S3O: A Dual-Phase Approach for Reconstructing Dynamic Shape and Skeleton of Articulated Objects from Single Monocular Video

Hao Zhang [1]   **Fang Li** [1]   **Samyak Rawlekar** [1]   **Narendra Ahuja** [1]

Abstract

Reconstructing dynamic articulated objects from a singular monocular video is challenging, requiring joint estimation of shape, motion, and camera parameters from limited views. Current methods typically demand extensive computational resources and training time, and require additional human annotations such as predefined parametric models, camera poses, and key points, limiting their generalizability. We propose Synergistic Shape and Skeleton Optimization (S3O), a novel two-phase method that forgoes these prerequisites and efficiently learns parametric models including visible shapes and underlying skeletons. Conventional strategies typically learn all parameters simultaneously, leading to interdependencies where a single incorrect prediction can result in significant errors. In contrast, S3O adopts a phased approach: it first focuses on learning coarse parametric models, then progresses to motion learning and detail addition. This method substantially lowers computational complexity and enhances robustness in reconstruction from limited viewpoints, all without requiring additional annotations. To address the current inadequacies in 3D reconstruction from monocular video benchmarks, we collected the PlanetZoo dataset. Our experimental evaluations on standard benchmarks and the PlanetZoo dataset affirm that S3O provides more accurate 3D reconstruction, and plausible skeletons, and reduces the training time by approximately 60% compared to the state-of-the-art, thus advancing the state of the art in dynamic object reconstruction. The code is available on GitHub at: https://github.com/haoz19/LIMR.

Table 1. Difference between S3O and existing methods. Detailed comparisons are provided in Supp. Sec.A.

Method	Shape Template	Skeleton Template	Rigging	Multiple Videos	Camera Pose	Learn Skeleton
SMAL	✓	✓	skeleton	✗	✗	✗
LASR	✗	✗	bones	✗	✗	✗
ViSER	✗	✗	bones	✗	✗	✗
BANMo	✗	✗	bones	✓	✓	✗
CASA	✓	✓	skeleton	✗	✗	✗
WIM	✗	✗	skeleton	✓	✓	✓
CAMM	✗	✓	skeleton	✓	✓	✗
RAC	✗	✓	skeleton	✓	✓	✗
MagicPony	✗	✓	skeleton	✓	✗	✗
S3O	✗	✗	skeleton	✗	✗	✓

1. Introduction

Given a single monocular video of an object undergoing articulated motion caused by an underlying skeleton of rigid bones (pairs of bones connected at joints and each bone surrounded by nonrigid skin), our goal is to estimate the visible dynamic 3D shape and invisible skeleton pose. We represent the motion of the objects using a physical 3D skeleton (Magnenat et al., 1988; Baran & Popović, 2007b; Wang et al., 2007; Xu et al., 2020) with certain poses. This task is highly challenging as it requires concurrent learning of shape, skeleton, motion, and camera parameters from a limited number of camera views. An error in any one of them can seriously impact the reconstruction results. Recent methods (Weng et al., 2022; Yang et al., 2021b; 2022; Luvizon et al., 2019) make use of virtual bones to represent the motion. These virtual bones do not model the physical skeleton as well as our physical bones (Fig.3 and Sec.4.3). Some methods (Yang et al., 2021a;b; 2022) calculate the displacement of each vertex in terms of the motion of the virtual bones and skinning weights using the blend skinning technique. They assign a bone to a group of nearby surface vertices without regard to the underlying structure of the mesh. As a result, these methods often fail to achieve physically plausible bone distribution as shown in Fig.3 (d). Other, model-based methods (Yang et al., 2023; Wu et al., 2023b; Kuai et al., 2023; Wu et al., 2022; Goel et al., 2023) use pre-defined shape and skeleton templates to attain a physically plausible skeleton and bone distribution; however, they require these as category-specific ground truth inputs, instead of estimating them from the video. This diminishes their generalizability to out-of-distribution (OOD)

objects. In terms of the amount of input, many recent NeRF-based methods (Yang et al., 2022; Noguchi et al., 2022; Wu et al., 2023b) require multiple videos, captured from a wide range of viewpoints, along with accurate camera calibration. Most current methods require substantial computational resources to obtain reasonable results, taking $8 - 30$ hours on A100 GPUs to train on a single object, which again reduces their usability.

In this work, we introduce Synergistic Shape and Skeleton Optimization (S3O), a two-phase methodology that (1) learns visible shape and the underlying skeleton, without reliance on pre-specified parametric models or additional human annotations, and (2) significantly reduces the training time by segregating the learning of distinct parameters such as shape, motion, and camera parameters in two phases: a coarse parametric model learning phase, and a motion and fine model learning phase.

In phase one, we automatically extract a canonical frame from the entire video, which depicts the object in standard poses (e.g., side view) via the horizontal-to-vertical distance ratio, lift its 2D skeleton to 3D, and learn a coarse rest shape. (Sec.3.2). The second phase is dedicated to learning the time-varying parameters, including the motion and camera parameters, and refining the shape and skeleton details. Here, we employ a method akin to the Expectation-Maximization (EM) algorithm, alternating the update of shape and skeleton. During the E-step, we utilize the current skeleton to deduce the bones and update the shape, time-varying camera parameters, and temporal transformation of each bone (motion). In the M-step, we begin by adapting the skeleton to align with the deformed mesh using skinning weights (Sec.3.4). Subsequently, utilizing the current mesh and time-varying parameters, we calculate the 2D motion and length of each bone, thereby forming physical constraints that guide the update of the skeleton's structure. Since current datasets mostly include minimal camera motion, they fail to test an algorithm's robustness against extensive camera movement. To overcome this, we introduce a new dataset, PlanetZoo, designed to assess algorithm performance under significant, long-range camera movements. Key **contributions** of our work include:

- We introduce S3O which estimates the 3D shape and skeleton of articulated objects from a single monocular video, without prior templates and ground truth camera poses, while significantly reducing training time and computational complexity over SOTA.

- We introduce a new dataset called PlanetZoo, to help evaluate 3D reconstruction methods from single monocular video.

- Experiments on standard and PlanetZoo videos show that S3O outperforms SOTA methods in shape recon-

struction quality while requiring 60% less training time.

- S3O displays superior generalization capabilities over RigNet (Xu et al., 2020) and Skeletor (Au et al., 2008) in reconstructing skeletons.

2. Related Work

3D Skeleton Extraction for Reconstruction. Skeletons lack a uniform definition and are subject to varying requirements across different applications. Some efforts aim to capture the object's medial axis (Au et al., 2008), a process known as curve-skeleton extraction. For instance, voxel-thinning methods (Ma et al., 2002; Palágyi & Kuba, 1999; Wang & Lee, 2008) achieve this by iteratively pruning boundary voxels while preserving topology. In contrast, distance-field methods (Hassouna & Farag, 2005; Zhou & Toga, 1999; Wade, 2000; Bitter et al., 2001; Cornea et al., 2005; Ma et al., 2003) involve computing a distance transform for each interior point of a 3D object and then identifying ridges in the field to select a set of candidate voxels. Recent works (Xu et al., 2020; Baran & Popović, 2007a; Yao et al., 2023; Yu et al., 2017; Vlasic et al., 2008b) focus on learning skeletons, tailored to reflect the mobility of an articulated shape. These physical skeletons, influenced by more than just shape geometry, incorporate the placement of joints and bones based on the animator's understanding of the character's anatomy and anticipated deformations.

Model-Based and Class-Specific Approach. The model-based approach employs a parametric framework, incorporating structured mesh templates and associated skinning weights, to tackle the challenge of estimating 3D shape and pose. Extensive research (Zuffi et al., 2017; Xiang et al., 2019; Loper et al., 2023) dedicated to 3D reconstruction of human and animal forms makes use of these parametric models, which are informed by 3D scans of actual human figures or animal replicas. The latest regression techniques (Yang et al., 2023; Zuffi et al., 2019; Biggs et al., 2020; Badger et al., 2020; Kocabas et al., 2020) devised to deduce pose parameters have shown to yield satisfactory reconstruction outputs, though they typically depend on manually provided annotations like pose, keypoints, and rely on predefined parametric models for their training processes. Consequently, these model-based approaches encounter difficulties when applied to the reconstruction of objects in an open-world context and exhibit limited generalizability in the absence of comprehensive annotations. Current trends aim to minimize reliance on 3D annotations, opting for 2D annotations like silhouettes and key points (Goel et al., 2020; Kanazawa et al., 2018; Li et al., 2020b). Single-view image reconstruction achieves impressive results without explicit 3D annotations (Li et al., 2020a; Kanazawa et al., 2018; Kulkarni et al., 2020). MagicPony (Wu et al., 2023b)

learns 3D models for objects such as horses and birds from single-view images, but faces challenges with fine details and substantial deformations, especially for unfamiliar objects.

Reconstruction from Video. Instead of leveraging predefined parametric models, some works (Yang et al., 2021a;b; 2022) jointly learn the skin weights, shape, camera parameters, and motion from monocular video(s). This method requires only mask annotations, eliminating the need for additional labeling and significantly enhancing its generalizability in real-world scenarios. However, due to the absence of structural information about the object and annotations regarding camera parameters, this approach tends to be effective only under certain conditions, such as with videos that do not involve extensive camera movement and that contain a multitude of standard viewpoints, including side and front views. Moreover, the Non-Rigid Structure from Motion (NRSfM) technique (Torresani et al., 2008; Kumar, 2019) is capable of functioning without explicit 3D data. It processes 2D keypoints and optical flow derived from readily available methods (Teed & Deng, 2020). However, this process necessitates persistent point tracking, which is often impractical or unavailable/unreliable in real-world settings.

