Contact and Human Dynamics from Monocular Video

Davis Rempe[1,2], Leonidas J. Guibas[1], Aaron Hertzmann[2], Bryan Russell[2], Ruben Villegas[2], and Jimei Yang[2]

[1] Stanford University
[2] Adobe Research
geometry.stanford.edu/projects/human-dynamics-eccv-2020

Abstract. Existing deep models predict 2D and 3D kinematic poses from video that are approximately accurate, but contain visible errors that violate physical constraints, such as feet penetrating the ground and bodies leaning at extreme angles. In this paper, we present a physics-based method for inferring 3D human motion from video sequences that takes initial 2D and 3D pose estimates as input. We first estimate ground contact timings with a novel prediction network which is trained without hand-labeled data. A physics-based trajectory optimization then solves for a physically-plausible motion, based on the inputs. We show this process produces motions that are significantly more realistic than those from purely kinematic methods, substantially improving quantitative measures of both kinematic and dynamic plausibility. We demonstrate our method on character animation and pose estimation tasks on dynamic motions of dancing and sports with complex contact patterns.

1 Introduction

Recent methods for human pose estimation from monocular video [1,20,34,47] estimate accurate overall body pose with small absolute differences from the true poses in body-frame 3D coordinates. However, the recovered motions in world-frame are visually and physically implausible in many ways, including feet that float slightly or penetrate the ground, implausible forward or backward body lean, and motion errors like jittery, vibrating poses. These errors would prevent many subsequent uses of the motions. For example, inference of actions, intentions, and emotion often depends on subtleties of pose, contact and acceleration, as does computer animation; human perception is highly sensitive to physical inaccuracies [16,38]. Adding more training data would not solve these problems, because existing methods do not account for physical plausibility.

Physics-based trajectory optimization presents an appealing solution to these issues, particularly for dynamic motions like walking or dancing. Physics imposes important constraints that are hard to express in pose space but easy in terms of dynamics. For example, feet in static contact do not move, the body moves smoothly overall relative to contacts, and joint torques are not large. However,

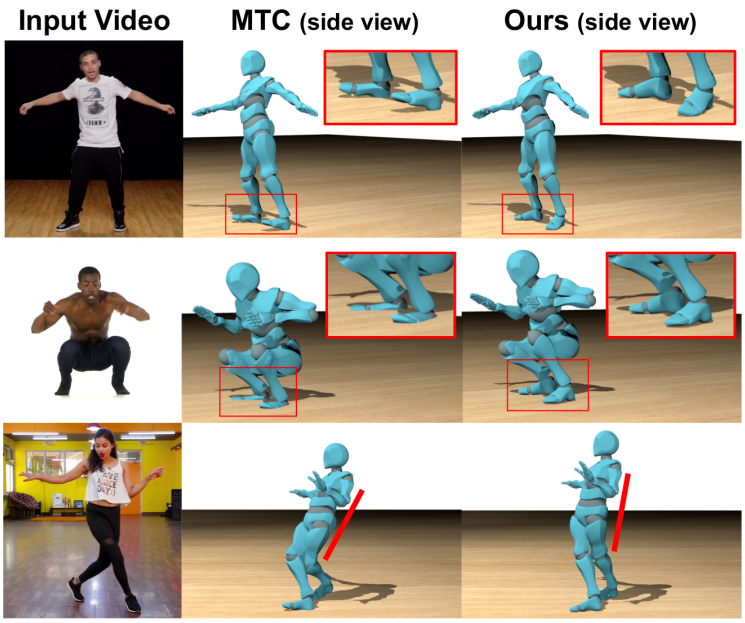

Fig. 1. Our contact prediction and physics-based optimization corrects numerous physically implausible artifacts common in 3D human motion estimations from, e.g., Monocular Total Capture (MTC) [47] such as foot floating (top row), foot penetrations (middle), and unnatural leaning (bottom).

full-body dynamics is notoriously difficult to optimize [40], in part because contact is discontinuous, and the number of possible contact events grows exponentially in time. As a result, combined optimization of contact and dynamics is enormously sensitive to local minima.

This paper introduces a new strategy for extracting dynamically valid full-body motions from monocular video (Figure 1), combining learned pose estimation with physical reasoning through trajectory optimization. As input, we use the results of kinematic pose estimation techniques [4,47], which produce accurate overall poses but inaccurate contacts and dynamics. Our method leverages a reduced-dimensional body model with centroidal dynamics and contact constraints [9,46] to produce a physically-valid motion that closely matches these inputs. We first infer foot contacts from 2D poses in the input video which are then used in a physics-based trajectory optimization to estimate 6D center-of-mass motion, feet positions, and contact forces. We show that a contact prediction network can be accurately trained on synthetic data. This allows us to separate initial contact estimation from motion optimization, making the optimization more tractable. As a result, our method is able to handle highly dynamic motions without sacrificing physical accuracy.

We focus on single-person dynamic motions from dance, walking, and sports. Our approach substantially improves the realism of inferred motions over state-of-the-art methods, and estimates numerous physical properties that could be useful for further inference of scene properties and action recognition. We primarily demonstrate our method on character animation by retargeting captured motion from video to a virtual character. We evaluate our approach using numerous kinematics and dynamics metrics designed to measure the physical plausiblity of the estimated motion. The proposed method takes an important step to incorporating physical constraints into human motion estimation from video, and shows the potential to reconstruct realistic, dynamic sequences.

2 Related Work

We build on several threads of work in computer vision, animation, and robotics, each with a long history [11]. Recent vision results are detailed here.

Recent progress in pose estimation can accurately detect 2D human keypoints [4,14,31] and infer 3D pose [1,20,34] from a single image. Several recent methods extract 3D human motions from monocular videos by exploring various forms of temporal cues [21,30,48,47]. While these methods focus on explaining human motion in pixel space, they do not account for physical plausibility. Several recent works interpret interactions between people and their environment in order to make inferences about each [7,13,49]; each of these works uses only static kinematic constraints. Zou et al. [50] infer contact constraints to optimize 3D motion from video. We show how dynamics can improve inference of human-scene interactions, leading to more physically plausible motion capture.

Some works have proposed physics constraints to address the issues of kinematic tracking. Brubaker et al. [3] propose a physics-based tracker based on a reduced-dimensional walking model. Wei and Chai [45] track body motion from video, assuming keyframe and contact constraints are provided. Similar to our own work, Brubaker and Fleet [2] perform trajectory optimization for full-body motion. To jointly optimize contact and dynamics, they use a continuous approximation to contact. However, soft contact models introduce new difficulties, including inaccurate transitions and sensitivity to stiffness parameters, while still suffering from local minima issues. Moreover, their reduced-dimensional model includes only center-of-mass positional motion, which does not handle rotational motion well. In contrast, we obtain accurate contact initialization in a preprocessing step to simplify optimization, and we model rotational inertia.

Li et al. [27] estimate dynamic properties from videos. We share the same overall pipeline of estimating pose and contacts, followed by trajectory optimization. Whereas they focus on the dynamics of human-object interactions, we focus on videos where the human motion itself is much more dynamic, with complex variation in pose and foot contact; we do not consider human-object interaction. They use a simpler data term, and perform trajectory optimization in full-body dynamics unlike our reduced representation. Their classifier training requires hand-labeled data, unlike our automatic dataset creation method.

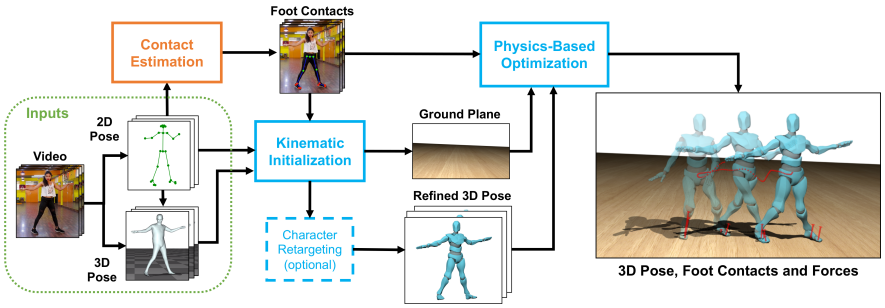

Fig. 2. Method overview. Given an input video, our method starts with initial estimates from existing 2D and 3D pose methods [4,47]. The lower-body 2D joints are used to infer foot contacts (orange box). Our optimization framework contains two parts (blue boxes). Inferred contacts and initial poses are used in a kinematic optimization that refines the 3D full-body motion and fits the ground. These are given to a reduced-dimensional physics-based trajectory optimization that applies dynamics.

Prior methods learn character animation controllers from video. Vondrak et al. [42] train a state-machine controller using image silhouette features. Peng et al. [36] train a controller to perform skills by following kinematically-estimated poses from input video sequences. They demonstrate impressive results on a variety of skills. They do not attempt accurate reconstruction of motion or contact, nor do they evaluate for these tasks, rather they focus on control learning.

