PPR: Physically Plausible Reconstruction from Monocular Videos

Gengshan Yang Shuo Yang John Z. Zhang Zachary Manchester Deva Ramanan
Carnegie Mellon University

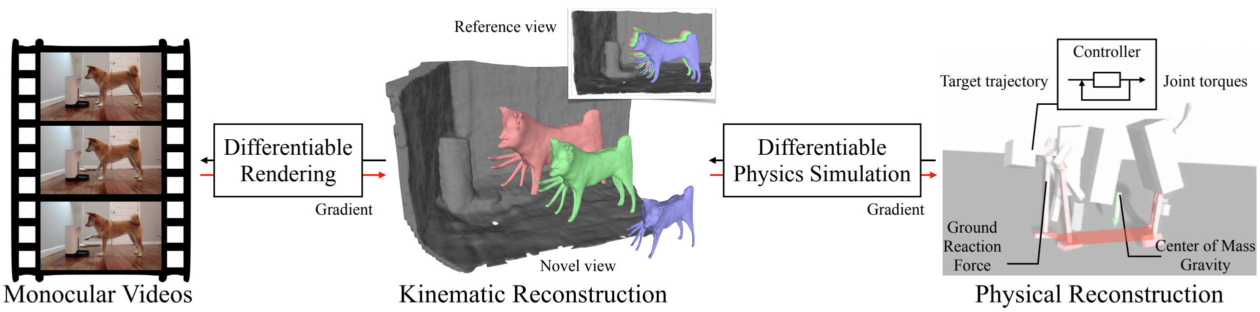


Figure 1: Given casually-captured monocular videos (left), PPR builds 3D models of articulated objects and the surrounding environment. Naive kinematic reconstruction (middle) generates a family of solutions, some containing inconsistent physical support and contact dynamics (blue and green color), such as floating or walking with sliding feet. We show that differentiable physics simulation acts as an effective regularizer for improving the physical plausibility of visual reconstruction algorithms. As PPR reconstructs the dynamics scene, it also drives a ragdoll in a physics simulator to track the kinematic reconstruction. This ensures the reconstructions are statically stable with ground contact (right), and the center of mass is projected within the support polygon (marked with red). PPR also reports physics estimations, such as ground reaction forces (red arrows) and center of mass (green arrow).

Abstract

Given monocular videos, we build 3D models of articulated objects and environments whose 3D configurations satisfy dynamics and contact constraints. At its core, our method leverages differentiable physics simulation to aid visual reconstructions. We couple differentiable physics simulation with differentiable rendering via coordinate descent, which enables end-to-end optimization of, not only 3D reconstructions, but also physical system parameters from videos. We demonstrate the effectiveness of physics-informed reconstruction on monocular videos of quadruped animals and humans. It reduces reconstruction artifacts (e.g., scale ambiguity, unbalanced poses, and foot swapping) that are challenging to address by visual cues alone, and produces better foot contact estimation.

1. Introduction

Given casually-captured monocular RGB videos, we aim to build 3D models of articulated objects and the environment, whose configurations (geometry, motion trajectory, force, and mass distribution) satisfy physics constraints, and can be replayed in a physics simulator.

Reconstructing dynamic 3D structures from monocular videos is challenging due to the under-constrained nature of the problem. Prior works often leverage *first order* constraints. For instance, Nonrigid-SfM explores temporal smoothness and low-rank priors [4] to constrain the problem. Recent works on differentiable rendering and dynamic NeRF utilize divergence-free motion fields [42] or as-rigid-as-possible priors [20]. Although those methods are able to obtain visually appealing reconstruction results from the reference viewpoint, physically-implausible configurations, such as foot sliding, statically-unstable poses, etc., are often observed from a novel viewpoint. An illustrative example is shown in Fig. 2.

Physics as a prior. We seek a more principled way to model the time-varying behavior of an object and its interaction with the environment using physics constraints. Physical priors tend to be relatively unexplored as a tool for aiding reconstruction, though important exceptions exist in the domain of monocular human motion capture [8, 50, 53]. One reason that such methods are not more widespread is that they often make strong assumptions about the target and the scene, for instance, accurate 2D/3D keypoint tracking, known ground plane, and contact state annotations. Moreover, operationalizing such constraints requires the use of

(a) Relative scale (b) 3D tracking with occlusion (c) Ground contact

Figure 2: **Physics helps monocular reconstruction.** Naive reconstructions of dynamic scenes from monocular videos are often physically implausible. (**a**) Although the 2D projections of both the red and blue reconstructions align with the input frame (top), their scales and 3D motions are wildly different due to the inherent ambiguity between the camera and object motion. Solution 1 (red) correctly touches the ground while solution 2 (blue) appears smaller than its true size and floats in the air. (**b**) Tracking is challenging when the feet of the cat are occluded during walking (top). PPR tracks dense surface points leveraging rigid body dynamics constraints. As a result, the left-rear feet and tail of the cat are correctly tracked in world coordinates despite undergoing heavy self-occlusion (bottom). The contact constraints also effectively reduce infeasible motion such as foot skating. (**c**) The kinematic solution does not often produce foot trajectories in contact with the ground (black horizontal line). Applying ground-fitting (to satisfy joints being above the ground) still produces a human that is floating and titled. PPR jointly optimizes scale, reconstructions, and physics to produce an upright human in contact with the ground.

heavyweight simulators that may be difficult to integrate with visual reconstruction algorithms.

In this work, we couple differentiable rendering with differentiable physics-based simulation to jointly solve for the object geometry, motion, background scene, and physics parameters including body mass distributions and control parameters. We posit that just as differentiable renderers have lowered the barrier of entry for (neural) 3D modeling, differentiable simulators are also lowering the barrier for incorporating physical constraints. Compared to prior approaches that rely on strong human priors to estimate 3D pose and ground contact, PPR works for more unconstrained settings including both humans and animals in an unknown environment, enabled by end-to-end optimization from videos to physics.