3. Method

3.1. Terminology and Overview

Towards articulated object reconstruction from video sequences, we propose a method, named Synergistic Shape and Skeleton Optimization (S3O), that concurrently learns parametric models and time-varying motion parameters. The parametric model includes canonical shape $\mathbf{M}_c$, underlying skeleton $\mathbf{S_c} = \{\mathbf{B} \in \mathbb{R}^{B \times 13}, \mathbf{J} \in \mathbb{R}^{J \times 5}\}$, and skinning weights $\mathbf{W} \in \mathbb{R}^{N \times B}$. The time-varying motion parameters include the camera parameters and articulation $\mathbf{T}^t = \{\mathbf{T}_0^t, \mathbf{T}_1^t, ..., \mathbf{T}_B^t\}, \mathbf{T}_b^t \in SE(3)$, where $\mathbf{T}_0^t$ and $\mathbf{T}_{b>0}^t$ are the root body and B bone transformations. The quantities B, and J represent the count of bones and joints. **Skeleton Definition.** In the skeleton, bones $\mathbf{B}$ are further defined by Gaussian centers $\mathbf{B}[:, :3]$, precision matrices $\mathbf{B}[:, 3:12]$, and bone lengths $\mathbf{B}[:, 12]$. Joints $\mathbf{J}$ are specified by connections between bones $\mathbf{J}[:, :2]$ and their spatial coordinates $\mathbf{J}[:, 2:]$.

Overview of S3O. As illustrated in Algorithm.1, the algorithm commences with a Coarse Shape Phase, wherein it selects one canonical frame from the video (Sec.3.2) to extract a 2D skeleton and derives an initial coarse 3D parametric model (Sec.3.2). This model forms the basis for the subsequent Motion & Fine Shape Phase, a process of time-varying parameters optimization and 3D shape and skeleton refinement (Sec.3.4). Similar to an Expectation-

Maximization approach, the algorithm alternates between adapting the skeletal structure to optimize shape and applying physical constraints to ensure the physical plausibility of the model. The optimization cycles through these steps, gradually improving the representation until convergence is achieved (the skeleton doesn't change), thus providing an accurate and dynamic 3D reconstruction of the articulated subjects in the video. Details of each step and the associated physical constraints are delineated in the respective sections of the paper below.

3.2. Coarse Model Learning

Canonical Frames and 2D Skeleton Extraction. Given a set of video frames, we first apply the medial axis transform (Lee, 1982; Ahuja & Chuang, 1997) to their silhouettes where we gradually remove the pixels from the border until we obtain a single-pixel-wide potential skeleton that preserves the original region's extent and connectivity. Along the 2D skeleton curve, we categorize the points into three distinct groups: junctions (points with three or more neighbors), endpoints (points with only one neighbor), and connections (points with two neighbors). Next, we remove the connections and reestablish connections between the remaining endpoints and junctions while preserving the overall topology as shown in Fig.1. By computing the skeleton's horizontal-to-vertical distance ratio across all frames, the frame with the highest ratio is selected as the canonical frame.

Generating Coarse 3D Skeleton and Mesh. We leverage the 2D skeleton of the canonical frame as the foundation for generating the coarse 3D skeleton following (Yao et al., 2022; 2023). The key idea in this transformation is the utilization of the plane of symmetry found in most articulated objects. Starting with the silhouette and the 2D skeleton, we create an ellipsoid around each edge (blue lines in Fig.1), ensuring it fits the silhouette as shown in Fig.14. Next, we identify the symmetric parts by the following features: (1) sum of average DINO features, (2) bone length, and (3) average radius of the ellipsoid similar to (Yao et al., 2023). For the symmetric parts, we set the z-coordinates to be on the opposite side of the symmetry plane. Next, we deform the neural surface to obtain mesh. We follow the deformation methods from (Lindell et al., 2022), which performs analysis-by-synthesis to supervise the shape reconstruction. To supervise deformations, we rely on silhouette, part silhouette, and semantic consistency loss described in Sec.D.

3.3. Time-Varying Articulation

Skinning Weights. $\mathbf{W}$ is designed to represent the probabilities that each vertex corresponds to B semi-rigid parts. Following LASR (Yang et al., 2021a), the skinning weights
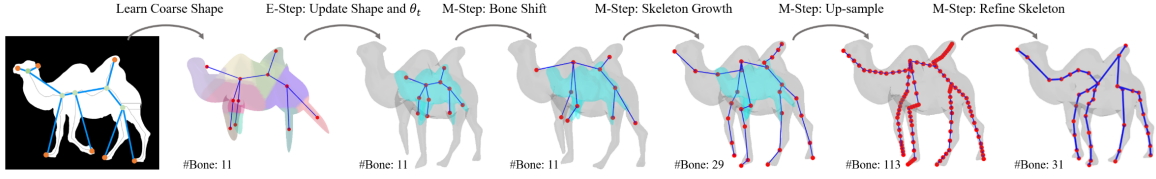
Figure 1. **Overview of Skeleton Learning of S3O**: We start by deriving a 3D skeleton from 2D inputs and learning a coarse shape, represented in 'cyan' for the coarse and 'gray' for the current shape. The next step involves refining this shape by adjusting the skeleton, shifting bones, and expanding the structure. We then upsample the skeleton points for greater detail before applying final physical constraints to ensure proper results.

are modeled by the mixture of B Gaussian ellipsoids (semi-rigid parts) as: $\mathbf{W}_{n,b} = \mathcal{F}e^{-\frac{1}{2}(\mathbf{X}_n - \mathbf{C}_b)^T \mathbf{Q}_b(\mathbf{X}_n - \mathbf{C}_b)}$, where $\mathcal{F}$ is the factor of normalization and precision matrices $\mathbf{Q}_b = \mathbf{V}_b^T \mathbf{\Lambda}_b \mathbf{V}_b$. In each Gaussian ellipsoid, $\mathbf{C} \in \mathbb{R}^{B \times 3}$ denotes Gaussian centers, $\mathbf{V} \in \mathbb{R}^{B \times 3 \times 3}$ defines the orientation and $\mathbf{\Lambda} \in \mathbb{R}^{B \times 3 \times 3}$ denotes the diagonal scale matrix. $\mathbf{X}_n$ is the 3D location of vertex n.

Dynamic Rigidity. To guarantee the flexibility of the deformed articulated mesh without losing stability, following LIMR (Zhang et al., 2023), we leverage Dynamic Rigidity (DR) loss $\mathcal{L}_{\text{DR}}$ to provide more freedom for the vertices close to the joints and less freedom for the ones between adjacent joints (along the bones), in that the vertices around joints are always more likely to be deformed. The freedom rules are derived from the proposed Rigidity Coefficient $\mathbf{R} \in \mathbb{R}^{N \times N}$ calculated by skinning weights $\mathbf{W} \in \mathbb{R}^{N \times B}$.

Rigidity Coefficient $\mathbf{R} \in \mathbb{R}^{N \times N}$ is proposed to represent the freedom of each vertex relative to each neighbor vertex based on the a priori assumption on skinning weights $\mathbf{W}$. For example, given $\mathbf{W} \in \mathbb{R}^{N \times B}$, each vertex $\mathbf{X}_n$ is provided a discrete probability distribution regarding all bones. The denser the distribution, the higher the likelihood that the vertex is assigned to the core of only one specific semi-rigid part (bone). In stark contrast, the sparser the distribution, the more probable the vertex is attributed to the border between two semi-rigid parts, which is actually around the joint. We model the density of distribution by entropy $\mathcal{E}$ and simulate the rigidity coefficient $\mathbf{R}$ by the product of entropies:

$$\mathbf{R}_{i,j} = \mathcal{E}(i)^{-1} \mathcal{E}(j)^{-1}, \tag{1}$$
$$\mathcal{E}(i) = \sum_{b=0}^{B-1} \mathbf{W}_{i,b} \log_2 \mathbf{W}_{i,b} + \lambda.$$

Here, $\mathbf{R}$ is a skew-symmetric matrix so that $\mathbf{R}_{i,j} = \mathbf{R}_{j,i}$ and $\mathbf{R}_{i,j}$ signifies the rigidity coefficient between vertices i and j. Besides, the stabilization constant is set as $\lambda = 0.1$ by default avoiding being divided by zero. Notably, in practice, $\mathbf{R}_{i,j}$ will be set to 0 if there is not an edge connecting vertices i and j.

Given Rigidity Coefficient $\mathbf{R}$, we define the Dynamic Rigid-

ity (DR) loss $\mathcal{L}_{\text{DR}}$ as following:

$$\mathcal{L}_{\text{DR}} = \sum_{i=1}^{n} \sum_{j \in N_i} \mathbf{R}_{i,j} \left| \left\| \mathbf{X}_i^t - \mathbf{X}_j^t \right\|_2 - \left\| \mathbf{X}_i^{t+1} - \mathbf{X}_j^{t+1} \right\|_2 \right|, \tag{2}$$

where $\mathbf{X}_i^t$ means the coordinate of vertex i at time t. DR loss allows more adaptive freedom to deformation compared to the conventional ARAP (As Rigid As Possible) loss (Yang et al., 2021a). The latter encourages the constant distance between the adjacent vertices, which limits the movement and deformation of the vertices near the joints.

Blend Skinning. The mapping from surface vertex $\mathbf{X}_n^0$ in canonical space (time 0 by default) to $\mathbf{X}_n^t$ at time t in camera space is designed by blend skinning. The forward blend skinning is shown below:

$$\mathbf{X}_n^t = \mathbf{T}_0^t \left(\sum_{b=1}^{B} \mathbf{W}_{n,b} \mathbf{T}_b^t \right) \mathbf{X}_n^0, \tag{3}$$

including the number of bones $\mathbf{B}$, skinning weights $\mathbf{W}$ and the transformation $\mathbf{T}^t = \{\mathbf{T}_b^t\}_{b=0}^{B}$ at time t, where $\mathbf{T}_0^t$ represents the root body transformation (Yang et al., 2021a) and $\mathbf{T}_{b>0}^t$ describes the bone transformation. Each $\mathbf{X}_n^0$ is first transformed by the weighted sum of each bone transformation $\mathbf{T}_{b>0}^t$ and then transformed by root body transformation $\mathbf{T}_0^t$ to achieve $\mathbf{X}_n^t$. On the other hand, the backward blend skinning maps $\mathbf{X}_n^t$ back to $\mathbf{X}_n^0$ utilizing $\mathbf{T}^{-1}$ instead of $\mathbf{T}$.