Our optimization is related to physics-based methods in computer animation, e.g., [10,19,24,28,29,37,44]. Two unique features of our optimization are the use of low-dimensional dynamics optimization that includes 6D center-of-mass motion and contact constraints, thereby capturing important rotational and footstep quantities without requiring full-body optimization, and the use of a classifier to determine contacts before optimization.

3 Physics-Based Motion Estimation

This section describes our approach, which is summarized in Figure 2. The core of our method is a physics-based trajectory optimization that enforces dynamics on the input motion (Section 3.1). Foot contact timings are estimated in a preprocess (Section 3.2), along with other inputs to the optimization (Section 3.3). Similar to previous work [27,47], in order to recover full-body motion we assume there is no camera motion and that the full body is visible.

3.1 Physics-Based Trajectory Optimization

The core of our framework is an optimization which enforces dynamics on an initial motion estimate given as input (see Section 3.3). The goal is to improve the plausibility of the motion by applying physical reasoning through the objective

and constraints. We aim to avoid common perceptual errors, e.g., jittery, unnatural motion with feet skating and ground penetration, by generating a smooth trajectory with physically-valid momentum and static feet during contact.

The optimization is performed on a reduced-dimensional body model that captures overall motion, rotation, and contacts, but avoids the difficulty of optimizing all joints. Modeling rotation is necessary for important effects like arm swing and counter-oscillations [15,24,29], and the reduced-dimensional *centroidal* dynamics model can produce plausible trajectories for humanoid robots [5,9,32]. Our method is based on a recent robot motion planning algorithm from Winkler et al. [46] that leverages a simplified version of centroidal dynamics, which treats the robot as a rigid body with a fixed mass and moment of inertia. Their method finds a feasible trajectory by optimizing the position and rotation of the center-of-mass (COM) along with feet positions, contact forces, and contact durations as described in detail below. We modify this algorithm to suit our computer vision task: we use a temporally varying inertia tensor which allows for changes in mass distribution (swinging arms) and enables estimating the dynamic motions of interest, we add energy terms to match the input kinematic motion and foot contacts, and we add new kinematics constraints for our humanoid skeleton.

Inputs. The method takes initial estimates of: COM position $\bar{\mathbf{r}}(t) \in \mathbb{R}^3$ and orientation $\bar{\boldsymbol{\theta}}(t) \in \mathbb{R}^3$ trajectories, body-frame inertia tensor trajectory $\mathbf{I}_b(t) \in \mathbb{R}^{3\times3}$, and trajectories of the foot joint positions $\bar{\mathbf{p}}_{1:4}(t) \in \mathbb{R}^3$. There are four foot joints: left toe base, left heel, right toe base, and right heel, indexed as $i \in \{1, 2, 3, 4\}$. These inputs are at discrete timesteps, but we write them here as functions for clarity. The 3D ground plane height h_{floor} and upward normal is provided. Additionally, for each foot joint at each time, a binary label is provided indicating whether the foot is in contact with the ground. These labels determine initial estimates of contact durations for each foot joint $\bar{T}_{i,1}, \bar{T}_{i,2}, \ldots, \bar{T}_{i,n_i}$ as described below. The distance from toe to heel ℓ_{foot} and maximum distance from toe to hip ℓ_{leg} are also provided. All quantities are computed from video input as described in Sections 3.2 and 3.3, and are used to both initialize the optimization variables and as targets in the objective function.

Optimization Variables. The optimization variables are the COM position and Euler angle orientation $\mathbf{r}(t), \boldsymbol{\theta}(t) \in \mathbb{R}^3$, foot joint positions $\mathbf{p}_i(t) \in \mathbb{R}^3$ and contact forces $\mathbf{f}_i(t) \in \mathbb{R}^3$. These variables are continuous functions of time, represented by piece-wise cubic polynomials with continuity constraints. We also optimize contact timings. The contacts for each foot joint are independently parameterized by a sequence of phases that alternate between contact and flight. The optimizer cannot change the type of each phase (contact or flight), but it can modify their durations $T_{i,1}, T_{i,2}, \ldots, T_{i,n_i} \in \mathbb{R}$ where n_i is the number of total contact phases for the ith foot joint.

Objective. Our complete formulation is shown in Figure 3. E_{data} and E_{dur} seek to keep the motion and contacts as close as possible to the intial inputs, which

$$
\begin{aligned}
\min \quad & \textstyle\sum_{t=0}^{T} \Big(E_{data}(t) + E_{vel}(t) + E_{acc}(t) \Big) + E_{dur} \\
\text{s.t.} \quad & m\ddot{\mathbf{r}}(t) = \textstyle\sum_{i=1}^{4} \mathbf{f}_i(t) + m\mathbf{g} && \text{(dynamics)} \\
& \mathbf{I}_w(t)\dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}(t) \times \mathbf{I}_w(t)\boldsymbol{\omega}(t) = \textstyle\sum_{i=1}^{4} \mathbf{f}_i(t) \times (\mathbf{r}(t) - \mathbf{p}_i(t)) \\
& \dot{\mathbf{r}}(0) = \dot{\bar{\mathbf{r}}}(0), \dot{\mathbf{r}}(T) = \dot{\bar{\mathbf{r}}}(T) && \text{(velocity boundaries)} \\
& ||\mathbf{p}_1(t) - \mathbf{p}_2(t)|| = ||\mathbf{p}_3(t) - \mathbf{p}_4(t)|| = \ell_{foot} && \text{(foot kinematics)} \\
& \text{for every foot joint } i: \\
& \quad ||\mathbf{p}_i(t) - \mathbf{p}_{hip,i}(t)|| \leq \ell_{leg} && \text{(leg kinematics)} \\
& \quad \textstyle\sum_{j=1}^{n_i} T_{i,j} = T && \text{(contact durations)} \\
& \text{for foot joint } i \text{ in contact at time } t: \\
& \quad \dot{\mathbf{p}}_i(t) = 0 && \text{(no slip)} \\
& \quad p_i^z(t) = h_{floor}(\mathbf{p}_i^{xy}) && \text{(on floor)} \\
& \quad 0 \leq \mathbf{f}_i(t)^T \hat{\mathbf{n}} \leq f_{max} && \text{(pushing/max force)} \\
& \quad |\mathbf{f}_i(t)^T \hat{\mathbf{t}}_{1,2}| < \mu \mathbf{f}_i(t)^T \hat{\mathbf{n}} && \text{(friction pyramid)} \\
& \text{for foot joint } i \text{ in flight at time } t: \\
& \quad p_i^z(t) \geq h_{floor}(\mathbf{p}_i^{xy}) && \text{(above floor)} \\
& \quad \mathbf{f}_i(t) = 0 && \text{(no force in air)}
\end{aligned}
$$

Fig. 3. Physics-based trajectory optimization formulation. Please see text for details.

are derived from video, at discrete steps over the entire duration T:

$$
\begin{aligned}
E_{data}(t) = w_r ||\mathbf{r}(t) - \overline{\mathbf{r}}(t)||^2 + w_\theta ||\boldsymbol{\theta}(t) - \overline{\boldsymbol{\theta}}(t)||^2 \\
+ w_p \sum_{i=1}^{4} ||\mathbf{p}_i(t) - \overline{\mathbf{p}}_i(t)||^2
\end{aligned} \tag{1}
$$

$$
E_{dur} = w_d \sum_{i=1}^{4} \sum_{j=1}^{n_i} (T_{i,j} - \bar{T}_{i,j})^2 \tag{2}
$$

We weigh these terms with $w_d = 0.1$, $w_r = 0.4$, $w_\theta = 1.7$, $w_p = 0.3$.

The remaining objective terms are regularizers that prefer small velocities and accelerations resulting in a smoother optimal trajectory:

$$
E_{vel}(t) = \gamma_r ||\dot{\mathbf{r}}(t)||^2 + \gamma_\theta ||\dot{\boldsymbol{\theta}}(t)||^2 + \gamma_p \sum_{i=1}^{4} ||\dot{\mathbf{p}}_i(t)||^2 \tag{3}
$$

$$
E_{acc}(t) = \beta_r ||\ddot{\mathbf{r}}(t)||^2 + \beta_\theta ||\ddot{\boldsymbol{\theta}}(t)||^2 + \beta_p \sum_{i=1}^{4} ||\ddot{\mathbf{p}}_i(t)||^2 \tag{4}
$$

with $\gamma_r = \gamma_\theta = 10^{-3}$, $\gamma_p = 0.1$ and $\beta_r = \beta_\theta = \beta_p = 10^{-4}$.

Constraints. The first set of constraints strictly enforce valid rigid body mechanics, including linear and angular momentum. This enforces important properties

of motion, for example, during flight the COM must follow a parabolic arc according to Newton's Second Law. During contact, the body motion acceleration is limited by the possible contact forces e.g., one cannot walk at a 45° lean.