Specifically, we (1) introduce an end-to-end framework for reconstructing physically-plausible dynamic objects and scene configurations from monocular videos; (2) propose an alternating-direction 3D-reconstruction formulation by coupling differentiable physics simulation and differentiable rendering; (3) demonstrate improved reconstruction quality on examples including humans and animal videos. To our knowledge, PPR is the first attempt at a generalized, end-to-end framework for *jointly* optimizing dynamic 3D reconstructions and physical systems from monocular videos.

2. Related Work

Parametric Body Models. A large body of work in human and animal reconstruction uses parametric models [31, 43, 61, 66, 82, 83], which are built from registered 3D scans of human or animals, and serve to recover 3D shapes given a single image or video at test time [1, 2, 22, 81]. Although parametric body models achieve great success in reconstructing humans with large amounts of ground-truth 3D data, it is challenging to apply the same methodology to categories with limited 3D data, such as animals.

Nonrigid Reconstruction from Imagery. Non-rigid structure from motion (NRSfM) methods [3, 12, 24, 25, 54] reconstruct non-rigid 3D shapes from 2D point trajectories in a class-agnostic way. However, due to difficulties in estimating long-range correspondences [51, 57], they do not work well for videos in the wild. Recent works apply differentiable rendering to reconstruct articulated objects from videos [46, 65, 71, 72, 73] or images [11, 20, 23, 27, 28, 65, 76]. However, their recovered 3D configurations are often physically implausible due to the ill-posed nature of the monocular reconstruction problem.

Physics-informed 3D Reconstruction. Prior works have shown that the physical realism of 3D human reconstruction can be improved by differentiable physics simulation [8] and soft physics constraints [50, 53, 61]. However, their methods typically require pre-computed 3D human poses and a template body shape, mass, and skeletons [31, 67], which makes it difficult to generalize to non-human categories. Most of them also require ground plane and foot contact annotations [52, 53]. Recently, DiffPhy [8] optimizes control parameters that generate the motion through a physics simulator that removes the dependency on contact annotations; however, it still relies on the 3D human pose and assumes a fixed camera frame. Beyond the human category, some recent works [19, 36] explore more generic physic priors to regularize shape and cloth deformation.

Differentiable Simulation with Contact. Differentiable contact reasoning in graphics and robotics has seen great advancement in recent years [14, 16, 34, 64, 80]. A crucial challenge for contact simulation and gradient computation arises from its non-smoothness nature. Some methods solve a set of complementarity problems governing contact forces via optimization and derive the gradients [14, 15, 48, 55, 64]. An alternative approach is to soften contact forces by allowing inter-penetration that produces elastic forces to push collided objects away [7, 34]. We leverage the soft contact model that can be easily parallelized on GPUs, and couple differentiable contact physics with differentiable rendering to jointly reason about 3D reconstruction and physics from videos.

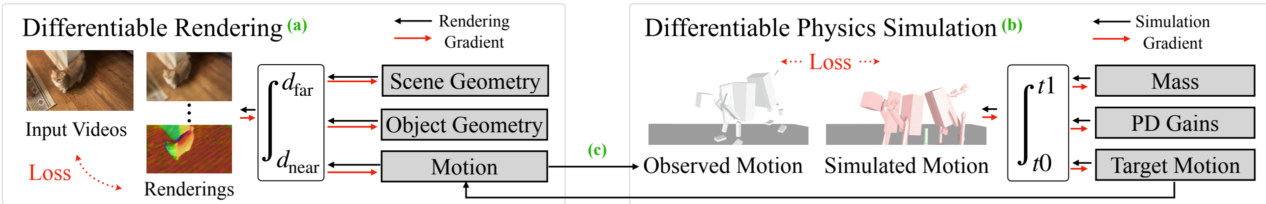


Figure 3: **Method overview.** Given monocular videos of an articulated object, PPR solves for a physically plausible representation of the object and the surrounding environment. **(a)** It leverages differentiable volume rendering to optimize the object and the background geometry (Sec. 3.1), as well as the object and camera motion parameters (Sec. 3.2). **(b)** It leverages a differentiable physics simulator to recover the underlying parameters of an articulated body system, including mass distribution, parameters of a controller, and the target motion tracked by the controller (Sec. 3.3). **(c)** We alternate between the differentiable rendering optimization and differentiable physics optimization, such that the reconstructions are consistent with visual observations while satisfying physics constraints (Sec. 3.4).

3. Approach

Given casually-taken videos of an articulated object, we apply differentiable rendering (DR) to solve for a kinematic reconstruction of the object and the surrounding environment that faithfully explains the input videos (Sec. 3.1 and Sec. 3.2). Meanwhile, we perform differentiable physics (DP) optimization to constrain the kinematic estimation to be physically plausible, such that it can be replayed in a simulator (Sec. 3.3). We alternate between DR and DP using coordinate descent until the optimization converges (Sec. 3.4). An overview is shown in Fig 3.

3.1. Object and Scene Model

To enable physics-based modeling of the interactions between the object and the environment from videos, we build a dense 3D representation of their kinematic states. We model the scene $\mathbf{T}$ as the composition of a dynamic object and a rigid background, each of which is represented as neural fields defined in their respective canonical space.

Object Fields $\mathbf{T}^o$. Our canonical model for the object is similar to BANMo [73]. A 3D point $\mathbf{X} \in \mathbb{R}^3$ is associated with three properties: signed distance $d \in \mathbb{R}$, color $\mathbf{c} \in \mathbb{R}^3$, and canonical features $\boldsymbol{\psi} \in \mathbb{R}^{16}$, which are used to register pixel observations to the canonical space. These properties are predicted by multi-layer perceptron (MLP) networks:

$$(d, \mathbf{c}^t) = \mathbf{MLP}_{\sigma}(\mathbf{X}, \boldsymbol{\omega}_a), \quad (1)$$

$$\boldsymbol{\psi} = \mathbf{MLP}_{\boldsymbol{\psi}}(\mathbf{X}). \quad (2)$$

Besides the 3D point locations, we further condition the color on an appearance code $\boldsymbol{\omega}_a \in \mathbb{R}^{64}$ that captures frame-specific appearance such as shadows and illumination changes [35]. To capture articulations and soft deformations, we use a deformable variant of NeRF [73]. For a given time t, it defines a forward warping field $\mathcal{W}^{t,\rightarrow}$ that transforms 3D points from the canonical space to the specified time instance, and a backward warping field $\mathcal{W}^{t,\leftarrow}$ to transform points in the inverse direction. The warping fields are further explained in Sec. 3.2.