2D Optical Flow Warping for Bone Motion. Our task is to discern the 2D motion trajectory of each semi-rigid segment covering a bone, from time t to $t + 1$. 2D optical flow is represented as $\mathbf{F}^{2D,t}$, and the specific motion path for a bone is denoted as $\mathbf{F}^{B,t} \in \mathbb{R}^{B \times 2}$. Essentially, a bone's movement is a cumulative effect of the movements of its exterior vertices. As shown in Fig.8, this process involves back-projecting each pixel from the 2D optical flow into the 3D canonical space. Due to the imperfect alignment between vertices and pixels, the surface motion trajectory $\mathbf{F}^{S,t} \in \mathbb{R}^{N \times 2}$ is calculated through bilinear interpolation.

A critical aspect is the inaccuracy of applying optical flow directions to vertices on the obscured side of the surface.
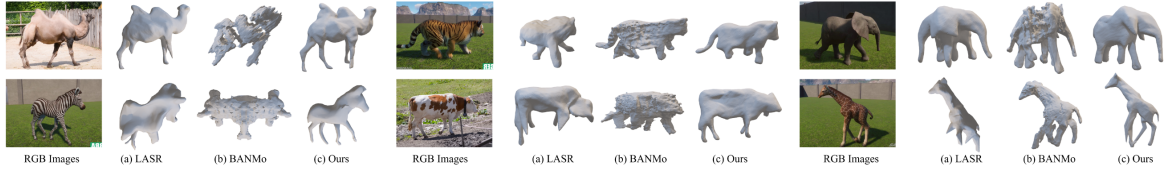
Figure 2. **Mesh Results.** We show the reconstruction results of (a) LASR, (b) BANMo, and (c) Ours in the DAVIS's `camel`, `cows` and PlanetZoo's `zebra`, `elephant`, `tiger`, and `giraffe`. Top view of the reconstruction results are shown in Fig.9.
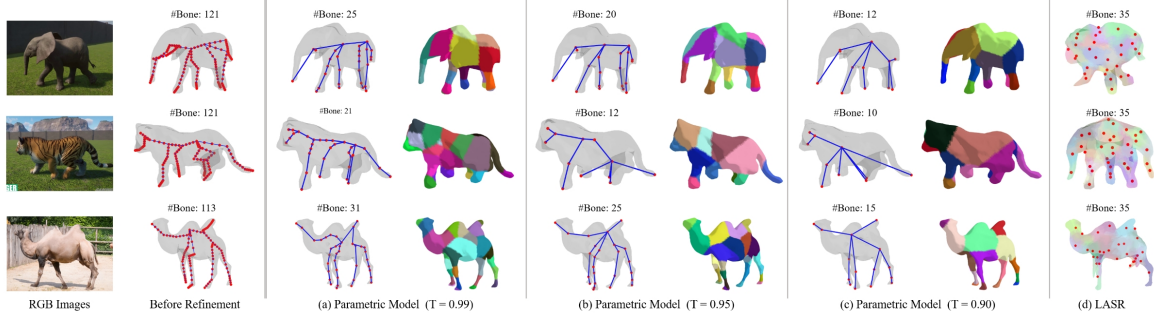

Figure 3. **Skeleton and Skinning Weight Results.** (a), (b), and (c) show the skeleton and skinning weights results from S3O with different bone motion similarity thresholds (0.99, 0.95, 0.90). (d) shows the bone distribution and skinning weights from LASR. Since each vertex on the surface is assigned to multiple bones (each given a different color) via the skinning weights, here we show the vertex in the color of the bone which has the maximum skinning weight for the vertex.

To rectify this, a visibility matrix $\mathcal{V}^t \in \mathbb{R}^{N \times 1}$ is created, reflecting the current mesh configuration and viewpoint at time t, determined via ray-casting. Optical flow for obscured vertices is thus disregarded.

The bone motion trajectory is then approximated by blending the optical flow of surface vertices. Considering the predefined skinning weights $\mathbf{W}$ and the derived surface flow path $\mathbf{F}^{S,t}$, the bone motion direction is expressed as:

$$\mathbf{F}_b^{B,t} = \sum_{n=0}^{N-1} \mathbf{W}_{n,b} \mathbf{F}_n^{S,t} \mathcal{V}_n^t, \quad \mathbf{F}^{B,t} = \{\mathbf{F}_0^{B,t}, ..., \mathbf{F}_{B-1}^{B,t}\}$$

(4)

Here, $\mathbf{F}_b^{B,t}$ represents the directional flow linked with bone b, while $\mathbf{F}_n^{S,t}$ and $\mathcal{V}_n^t$ specify the directional flow and the visibility of vertex n, respectively.

3.4. Motion and Fine Shape Phase

Our approach is grounded in two interdependent sets of learnable parameters. The first set includes 3D shape and time-varying motion parameters. The second set pertains to the underlying skeleton, encompassing the bones and articulation joints. To achieve a harmonized optimization of these parameter sets, we update both parameter sets iteratively similar to the EM algorithm. During the **E**-step, the skeleton remains fixed, and the first set of parameters is then updated based on the reconstruction loss. Subsequently, using the updated parameters, in the **M**-step we adjust the skeleton to match the current mesh and then determine the bone motion direction and bone lengths for constructing

physical constraints. Such constraints are utilized for optimizing the skeleton. The skeleton optimization consists of three operations: (1) bone shifting, (2) skeleton growth, and (3) physically constrained refinement. Since the coarse shape phase typically yields only an approximate shape, the second phase involves significant deformations and scale changes to the mesh in each iteration, as shown in Fig.1. The purpose of the Bone Shifting and Skeleton Growing operations is to automatically update the skeleton so that its structure aligns with the revised mesh.

Bone Shifting. We start with the assumption that, although each vertex of the mesh undergoes displacement, their relative position remains unchanged as the coarse shape already being closely representative of the actual shape. Therefore, we update the Gaussian center for the bone with the centroid of the corresponding semi-rigid part:
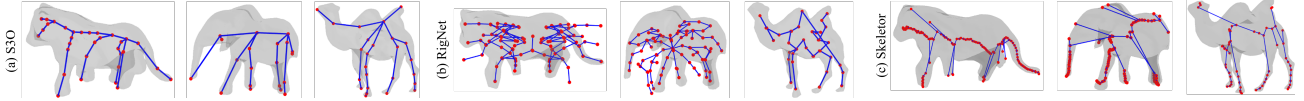
$$\mathbf{B}[b, :3] = \frac{1}{|\mathbf{I}_b|} \sum_{i \in \mathbf{I}_b} \mathbf{M}_c[i],$$

(5)

where b is the index of bone, $\mathbf{I}_b$ is a set of vertex indexes that belong to the semi-rigid part corresponding to bone b. $\mathbf{W}_{\text{one-hot}}[i, :] = \mathbf{W}[\mathbf{W}[i, :] == \arg\max_i \mathbf{W}[I, :]]$, $\mathbf{W}_{\text{one-hot}} \in \{0, 1\}^{N \times B}$, and $\mathbf{I}_b = \mathbf{W}_{\text{one-hot}} == 1$.

Skeleton Growing. The coarse shape often fails to accurately reconstruct the fine structures at the extremities of an object, such as limbs, which mostly need to be built during the second phase. This results resulting in an incomplete coarse skeleton. To complete it during the second phase, we introduce the skeleton growth that automatically extends the

Table 2. **2D Keypoint Transfer Accuracy.** This table presents the 2D keypoint transfer accuracy on DAVIS and PlanetZoo datasets. For classes from DAVIS (`camel, cow`) and PlanetZoo (`giraffe, tiger`), we carry out multiple executions and report the mean and standard deviation of the observed accuracy in each case. Time is in GPU hours on A100. Note that the results for LASR and ViSER here are from our executions of the codes provided by the authors. The authors in their papers provide a single number for accuracy. It is not clear if this represents the results of a single run, or mean/min/max of multiple runs. This makes our numbers hard to compare with those reported by the authors in their papers.

Method	DAVIS							PlanetZoo							
	camel	cow	dog	bear	dance-twirl	Ave.	Ave. time	giraffe	tiger	elephant	bear	zebra	deer	Ave.	Ave. time
ViSER	$71.7_{\pm4.4}$	$73.7_{\pm4.1}$	65.1	72.7	78.3	72.3	10.4	$51.2_{\pm4.6}$	$68.4_{\pm4.3}$	68.9	60.3	55.7	57.3	60.3	12.1
LASR	$75.2_{\pm2.9}$	$80.3_{\pm2.4}$	60.3	83.1	55.3	70.8	9.1	$56.3_{\pm3.7}$	$70.4_{\pm2.1}$	69.5	63.1	57.4	60.3	62.8	10.2
S3O	$\mathbf{79.8}_{\pm1.1}$	$\mathbf{83.2}_{\pm0.7}$	**67.3**	**86.7**	**84.7**	$\mathbf{80.4}_{\uparrow5.7}$	$\mathbf{3.8}_{\downarrow5.3}$	$\mathbf{63.1}_{\pm0.8}$	$\mathbf{72.8}_{\pm0.6}$	**71.3**	**66.4**	**61.5**	62.1	$\mathbf{66.2}_{\uparrow3.4}$	$\mathbf{4.2}_{\downarrow6.0}$

Figure 4. **Skeleton Comparision.** We compared the learned skeleton for `tiger, elephant, camel` from our approach with RigNet (Xu et al., 2020) and Skeletor (Au et al., 2008).

skeleton based on the new extremity structures formed by the mesh, ensuring the skeleton's growth is both systematic and reflects the nuanced changes in the mesh structure. The skeleton growth operates by identifying end parts [1] and subdividing them into 2 distinct parts along the direction of the skeletal edge if the maximum distance within the end part is more than twice of the initial value. Once identified, the algorithm computes the direction vector of the growth by taking the normalized vector pointing to the endpoint from the connected point on the skeleton. The vertices assigned to the endpoint are then projected onto this direction vector, and their distribution along the edge is used to subdivide them into k sections. Each section's centroid is calculated, and new skeletal points (bones) are added at these centroid locations. These points are interconnected to form new edges that replace the original edge, effectively growing the skeleton.

