PPR: Physically Plausible Reconstruction from Monocular Videos

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

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 quadrupedal 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-imausible 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
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.
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 $T$ 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 $T^o$. Our canonical model for the object is similar to BANMo [73]. A 3D point $X \in \mathbb{R}^3$ is associated with three properties: signed distance $d \in \mathbb{R}$, color $c \in \mathbb{R}^3$, and canonical features $\psi \in \mathbb{R}^{10}$, which are used to register pixel observations to the canonical space. These properties are predicted by multi-layer perceptron (MLP) networks:

\[
(d, c^t) = \text{MLP}_\sigma (X, \omega_o), \tag{1}
\]
\[
\psi = \text{MLP}_\psi (X). \tag{2}
\]

Besides the 3D point locations, we further condition the color on an appearance code $\omega_o \in \mathbb{R}^{64}$ 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 $\mathcal{W}^{f,t}$ that transforms 3D points from the canonical space to the specified time instance, and a backward warping field $\mathcal{W}^{b,t}$ to transform points in the inverse direction. The warping fields are further explained in Sec. 3.2.

Scene Fields $T^s$. 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 $G_o$, skeleton articulations $A = \{J, W, Q\}$ and soft deformations $S$.

Global Movement $\{G_b, G_o\}$. We model the background-to-camera transformations $G_b$ and object root-to-camera transformations $G_o$ as per-frame $\text{SE}(3)$ represented as Fourier-based MLP networks.

Skeleton Articulation $\{J, Q, W\}$. 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 $J \in \mathbb{R}^{3 \times (B-1)}$ and Gaussian bones of size $L \in \mathbb{R}^{3 \times B}$. We set $B=26$ for quadrupeds and $B=19$ for humans. The skeleton topology is fixed through optimization but $J$ and $L$ are specialized to input videos. To model time-varying skeletal
movements, we define per-frame joint angles:

$$q = \text{MLP}_A(\theta) \in \mathbb{R}^{3 \times (B-1)},$$

where $\theta \in \mathbb{R}^{16}$ is a low-dimensional articulation code. Given joint angles and the per-video joint locations, we compute bone transformations $G \in \mathbb{R}^{3 \times 4 \times B}$ 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 $X$ [56, 71],

$$W = \sigma_{\text{softmax}}(d_\sigma(X, \theta) + \text{MLP}_W(X, \theta)) \in \mathbb{R}^B,$$

where $\theta$ is a pose code and $d_\sigma(X, \theta)$ is the Mahalanobis distance between $X$ and Gaussian bones under pose $\theta$, redefined 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,

$$X(\theta) = \left( \sum_{b=1}^B W_b G_b \right) X.$$

**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] $S(\cdot)$ 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],

$$X(\omega_d) = S(X, \omega_d),$$

where $\omega_d$ is a per-frame soft deformation code. Compared to applying $S$ 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 $\{W^{f,\rightarrow}, W^{f,\leftarrow}\}$ 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 $G$ 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 $X$, can be transformed into world space as:

$$X_w = G_{o \rightarrow w} X = G_b \cdot w G_b^{-1} G_o X,$$

where $G_o$ is the object root to camera transform and $G_b$ is the background to camera transform, both of which can be estimated with differentiable rendering optimization [73]. $G_{b \rightarrow w}$ 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, $T_b = s T_{c \rightarrow b}$, 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]:

$$\dot{q} = M^{-1} (S \tau + J_c(q)^T f - c(q, \dot{q})), $$

where $q = [G_{o \rightarrow w}, Q] \in \mathbb{R}^{6 + 3B}$ contains the generalized coordinates of the ragdoll. $G_{o \rightarrow w}$ is the root SE(3) transformation in Eq. (7) and $Q \in \mathbb{R}^{3B}$ contains the joint angles produced by Eq. (3). $M$ is the generalized mass matrix, $J_c$ is the contact Jacobian computed by forward kinematics, $f$ represents contact forces, $c$ includes Coriolis force and gravity, and $\tau \in \mathbb{R}^{3B}$ represents the joint torque actuation, which is mapped to the generalized coordinates using a selection matrix $S$ [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 $(q, \dot{q})$, which is differentiable. To ensure differ-
Differentiable Rendering (DR) Losses. Similar to rendering to update both the neural fields $T$ and the kinematic parameters $D$. Reconstruction losses are similar to those in existing differentiable rendering pipelines \cite{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:

$$L_{\text{DR}}(T, D) = L_{\text{sil}} + L_{\text{rgb}} + L_{\text{OF}} + L_{\text{feat}} + L_{\text{Reg}}.$$  

(10)