Scene Fields $\mathbf{T}^s$. We leverage VolSDF [74] to build a high-quality background reconstruction. One crucial modification is that we supervise the background fields with not only RGB, but also optical flow and surface normal estimations. Optical flow supervision acts similarly to direct bundle adjustment in DroidSLAM [60], which effectively optimizes camera parameters and scene geometries in challenging conditions (e.g., low-texture, large camera motion). Surface normal supervision [62, 77] provides a signal to regularize geometry and reconstruct the background when there is little to no motion parallax. Those supervisions are extracted from pre-trained optical flow [59, 70] and surface normal [6] predictors.

Composite Rendering. To render images, we compose the density and color of the object and the scene fields in the view space [41], and compute the expected color, optical flow, and surface normal maps. During optimization, we minimize the difference between the rendered and observed color, flow, and surface normal values.

3.2. Motion Representation

The warping function $\mathcal{W}$ models object motion is modeled at three levels: root body movements $\mathbf{G}_o$, skeleton articulations $\mathbf{A} = \{\mathbf{J}, \mathbf{W}, \mathbf{Q}\}$ and soft deformations $\mathbf{S}$.

Global Movement $\{\mathbf{G}_\mathbf{b}, \mathbf{G}_o\}$. We model the background-to-camera transformations $\mathbf{G}_b$ and object root-to-camera transformations $\mathbf{G}_o$ as per-frame SE(3) represented as Fourier-based MLP networks.

Skeleton Articulation $\{\mathbf{J}, \mathbf{Q}, \mathbf{W}\}$. The coarse-level motion of the object is controlled by an articulated skeleton model. The skeleton has bones connected by 3-DoF spherical joints, specifically a tree topology with joint locations $\mathbf{J} \in \mathbb{R}^{3\times(B-1)}$ and Gaussian bones of size $\mathbf{L} \in \mathbb{R}^{9\times B}$. We set B=26 for quadrupeds and B=19 for humans. The skeleton topology is fixed through optimization but $\mathbf{J}$ and $\mathbf{L}$ are specialized to input videos. To model time-varying skeletal movements, we define per-frame joint angles:

$$\mathbf{Q} = \mathbf{MLP_A}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times(B-1)}, \tag{3}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{16}$ is a low-dimensional articulation code. Given joint angles and the per-video joint locations, we compute bone transformations $\mathbf{G} \in \mathbb{R}^{3\times4\times B}$ in the object root coordinate via forward kinematics [39].

To drive the space deformation with bone articulations, we define the skinning weights as the membership probability of bones assigned to 3D points $\mathbf{X}$ [56, 71],

$$\mathbf{W} = \sigma_{\text{softmax}}\big(d_\sigma(\mathbf{X},\boldsymbol{\theta}) + \mathbf{MLP_W}(\mathbf{X},\boldsymbol{\theta})\big) \in \mathbb{R}^B, \tag{4}$$

where θ is a pose code and $d_\sigma(\mathbf{X},\boldsymbol{\theta})$ is the Mahalanobis distance between $\mathbf{X}$ and Gaussian bones under pose θ, refined by a delta skinning MLP [73]. Each bone has three parameters for center, orientation, and scale respectively. The orientations are determined by the parent joints, and the centers as well as the scales are optimized. Then the motion of a spatial point is driven by blending skinning,

$$\mathbf{X}(\theta) = \left(\sum_{b=1}^{B} \mathbf{W}_b \mathbf{G}_b\right)\mathbf{X}. \tag{5}$$

Soft Deformation S. To account for the deformation caused by non-skeletal movements (such as the clothes of humans), we add a neural deformation field [26, 42] $\mathbf{S}(\cdot)$ that is capable of representing highly nonrigid deformations. We use real-NVP [5] to produce 3D deformation fields that are invertible by construction. The soft deformation is applied to the canonical space similar to Human-NeRF [63],

$$\mathbf{X}(\omega_d) = \mathbf{S}(\mathbf{X},\omega_d), \tag{6}$$

where ω_d is a per-frame soft deformation code. Compared to applying $\mathbf{S}$ to the articulated space, we found this formulation to be easier to optimize since it operates on the fixed canonical space.

Invertibility of Warping Fields. To summarize, both the forward and backward warping fields $\{\mathcal{W}^{t,\rightarrow}, \mathcal{W}^{t,\leftarrow}\}$ include an articulation operation in Eq. (3) and a deformation operation in Eq. (6). Notably, we only need to define each operation in the forward direction. The deformation operation is invertible by construction. To invert the articulations, we invert the SE(3) transformations $\mathbf{G}$ in the blend skinning equation, and compute the skinning weights with Eq. (4) using the corresponding articulation codes. To ensure the articulation fields are self-consistent, we use a 3D cycle loss following prior works [29, 73].

3.3. Physics-Informed Reconstruction

We define a differentiable physics simulation module to constrain the scene and object representations.

Coordinate Transforms. To simulate physics, we define a world coordinate system where gravity points in the -y direction and the ground is located at the x-z plane. A point from object space, denoted by $\mathbf{X}$, can be transformed into world space as:

$$\mathbf{X}_w = \mathbf{G}_{o\rightarrow w}\mathbf{X} = \mathbf{G}_{b\rightarrow w}{\mathbf{G}_b}^{-1}\mathbf{G}_o\mathbf{X}, \tag{7}$$

where $\mathbf{G}_o$ is the object root to camera transform and $\mathbf{G}_b$ is the background to camera transform, both of which can be estimated with differentiable rendering optimization [73]. $\mathbf{G}_{b\rightarrow w}$ is the background to world transform, and we solve it by fitting ground planes to the scene geometry (extracted from density fields by marching cubes [32]) per video.