Physically Constrained Refinement. The skeleton is refined by adhering to two physical principles: (1) The fusion of bones $\mathbf{B}_b$ and $\mathbf{B}_{b'}$ is executed when they exhibit synchronized movements throughout all $\mathbf{H}$ frames $\{\mathbf{I}_f\}_{f=0}^{\mathbf{H}}$, suggesting they form a single semi-rigid structure. (2) A joint is established between $\mathbf{J}_j$ and $\mathbf{J}_{j'}$ upon observing significant distance variations in certain frames.

Accordingly,

$$\text{if } \max_f \text{Length}(\mathbf{B}_{b'}^f) - \min_f \text{Length}(\mathbf{B}_{b'}^f) > t_d \quad (6)$$

$$\Rightarrow \text{create } \widetilde{\mathbf{J}}; \quad (7)$$

$$\text{if } \min_f \mathcal{S}(\mathbf{F}_B^{b',f}, \mathbf{F}_B^{b'',f}) > t_o \quad (8)$$

$$\Rightarrow \text{merge} \mathbf{B}_{b'} \text{ with } \mathbf{B}_{b''} \text{ to generate } \widetilde{\mathbf{B}}, \quad (9)$$

---

[1]Vertices that belong to the endpoints of the existing skeleton.

where $\mathcal{S}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||_2 \times ||\mathbf{b}||_2}$ calculates the cosine similarity. Moreover, the position of $\widetilde{\mathbf{J}}$ is determined by averaging the coordinates of $\mathbf{J}_j$ and $\mathbf{J}_{j'}$, after which their connection is removed and redirected to $\widetilde{\mathbf{J}}$. The formulation $\widetilde{\mathbf{B}} = w_b' \mathbf{B}_{b'} + w_{b''} \mathbf{B}_{b''}$, where $w_b' = \frac{\exp \sum_{b=b'} \mathbf{W}_{n,b}}{\exp \sum_{b=b'} \mathbf{W}_{n,b} + \exp \sum_{b=b''} \mathbf{W}_{n,b}}$, is applied to Gaussian centers $\mathbf{C}$ and precision matrices $\mathbf{Q}$. However, the new bone length is directly derived from the joint coordinates at the extremities of the two bones.

Joint Positions and Bone Lengths Calculation. Utilizing the current bone coordinates $\mathbf{B}$, the inter-bone links $\mathbf{J}[:, :2]$, and the skinning weights $\mathbf{W}$, the procedure begins by locating the vertices $V_{i,j}$ situated directly above the joint $\mathbf{J}_{i,j}$ that connect bones i and j. A vertex v_n is classified as part of $V_{i,j}$ if its corresponding weights for bones i and j ($\mathbf{W}_{n,i}$ and $\mathbf{W}_{n,j}$) meet or exceed a threshold value t_r (commonly 0.4). The position of the joint $\mathbf{J}_{i,j}$ is then ascertained by averaging the coordinates of the vertices in $V_{i,j}$, formulated as $\mathbf{J}_{i,j}[:, 2 :] = \frac{\sum_{v_n \in V_{i,j}} \mathbf{X}_n}{|V_{i,j}|}$, where $\mathbf{X}_n$ denotes the coordinates of v_n. Finally, upon acquiring each joint's coordinates, we compute the lengths of bones by determining the distance between the coordinates of joint pairs.

3.5. Losses and Regularization

In the coarse shape phase, S3O leverages a reconstruction loss, which consists of silhouette loss, optical flow loss, texture loss, perceptual loss, DINO feature loss, and smooth, motion, and symmetric regularizations. In the fine shape phase, unlike the coarse shape phase, we eliminate the DINO feature loss. While the use of DINO features effectively aids in quickly learning a coarse model in the initial phase, their limited accuracy eventually hampers the acquisition of a more detailed shape in later stages. More
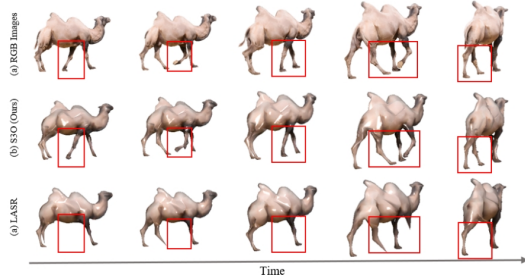
6

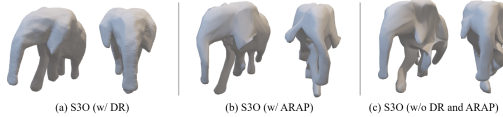*Figure 5.* **Rendering Results** of S3O and LASR on `camel` video.

Figure 6. **Enhanced Physical Plausibility with Dynamic Rigidity.** (a), (b), and (c) show the results of S3O with Dynamic Rigidity (DR), with As-Rigid-As-Possible (ARAP), and without both regularization terms.

details and discussion are in Sec.D.

4. Experiments and Discussion

In this section, we conduct a detailed evaluation of our approach, focusing on two main areas: 3D Shape Reconstruction (Sec.4.2) and 3D Skeleton Reconstruction (Sec.4.3). For shape reconstruction, we compare our method with state of art techniques including Hi-lASSIE (Yao et al., 2023), LASR (Yang et al., 2021a), and BANMo (Yang et al., 2022). All the requirements for S3O, LASR, and Hi-LASSIE are the same while BANMo needs an extra pre-trained PoseNet to obtain the initial camera pose. For skeleton reconstruction, we use RigNet (Xu et al., 2020) and Skeletor (Au et al., 2008) as benchmarks. Note that RigNet is trained with ground truth 3D annotations, such as skinning weights and skeletons, while S3O and Skeletor (Au et al., 2008) do not require any 3D annotation. More experimental results and comparisons are given in Sec.E. Details and results for the benchmarks and implementation are given in Sec.B.

4.1. Datasets

PlanetZoo. We introduce here a curated dataset named PlanetZoo, which includes RGB synthetic videos of different animals with around 100 frames each. PlanetZoo covers a 180-degree visual field, captured by a moving camera to allow better evaluation of 3D reconstruction when imaging parameters must also be dynamically estimated due to the moving camera, and over a large visual field. In addition, following BADJA (Biggs et al., 2018), we also provide 2D key point annotations. More details about PlanetZoo, as well as DAVIS (Perazzi et al., 2016) and AMA-Human (Vlasic et al., 2008a) are provided in Sec.C.

4.2. Shape Reconstruction Comparison

Qualitative Comparisons. As depicted in Fig.2,9 (b) and 10, existing NeRF-based (Neural Radiance Fields) approaches, often require a substantial number of input videos, acquired from a wide range of viewpoints and precise camera poses. Therefore, when the input is a monocular short video, these methods struggle to produce plausible outcomes. We present the results of BANMo (Yang et al., 2022) in Fig.2. MagicPony (Wu et al., 2023b), RAC (Yang et al., 2023), and CAMM (Kuai et al., 2023) also have nearly the same results as BANMo because they can be seen to be an extension of BANMo utilizing a pre-trained RigNet (Xu et al., 2020) to generate skeleton or directly providing predefined skeleton. However, when the mesh quality is poor, the reconstruction results are also significantly compromised. Conversely, for monocular short videos, methods based on NMR (Neural Mesh Rendering) like LASR (Yang et al., 2021a) (Fig.2 (a)) and Hi-LASSIE (Yao et al., 2023) (Fig.15 (b)) demonstrate superior performance. Nevertheless, due to a lack of structural information about the object, LASR also falls short of achieving ideal results. Hi-Lassie only can provide coarse shape and motion due to the lack of temporal information and refinement operations. However, S3O, without using any additional annotations, outperforms both methods, as illustrated in Fig.2 (c) and Fig.5.

Quantitative Comparisons. Furthermore, for a comprehensive comparison, we examine the performance of S3O against the best existing methods, LASR and ViSER, in terms of 2D keypoint transfer accuracy, as shown in Tab.2. S3O consistently surpasses both LASR and ViSER across all animal subjects and requires approximately 40% less training time. Another point worth noting is the superior stability of S3O compared to the other two methods. We conduct multiple trials on DAVIS `camel`, `cows`, and Planet ZOO `giraffe`, `tiger`, noting standard deviations as a measure of the stability of the method. We observe that LASR and ViSER exhibit poor stability; for instance, in the `camel` experiments with LASR, results vary per trial, sometimes erroneously producing three or five legs. In contrast, S3O, by learning skeletal structural information, effectively avoids such inconsistencies, yielding reliable results.

Another important point to note is that when tested on the PlanetZoo dataset, LASR/BANMo tends to learn shapes that are reasonable only in a side view but extremely incorrect in other views. For example, as seen in Fig.2 and 9, the shapes of the giraffe, tiger, and elephant look acceptable from the side view, but from the top view, their shapes are completely incorrect. This issue stems from errors in learning the camera pose, leading to a situation where, despite each frame's predicted parameters aligning the rendered results with the ground truth mask, the incorrect camera pose results in all other parameters being wrong.

Also, we compared S3O with BANMo on the AMA-Human dataset. This dataset contains multi-view RGB videos by 8 synchronized cameras and provides ground truth camera parameters and 3D meshes. However, we use only the videos; we do not use the camera parameters and 3D meshes, since we estimate them as a part of our S3O. However, as shown in Tab.3, S3O still shows superior performance on all three metrics (Chamfer Distance, F-score, and mIoU) with much less training time.

4.3. Skeleton Reconstruction Comparison

We outline the essential features of an effective skeleton in 3D modeling. (1) Bone distribution should be logical and tailored to the object's movement complexity, with denser bones in areas of high skeletal motion like limbs. (2) The skeleton should be sufficiently detailed to depict the structure of the object while avoiding unnecessary complexities due to local irregularities.

S3O vs RigNet and Skeletor. We compare the skeletons generated by S3O and RigNet (Xu et al., 2020) and Skeletor (Au et al., 2008), as illustrated in Fig.4. [2] S3O achieves better generalization than RigNet without 3D supervision. Although RigNet leverages extensive 3D models, ground truth skeletons, and skinning weights as supervision, it fails on out-of-distribution (OOD) objects, e.g., for the cases of `tiger` and `elephant`. Further, S3O effectively utilizes motion information from videos to produce physically plausible skeletons. It allocates more bones to areas with denser motion and fewer bones to less dynamic regions. This contrasts with RigNet and Skeletor, which lack such motion information. For instance, Skeletor assigns an excessive number of redundant bones to the torso, which exhibits much less movement than legs. This discrepancy underscores the advanced capability of S3O in generating more physically meaningful skeletal structures, highlighting its practicality for dynamic object analysis in video data.