At each timestep, we use the world-frame inertia tensor $\mathbf{I}_w(t)$ computed from the input $\mathbf{I}_b(t)$ and the current orientation $\boldsymbol{\theta}(t)$. This assumes that the final output poses will not be dramatically different from those of the input: a reasonable assumption since our optimization does not operate on upper-body joints and changes in feet positioning are typically small (though perceptually important). We found that using a constant inertia tensor (as in Winkler et al. [46]) made convergence difficult to achieve. The gravity vector is $\mathbf{g} = -9.8\hat{\mathbf{n}}$, where $\hat{\mathbf{n}}$ is the ground normal. The angular velocity $\boldsymbol{\omega}$ is a function of the rotations $\boldsymbol{\theta}$ [46].

The contact forces are constrained to ensure that they push away from the floor but are not greater than $f_{max} = 1000$ N in the normal direction. With 4 feet joints, this allows 4000 N of normal contact force: about the magnitude that a 100 kg (220 lb) person would produce for extremely dynamic dancing motion [23]. We assume no feet slipping during contact, so forces must also remain in a friction pyramid defined by friction coefficient $\mu = 0.5$ and floor plane tangents $\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2$. Lastly, forces should be zero at any foot joint not in contact.

Foot contact is enforced through constraints. When a foot joint is in contact, it should be stationary (no-slip) and at floor height h_{floor}. When not in contact, feet should always be on or above the ground. This avoids feet skating and penetration with the ground.

In order to make the optimized motion valid for a humanoid skeleton, the toe and heel of each foot should maintain a constant distance of ℓ_{foot}. Finally, no foot joint should be farther from its corresponding hip than the length of the leg ℓ_{leg}. The hip position $\mathbf{p}_{hip,i}(t)$ is computed from the COM orientation at that time based on the hip offset in the skeleton detailed in Section 3.3.

Optimization Algorithm. We optimize with IPOPT [43], a nonlinear interior point optimizer, using analytical derivatives. We perform the optimization in stages: we first use fixed contact phases and no dynamics constraints to fit the polynomial representation for COM and feet position variables as close as possible to the input motion. Next, we add in dynamics constraints to find a physically valid motion, and finally we allow contact phase durations to be optimized to further refine the motion if possible.

Following the optimization, we compute a full-body motion from the physically-valid COM and foot joint positions using Inverse Kinematics (IK) on a desired skeleton $\mathbf{S}_{tgt}$ (see supplementary Appendix C).

3.2 Learning to Estimate Contacts

Before performing our physics-based optimization, we need to infer when the subject's feet are in contact with the ground, given an input video. These contacts are a target for the physics optimization objective and their accuracy is crucial to its success. To do so, we train a network that, for each video frame, classifies whether the toe and heel of each foot are in contact with the ground.

The main challenge is to construct a suitable dataset and feature representation. There is currently no publicly-available dataset of videos with labeled foot contacts and a wide variety of dynamic motions. Manually labeling a large, varied dataset would be difficult and costly. Instead, we generate synthetic data using motion capture (mocap) sequences. We automatically label contacts in the mocap and then use 2D joint position features from OpenPose [4] as input to our model, rather than image features from the raw rendered video frames. This allows us to train on synthetic data but then apply the model to real inputs.

Dataset. To construct our dataset, we obtained 65 mocap sequences for the 13 most human-like characters from `www.mixamo.com`, ranging from dynamic dancing motions to idling. Our set contains a diverse range of mocap sequences, retargeted to a variety of animated characters. At each time of each motion sequence, four possible contacts are automatically labeled by a heuristic: a toe or heel joint is considered to be in contact when (i) it has moved less than 2 cm from the previous time, and (ii) it is within 5 cm from the known ground plane. Although more sophisticated labeling [17,25] could be used, we found this approach sufficiently accurate to learn a model for the videos we evaluated on.

We render these motions (see Figure 5(c)) on their rigged characters with motion blur, randomized camera viewpoint, lighting, and floor texture. For each sequence, we render two views, resulting in over 100k frames of video with labeled contacts and 2D and 3D poses. Finally, we run a 2D pose estimation algorithm, OpenPose [4], to obtain the 2D skeleton which our model uses as input.

Model and Training. The classification problem is to map from 2D pose in each frame to the four contact labels for the feet joints. As we demonstrate in Section 4.1, simple heuristics based on 2D velocity do not accurately label contacts due to the ambiguities of 3D projection and noise.

For a given time t, our labeling neural network takes as input the 2D poses over a temporal window of duration w centered on the target frame at t. The 2D joint positions over the window are normalized to place the root position of the target frame at $(0, 0)$, resulting in relative position and velocity. We set $w = 9$ video frames and use the 13 lower-body joint positions as shown in Figure 4. Additionally, the OpenPose confidence c for each joint position is included as input. Hence, the input to the network is a vector of (x, y, c) values of dimension $3 * 13 * 9 = 351$. The model outputs four contact labels (left/right toe, left/right heel) for a window of 5 frames centered around the target. At test time, we use majority voting at overlapping predictions to smooth labels across time.

We use a five-layer multilayer perceptron (MLP) (sizes 1024, 512, 128, 32, 20) with ReLU non-linearities [33]. We train the network entirely on our synthetic dataset split 80/10/10 for train/validation/test based on motions per character, i.e., no motion will be in both train and test on the same character, but a training motion may appear in the test set retargeted to a different character. Although 3D motions may be similar in train and test, the resulting 2D motions (the network input) will be very different after projecting to differing camera viewpoints. The network is trained using a standard binary cross-entropy loss.

3.3 Kinematic Initialization

Along with contact labels, our physics-based optimization requires as input a ground plane and initial trajectories for the COM, feet, and inertia tensor. In order to obtain these, we compute an initial 3D full-body motion from video. Since this stage uses standard elements, e.g., [12], we summarize the algorithm here, and provide full details in Appendix B.

First, Monocular Total Capture [47] (MTC) is applied to the input video to obtain an initial noisy 3D pose estimate for each frame. Although MTC accounts for motion through a texture-based refinement step, the output still contains a number of artifacts (Figure 1) that make it unsuitable for direct use in our physics optimization. Instead, we initialize a skeleton $\mathbf{S}_{src}$ containing 28 body joints from the MTC input poses, and then use a kinematic optimization to solve for an optimal root translation and joint angles over time, along with parameters of the ground plane. The objective for this optimization contains terms to smooth the motion, ensure feet are stationary and on the ground when in contact, and to stay close to both the 2D OpenPose and 3D MTC pose inputs.

We first optimize so that the feet are stationary, but not at a consistent height. Next, we use a robust regression to find the ground plane which best fits the foot joint contact positions. Finally, we continue the optimization to ensure all feet are on this ground plane when in contact.

The full-body output motion of the kinematic optimization is used to extract inputs for the physics optimization. Using a predefined body mass (73 kg for all experiments) and distribution [26], we compute the COM and inertia tensor trajectories. We use the orientation about the root joint as the COM orientation, and the feet joint positions are used directly.

4 Results

Here we present extensive qualitative and quantitative evaluations of our contact estimation and motion optimization.

4.1 Contact Estimation

We evaluate our learned contact estimation method and compare to baselines on the synthetic test set (78 videos) and 9 real videos with manually-labeled foot contacts. The real videos contain dynamic dancing motions and include 700 labeled frames in total. In Table 1, we report classification accuracy for our method and numerous baselines.

We compare to using a velocity heuristic on foot joints, as described in Section 3.2, for both the 2D OpenPose and 3D MTC estimations. We also compare to using different subsets of joint positions. Our MLP using all lower-body joints is substantially more accurate on both synthetic and real videos than all baselines. Using upper-body joints down to the knees yields surprisingly good results.

In order to test the benefit of contact estimation, we compared our full optimization pipeline on the synthetic test set using network-predicted contacts

Table 1. Classification accuracy of estimating foot contacts from video. Left: comparison to various baselines, Right: ablations using subsets of joints as input features.

Baseline Method	Synthetic Accuracy	Real Accuracy	MLP Input Joints	Synthetic Accuracy	Real Accuracy
Random	0.507	0.480	Upper down to hips	0.919	0.692
Always Contact	0.677	0.647	Upper down to knees	0.935	0.865
2D Velocity	0.853	0.867	Lower up to ankles	0.933	0.923
3D Velocity	0.818	0.875	**Lower up to hips**	**0.941**	**0.935**

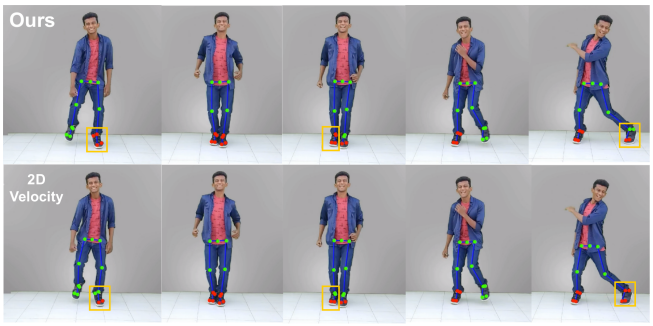

Fig. 4. Foot contact estimation on a video using our learned model compared to a 2D velocity heuristic. All visualized joints are used as input to the network which outputs four contact labels (left toes, left heel, right toes, right heel). Red joints are labeled as contacting. Key differences are shown with orange boxes.

versus contacts predicted using a velocity heuristic on the 3D joints from MTC input. Optimization using network-predicted contacts converged for 94.9% of the test set videos, compared to 69.2% for the velocity heuristic. This illustrates how contact prediction is crucial to the success of motion optimization.