Relative Scale Ambiguity. Notably, there is scale ambiguity between each independently moving scene element [13, 78], including the objects and the background (Fig. 2). For instance, one may reconstruct a regular-sized cat walking on the ground, or a tiny cat floating in the air, such that both interpretations explain the input video. To tackle the relative scale ambiguity, we use physics cues to optimize a scale factor s applied to the background geometry and camera translation, $\mathbf{T}_b^* = s\mathbf{T}_{c\rightarrow b}$, such that the total scene can be reconstructed up to a global scale factor.

During the physics optimization, s is updated to enable the simulated ragdoll to follow the reconstructed kinematics under gravity and contact constraints. Specifically, floating objects (which correspond to an overly large background scale) are penalized because they lead to a falling motion that is inconsistent with the kinematic reconstruction. Similarly, ground penetrations (which correspond to an overly small background scale) are also penalized because they lead to an inconsistent "bounce" from ground reaction forces.

Differentiable Ragdoll Simulation. We construct an articulated body dynamics model of a ragdoll using standard Newtonian dynamics [30, 38]:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{S}\boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^T\mathbf{f} - \mathbf{c}(\mathbf{q},\dot{\mathbf{q}})), \tag{8}$$

where $\mathbf{q} = [\mathbf{G}_{o\rightarrow w}, \mathbf{Q}] \in \mathbb{R}^{6+3B}$ contains the generalized coordinates of the ragdoll. $\mathbf{G}_{o\rightarrow w}$ is the root SE(3) transformation in Eq. (7) and $\mathbf{Q} \in \mathbb{R}^{3B}$ are joint angles produced by Eq. (3). $\mathbf{M}$ is the generalized mass matrix, $\mathbf{J}_c$ is the contact Jacobian computed by forward kinematics, $\mathbf{f}$ represents contact forces, $\mathbf{c}$ includes Coriolis force and gravity, and $\boldsymbol{\tau} \in \mathbb{R}^{3B}$ represents the joint torque actuation, which is mapped to the generalized coordinates using a selection matrix $\mathbf{S}$ [38]. Intuitively, Eq. (8) is the generalization of Newton's "F=MA" for rotating rigid bodies under contact. We *differentiably* simulate ragdoll rigid body dynamics with environmental contact using Warp [34, 69]. Warp performs semi-implicit Euler integration to compute the updated system state $(\mathbf{q}, \dot{\mathbf{q}})$, which is differentiable. To ensure differ-

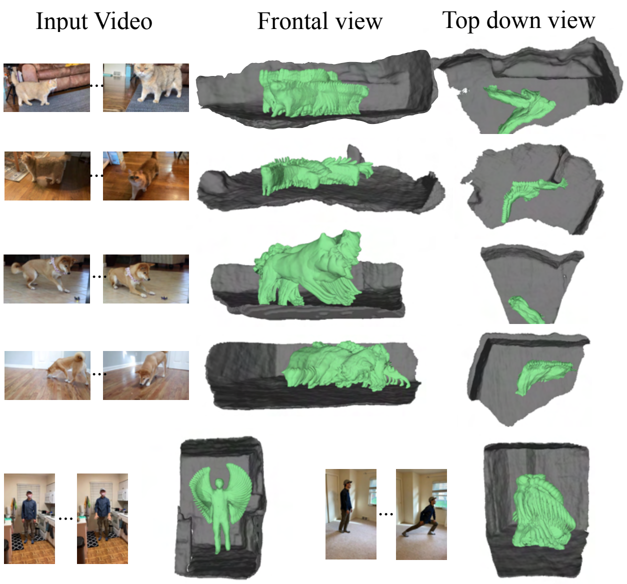


Figure 4: **Reconstruction given monocular videos.** We show physically-plausible reconstructions of the articulated shapes (green) and the surrounding environment (gray). Please see the supplement website for video results.

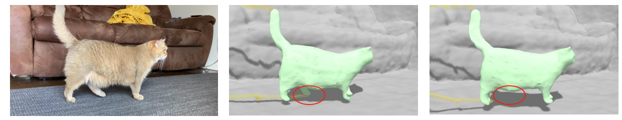
entiability through contact, Warp uses the frictional contact model that approximates Coulomb friction with a linear step function [10]. Additionally, it incorporates the contact damping force formulation to provide better smoothness in contact dynamics [68]. We refer readers to [69] for details.
Control. Rather than directly optimizing for time-varying joint torque profiles $\boldsymbol{\tau}_t$, we optimize for gain parameters of a Proportional Derivative (PD) Controller [39], and the time-varying target motion. Given target joint angles $\mathbf{Q}^t$, currently simulated joint angles $\mathbf{Q}^s$, and their derivatives at every frame, the PD controller computes joint torques to reach the target:

$$\boldsymbol{\tau}_t = \mathbf{K}_p(\mathbf{Q}^t - \mathbf{Q}^s) + \mathbf{K}_d(\dot{\mathbf{Q}}^t - \dot{\mathbf{Q}}^s), \tag{9}$$

where $\mathbf{K}_p \in \mathbb{R}^{3B}$ and $\mathbf{K}_d \in \mathbb{R}^{3B}$ are PD gains. During optimization (described in the next section), both the gains $(\mathbf{K}_p, \mathbf{K}_d)$ and the targets $(\mathbf{Q}^t, \dot{\mathbf{Q}}^t)$ are updated. $\mathbf{Q}^t$ is initialized as the most recent kinematic reconstruction.