4.4. Diagnostics

Fine Granularity of the Skeleton. Our approach allows for the generation of skeletons with varying levels of granularity by adjusting the threshold for merging bone motion. As demonstrated in Fig.3, by employing thresholds of 0.99, 0.95, and 0.90, we progressively reduce the granularity of the skeleton predictions, while still preserving the most critical bones and structural integrity. It enables the creation of a skeleton with differing levels of detail.

Effectiveness of Dynamic Rigidity. As shown in Fig.6, incorporating Dynamic Rigidity into our reconstruction process yields a shape that adheres more closely to physical

---

[2] We apply pre-trained RigNet and Skeletor on the final mesh results generated by S3O to get their skeleton results.

reality compared to methods utilizing As-Rigid-As-Possible (ARAP) modeling and without both regularization terms. The resulting structure maintains geometric integrity and behaves in a manner consistent with real-world physics.

Bone Motion Estimation. We use optical flow warp to estimate bone motion, as illustrated in Fig.8. We experimented with using the estimated bone formation (SE(3)) for each bone as the basis for determining bone motion, similar to the method used in WIM (Noguchi et al., 2022). Intuitively, utilizing the SE(3) of each bone, which encompasses 3D information, should be more effective than relying on 2D optical flow, which lacks depth perception. However, our results were not as anticipated for two main reasons: (1) The SE(3) of bones are predicted, meaning the motion calculated for each part is only accurate when these predictions are highly stable and precise. This explains why the bone merge stage in WIM is more akin to a post-processing step, where accurate learning of the mesh and pose of each part is prioritized before the merge. In contrast, our method synchronously updates the skeleton and other learnable parameters, merging or generating bones at specific epochs. The model then needs to relearn the SE(3) and other parameters for these new bones. In such cases, the fluctuating SE(3) values may not be as reliable as the consistent 2D optical flow, which can be derived using existing models. (2) From a degenerate perspective, consider a scenario where the rotation of the bones occurs in a plane that projects as a line in the image plane. In this case, the motion directions of two adjoining bones (across a potential joint) align and become indistinguishable. However, this degenerate scenario also impedes the estimation of SE(3) from images, rendering both SE(3) and optical flow ineffective. Therefore, for arbitrary views, the direction indicated by optical flow is sufficient, and the use of SE(3) becomes unnecessary.

Physical Skeletons vs Virtual Bones. In this discussion, we explore why using a physical skeleton outperforms virtual bones for mesh pose and motion manipulation. As shown in Fig.2 (d), methods like LASR, ViSER, and BANMo use k-means clustering on meshes to determine virtual bone positions. This often results in larger areas like the torso getting more bones, while slender limbs have fewer. This allocation contradicts the body's motion dynamics, where more bones should be in frequently deforming areas like legs and fewer in more stable areas like the torso. Such skewed distribution leads to a rigid treatment of entire limbs, as seen in LASR, preventing natural bending.

5. Conclusions and Limitations

We have presented S3O, an approach to concurrently learn the shape, time-varying parameters, and skeleton of a dynamically articulated object from a single monocular video, without any 3D supervision. We propose the concept of

Dynamic Rigidity as a solution to the limitations inherent in existing As-Rigid-As-Possible (ARAP) formulations. Our experiments show that S3O exhibits enhanced generalization in the generation of skeletons, compared to RigNet which also uses 3D supervision. Further, the ability of S3O to leverage the 3D structure it captures helps it surpass current state-of-the-art methods in shape reconstruction performance, while significantly reducing training time. A **limitation** of most methods, including ours, is the excessive computation time required, rendering them impractical for everyday applications. Recognizing this, our future work will focus on utilizing large-scale pre-trained text/image-to-3D models as 3D priors to expedite the training process.

Impact Statement

The proposed Synergistic Shape and Skeleton Optimization (S3O) method significantly advances the field of 3D reconstruction from single monocular videos by addressing the limitations of current techniques that require extensive computational resources and additional human annotations. By adopting a dual-phase approach, S3O learns visible shapes and underlying skeletons without predefined models, camera poses, or key points. This innovation reduces training time by 60% compared to state-of-the-art methods and improves reconstruction accuracy and robustness from limited viewpoints. The introduction of the PlanetZoo dataset further demonstrates S3O's capability to handle dynamic camera movements and diverse objects, setting a new benchmark for 3D reconstruction from monocular videos.

Acknowledgement

References

Ahuja, N. and Chuang, J.-H. Shape representation using a generalized potential field model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):169–176, 1997.

Au, O. K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., and Lee, T.-Y. Skeleton extraction by mesh contraction. *ACM transactions on graphics (TOG)*, 27(3):1–10, 2008.

Badger, M., Wang, Y., Modh, A., Perkes, A., Kolotouros, N., Pfrommer, B. G., Schmidt, M. F., and Daniilidis, K. 3d bird reconstruction: a dataset, model, and shape recovery from a single view. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2020.

Baran, I. and Popović, J. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007a.

Baran, I. and Popović, J. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007b.

Biggs, B., Roddick, T., Fitzgibbon, A., and Cipolla, R. Creatures great and smal: Recovering the shape and motion of animals from video, 2018.

Biggs, B., Roddick, T., Fitzgibbon, A., and Cipolla, R. Creatures great and smal: Recovering the shape and motion of animals from video. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, pp. 3–19. Springer, 2019.

Biggs, B., Boyne, O., Charles, J., Fitzgibbon, A., and Cipolla, R. Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 195–211. Springer, 2020.

Bitter, I., Kaufman, A. E., and Sato, M. Penalized-distance volumetric skeleton algorithm. *IEEE Transactions on Visualization and computer Graphics*, 7(3):195–206, 2001.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Cornea, N. D., Silver, D., Yuan, X., and Balasubramanian, R. Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer*, 21:945–955, 2005.

Goel, S., Kanazawa, A., and Malik, J. Shape and viewpoint without keypoints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pp. 88–104. Springer, 2020.

Goel, S., Pavlakos, G., Rajasegaran, J., Kanazawa, A., and Malik, J. Humans in 4d: Reconstructing and tracking humans with transformers. *arXiv preprint arXiv:2305.20091*, 2023.

Hassouna, M. S. and Farag, A. A. Robust centerline extraction framework using level sets. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 458–465. IEEE, 2005.

Huang, J., Su, H., and Guibas, L. Robust watertight manifold surface generation method for shapenet models, 2018.

Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 371–386, 2018.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment anything. *arXiv:2304.02643*, 2023.

Kocabas, M., Athanasiou, N., and Black, M. J. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5253–5263, 2020.

Kuai, T., Karthikeyan, A., Kant, Y., Mirzaei, A., and Gilitschenski, I. Camm: Building category-agnostic and animatable 3d models from monocular videos, 2023.

Kulkarni, N., Gupta, A., Fouhey, D. F., and Tulsiani, S. Articulation-aware canonical surface mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 452–461, 2020.

Kumar, S. *Non-Rigid Structure from Motion*. PhD thesis, College of Engineering & Computer Science, The Australian National University, 2019.

Lee, D.-T. Medial axis transformation of a planar shape. *IEEE Transactions on pattern analysis and machine intelligence*, (4):363–369, 1982.

Li, X., Liu, S., De Mello, S., Kim, K., Wang, X., Yang, M.-H., and Kautz, J. Online adaptation for consistent mesh reconstruction in the wild. *Advances in Neural Information Processing Systems*, 33:15009–15019, 2020a.

Li, X., Liu, S., Kim, K., De Mello, S., Jampani, V., Yang, M.-H., and Kautz, J. Self-supervised single-view 3d reconstruction via semantic consistency. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 677–693. Springer, 2020b.

Lindell, D. B., Van Veen, D., Park, J. J., and Wetzstein, G. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16252–16262, 2022.

Liu, S., Li, T., Chen, W., and Li, H. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7708–7717, 2019.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., and Zhang, L. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866. 2023.

Luvizon, D. C., Tabia, H., and Picard, D. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, 2019.

Ma, C.-M., Wan, S.-Y., and Lee, J.-D. Three-dimensional topology preserving reduction on the 4-subfields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1594–1605, 2002.

Ma, W.-C., Wu, F.-C., and Ouhyoung, M. Skeleton extraction of 3d objects with radial basis functions. In *2003 Shape Modeling International.*, pp. 207–215. IEEE, 2003.

Magnenat, T., Laperrière, R., and Thalmann, D. Joint-dependent local deformations for hand animation and object grasping. Technical report, Canadian Inf. Process. Soc, 1988.

Noguchi, A., Iqbal, U., Tremblay, J., Harada, T., and Gallo, O. Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects, 2022.

Palágyi, K. and Kuba, A. A parallel 3d 12-subiteration thinning algorithm. *Graphical Models and Image Processing*, 61(4):199–221, 1999.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

Teed, Z. and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Torresani, L., Hertzmann, A., and Bregler, C. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892, 2008.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. Articulated mesh animation from multi-view silhouettes. In *Acm Siggraph 2008 papers*, pp. 1–9. 2008a.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. Articulated mesh animation from multi-view silhouettes. In *Acm Siggraph 2008 papers*, pp. 1–9. 2008b.

Wada, K. Labelme: Image Polygonal Annotation with Python. URL https://github.com/wkentaro/labelme.

Wade, L. *Automated generation of control skeletons for use in animation*. The Ohio State University, 2000.

Wang, R. Y., Pulli, K., and Popović, J. Real-time enveloping with rotational regression. In *ACM SIGGRAPH 2007 papers*, pp. 73–es. 2007.

Wang, Y.-S. and Lee, T.-Y. Curve-skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):926–936, 2008.

Weng, C.-Y., Curless, B., Srinivasan, P. P., Barron, J. T., and Kemelmacher-Shlizerman, I. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, pp. 16210–16220, 2022.