Qualitative results of our contact estimation method are shown in Figure 4. Our method is compared to the 2D velocity baseline which has difficulty for planted feet when detections are noisy, and often labels contacts for joints that are stationary but off the ground (e.g. heels).

4.2 Qualitative Motion Evaluation

Our method provides key qualitative improvements over prior kinematic approaches. We urge the reader to **view the supplementary video** in order to fully appreciate the generated motions. For qualitative evaluation, we demonstrate animation from video by retargeting captured motion to a computer-animated character. Given a target skeleton $\mathbf{S}_{tgt}$ for a character, we insert an IK retargeting step following the kinematic optimization as shown in Figure 2 (see Appendix D for details), allowing us to perform the usual physics-based

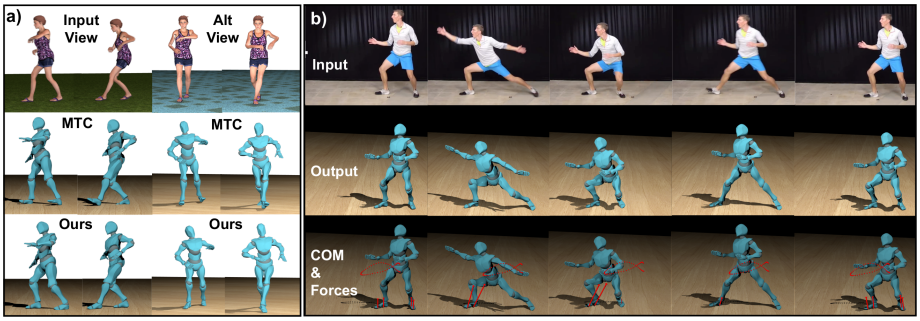

Fig. 5. Qualitative results on synthetic and real data. a) results on a synthetic test video with a ground truth alternate view. Two nearby frames are shown for the input video and the alternate view. We fix penetration, floating and leaning prevalent in our method's input from MTC. b) dynamic exercise video (top) and the output full-body motion (middle) and optimized COM trajectory and contact forces (bottom).

optimization on this new skeleton. We use the same IK procedure to compare to MTC results directly targeted to the character.

Figure 1 shows that our proposed method fixes artifacts such as foot floating (top row), foot penetrations (middle), and unnatural leaning (bottom). Figure 5(a) shows frames comparing the MTC input to our final result on a synthetic video for which we have a ground truth alternate view. For this example only, we use the true ground plane as input to our method for a fair comparison (see Section 4.3). From the input view, our method fixes feet floating and penetration. From the first frame of the alternate view, we see that the MTC pose is in fact extremely unstable, leaning backward while balancing on its heels; our method has placed the contacting feet in a stable position to support the pose, better matching the true motion.

Figure 5(b) shows additional qualitative results on a real video. We faithfully reconstruct dynamic motion with complex contact patterns in a physically accurate way. The bottom row shows the outputs of the physics-based optimization stage of our method at multiple frames: the COM trajectory and contact forces at the heel and toe of each foot.

4.3 Quantitative Motion Evaluation

Quantitative evaluation of high-quality motion estimation presents a significant challenge. Recent pose estimation work evaluates average positional errors of joints in the local body frame up to various global alignment methods [35]. However, those pose errors can be misleading: a motion can be pose-wise close to ground truth on average, but produce extremely implausible dynamics, including vibrating positions and extreme body lean. These errors can be perceptually objectionable when remapping the motion onto an animated character, and prevent the use of inferred dynamics for downstream vision tasks.

Therefore, we propose to use a set of metrics inspired by the biomechanics literature [2,15,19], namely, to evaluate *plausibility* of physical quantities based on known properties of human motion.

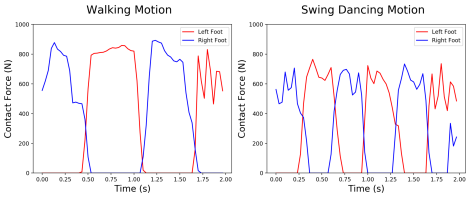

Fig. 6. Contact forces from our physics-based optimization for a walking and dancing motion. The net contact forces around 1000 N are 140% of the assumed body weight (73 kg), a reasonable estimate compared to prior force plate data [2].

We use two baselines: MTC, which is the state-of-the-art for pose estimation, and our kinematic-only initialization (Section 3.3), which transforms the MTC input to align with the estimated contacts from Section 3.2. We run each method on the synthetic test set of 78 videos. For these quantitative evaluations only, we use the ground truth floor plane as input to our method to ensure a fair comparison. Note that our method does not *need* the ground truth floor, but using it ensures a proper evaluation of our primary contributions rather than that of the floor fitting procedure, which is highly dependent on the quality of MTC input (see Appendix E for quantitative results using the estimated floor).

Dynamics Metrics. To evaluate dynamic plausibility, we estimate net ground reaction forces (GRF), defined as $\mathbf{f}_{GRF}(t) = \sum_i \mathbf{f}_i(t)$. For our full pipeline, we use the physics-based optimized GRFs which we compare to implied forces from the kinematic-only initialization and MTC input. In order to infer the GRFs implied by the kinematic optimization and MTC, we estimate the COM trajectory of the motion using the same mass and distribution as for our physics-based optimization (73 kg). We then approximate the acceleration at each time step and solve for the implied GRFs for all time steps (both in contact and flight).

We assess plausibility using GRFs measured in force plate studies, e.g., [2,15,39]. For walking, GRFs typically reach 80% of body weight; for a dance jump, GRFs can reach up to about 400% of body weight [23]. Since we do not know body weights of our subjects, we use a conservative range of 50kg–80kg for evaluation. Figure 6 shows the optimized GRFs produced by our method for a walking and swing dancing motion. The peak GRFs produced by our method match the data: for the walking motion, 115–184% of body weight, and 127–204% for dancing. In contrast, the kinematic-only GRFs are 319–510% (walking) and 765–1223% (dancing); these are implausibly high, a consequence of noisy and unrealistic joint accelerations.

We also measure GRF plausibility across the whole test set (Table 2(left)). GRF values are measured as a percentage of the GRF exerted by an idle 73 kg person. On average, our estimate is within 1% of the idle force, while the kinematic motion implies GRFs as if the person were 24.4% heavier. Similarly, the peak force of the kinematic motion is equivalent to the subject carrying an extra 830 kg of weight, compared to only 174 kg after physics optimization. The Max GRF for MTC is even less plausible, as the COM motion is jittery

Table 2. Physical plausibility evaluation on synthetic test set. *Mean/Max GRF* are contact forces as a proportion of body weight; see text for discussion of plausible values. *Ballistic GRF* are unexplained forces during flight; smaller values are better. Foot position metrics measure the percentage of frames containing typical foot contact errors per joint; smaller values are better.

	Dynamics (Contact forces)			Kinematics (Foot positions)		
Method	**Mean GRF**	**Max GRF**	**Ballistic GRF**	**Floating**	**Penetration**	**Skate**
MTC [47]	143.0%	9055.3%	115.6%	58.7%	21.1%	16.8%
Kinematics (ours)	124.4%	1237.5%	255.2%	**2.3%**	2.8%	**1.6%**
Physics (ours)	**99.0%**	**338.6%**	**0.0%**	8.2%	**0.3%**	3.6%

before smoothing during kinematic and dynamics optimization. *Ballistic GRF* measures the median GRF on the COM when no feet joints should be in contact according to ground truth labels. The GRF should be exactly 0%, meaning there are no contact forces and only gravity acts on the COM; the kinematic method obtains results of 255%, as if the subject were wearing a powerful jet pack.

Kinematics Metrics. We consider three kinematic measures of plausibility (Table 2(right)). These metrics evaluate accuracy of foot contact measurements. Specifically, given ground truth labels of foot contact we compute instances of foot *Floating*, *Penetration*, and *Skate* for heel and toe joints. *Floating* is the fraction of foot joints more than 3 cm off the ground when they should be in contact. *Penetration* is the fraction penetrating the ground more than 3 cm at any time. *Skate* is the fraction moving more than 2 cm when in contact.

After our kinematics initialization, the scores on these metrics are best (lower is better for all metrics) and degrade slightly after adding physics. This is due to the IK step which produces full-body motion following the physics-based optimization. Both the kinematic and physics optimization results substantially outperform MTC, which is rarely at a consistent foot height.