3.4. Optimization and Losses

Given monocular videos of an articulated target, we optimize the geometric parameters including the object and scene radiance fields $\mathbf{T}$, kinematic (or motion) parameters $\mathbf{D} = \{\mathbf{G}_o, \mathbf{G}_b, \mathbf{A}, \mathbf{S}\}$, as well as physics parameters $\phi = \{s, \mathbf{M}, \mathbf{K}, \mathbf{Q}^t\}$ as described above. The model is learned by minimizing two types of losses: differentiable rendering losses and differentiable physics losses.
Differentiable Rendering (DR) Losses. Similar to BANMo [73], the DR losses leverage differentiable volume



Input View PPR-full PPR w/o physics

Figure 5: **Differentiable physics simulation helps 3D tracking.** We draw trajectories for points on the left-rear foot of the cat over time (yellow lines). Feet undergoing walking motion are difficult to track using visual cues, due to similar textures and occlusion by the other body parts. PPR uses dynamics priors via physics simulation to bias the solution towards avoiding sudden changes of velocity, and tracks the left-rear foot undergoing occlusion.

rendering to update both the neural fields $\mathbf{T}$ and the kinematic parameters $\mathbf{D}$. Reconstruction losses are similar to those in existing differentiable rendering pipelines [37, 75], where the goal is the minimize the difference between the rendered images (including object silhouette, color, optical flow, and pixel features) and the observed ones:

$$\mathcal{L}_{\text{DR}}(\mathbf{T}, \mathbf{D}) = \mathcal{L}_{\text{sil}} + \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{OF}} + \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{Reg}}. \tag{10}$$

We refer readers to the supplementary material for the regularization terms, and BANMo [73] for the volume rendering equations for each rendered image quantity. To disentangle the object from the background, we use off-the-shelf estimates of object segmentation [21] as supervision to kick-start the optimization. To account for errors in the off-the-shelf segmentation, we set the weight of the silhouette term to 0 after several iterations of optimization, while the composite rendering of foreground and background itself is capable of disentangling the object and the non-object components by matching the image evidence.
Differentiable Physics (DP) Losses. While image reconstruction losses alone can achieve visually appealing results from the reference viewpoint, the resulting poses can be physically implausible (e.g., Fig. 2), particularly for the relative scale and the non-visible body parts. To address this ambiguity, we use a differentiable physics simulator to regularize the solution. The physics term is defined as the difference between the *observed* kinematics $\mathbf{q}$ and a *simulated* trajectory $\mathbf{q}^s$ that is by construction physically-plausible:

$$\mathcal{L}_{\text{DP}}(\mathbf{D}, \phi) = \sum_{t=t_0}^{t_0+T} \|\mathbf{q}_t(\mathbf{D}) - \mathbf{q}_t^s(\phi)\| \quad \text{such that}$$
$$\mathbf{q}_{t+1}^s = \mathcal{I}(\mathbf{q}_t^s, \phi). \tag{11}$$

Here, the observed kinematics $\mathbf{q}$ are a function of reconstructed root coordinates in Eq. (7) and joint angles in Eq. (3), while the simulated trajectory $\mathbf{q}^s$ is a function of physical parameters ϕ including scale, body mass and control. Notably, $\mathbf{q}^s$ is *constrained* to be physically plausible

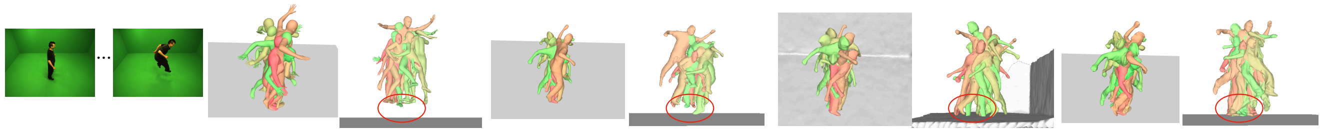


Figure 6: **Qualitative comparison on `AMA-bouncing`.** We visualize the reconstructed meshes in world coordinates, colored by their timestamp (red: early; green: late). From the input viewpoint (left), the reconstruction of PPR looks comparable to the baselines. From a novel viewpoint (as if viewing the scene from the right), PPR produces more physically plausible poses. HuMoR [49] reconstructs a floating person with physically-implausible ground contact. BANMo with ground plane fitting (following NeuMan [18]) estimates a rough scale of the person, but produces inaccurate foot contact (note the feet make contact with the ground only once over the video). PPR jointly optimizes 3D pose and relative scale under dynamics and contact constraints, producing accurate foot contact and upright human poses. Please find more visual comparisons in the supplement.

since it is the output of a physics simulator $\mathcal{I}$, which is also *differentiable* and therefore allows one to compute $\frac{\partial \mathcal{I}(\cdot)}{\phi}$.

Coordinate Descent Optimization. In theory, the overall $\text{Loss}(\mathbf{T}, \mathbf{D}, \boldsymbol{\phi}) = \mathcal{L}_{\text{DR}}(\mathbf{T}, \mathbf{D}) + \mathcal{L}_{\text{DP}}(\mathbf{D}, \phi)$ could be directly optimized overall photometric, geometric, and physical parameters [33]. However, differentiable rendering and physical simulation tend to require different sampling strategies [47]. For instance, differentiable rendering is more efficient when sampling pixels uniformly from a dataset to encourage batch diversity, while differentiable physics requires samples of long time intervals that support physical dynamic computations. As a result, joint optimization is rather sample-inefficient. Instead, we optimize by repeating the following two steps of coordinate descent:

1. $\min_{\mathbf{T},\mathbf{D}} \text{Loss}(\mathbf{T}, \mathbf{D}, \boldsymbol{\phi})$ [Differentiable Rendering]
2. $\min_{\boldsymbol{\phi}} \text{Loss}(\mathbf{T}, \mathbf{D}, \boldsymbol{\phi})$ [Differentiable Physics]

Step 1 corresponds to a standard differentiable rendering optimization where the kinematics $\mathbf{D}$ are *regularized* to be similar to the most recent simulated trajectory $\mathbf{q}^s(\boldsymbol{\phi})$. Step 2 corresponds to a differentiable physics optimization that solves for physical parameters that closely *targets* (or tracks) the most recent kinematic reconstruction $\mathbf{D}$.