We refer readers to the supplementary material for the regularization terms, and BANMo \cite{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 \cite{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 $q$ and a simulated trajectory $q^s$ that is by construction physically-plausible:

$$L_{\text{DP}}(D, \phi) = \sum_{t=t_0}^{t_0+T} \|q_t(D) - q^s_t(\phi)\|$$ such that

$$q^s_{t+1} = \mathcal{I}(q^s_t, \phi).$$  

(11)

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

3.4. Optimization and Losses

Given monocular videos of an articulated target, we optimize the geometric parameters including the object and scene radiance fields $T$, kinematic (or motion) parameters $D = \{G_o, G_b, A, S\}$, as well as physics parameters $\phi = \{s, M, K, Q^t\}$ 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 \cite{73}, the DR losses leverage differentiable volume rendering to update both the neural fields $T$ and the kinematic parameters $D$. Reconstruction losses are similar to those in existing differentiable rendering pipelines \cite{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:

$$L_{\text{DR}}(T, D) = L_{\text{sil}} + L_{\text{rgb}} + L_{\text{OF}} + L_{\text{feat}} + L_{\text{Reg}}.$$  

(10)

We refer readers to the supplementary material for the regularization terms, and BANMo \cite{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 \cite{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 $q$ and a simulated trajectory $q^s$ that is by construction physically-plausible:

$$L_{\text{DP}}(D, \phi) = \sum_{t=t_0}^{t_0+T} \|q_t(D) - q^s_t(\phi)\|$$ such that

$$q^s_{t+1} = \mathcal{I}(q^s_t, \phi).$$  

(11)

Here, the observed kinematics $q$ are a function of re-constructed root coordinates in Eq. (7) and joint angles in Eq. (3), while the simulated trajectory $q^s$ is a function of physical parameters $\phi$ including scale, body mass and control. Notably, $q^s$ is constrained to be physically plausible.
since it is the output of a physics simulator $I$, which is also differentiable and therefore allows one to compute $\frac{\partial L}{\partial \psi}$.

**Coordinate Descent Optimization.** In theory, the overall loss $L(T, D, \psi) = L_{DR}(T, D) + L_{DP}(D, \psi)$ could be directly minimized to optimize 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. $\min_{T, D} L(T, D, \psi)$ [Differentiable Rendering]
2. $\min_{\psi} L(T, D, \psi)$ [Differentiable Physics]

Step 1 corresponds to a standard differentiable rendering optimization where the kinematics $D$ are regularized to be similar to the most recent simulated trajectory $\psi^s(\cdot)$. Step 2 corresponds to a differentiable physics optimization that solves for physical parameters that closely target (or tracks) the most recent kinematic reconstruction $D$.

**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 $N_{DR} = 10$ iterations of SGD. Because the reconstructed kinematics will be heavily regularized to lie close to the most recent physically-plausible trajectory $\psi^s(\cdot)$, we found faster convergence by initializing $D$ to $\arg\min_D L_{DR}(D, \psi)$, 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 $N_{DP} = 10$ iterations of SGD per cycle. In practice, we perform 50-100 cycles of coordinate desents.

**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 $M$ 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
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 $\mathcal{W}^{t_{\text{ref}}, \rightarrow}$, 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 $d \in \{5\text{cm, 2cm}\}$ [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 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

![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.](image)

Table 1: **Surface reconstruction evaluation on AMA.** 3D Chamfer distance (cm, ↓) and F-score (%,$\uparrow$) 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>samba</th>
<th>bouncing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CD</td>
<td>F@5cm</td>
</tr>
<tr>
<td>HuMoR</td>
<td>10.5</td>
<td>65.3</td>
</tr>
<tr>
<td>BANMo</td>
<td>11.8</td>
<td>56.7</td>
</tr>
<tr>
<td>+skel</td>
<td>8.9</td>
<td>68.1</td>
</tr>
<tr>
<td>PPR-Ours</td>
<td>8.3</td>
<td>73.4</td>
</tr>
</tbody>
</table>
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 both methods. 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, even though HuMoR has 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>casual-cat Contact</th>
<th>casual-cat Skate</th>
<th>samba Contact</th>
<th>samba Skate</th>
<th>bouncing Contact</th>
<th>bouncing Skate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HuMoR</td>
<td>N.A.</td>
<td>44.6</td>
<td>0.4</td>
<td>54.8</td>
<td>2.6</td>
<td>2.6</td>
</tr>
<tr>
<td>BANMo*</td>
<td>68.6</td>
<td>2.8</td>
<td>24.5</td>
<td>1.6</td>
<td>8.3</td>
<td></td>
</tr>
<tr>
<td>PPR-Ours</td>
<td>93.1</td>
<td>2.1</td>
<td>67.4</td>
<td>1.2</td>
<td>85.4</td>
<td>7.4</td>
</tr>
</tbody>
</table>

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 in 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.

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 K and 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


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


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

[28] Xuejting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. ECCV, 2020. 2


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


[80] Yaoqeng 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