Positional Metrics. For completeness, we evaluate the 3D pose output of our method on variations of standard positional metrics. Results are shown in Table 3. In addition to our synthetic test set, we evaluate on all walking sequences from the training split of HumanEva-I [41] using the known ground plane as input. We measure the mean **global** per-joint position error (mm) for ankle and toe joints (*Feet* in Table 3) and over all joints (*Body*). We also report the error after aligning the root joint of only the first frame of each sequence to the ground truth skeleton (*Body-Align 1*), essentially removing any spurious constant offset from the predicted trajectory. Note that this differs from the common practice of aligning the roots at every frame, since this would negate the effect of our trajectory optimization and thus does not provide an informative performance measure. The errors between all methods are comparable, showing at most a difference of 5 cm which is very small considering global joint position. Though

Table 3. Pose evaluation on synthetic and HumanEva-I walking datasets. We measure mean global per-joint 3D position error (no alignment) for feet and full-body joints. For full-body joints, we also report errors after root alignment on only the first frame of each sequence. We remain competitive while providing key physical improvements.

	Synthetic Data			HumanEva-I Walking		
Method	**Feet**	**Body**	**Body-Align 1**	**Feet**	**Body**	**Body-Align 1**
MTC [47]	581.095	**560.090**	**277.215**	511.59	532.286	**402.749**
Kinematics (ours)	573.097	562.356	281.044	**496.671**	525.332	407.869
Physics (ours)	**571.804**	573.803	323.232	508.744	**499.771**	421.931

the goal of our method is to improve physical plausibility, it does not negatively affect the pose on these standard measures.

5 Discussion

Contributions. The method described in this paper estimates physically-valid motions from initial kinematic pose estimates. As we show, this produces motions that are visually and physically much more plausible than the state-of-the-art methods. We show results on retargeting to characters, but it could also be used for further vision tasks that would benefit from dynamical properties of motion.

Estimating accurate human motion entails numerous challenges, and we have focused on one crucial sub-problem. There are several other important unknowns in this space, such as motion for partially-occluded individuals, and ground plane position. Each of these problems and the limitations discussed below are an enormous challenge in their own right and are therefore reserved for future work. However, we believe that the ideas in this work could contribute to solving these problems and open multiple avenues for future exploration.

Limitations. We make a number of assumptions to keep the problem manageable, all of which can be relaxed in future work: we assume that feet are unoccluded, there is a single ground plane, the subject is not interacting with other objects, and we do not handle contact from other body parts like knees or hands. These assumptions are permissible for the character animation from video mocap application, but should be considered in a general motion estimation approach. Our optimization is expensive. For a 2 second (60 frame) video clip, the physical optimization usually takes from 30 minutes to 1 hour. This runtime is due primarily to the adapted implementation from prior work [46] being ill-suited for the increased size and complexity of human motion optimization. We expect a specialized solver and optimized implementation to speed up execution.

Acknowledgments. This work was in part supported by NSF grant IIS-1763268, grants from the Samsung GRO program and the Stanford SAIL Toyota Research Center, and a gift from Adobe Corporation. We thank the following YouTube channels that contributed video data: Dance FreaX, Dancercise Studio, Fencer's Edge, MihranTV, DANCE TUTORIALS, Deepak Tulsyan, Gibson Moraes, and pigmie.

References

1. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In: European Conference on Computer Vision (ECCV). pp. 561–578 (2016)
2. Brubaker, M.A., Sigal, L., Fleet, D.J.: Estimating contact dynamics. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2389–2396 (2009)
3. Brubaker, M.A., Fleet, D.J., Hertzmann, A.: Physics-based person tracking using the anthropomorphic walker. International Journal of Computer Vision **87**(1), 140–155 (2010)
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
5. Carpentier, J., Mansard, N.: Multicontact locomotion of legged robots. IEEE Transactions on Robotics **34**(6), 1441–1460 (2018)
6. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: IEEE International Conference on Computer Vision (ICCV) (2019)
7. Chen, Y., Huang, S., Yuan, T., Qi, S., Zhu, Y., Zhu, S.C.: Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In: The IEEE International Conference on Computer Vision (ICCV). pp. 8648–8657 (2019)
8. Choi, K.J., Ko, H.S.: On-line motion retargetting. In: Pacific Conference on Computer Graphics and Applications. pp. 32– (1999)
9. Dai, H., Valenzuela, A., Tedrake, R.: Whole-body motion planning with centroidal dynamics and full kinematics. In: IEEE-RAS International Conference on Humanoid Robots. pp. 295–302 (2014)
10. Fang, A.C., Pollard, N.S.: Efficient synthesis of physically valid human motion. ACM Trans. Graph. **22**(3), 417–426 (2003)
11. Forsyth, D.A., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational studies of human motion: Part 1, tracking and motion synthesis. Foundations and Trends in Computer Graphics and Vision **1**(2–3), 77–254 (2006)
12. Gleicher, M.: Retargetting motion to new characters. In: SIGGRAPH. pp. 33–42 (1998)
13. Hassan, M., Choutas, V., Tzionas, D., Black, M.J.: Resolving 3d human pose ambiguities with 3d scene constraints. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2282–2292 (2019)
14. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2961–2969 (2017)
15. Herr, H., Popovic, M.: Angular momentum in human walking. Journal of Experimental Biology **211**(4), 467–481 (2008)
16. Hoyet, L., McDonnell, R., O'Sullivan, C.: Push it real: Perceiving causality in virtual interactions. ACM Trans. Graph. **31**(4), 90:1–90:9 (2012)

17. Ikemoto, L., Arikan, O., Forsyth, D.: Knowing when to put your foot down. In: Symposium on Interactive 3D Graphics and Games (I3D). pp. 49–53 (2006)
18. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(7), 1325–1339 (2014)
19. Jiang, Y., Van Wouwe, T., De Groote, F., Liu, C.K.: Synthesis of biologically realistic human motion using joint torque actuation. ACM Trans. Graph. **38**(4), 72:1–72:12 (2019)
20. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7122–7131 (2018)
21. Kanazawa, A., Zhang, J.Y., Felsen, P., Malik, J.: Learning 3d human dynamics from video. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5614–5623 (2019)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
23. Kulig, K., Fietzer, A.L., Jr., J.M.P.: Ground reaction forces and knee mechanics in the weight acceptance phase of a dance leap take-off and landing. Journal of Sports Sciences **29**(2), 125–131 (2011)
24. de Lasa, M., Mordatch, I., Hertzmann, A.: Feature-based locomotion controllers. In: SIGGRAPH. pp. 131:1–131:10 (2010)
25. Le Callennec, B., Boulic, R.: Robust kinematic constraint detection for motion data. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA). pp. 281–290 (2006)
26. de Leva, P.: Adjustments to zatsiorsky-seluyanov's segment inertia parameters. Journal of Biomechanics **29**(9), 1223 – 1230 (1996)
27. Li, Z., Sedlar, J., Carpentier, J., Laptev, I., Mansard, N., Sivic, J.: Estimating 3d motion and forces of person-object interactions from monocular video. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8640–8649 (2019)
28. Liu, C.K., Hertzmann, A., Popović, Z.: Learning physics-based motion style with nonlinear inverse optimization. ACM Trans. Graph. **24**(3), 1071–1081 (2005)
29. Macchietto, A., Zordan, V., Shelton, C.R.: Momentum control for balance. In: SIGGRAPH. pp. 80:1–80:8 (2009)
30. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D., Theobalt, C.: Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Trans. Graph. **36**(4) (2017)
31. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision (ECCV). pp. 483–499 (2016)
32. Orin, D.E., Goswami, A., Lee, S.H.: Centroidal dynamics of a humanoid robot. Autonomous Robots **35**(2-3), 161–176 (2013)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 8026–8037 (2019)
34. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10975–10985 (2019)

35. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7753–7762 (2019)
36. Peng, X.B., Kanazawa, A., Malik, J., Abbeel, P., Levine, S.: Sfv: Reinforcement learning of physical skills from videos. ACM Trans. Graph. **37**(6), 178:1–178:14 (2018)
37. Popović, Z., Witkin, A.: Physically based motion transformation. In: SIGGRAPH. pp. 11–20 (1999)
38. Reitsma, P.S.A., Pollard, N.S.: Perceptual metrics for character animation: Sensitivity to errors in ballistic motion. ACM Trans. Graph. **22**(3), 537–542 (2003)
39. Robertson, D.G.E., Caldwell, G.E., Hamill, J., Kamen, G., Whittlesey, S.N.: Research Methods in Biomechanics. Human Kinetics (2004)
40. Safonova, A., Hodgins, J.K., Pollard, N.S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In: SIGGRAPH. pp. 514–521. ACM (2004)
41. Sigal, L., Balan, A., Black, M.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International Journal of Computer Vision **87**, 4–27 (2010)
42. Vondrak, M., Sigal, L., Hodgins, J., Jenkins, O.: Video-based 3d motion capture through biped control. ACM Trans. Graph. **31**(4), 27:1–27:12 (2012)
43. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006)
44. Wang, J.M., Hamner, S.R., Delp, S.L., Koltun, V.: Optimizing locomotion controllers using biologically-based actuators and objectives. ACM Trans. Graph. **31**(4) (2012)
45. Wei, X., Chai, J.: Videomocap: Modeling physically realistic human motion from monocular video sequences. In: SIGGRAPH. pp. 42:1–42:10 (2010)
46. Winkler, A.W., Bellicoso, D.C., Hutter, M., Buchli, J.: Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. IEEE Robotics and Automation Letters (RA-L) **3**, 1560–1567 (2018)
47. Xiang, D., Joo, H., Sheikh, Y.: Monocular total capture: Posing face, body, and hands in the wild. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10965–10974 (2019)
48. Xu, W., Chatterjee, A., Zollhöfer, M., Rhodin, H., Mehta, D., Seidel, H.P., Theobalt, C.: Monoperfcap: Human performance capture from monocular video. ACM Trans. Graph. **37**(2), 27:1–27:15 (2018)
49. Zanfir, A., Marinoiu, E., Sminchisescu, C.: Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2148–2157 (2018)
50. Zou, Y., Yang, J., Ceylan, D., Zhang, J., Perazzi, F., Huang, J.B.: Reducing footskate in human motion reconstruction with ground contact constraints. In: The IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 459–468 (2020)

Appendices

A Contact Estimation Details

Here we detail the contact estimation model and the dataset from Section 3.2.