SGD. In practice, each coordinate step is implemented via a fixed number of stochastic gradient descent (SGD) steps, initialized from the values of the variables in the previous descent cycle. Specifically, Step 1 constructs a batch of $N_{px} = 4096$ random pixels for optimization, performing $\mathcal{N}_{\text{DR}} = 10$ iterations of SGD. Because the reconstructed kinematics will be heavily regularized to lie close to the most recent physically-plausible trajectory $\mathbf{q}^s(\boldsymbol{\phi})$, we found faster convergence by initializing $\mathbf{D}$ to $\arg\min_{\mathbf{D}} \mathcal{L}_{\text{DP}}(\mathbf{D}, \boldsymbol{\phi})$, rather than its value from the previous DR cycle. Finally, for SGD optimization of Step 2, we construct random batches of $N_{int} = 30$ random *time intervals* of length 2.4 seconds, performing $\mathcal{N}_{\text{D}P} = 10$ iterations of SGD per cycle. In practice, we perform 50-100 cycles of coordinate descents.

Comparison to Prior Methods. PPR differs from prior work [8, 9, 50, 53] in several ways. First, prior art can be conceptualized as one cycle of coordinate descent, by producing a single kinematic reconstruction (Step 1) to which one fits a ragdoll simulation (Step 2). Second, the simulator often solves for forces assuming known physical parameters (such as generalized mass $\mathbf{M}$ and control parameters K), while PPR optimizes over such parameters. We ablate such design choices in our experiments. Intuitively, multiple descent cycles allow for different kinematic reconstructions, rather than a single (potentially inaccurate) point estimate of geometry and kinematics. Finally, from a control perspective, one can view Step 2 as an instance of model-predictive control (MPC) with stochastic batch sampling over small time intervals [45].

4. Experiments

Dataset. We test PPR on the articulated-mesh-animation (`AMA`) dataset, the `casual-videos` dataset from BANMo, and a newly collected `RGBD-pet` dataset.

`AMA` contains videos of human performance with 3D mesh ground-truth, and we use it to evaluate surface reconstruction accuracy. To test our method, we select 4 videos of `samba` and 4 videos of `bouncing`. Although the videos are calibrated multi-view captures, we treat them as monocular videos and *do not* use the time-synchronization or camera extrinsic parameters.

`Casual-videos` includes monocular videos of the same instance captured from different viewpoints, locations, and times. It contains 11 videos of a cat, 11 videos of a dog, and 10 videos of a human. We manually annotate per-frame binary foot contact labels and use them to evaluate contact reconstruction accuracy.

`RGBD-pet` dataset contains videos of a cat and a dog, captured by an iPad with an RGBD sensor. We use it to evaluate scene reconstruction performance (please see supplement).

Implementation Details. Our differentiable rendering pipeline is implemented with Pytorch. The object neural fields follow BANMo [73], and the background neural fields follow VolSDF [74]. We modify the neural blend skinning of BANMo to represent skeletal deformation and

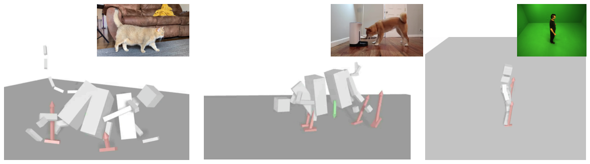


Figure 7: **Visualization of ground reaction force.** We show the ground contact returned by the simulator. The body part in contact with the ground is colored in red and the ground reaction forces are marked with red arrows. Gravity is represented by a green arrow. Note the contact modes are aligned with the image evidence.

follow CaDeX [26] to represent soft deformation as invertible 3D flow fields. We use Warp [34] for differentiable physics simulation. It implements articulated body dynamics as well as contact physics and integrates kinematics over time using the semi-implicit Euler scheme. The step size of the simulator is set to $5e^{-4}$s. The simulation operations are automatically differentiated and parallelized at the CUDA kernel level. We write wrappers to pass gradients between Warp and Pytorch.

Hyper-parameters. We use AdamW optimizer and optimize the model for 36k iterations (taking around 18 hours on 2 NVIDIA GeForce RTX 3090 GPUs). The weights of loss terms are tuned to have similar initial magnitudes. We first pre-train a background field [77], and jointly optimize an object field with differentiable composite rendering [41]. The object root poses are initialized with a viewpoint network following BANMo.

Extracting Registered Meshes. To evaluate surface reconstruction accuracy, we extract the canonical mesh by finding the zero-level set of SDF with running marching cubes on a 256^3 grid. To get the shape at a specific time instance, the canonical mesh is forward warped with $\mathcal{W}^{t,\rightarrow}$, which defines an articulation and deformation operation.

4.1. Surface Reconstruction

Metrics. To evaluate the reconstruction quality, we report both Chamfer distance and F-scores. Chamfer distance computes the average distance between the ground truth and the estimated surface points by finding the nearest neighbor matches, but it is sensitive to outliers. F-score at distance thresholds $d \in \{5\text{cm}, 2\text{cm}\}$ [58] provides a more informative quantification of surface reconstruction error at different granularity. To account for the scale ambiguity, we fit a per-video scale factor by aligning the predicted mesh with the ground truth in the view space.

Baselines. We compare PPR against HuMoR and BANMo for human reconstruction. HuMoR [49] learns human motion priors (in the world coordinate) from large-scale motion capture datasets. Given an input video, it performs test-time optimization leveraging OpenPose keypoint detections

Table 1: **Surface reconstruction evaluation on `AMA`.** 3D Chamfer distance (cm, ↓) and F-score (%, ↑) are averaged over all the frames. The best results are in bold. We align the reconstructed meshes with the ground-truth meshes by a per-sequence scale factor and SE(3) transformation. PPR outperforms HuMoR and BANMo in all metrics. Replacing BANMo's control point deformation with our skeleton deformation significantly improves results on `samba` but made results worse on `bouncing`. Further enforcing dynamics and contact constraints via differentiable physics simulation (PPR-Ours) significantly improves results on both sequences.