Wu, S., Li, R., Jakab, T., Rupprecht, C., and Vedaldi, A. Magicpony: Learning articulated 3d animals in the wild, 2023a.

Wu, S., Li, R., Jakab, T., Rupprecht, C., and Vedaldi, A. Magicpony: Learning articulated 3d animals in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8792–8802, 2023b.

Wu, Y., Chen, Z., Liu, S., Ren, Z., and Wang, S. Casa: Category-agnostic skeletal animal reconstruction. *Advances in Neural Information Processing Systems*, 35: 28559–28574, 2022.

Xiang, D., Joo, H., and Sheikh, Y. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10974, 2019.

Xu, Z., Zhou, Y., Kalogerakis, E., Landreth, C., and Singh, K. Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559*, 2020.

Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Chang, H., Ramanan, D., Freeman, W. T., and Liu, C. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15980–15989, 2021a.

Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Liu, C., and Ramanan, D. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. *Advances in Neural Information Processing Systems*, 34:19326–19338, 2021b.

Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., and Joo, H. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2863–2873, 2022.

Yang, G., Wang, C., Reddy, N. D., and Ramanan, D. Reconstructing animatable categories from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16995–17005, 2023.

Yao, C.-H., Hung, W.-C., Li, Y., Rubinstein, M., Yang, M.-H., and Jampani, V. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. *Advances in Neural Information Processing Systems*, 35: 15296–15308, 2022.

Yao, C.-H., Hung, W.-C., Li, Y., Rubinstein, M., Yang, M.-H., and Jampani, V. Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4853–4862, June 2023.

Yu, T., Guo, K., Xu, F., Dong, Y., Su, Z., Zhao, J., Li, J., Dai, Q., and Liu, Y. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 910–919, 2017.

Zhang, H., Li, F., Rawlekar, S., and Ahuja, N. Learning implicit representation for reconstructing articulated objects. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhang, H., Li, F., Qi, L., Yang, M.-H., and Ahuja, N. Csl: Class-agnostic structure-constrained learning for segmentation including the unseen. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7078–7086, 2024.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhou, Y. and Toga, A. W. Efficient skeletonization of volumetric objects. *IEEE Transactions on visualization and computer graphics*, 5(3):196–209, 1999.

Zuffi, S., Kanazawa, A., Jacobs, D. W., and Black, M. J. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6365–6373, 2017.

Zuffi, S., Kanazawa, A., Berger-Wolf, T., and Black, M. J. Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild". In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5359–5368, 2019.
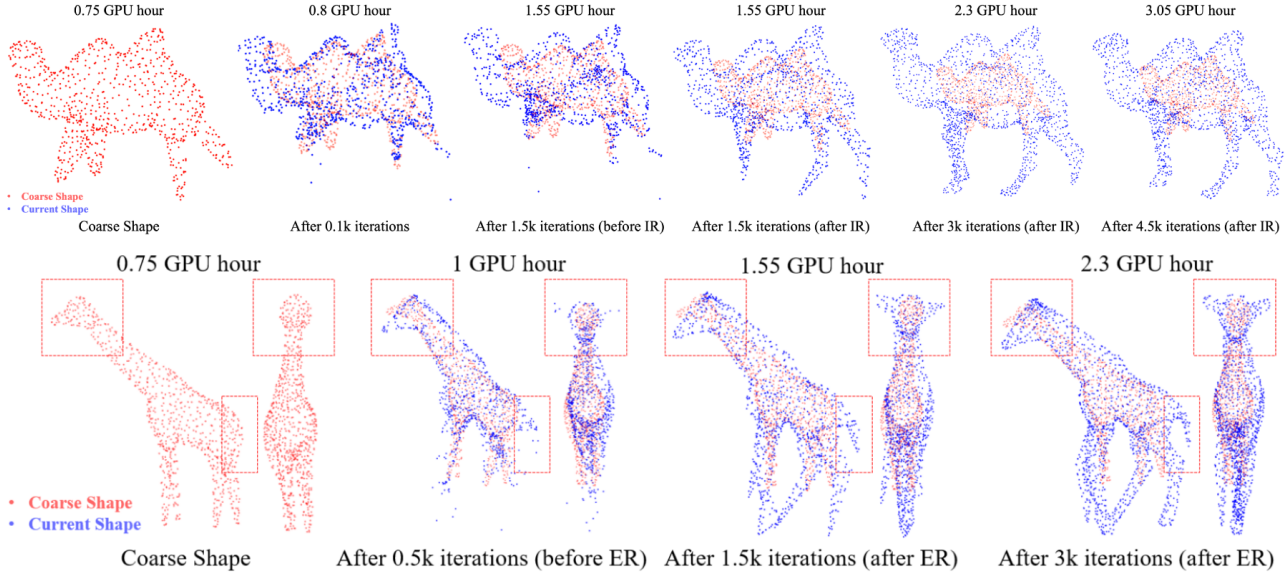
Figure 7. **Coarse-to-Fine Deformation.** The second phase of S3O focuses on intricacies, notably improving head and tail details that are less defined in the initial coarse learning stage. Points marked in red denote uniformly sized coarse shapes. ER represents Even Resampling, a process that makes vertices evenly distributed while maintaining the integrity of the original shape. Note the gradual improvement from coarse to fine stages as explained in Sec.E. Details of ER operation are presented in Sec.B.

In this section, we aim to provide additional information not included in the main text due to length constraints. This supplemental content includes **(1)** A comparison between S3O and existing methods (Sec.A) **(2)** Implementation details (Sec.B), **(3)** Information about the dataset used for evaluation (Sec.C), **(4)** an In-depth explanation of loss and regularization terms (Sec.D), **(5)** Further experimental results (human dataset in Tab.3, reconstruction from top view in Fig.9, reconstruction results of MagicPony and Hi-LASSIE in Fig.10 and 15, procedure of coarse shape phase in Fig.7) along with discussions (Sec.E), and **(6)** Table of Notations (Tab.4).

A. Difference of S3O with existing methods

In Tab.1 from the main paper, we compare the proposed S3O method with existing methods under various settings. S3O addresses more complex yet realistic scenarios, which include: 1) In real-world applications, ground truth (GT) 3D shapes, skeleton templates, and camera poses are often not readily available. 2) It's not always feasible to have multiple videos of moving objects showcasing different viewpoints.

Methods like CASA (Wu et al., 2022), SMAL (Zuffi et al., 2017), and others that rely on instance-specific 3D shapes, skeletons, or both as templates for motion modeling face limitations in generalizing to Out-Of-Distribution objects lacking GT 3D data. Similarly, approaches such as WIM (Noguchi et al., 2022), CAMM (Kuai et al., 2023), RAC (Yang et al., 2023), and BANMo (Yang et al., 2022) require accurate camera poses and extensive video data (exceeding 1000 frames) from varied viewpoints to produce satisfactory results. Monocular video from limited viewing directions lead to unsatisfying results, shown in Fig.2, 10, and 9.

Alternatively, while methods like LASR (Yang et al., 2021a) and VISER (Yang et al., 2021b) do not require shape and skeleton templates, the apparent surface motion does not match the apparent ground truth motion, due to errors in joint estimates of surface shape and motion, which is because they do not use any structural information (skeleton) about the object. In contrast, S3O also learns a physical skeleton to model the articulated motions of objects more effectively than the virtual bones used in LASR (Yang et al., 2021a), BANMo (Yang et al., 2022), and ViSER (Yang et al., 2021b). This allows our S3O to not require inputs of shape and skeleton templates, and camera pose, yielding a less demanding solution.

Algorithm 1 Dual-Phase Synergistic Shape and Skeleton Optimization (S3O)

Given a set of frames F from a video.

Select canonical frame f_c. (Sec.3.2)

• **Coarse Shape Phase**

Extract 2D skeleton from f_c (Sec.3.2)

Learn coarse parametric model including canonical 3D shape $\mathbf{M}_c^{(0)}$ and skeleton $\mathbf{S_T}^{(0)}$ from F (Sec.3.2)

• **Motion & Fine Shape Phase**

Initialize the camera & motion parameters $\theta^{(0)}$

for $e = 1$ **to** E **do**

  if $e < e_1$ **then**

    Articulation $\mathbf{M}_t^{(e)} \leftarrow \mathbf{M}_c^{(e)}$ (Sec.3.3)

    Compute the reconstruction loss.

    Update canonical shape and time-varying parameter: $\theta^{(e+1)}, \mathbf{M}_c^{(e+1)} \leftarrow \theta^{(e)}, \mathbf{M}_c^{(e)}$

    Adapt skeleton $\mathbf{S_c}^{(e)}$ to $\mathbf{M}_c^{(e+1)}$ (Sec.3.4)

  else if $e = e_1$ **then**

    Upsample Skeleton

  else

    Calculate physical constraints (Sec.3.4)

    Update Skeleton: $\mathbf{S_c}^{(e+1)} \leftarrow \mathbf{S_c}^{(e)}$
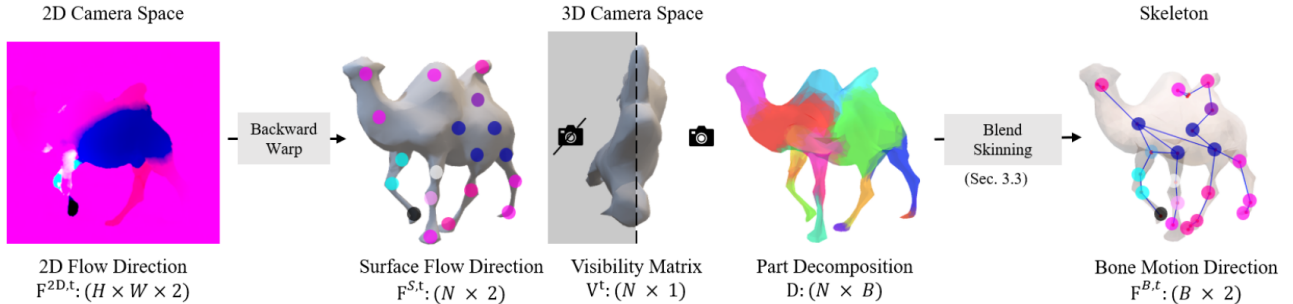
  end if

end for

Figure 8. **Optical Flow Warp.** We estimate the bone motion from the 2D optical flow.