A.1 Synthetic Dataset

Our synthetic dataset was rendered using Blender[3] and includes 13 characters performing 65 different motion capture sequences retargeted to each character taken from `www.mixamo.com`. Each motion is recorded from 2 camera viewpoints resulting in 1690 videos and 101k frames of data. The motions include: samba, swing, and salsa dancing, boxing, football, and baseball actions, walking, and idle poses. Videos are rendered at 1280x720 with motion blur, and are 2 seconds long at 30 fps. Example frames from the dataset are shown in Figure A1. In addition to RGB frames, at each timestep the dataset includes 2D OpenPose [4] detections, the 3D pose in the form of the character's skeleton (skeletons are different for each character, and pose is provided in a .bvh motion capture file), foot contact labels for the heel and toe base of each foot as described in the main paper, and camera parameters.

For each video, many parameters are randomized. The camera is placed at a uniform random distance in $[4.5, 7.5]$ and Gaussian random height with $\mu = 0.9$, and $\sigma = 0.3$ but clamped to be in $[0.3, 1.75]$, all in meters. The camera is placed at a random angle within 90 degrees offset from the front of the character but always looks at roughly character hip height. The camera does not move during the video. Floor texture is randomly chosen from solid colors and 26 other textures with various wood, grass, tile, metal, and carpet. Four lights in the scene are randomly toggled on and off, given random energies, and randomly offset from default positions, resulting in many shadow variations across videos.

A.2 Model Details

We implement our contact estimation MLP (sizes 1024, 512, 128, 32, 20) in PyTorch [33]. All but the last layer are followed by batch normalization, and we use a single dropout layer before the size-128 layer (dropout $p = 0.3$). To train, we optimize the binary cross-entropy loss using Adam [22] with a learning rate of 10^{-4}. We apply an L2 weight decay with a weight of 10^{-4} and use early stopping based on the validation set loss. We scale all 2D joint inputs to be largely between $[-1, 1]$. During training, we also add Gaussian noise to the normalized joints with $\sigma = 0.005$.

Because our network classifies contacts jointly over 5 frames for every target frame (the frame at the center of the window), there are many overlapping predictions at test time. When inferring contacts for an entire video at test time, we first use every frame as a target and then collect votes from overlapping

[3] https://www.blender.org/

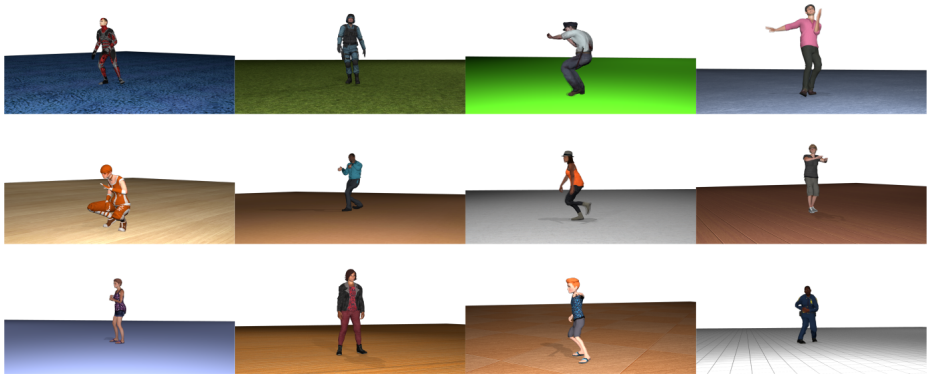

Fig. A1. Example RGB frames from our synthetic dataset for contact estimation learning. The data also contains 2D OpenPose [4] detections, 3D pose in the form of the character's skeleton, and automatically labeled contacts for the toe base and heels.

predictions. A joint is marked in contact at a frame if a majority of the votes for that frame classify it as in contact.

B Kinematic Optimization Details

Here detail the kinematic optimization procedure used to initialize our physics-based optimization. See Section 3.3 for an overview.

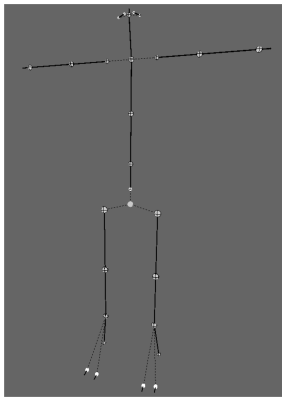

Fig. A2. Skeleton. Bone lengths and pose are initialized from MTC input for each motion sequence before kinematic optimization.

B.1 Inputs and Initialization

Our kinematic optimization takes in J body joints over T timesteps that make up the motion. Specifically, for $j \in J$ and $t \in T$ we have the 2D pose detection from OpenPose $\mathbf{x}_{j,t} \in \mathbb{R}^2$ with confidence $\sigma_{j,t}$, contacts estimated in the previous stage of our pipeline $c_{j,t} \in \{0, 1\}$ where $c_{j,t} = 1$ if joint j is in contact with the ground at time t, and finally the full-body 3D pose from MTC.

We use the MTC input to initialize a custom skeleton, called $\mathbf{S}_{src}$ in the main paper, which contains $J = 28$ joints. In particular, we use the MTC COCO regressor to obtain 19 body joint positions (excluding feet) over the sequence, and vertices on the Adam mesh for the 6 foot joints as in the original MTC paper. We choose these 25 joints in order to use a re-projection error in our optimization objective, as described below. To better map to the character rigs used

during animation retargeting, we additionally use 3 spine joint positions directly from the MTC body model giving the final total of 28 joints. Note, we do not use any hand or face information from MTC. Our skeleton has fixed bone lengths which are determined based on these input 3D joint locations throughout the motion sequence: the median distance between two joints over the entire motion sequence defines the bone length. Our skeleton (before fitting bone lengths to the input data) is visualized in Figure A2.

We normalize the input positions to get the root-relative positions of each joint $\overline{\mathbf{q}}_{j,t} \in \mathbb{R}^3,\ j = 1, \ldots, J,\ t = 1, \ldots, T$ which we will target during optimization, and let the global translation be our initial root translation $\overline{\mathbf{p}}_{root,t}$. All these positions are preprocessed to remove obviously incorrect frames based on OpenPose detection confidence: for the 25 joints with corresponding 2D OpenPose detections (all non-spine joints in our skeleton), if the confidence is below 0.3, then the frame is replaced with a linear interpolation between the closest adjacent frames with sufficient confidence.

Because we optimize for the joint angles of our skeleton (see below), next we must find initial joint angles to match the MTC joint position inputs. We roughly initialize the joint angles of our skeleton by copying those from the MTC body model, and finally perform inverse kinematics (IK) targeting the preprocessed joint positions which results in a reconstruction of the MTC input on our skeleton. We use a Jacobian-based full body IK solver based on [8]. This is the skeleton which is optimized throughout our kinematic initialization.

B.2 Optimization Variables

We optimize over global 3D root translation $\mathbf{p}_{root,t} \in \mathbb{R}^3$ and skeleton joint Euler angles $\boldsymbol{\theta}_{j,t} \in \mathbb{R}^3$ with $j = 1, \ldots, J,\ t = 1, \ldots, T$. We also find ground plane parameters $\hat{\mathbf{n}}, \mathbf{p}_{floor} \in \mathbb{R}^3$ which are the normal vector and some point on the plane. As described below, we do not jointly optimize all of these at once; we do it in stages and fit the floor separately.

B.3 Problem Formulation

We seek to minimize the following objective function:

$$\alpha_{proj}E_{proj}+\alpha_{vel}E_{vel}+\alpha_{ang}E_{ang}+\alpha_{acc}E_{acc}+\alpha_{data}E_{data}+\alpha_{cont}E_{cont}+\alpha_{floor}E_{floor}$$

where the α are constant weights. We use $\alpha_{proj} = 0.5$, $\alpha_{vel} = \alpha_{ang} = 0.1$, $\alpha_{acc} = 0.5$, $\alpha_{data} = 0.3$, and $\alpha_{cont} = \alpha_{floor} = 10$. We now detail each of these energy terms.