Method	`samba`			`bouncing`		
	CD	F@5cm	F@2cm	CD	F@5cm	F@2cm
HuMoR	10.5	65.3	31.7	14.8	49.0	18.9
BANMo	11.8	56.7	27.2	12.8	56.0	25.6
+skel	8.9	68.1	32.0	14.1	51.3	23.9
PPR-Ours	**8.3**	**73.4**	**35.4**	**9.1**	**68.3**	**32.8**

and the learned humor motion priors. Processing a 170-frame video takes around 2 hours on a Titan-X machine. BANMo [73] is a template-free method for video-based deformable shape reconstruction. It relies on differentiable rendering optimization given optical flow correspondence and DensePose features [40]. Running BANMo on around 1k frames takes 10 hours on two V100 GPUs.

Results. We show qualitative results in Fig. 6, and quantitative results in Tab. 1. HuMoR produces accurate and consistent reconstructions for videos with common motion (such as the samba sequence). However, human motion prior fails for certain athletic movements (such as the bouncing sequence) due to the lack of those motions in the human MoCap dataset. BANMo reconstructions look reasonable from the reference viewpoint, but the invisible body parts often appear distorted or tilted from a novel viewpoint due to the fundamental depth ambiguity. In contrast, PPR's differentiable rendering module alone improves the reconstruction (CD: 8.9% vs 11.8% for samba) by constraining the body deformation with a skeleton. Our physics-informed optimization further improves the reconstruction (CD: 8.3% vs 8.9% for samba) by inferring root pose and body motions that satisfy contact and dynamics constraints in a differentiable physics simulator.

4.2. Contact Estimation

Evaluation Protocol. To evaluate the physical plausibility of the reconstructions, we follow DiffPhy [8] to design contact metrics, including contact accuracy and foot skate. The contact accuracy is defined as the F-score of contact state estimation averaged over all frames. We further measure the amount of foot skate as the average distance feet move over adjacent contact frames, frames that are marked as in contact with the ground according to the ground truth. To predict contact state, we mark a foot to be in contact with

Table 2: **Foot contact estimation on `casual-cat` and AMA.** Foot contact F-score (%, ↑) and skating (cm, ↓) are averaged over all frames. The best results are in bold. *To compare with BANMo that only produces camera-space reconstructions, we take a step further to reconstruct the background and use ground prior to finding the scale of camera motion following NeuMan [18], and decouple it from BANMo results to produce world-space reconstructions. By enforcing physics constraints, PPR-Ours outperforms the baselines in contact estimation. It produces more foot skates than HuMoR, but fewer foot skates than BANMo.

Method	`casual-cat`		`samba`		`bouncing`	
	Contact	Skate	Contact	Skate	Contact	Skate
HuMoR	N.A.		44.6	**0.4**	54.8	**2.6**
BANMo*	68.6	2.8	24.5	1.6	1.3	8.3
PPR-Ours	**93.1**	**2.1**	**67.4**	1.2	**85.4**	7.4

the ground if its distance to the ground is smaller than 5% of the body height.

Results. We show quantitative evaluation in Tab. 2. Our method with physics-informed optimization achieves the highest accuracy in contact estimation. BANMo with ground fitting solves a rough relative scale between the background and the object, and we found it to produce worse results than PPR (Contact: 68.6 v 93.1 for casual-cat). We posit that scale fitting suffers from inaccurate kinematic reconstructions, while PPR jointly improves both via differentiable physics simulation. In terms of foot skate, although PPR works better than BANMo, it still produces more foot skate than HuMoR, especially for the highly-dynamic bouncing sequence. Enforcing a stronger physics prior for PPR (e.g., simulating longer time intervals) may produce fewer foot skates. However, doing so makes the motion more challenging to track by a controller. Besides contact estimation, PPR also produces plausible ground reaction force estimation as shown in Fig. 7.

4.3. Ablation Study

We ablate design choices and show the results in Tab. 3.

Ground Prior vs Differentiable Physics. One commonly used approach to determine the object scale is to force the reconstructions to be above the ground plane except for a supporting region touching the ground [18]. However, this is sensitive to the accuracy of the visual reconstruction, often resulting inaccurate ground contact with tilted body poses, as illustrated in Fig. 2 (c). As a result, replacing differentiable physics simulation with ground prior makes contact estimation accuracy much worse (67.4% vs 26.3%).

One-cycle vs Coordinate Descent Optimization. Instead of alternating between visual reconstruction and physics-informed optimization, existing works [8, 9, 50, 53] only complete one cycle by first estimating the shape and motion (using DR or feed-forward networks) and then optimizing physics (using DP or trajectory optimization). Compared to using coordinate descent, the reconstruction error of one-cycle optimization significantly increases (8.3cm vs 46.0cm). In terms of contact metrics, the foot contact accuracy dropped (62.3% vs 67.4%), although the skating metric becomes slightly better. This validates the effectiveness of joint vision-physics optimization via coordinate descent: alternating between visual reconstruction and physics-informed optimization improves reconstruction quality while making the solution physically plausible.

Table 3: **Ablation study on AMA-`samba`.** Best results are in bold. Please see Sec. 4.3 for a detailed discussion.

Method	Contact (%, ↑)	Skate (cm, ↓)	CD (cm, ↓)
Full method	**67.4**	1.2	**8.3**
Phys→ground	26.3	1.3	8.9
One-cycle	62.3	**1.1**	46.0
PD→ open-loop	53.6	2.7	12.7
Freeze K/M	50.4	1.2	9.0

Feedback vs Open-loop Control. We ablate the necessity of using feedback control (specifically PD control in Sec. 3.3) during differentiable simulation, as some existing works [16, 17] directly optimize open-loop control without position and velocity feedback. We find that open-loop control fails to track the target kinematics from DR, and decreases both the contact accuracy and reconstruction quality. This suggests incorporating prior knowledge into the controller function improves reconstruction results and physical plausibility.

Optimizing Mass and PD Gains. We further investigate the effect of optimizing the mass of each body part, as well as the PD gains for each joint of the ragdoll. We found freezing $\mathbf{K}$ and $\mathbf{M}$ decreases contact estimation accuracy and reconstruction quality (even worse than without DP). This suggests jointly inferring the internal parameters of the ragdoll is important for physics-informed optimization.