B. Implementation Details

S3O follows Neural Mesh Rasterization (NMR) approaches that jointly learn the (1) parametric model, including the mesh and skeleton, which is used in calculating the skinning weights and changing the pose of the object via blending skinning operation, and (2) time-varying parameters, which include the bone transformation, and camera parameter.

Coarse Shape Phase. While different from existing shape-template-free methods such as LASR (Yang et al., 2021a), MagicPony (Wu et al., 2023a), and BANMo (Yang et al., 2022), which initializes the mesh with a sphere or an ellipsoid, we use the extraction of 2D skeletons from the canonical frame and DINO feature information to obtain an instance-specified initial mesh, similar to the work Hi-LASSIE (Yao et al., 2023), as shown in the Fig.14. Then, through the coarse shape phase (Sec.3.2 in the main paper), we can quickly learn a coarse shape. Although the quality of the mesh is still far from satisfactory, this coarse mesh can greatly assist us in learning accurate camera parameters more quickly (results are shown in Fig.14).

Analysis-by-Synthesis Strategy. Our approach adopts an analysis-by-synthesis strategy. In this strategy, we first obtain the object silhouette, optical flow, and pixel values by rendering (Liu et al., 2019) the deformed mesh to image space, defined as the synthesis process. Next, we analyze the predicted object silhouette, optical flow, and pixel values with respect to the video observations.
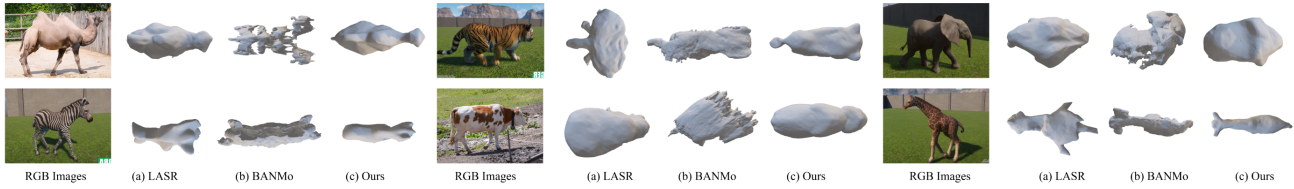
Figure 9. **Reconstructed Mesh Results (top view).** For (a) LASR, (b) BANMo, and (c) S30 (ours), for six animal videos: DAVIS's `camel`, `cows` and PlanetZoo's `zebra`, `elephant`, `tiger`, and `giraffe`.
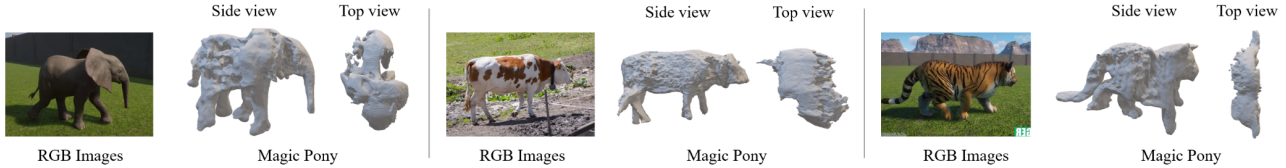

Figure 10. **Reconstructed Mesh Results (side and top views).** For MagicPony (Wu et al., 2023a), for three animal videos: DAVIS's `cows` and PlanetZoo's `elephant`, `tiger`.

Even Resampling. During the training process, as shown in Fig.7, only the outermost points on the camel's legs initially move downward, thereby elongating the entire leg to achieve a shape more akin to a real camel in the video. However, in this scenario, the distribution of vertices is highly uneven, with the lower half of the leg having very few points. Therefore, every 1,500 iterations, we perform an Evenly Resampling (2-Manifold Processing) (Huang et al., 2018), which makes the vertices roughly uniformly distributed on the geometry surface.

Mask and Optical Flow Generation. We generate object-level segmentation masks for input videos using CSL (Zhang et al., 2024), Grounding DINO (Liu et al., 2023) and Segment Anything Model (SAM) (Kirillov et al., 2023). By inputting text about the target object in the video, we leverage Grounding DINO to track that object across frames and then we input the detected bounding boxes to SAM to obtain the segmentation masks. Results are shown in Fig.12. For part-level segmentation, which is used during the coarse phase, we leverage DINO (Caron et al., 2021) to extract per-pixel embeddings and then cluster the embeddings to generate part-level masks similar to (Yao et al., 2023). The results are shown in Fig.13. Optical flow was estimated using existing flow estimators (Teed & Deng, 2020).

Experimental Setup. For our experiments on DAVIS and PlanetZoo, we utilized a single A100 40GB GPU. The batch size was set to 8, and the training consisted of 10 epochs compared to LASR and ViSER having 25 epochs. Each epoch comprised 1500 iterations. Similar to LASR, for the first 40% epochs, we estimate 16 cameras and keep the best one in the following epochs. For our experiment on AMA-Human, we follow the same training setting as the original BANMo, except we only use one single GPU and S3O requires 40% training epochs to achieve better reconstruction results compared to BANMo.

C. Datasets and Metrics

C.1. Datasets

DAVIS & BADJA. BADJA (Biggs et al., 2019) dataset is a derivative of DAVIS (Perazzi et al., 2016) which contains the RGB videos and the corresponding ground truth segmentation results. It contains typical quadrupeds like bears, camels, dogs, and horses. Besides, BADJA also provides 2D key points for all videos. In experiments, we show both numerical and visualization results on DAVIS & BADJA.

PlanetZoo. We obtain 2D key points for the ground truth using Labelme tools (Wada). Experiments on the PlanetZoo dataset containing quantitative and qualitative results are also shown in the paper. For annotation rules, we also adopt the same real physical morphology across frames in each video as DAVIS & BADJA. In detail, these points are located on the

Figure 11. Keypoints Annotations for PlanetZoo dataset.

Figure 12. Example frames and masks of `tiger` from PlanetZoo dataset.

paws, knees, ears, tails, jaws, and noses as shown in Fig.11. We plan to release the whole PlanetZoo dataset including the RGB videos, 2D ground truth segmentation, and 2D key points annotations in the future.

AMA human. (Vlasic et al., 2008a) records multi-view videos by 8 synchronized cameras for which it provides ground truth camera pose, mesh, and masks. To compute the Chamfer distance we use to evaluate our 3D reconstruction, We need to use only the 3D mesh.

C.2. Metrics

To evaluate the performance of S3O and the existing methods, we use 2D Keypoint Transfer Accuracy same as LASR (Yang et al., 2021a) and ViSER (Yang et al., 2021b). 2D Keypoint Transfer Accuracy evaluates results by the percentage of correct key point transfer (PCK-T) (Kanazawa et al., 2018; Kulkarni et al., 2020). If the distance between the transferred point and the corresponding ground truth is one is within a predefined threshold, this transferred point will be categorized as 'Correct'. This threshold d_{th} is defined by $0.2\sqrt{|S|}$, and $|S|$ means the area of ground truth mask (Biggs et al., 2019). We use 3D Chamfer distance, mIoU, and F-score on the AMA-Human dataset.

D. Losses and Regularization

The model is learned by minimizing the reconstruction loss, which includes seven terms. The first five terms are similar to those in existing differentiable rendering pipelines (Yang et al., 2021a; Yao et al., 2023; Yang et al., 2022; Goel et al., 2020). We define $\mathbf{S^t}, \mathbf{I^t}, \mathbf{F}^{2D,t}$ as the silhouette, input image, and optical flow of the input image, and their corresponding rendered counterparts as $\tilde{\mathbf{S}}^{\mathbf{t}}, \tilde{\mathbf{I}}^{\mathbf{t}}, \tilde{\mathbf{F}}^{2D,t}$. The following losses ensure the fitting between rendered and original:

$$\mathcal{L}_{\text{silhouette}} = \sum_{\mathbf{x}^t} \left\| \mathbf{s}(\mathbf{x}^t) - \hat{\mathbf{s}}(\mathbf{x}^t) \right\|_2, \qquad (10)$$
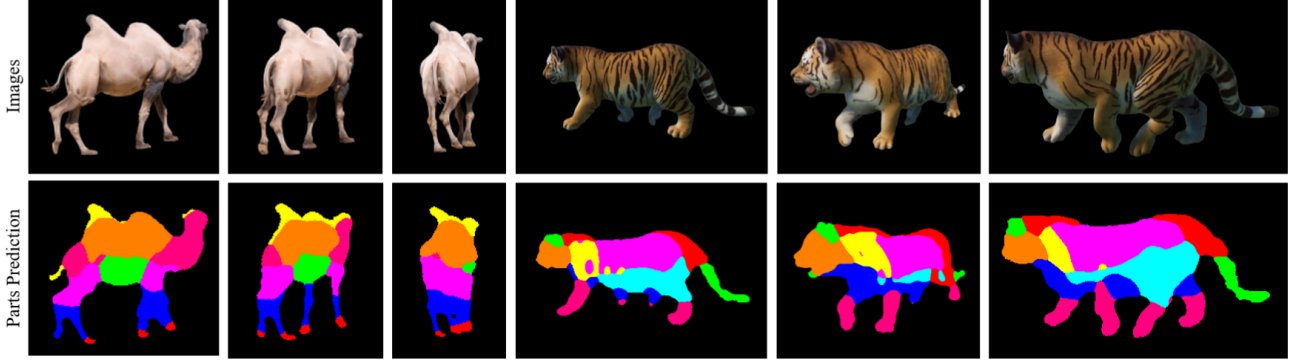
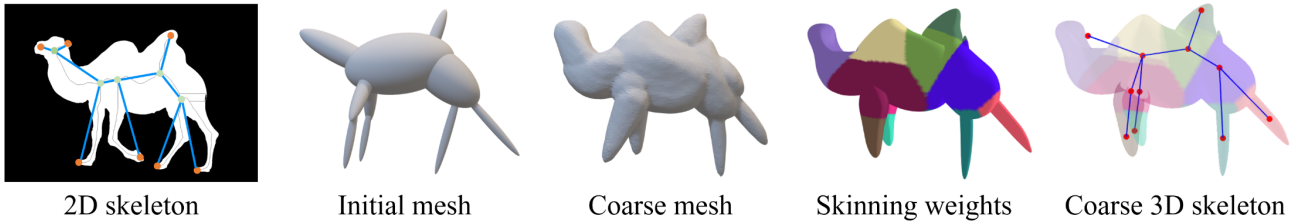Figure 13. **Part results** obtained from DINO features.

