowei Liu, Zhongzheng Ren, and Shenlong Wang. Casa: Category-agnostic skeletal animal reconstruction. In *NeurIPS*, 2022. 2

[63] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019. 1, 2

[64] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *CVPR*, 2020. 1, 2

[65] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019. 2

[66] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. In *SIGGRAPH*, 2020. 3, 5, 8

[67] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 6

[68] 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. 1, 2, 3, 5

[69] 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. 1, 2, 3, 4, 5, 6, 7

[70] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[71] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. In *NeurIPS*, 2022. 1, 2

[72] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 3

[73] 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

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

[75] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *CVPR*, 2021. 2

[76] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 1

[77] Haitian Zeng, Yuchao Dai, Xin Yu, Xiaohan Wang, and Yi Yang. Pr-rrn: pairwise-regularized residual-recursive networks for non-rigid structure-from-motion. In *ICCV*, 2021. 2

[78] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 1

[79] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, 2019. 2

[80] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *CVPR*, 2018. 1

[81] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *CVPR*, 2017. 1