Suppose $\mathbf{q}_{j,t} \in \mathbb{R}^3,\ j = 1, \ldots, J,\ t = 1, \ldots, T$ are the current root-relative joint position estimates during optimization which can be calculated using forward kinematics on $\mathbf{S}_{src}$ with the current joint angles $\boldsymbol{\theta}_{j,t}$. Then our energy terms are defined as follows.

– *2D Re-projection Error*: minimizes deviation of joints from corresponding OpenPose detections, weighted by detection confidence

$$E_{proj} = \sum_{t=1}^{T} \sum_{j=1}^{J} \sigma_{j,t} ||\Pi(\mathbf{q}_{j,t} + \mathbf{p}_{root,t}) - \mathbf{x}_{j,t}||^2 \tag{B1}$$

where Π is the perspective projection parameterized by focal length f (assumed to be 2000) and $[c_x, c_y]$.

– *Velocity Smoothing*: minimizes change in joint positions and angles over time

$$E_{vel} = \sum_{t=1}^{T-1} \sum_{j=1}^{J} ||\mathbf{q}_{j,t+1} - \mathbf{q}_{j,t}||^2 \tag{B2}$$

$$E_{ang} = \sum_{t=1}^{T-1} \sum_{j=1}^{J} ||\boldsymbol{\theta}_{j,t+1} - \boldsymbol{\theta}_{j,t}||^2. \tag{B3}$$

– *Linear Acceleration Smoothing*: minimizes change in joint linear velocity over time

$$E_{acc} = \sum_{t=1}^{T-2} \sum_{j=1}^{J} ||(\mathbf{q}_{j,t+2} - \mathbf{q}_{j,t+1}) - (\mathbf{q}_{j,t+1} - \mathbf{q}_{j,t})||^2. \tag{B4}$$

– *3D Data Error*: minimizes deviation from 3D MTC joint initialization

$$E_{data} = \sum_{t=1}^{T} \sum_{j=1}^{J} ||\mathbf{q}_{j,t} - \overline{\mathbf{q}}_{j,t}||^2. \tag{B5}$$

– *Contact Velocity Error*: encourages feet joints (toes and heels) to be stationary when labeled as in contact

$$E_{cont} = \sum_{t=1}^{T-1} \sum_{j \in J_F} ||c_{j,t} \left((\mathbf{q}_{j,t+1} + \mathbf{p}_{root,t+1}) - (\mathbf{q}_{j,t} + \mathbf{p}_{root,t}) \right)||^2. \tag{B6}$$

where J_F is the set of foot joints.

– *Contact Position Error*: encourages toe and heel joints to be on the ground plane when labeled as in contact

$$E_{floor} = \sum_{t=1}^{T} \sum_{j \in J_F} ||c_{j,t} \left(\hat{\mathbf{n}} \cdot (\mathbf{q}_{j,t} + \mathbf{p}_{root,t} - \mathbf{p}_{floor}) \right)||^2. \tag{B7}$$

B.4 Optimization Algorithm

We perform this optimization in three main stages. First, we enforce all objectives *except* the contact position error and solve only for skeleton root position and joint angles (no floor parameters). Next, we use a robust Huber regression

to find the floor plane that best matches the foot joint contact positions and reject outliers, i.e., joints labeled as in contact when they are far from the ground. Outlier contacts are re-labeled as non-contacts for all subsequent processing. Finally, we repeat the full-body optimization, now enabling the contact position objective to ensure feet are on the ground plane. We optimize using the Trust Region Reflective algorithm with analytical derivatives.

B.5 Extracting Inputs for Physics-Based Optimization

From the full-body output of this kinematic optimization, we need to extract inputs for the physics-based optimization (Section 3.1). To get the COM targets $\overline{\mathbf{r}}(t) \in \mathbb{R}^3$, we treat each body part as a point with a pre-defined mass [26]. This also allows the calculation of the body-frame inertia tensor at each time step $\mathbf{I}_b(t) \in \mathbb{R}^{3\times3}$ which is used to enforce dynamics constraints. Unless otherwise noted, we assume a body mass of 73 kg for the character. We use the orientation about the root joint as the COM orientation $\overline{\boldsymbol{\theta}}(t) \in \mathbb{R}^3$ and the feet joint positions $\overline{\mathbf{p}}_{1:4}(t) \in \mathbb{R}^3$ are directly taken from the full-body motion.

C Physics-Based Optimization Details

Here we detail the physics-based trajectory optimization from Section 3.1.

C.1 Polynomial Parameterization

COM position and orientation, foot positions during flight, and contact forces during stance are parameterized by a sequence of cubic polynomials as done in Winkler et al. [46]. These polynomials use a Hermite parameterization: we do not optimize over the polynomial coefficients directly, rather the duration, starting and ending positions, and boundary velocities.

The COM position and orientation use one polynomial every 0.1 s. Feet positions and forces always use at least 6 polynomials per phase, which we found necessary to accurately produce extremely dynamic motions. We adaptively add polynomials depending on the length of the phase. If the phase is longer than 2 s, additional polynomials are added commensurately. Foot positions during stance are a single constant value and contact forces during flight are constant 0 value. This ensures that the no slip and no force during flight constraints are met.

Please see Winkler et al. [46] for a more in-depth discussion of the polynomial parameterization along with the contact phase duration parameterization.

C.2 Constraint Parameters

Though the optimization variables are continuous polynomials, objective energies and constraints are enforced at discrete intervals. Leg and foot kinematic constraints are enforced at 0.08 s intervals, the above floor constraint at 0.1 s intervals, and dynamics constraints are enforced every 0.1 s. In practice, the

velocity boundary constraints try to match the *mean* initial velocity over the first(last) 5 frames to make it more robust to noisy motion.

Objective terms, including smoothing, are enforced at every step for which we have input data. For example, the synthetic dataset at 30 fps will provide an objective term at (1/30) s intervals.

C.3 Contact Timing Optimization

As explained in Section 3.1, our physics optimization is done in stages such that contact phase durations are not optimized until the very last stage. We found that allowing these durations to be optimized along with dynamics does not always result in a better solution as it is an inherently harder and less stable optimization. Therefore, in the presented results we use the better of the two solutions: either the solution using fixed input contact timings (from our neural network) or the solution after subsequently allowing phase durations to change, if the motion is improved.

C.4 Full-Body Output

Following the physics-based optimization, we must compute a full-body motion from the physically-valid COM and foot joint positions using IK. For the upper body (including the root), we calculate the offset of each joint from the COM in the input motion to the physics optimization, and use this offset added to the new optimal COM as the joint targets during IK. This means the upper-body motion will be essentially identical to the result of the kinematic optimization (though the posture may improve due to the new COM position). For the lower body, we target the toe and heel joints directly to the physically optimized output and let the remainder of the joints (i.e., ankles, knees, and hips) result from IK, which can be drastically different from the input. We use the same IK algorithm as in Appendix B.1.

D Retargeting to a New Character

In many cases, we wish to retarget the estimated motion to a new animated character mesh. We do this in the main paper in Section 4.2 for qualitative evaluation. One could apply physics-based motion retargeting methods to the output motion after an IK retargeting procedure, e.g., [37]. However, we avoid this extra step by directly performing our physics-based optimization on the target character skeleton.

Given a target skeleton $\mathbf{S}_{tgt}$, we insert an additional retargeting step following the kinematic optimization (see Figure 2). Namely, we uniformly scale $\mathbf{S}_{src}$ to the approximate size of our target skeleton, and then perform an IK optimization based on a predefined joint mapping to recover joint angles for $\mathbf{S}_{tgt}$. Then, the subsequent physics-based optimization and full-body upgrade are performed with this skeleton replacing $\mathbf{S}_{src}$. We use the same IK algorithm as in Appendix B.1.

Table E1. Precision, recall, and F1 Score (*Prec/Rec/F1*) of estimating foot contacts from video. Left: comparison to various baselines, Right: ablations using subsets of joints as features. Supplements Table 1.

Baseline Method	Synthetic Prec / Rec / F1	Real Prec / Rec / F1	MLP Input Joints	Synthetic Prec / Rec / F1	Real Prec / Rec / F1
Random	0.679 / 0.516 / 0.586	0.627 / 0.487 / 0.548	Upper to hips	0.940 / 0.941 / 0.940	0.728 / 0.837 / 0.779
Always Contact	0.677 / 1.000 / 0.808	0.647 / 1.000 / 0.786	Upper to knees	**0.958** / 0.946 / 0.952	0.926 / 0.859 / 0.892
2D Velocity	0.861 / 0.933 / 0.896	0.922 / 0.868 / 0.894	Lower to ankles	0.933 / 0.971 / 0.952	**0.963** / 0.916 / 0.939
3D Velocity	0.858 / 0.876 / 0.867	0.920 / 0.884 / 0.902	Lower to hips	0.941 / **0.973** / **0.957**	0.956 / **0.943** / **0.949**

Note for qualitative comparison to MTC, we perform a very similar procedure: we first fit our skeleton to the raw MTC input, similar to Appendix B.1 but without the preprocessing, and then perform the IK retargeting as described in this section. This provides a stronger baseline than a naive approach like directly copying joint angles from MTC to $\mathbf{S}_{tgt}$.