5. Conclusion

We have presented a method for 3D-capturing dynamic objects and environments from monocular videos. PPR combines differentiable rendering and differentiable physics simulation, where the former builds a faithful 3D model of the dynamic object and the rigid background scene, and the latter fixes the physically-implausible configurations, such as floating, unbalanced pose, foot staking, and part swapping. PPR can generate physically plausible trajectories, hence it has the potential to synthesize reference motions for legged robots [44, 79], and learn animal motion priors [49] from internet video collections. The assumption of the rigid body contact model limits PPR to terrestrial creatures making contact with a flat ground plane. Extending it to contact scenarios in a complex environment and between multiple agents will be interesting future work.

References

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

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

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

[4] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *IJCV*, 107(2):101–122, 2014. 1

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 4

[6] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, pages 10786–10796, 2021. 3

[7] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation, 2021. 2

[8] Erik Gärtner, Mykhaylo Andriluka, Erwin Coumans, and Cristian Sminchisescu. Differentiable dynamics for articulated 3d human motion reconstruction. In *CVPR*, pages 13190–13200, 2022. 1, 2, 6, 7, 8

[9] Erik Gärtner, Mykhaylo Andriluka, Hongyi Xu, and Cristian Sminchisescu. Trajectory optimization for physics-based reconstruction of 3d human pose from monocular video. In *CVPR*, pages 13106–13115, 2022. 6, 8

[10] Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. Add: Analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 5

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

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

[13] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 4

[14] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S Sukhatme. NeuralSim: Augmenting differentiable simulators with neural networks. In *ICRA*, 2021. 2

[15] Taylor A Howell, Simon Le Cleac'h, J Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable simulator for robotics. *arXiv preprint arXiv:2203.00806*, 2022. 2

[16] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *ICLR*, 2020. 2, 8

[17] Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jérôme Parent-Lévesque, Kevin Xie, Kenny Erleben, et al. gradsim: Differentiable simulation for system identification and visuomotor control. *ICLR*, 2021. 8

[18] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *ECCV*, pages 402–418. Springer, 2022. 6, 8

[19] Navami Kairanda, Edith Tretschk, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. f-sft: Shape-from-template with a physics-based deformation model. In *CVPR*, pages 3948–3958, 2022. 2

[20] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 1, 2

[21] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. 2019. 5

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

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

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

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

[26] Jiahui Lei and Kostas Daniilidis. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *CVPR*, 2022. 4, 7

[27] 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

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

[29] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 4

[30] C Karen Liu and Sumit Jain. A quick tutorial on multibody dynamics. *Online tutorial, June*, page 7, 2012. 4

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

[32] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 21(4):163–169, 1987. 4

[33] Pingchuan Ma, Tao Du, Joshua B Tenenbaum, Wojciech Matusik, and Chuang Gan. Risp: Rendering-invariant state predictor with differentiable simulation and rendering for cross-domain parameter estimation. *arXiv preprint arXiv:2205.05678*, 2022. 6

[34] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, March 2022. NVIDIA GPU Technology Conference (GTC). 2, 4, 7

[35] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi,

Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 3

[36] Mariem Mezghanni, Théo Bodrito, Malika Boulkenafed, and Maks Ovsjanikov. Physical simulation layer for accurate 3d modeling. In *CVPR*, pages 13514–13523, 2022. 2

[37] 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. 5

[38] Michael Mistry, Jonas Buchli, and Stefan Schaal. Inverse dynamics control of floating base systems using orthogonal decomposition. In *2010 IEEE International Conference on Robotics and Automation*, pages 3406–3412, 2010. 4

[39] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017. 4, 5

[40] Natalia Neverova, Artsiom Sanakoyeu, Patrick Labatut, David Novotny, and Andrea Vedaldi. Discovering relationships between object categories via universal canonical maps. In *CVPR*, 2021. 7

[41] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021. 3, 7

[42] 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, 4

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

[44] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020. 8

[45] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014. 6

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

[47] Yi-Ling Qiao, Alexander Gao, and Ming C. Lin. Neuphysics: Editable neural geometry and physics from monocular videos. In *NeurIPS*, 2022. 6

[48] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C Lin. Efficient differentiable simulation of articulated bodies. In *ICLR*, pages 8661–8671. PMLR, 2021. 2

[49] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, 2021. 6, 7, 8

[50] Davis Rempe, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *ECCV*, pages 71–87. Springer, 2020. 1, 2, 6, 8

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

[52] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt. Neural monocular 3d human motion capture with physical awareness. *ToG*, 40(4), aug 2021. 2

[53] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG)*, 39(6):1–16, 2020. 1, 2, 6, 8

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

[55] David E Stewart and J C Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996. 2

[56] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*. 2007. 4

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

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

[59] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 3

[60] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 34, 2021. 3

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

[62] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *ECCV*, pages 139–155. Springer, 2022. 3

[63] 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*, pages 16210–16220, 2022. 4

[64] Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and C Karen Liu. Fast and feature-complete differentiable physics for articulated rigid bodies with contact. *RSS*, 2021. 2

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

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

[67] 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*, pages 6184–6193, 2020. 2

[68] Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech

Matusik, Shinjiro Sueda, and Pulkit Agrawal. An end-to-end differentiable framework for contact-aware robot design. *RSS*, 2021. 5

[69] Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. *ICLR*, 2022. 4, 5

[70] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 3

[71] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021. 2, 4

[72] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. In *NeurIPS*, 2021. 2

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

[74] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 2021. 3, 6

[75] 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

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

[77] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 2022. 3, 7

[78] Chang Yuan, Gerard Medioni, Jinman Kang, and Isaac Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. In *TPAMI*, 2007. 4

[79] John Z Zhang, Shuo Yang, Gengshan Yang, Arun L Bishop, Deva Ramanan, and Zachary Manchester. Slomo: A general system for legged robot motion imitation from casual videos. *arXiv preprint arXiv:2304.14389*, 2023. 8

[80] Yaofeng Desmond Zhong, Jiequn Han, and Georgia Olympia Brikis. Differentiable physics simulations with contacts: Do they have correct gradients wrt position, velocity and control? In *ICML 2022 2nd AI for Science Workshop*. 2

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

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

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