| 2D skeleton | Initial mesh | Coarse mesh | Skinning weights | Coarse 3D skeleton |

Figure 14. **Procedure of coarse shape learning phase.**

$$\mathcal{L}_{\text{optical flow}} = \sigma \left\| (\tilde{\mathbf{F}}^{\mathbf{2D,t}}) - (\mathbf{F}^{\text{2D,t}}) \right\|_2^2, \tag{11}$$

$$\mathcal{L}_{\text{texture}} = \left\| \tilde{\mathbf{I}}^{\mathbf{t}} - \mathbf{I}^{\mathbf{t}} \right\|_1, \tag{12}$$

$$\mathcal{L}_{\text{perceptual}} = \text{pdist}(\tilde{\mathbf{I}}^{\mathbf{t}}, \mathbf{I}^t), \tag{13}$$

$$\mathcal{L}_{\text{dino feat}} = \mathcal{L}_{\text{part silhouette}} + \mathcal{L}_{\text{consistency}}, \tag{14}$$

where pdist(,) is the perceptual distance (Zhang et al., 2018), and $\mathcal{L}_{\text{dino feat}}$ leverages the dino features, which combines the part silhouettes and consistency loss that maintains consistency between 2D pixels on a part and corresponding 3D points on the part surface.

$$\mathcal{L}_{\text{part silhouette}} = \sum_p \left\| s_p^t - \hat{s}_p^t \right\|_2, \tag{15}$$

where s_p^t represents part silhouettes for part p and frame t. The consistency loss, denoted as $\mathcal{L}_{\text{consistency}}$, is designed to optimize the 3D surface coordinates. This optimization ensures that the aggregated 3D point features project more accurately onto corresponding pixel features within the image ensemble, following the methodology outlined in (Yao et al., 2023). In our experiments, we observed that the dino feature loss aids the model in more rapidly learning a coarse shape through part-wise optimization. However, due to the limited accuracy of part-level annotations, as shown in Fig.13, it is not guaranteed to maintain excellent consistency across all frames. Consequently, the dino feature loss restricts the model's capacity for more detailed deformation. Therefore, in the motion and fine shape phase, we removed the dino feature loss.

Also, we leverage three **regularization terms:** (1) Dynamic Rigidity term, which is introduced in the main paper Sec.3.3. (2) Symmetry term, which encourages the canonical mesh to be symmetric.

$$\mathcal{L}_{\text{symm}} = \sum_i \min_j \left\| v_j - \phi\left(v_i^{symm}\right) \right\|^2, \tag{16}$$

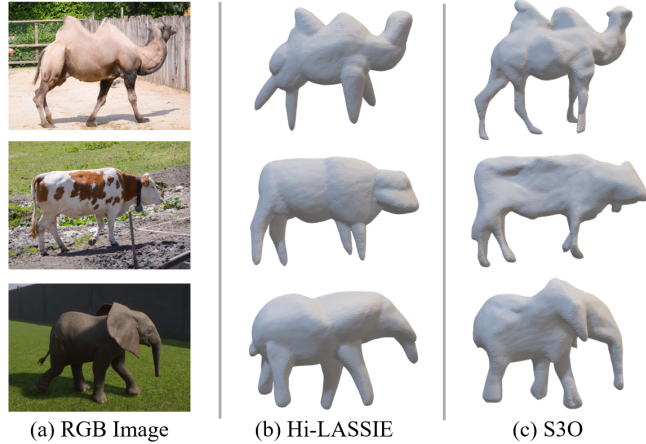| | (a) RGB Image | (b) Hi-LASSIE | (c) S3O |

Figure 15. **Mesh Results** of S3O compared with with Hi-LASSIE.

Table 3. Quantitative results on AMA's `swing` and `samba`. 3D Chamfer Distance, F-score, and mIoU are shown averaged over all frames. Note we use one A100 GPU with half batch size compared with the original BANMo (Yang et al., 2022), which leads to lower accuracy. For the AMA-Human dataset, we use more than one video for training, which is different from other experiments. (Time: GPU hours on A100)

method	AMA-samba				AMA-swing				Ave.			
	CD	F@2%	mIoU	Time	CD	F@2%	mIoU	Time	CD	F@2%	mIoU	Time
BANMo	15.3	53.1	61.2	12.5	13.8	54.8	62.4	12.7	14.6	53.9	61.8	12.6
S3O	12.9	55.2	62.5	5.1	12.3	56.4	63.6	4.8	12.9	55.8	62.5	4.9

where, v_j is the vertex j in the one side of the symmetry plane, and $v_i{}^{symm}$ is the vertex from the other side. ϕ is the reflection operation w.r.t to the symmetry plane. (3) Laplacian smoothing: We apply laplacian smoothing to generate smooth mesh surfaces.

$$\mathcal{L}_{\text{shape}} = \left\| \mathbf{X}_i^0 - \frac{1}{|N_i|} \sum_{j \in N_i} \mathbf{X}_j^0 \right\|^2,$$

(17)

where, $\mathbf{X}_i^0$ is coordinates of vertex i in canonical space. And N is the number of vertices

E. Discussion

Coarse-to-Fine Deformation. As illustrated in Fig.7, S3O starts with learning a coarse shape in the first phase, which takes approximately 15 minutes. This is followed by a warm-up stage at the onset of the second phase, where the mesh is kept constant while only the time-varying motion parameters are learned (30 minutes). Subsequently, training is conducted for all parameters. The coarse mesh initially struggles to accurately reconstruct finer extremities, such as the legs and head. These structures gradually emerge during the second phase. Points marked in red denote coarse shapes. ER denotes Evenly Resampling, a process that makes vertices evenly distributed while maintaining the integrity of the original shape.

Comparison between NeRF-based and NMR-based Methods. Methods based on Neural Radiance Fields (NeRF), such as BANMo and MagicPony, have demonstrated remarkable capabilities in reconstruction tasks. These methods involve learning a neural radiance field and subsequently extracting a mesh from this learned field. However, the process of learning a neural radiance field is data-intensive, often requiring multiple videos and viewpoints for optimal results. This becomes a limitation, as obtaining such extensive data can be challenging. The performance of NeRF-based methods significantly diminishes with restricted data and viewpoints, as evidenced in Fig.2, 10, and 9. In contrast, methods utilizing the Neural Mesh Renderer (NMR) focus on directly training models to produce a mesh. These approaches have been shown to yield satisfactory outcomes even with limited data availability, such as using a single monocular video with less than a hundred input frames. Our approach is based on NMR and demonstrates considerable improvements, both qualitatively and

quantitatively, compared with existing NMR-based methods.

Table 4. Table of Notations

Symbol	Description	Dimension
B	Number of Bones	—
E	Number of Edges in Surface Mesh	—
J	Number of Joints	—
Representation Parameters		
$\mathbf{M}$	Canonical Surface Mesh and Color	—
$\mathbf{P}_c^t$	Camera Parameters	—
$\mathbf{W}$	Skinning Weights	$\mathbf{W} \in \mathbb{R}^{N \times B}$
$\mathbf{R}$	Rigidity Coefficient	$\mathbf{R} \in \mathbb{R}^E$
$\mathbf{T}^t$	Time Varying Transformation	$\mathbf{T} \in SE(3)$
Skeleton Components		
$\mathbf{S_T}$	Skeleton	—
$\mathbf{B}$	Bones	$\mathbf{B} \in \mathbb{R}^{B \times 13}$
$\mathbf{J}$	Joints	$\mathbf{J} \in \mathbb{R}^{J \times 5}$
Bone Components		
$\mathbf{C}/\mathbf{B}[:,:3]$	Gaussian Centers Coordinates	$\mathbf{C} \in \mathbb{R}^{B \times 3}$
$\mathbf{Q}/\mathbf{B}[:,3:12]$	Precision Matrix	$\mathbf{Q} \in \mathbb{R}^{B \times 9}$
$\mathbf{L}/\mathbf{B}[:,12]$	Bone Length	$\mathbf{L} \in \mathbb{R}^B$
Joint Components		
$\mathbf{J}[:,:2]$	Index of Two connected Bones	$\mathbb{R}^{J \times 2}$
$\mathbf{J}[:,2:]$	Joint Coordinates	$\mathbb{R}^{J \times 3}$
Optical Flow Notations (at time t)		
$\mathbf{F}^{2\mathrm{D},t}$	2D Optical Flow	$\mathbf{F}^{2\mathrm{D},t} \in \mathbb{R}^{H \times W}$
$\mathbf{F}^{B,t}$	2D Bone Motion Direction	$\mathbf{F}^{B,t} \in \mathbb{R}^{B \times 2}$
$\mathbf{F}^{S,t}$	Surface Flow Direction	$\mathbf{F}^{S,t} \in \mathbb{R}^{S \times 2}$
$\mathcal{V}^t$	Visibility Matrix	$\mathcal{V}^t \in \mathbb{R}^{N \times 1}$
2D Notations		
$\mathbf{S}/\tilde{\mathbf{S}}$	Input silhouette/ Observed silhouette	$\mathbf{S}/\tilde{\mathbf{S}} \in \mathbb{R}^{H \times W}$
$\mathbf{I}/\tilde{\mathbf{I}}$	Input Image / Observed Image	$\mathbf{I}/\tilde{\mathbf{I}} \in \mathbb{R}^{H \times W}$
$\mathbf{F}^{2\mathrm{D}}/\tilde{\mathbf{F}}^{2\mathrm{D}}$	2D Input OF / 2D observed OF	$\mathbf{F}^{2\mathrm{D}}/\tilde{\mathbf{F}}^{2\mathrm{D}} \in \mathbb{R}^{H \times W}$