E Evaluation Details and Additional Results

Here we include additional results and details for evaluations.

E.1 Contact Estimation

Main results for contact estimation are presented in Section 4.1. Table E1 supplements the main results with the precision, recall, and F1 score of each method. These give additional insight compared to accuracy since data labels are slightly imbalanced (more in-contact frames than no-contact).

Table E2 shows an ablation study between different input and output window size combinations for our network. *Input Window* is the number of frames of 2D lower-body joints given to the network, and *Prediction Window* is the number of frames for which the network outputs foot contact classifications. We use an input window $w = 9$ and prediction window of 5 in our experiments since it achieves the best accuracy on real videos as shown in the table. In general, there is not a clear trend in *Prediction Window* size, but as the *Input Window* size increases, so does accuracy on the real dataset.

Table E2. Ablation study of input and output window sizes for learned contact estimation. Classification accuracy for many different combinations are shown.

Input Window	Prediction Window	Synthetic Accuracy	Real Accuracy
3	3	0.931	0.919
5	3	0.933	0.913
5	5	0.943	0.906
7	3	0.936	0.922
7	5	0.941	0.923
7	7	0.943	0.926
9	3	0.936	0.905
9	5	0.941	**0.935**
9	7	0.942	0.921
9	9	**0.946**	0.927

Figure E1 shows the accuracy of contact estimations over the entire prediction window of 5 frames on the synthetic test set. Though the target frame in this case is frame index 2, predictions on the off-target frames degrade only slightly and are still very accurate since the input windows is 9 frames. This motivates the use of the majority voting scheme at inference time.

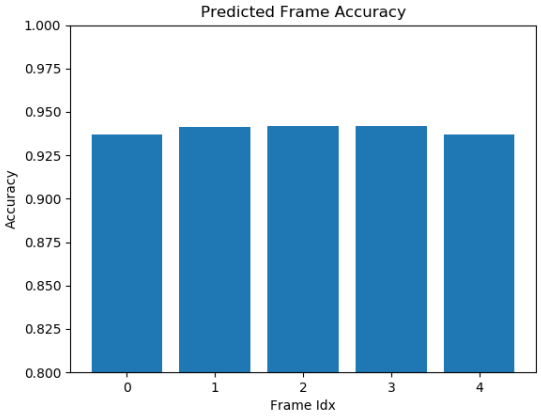

Fig. E1. Contact estimation classification accuracy for all frames in the 5-frame output window on the synthetic test set (given 9 frames as input). The center frame index 2 is the target frame, however off-target contact predictions are still accurate.

E.2 Qualitative Motion Evaluation

For extensive qualitative evaluation, see the supplementary video. For real data, we use videos from: publicly available datasets [6,36], YouTube videos that are licensed under Creative Commons or with permission from the content creators to be used in this publication, and licensed stock footage.

E.3 Quantitative Motion Evaluation

Primary quantitative results for motion reconstruction are presented in Section 4.3. These quantitative evaluations make use of the ground truth floor plane as input. Note that our method does not *need* the ground truth floor: our floor fitting procedure works well as demonstrated in all qualitative results on live action monocular videos. However, we performed quantitative evaluations on data that contains many cases of movement directly towards or away from the camera: a challenging case for MTC, which results in noisy feet joints as input to our method causing a poor floor fit. This makes optimization difficult and interferes with evaluating our primary contributions.

However, for completeness, here we include quantitative results using the floor fitting procedure (rather than taking the ground truth floor as input) on the synthetic test set. Table E3 shows kinematic and dynamics evaluations using the fitted floor while Table E4 shows the pose evaluation. Trends are similar to those in Tables 2 and 3 using the ground truth floor.

E.4 Discussion on Global Pose Estimation

For quantitative evaluations in Section 4.3, we do not compare to other methods in global human motion estimation but instead evaluate ablations of our own method: the kinematics-only version and initialization from MTC.

Table E3. Physical plausibility evaluation using the *estimated floor* on the synthetic test set. Supplements Table 2.

	Dynamics (Contact forces)			Kinematics (Foot positions)		
Method	**Mean GRF**	**Max GRF**	**Ballistic GRF**	**Floating**	**Penetration**	**Skate**
MTC [47]	142.7%	9036.7%	120.8%	19.1%	10.0%	16.5%
Kinematics (ours)	119.7%	1252.4%	103.6%	**1.5%**	1.8%	**1.3%**
Physics (ours)	**98.8%**	**293.2%**	**0.0%**	5.9%	**0.1%**	3.8%

Table E4. Pose evaluation on synthetic test set using the *estimated floor*. Supplements Table 3.

Method	**Feet**	**Body**	**Body-Align 1**
MTC [47]	585.303	**565.068**	**277.296**
Kinematics (ours)	582.400	565.097	281.416
Physics (ours)	**582.311**	587.627	319.517

The problem of predicting a temporally-consistent global motion (like MTC and this work does) is vastly under-explored so there are few comparable prior works. Many methods do traditional local 3D pose estimation or even predict the global root translation from the camera, but these rarely result in a coherent global motion.

For example, a recent work from Pavlakos et al. [34] called SMPLify-X estimates global camera extrinsics, local pose, and body shape, which gives global motion when applied to video. However, we found that MTC, which uses a temporal tracking procedure, gave better results which motivated its use in initializing our pipeline. Figure E2 shows a fixed side view of results from SMPLify-X and MTC on the same video clip. SMPLify-X is noisy and inconsistent especially in terms of global translation; MTC is much smoother and coherent.

E.5 Pose Estimation Evaluation Details

We quantitatively evaluate pose estimation in Section 4.3. We evaluate on our synthetic test set and HumanEva-I [41] walking sequences. Like many pose estimation benchmarks (e.g. Human3.6M [18]), few motions in HumanEva are dynamic with interesting foot contact patterns. Therefore, we evaluate on a subset containing the walking sequences which meet this criteria.

For the MTC baseline, we measure accuracy based directly on the regressed joints given as input to our method. For our method, we use the estimated joints after the full physics-based motion pipeline on our custom skeleton that is initially fit from the MTC input as described in Appendix B.1.

For the synthetic test set, we measure joint errors with respect to a subset of the known character rig that includes 16 joints: neck, shoulders, elbows, wrists, hips, knees, ankles, and toes (no spine joints). The "Feet" column of Table 3 includes ankle and toe joints only.

On the right side of Table 3, we evaluate methods on the walking sequences from the training split of HumanEva-I [41] (which includes subjects 1, 2, and 3). Following prior work [35], we first split the walking sequences into contiguous chunks by removing corrupted motion capture frames. We then further split these chunks into sequences of roughly 120 frames (about 2 seconds) to use as input to

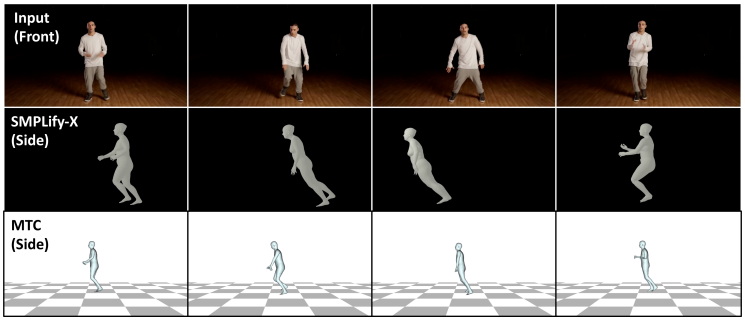

Fig. E2. A fixed side view is shown from SMPLify-X [34] and Monocular Total Capture (MTC) [47]. SMPLify-X gives noisy and inconsistent global motion whereas the tracking refinement of MTC gives smoother results.

Fig. E3. Our method can be applied to multiple characters with varying body masses and mass distributions. From left to right the animated characters are Ybot (body mass 73 kg), Ty (36.5 kg), and Skeleton Zombie (146 kg).

our method. We extract the ground truth floor plane using the camera extrinsics from the dataset and use this as input to our method. Joint errors are measured with respect to an adapted 15-joint skeleton [35] from the HumanEva ground truth motion capture which includes: root, head, neck, shoulders, elbows, wrists, hips, knees, and ankles. The "Feet" column of Table 3 includes ankle joints only.

E.6 Multi-Character Generalization

Following the procedure laid out in Appendix D, our physics-based optimization can be applied to many character skeletons with varying body and mass distributions. Figure E3 shows an example of estimating motion from the same video for three different characters: Ybot, Ty, and Skeleton Zombie. Ybot has a body mass of 73 kg with a typical human mass distribution [26]. Ty is much lighter at 36.5 kg and his distribution is modified such that 40% of his mass is in the head. Skeleton Zombie is much more massive at 146 kg and has 36% of its mass in its arms alone (due to the giant claws). Our physics-based optimization can handle these variations and still accurately recover the motion from the video. Please see the supplementary video for additional examples.