VideoMocap: Modeling Physically Realistic Human Motion from Monocular Video Sequences

Xiaolin Wei*
Texas A&M University

Jinxiang Chai†
Texas A&M University

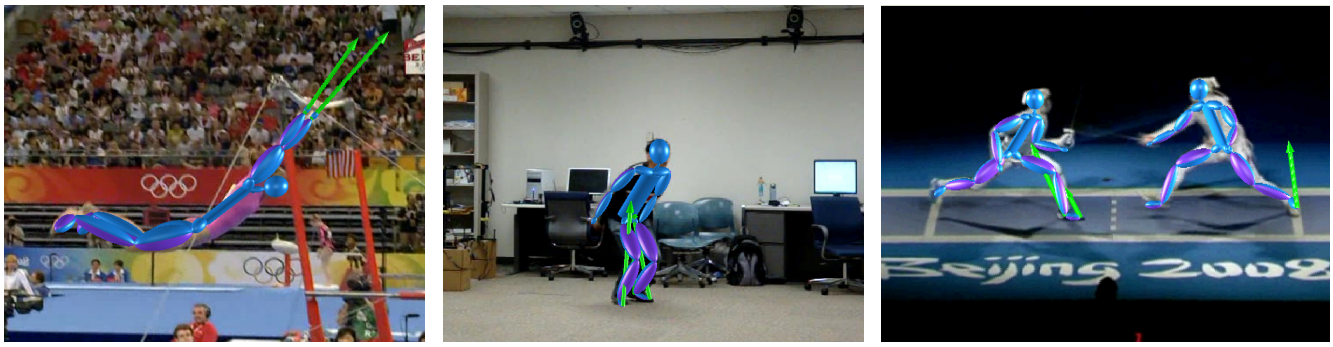


Figure 1: *Modeling physically realistic human motion from uncalibrated monocular video sequences.*

Abstract

This paper presents a video-based motion modeling technique for capturing physically realistic human motion from monocular video sequences. We formulate the video-based motion modeling process in an image-based keyframe animation framework. The system first computes camera parameters, human skeletal size, and a small number of 3D key poses from video and then uses 2D image measurements at intermediate frames to automatically calculate the "in between" poses. During reconstruction, we leverage Newtonian physics, contact constraints, and 2D image measurements to simultaneously reconstruct full-body poses, joint torques, and contact forces. We have demonstrated the power and effectiveness of our system by generating a wide variety of physically realistic human actions from uncalibrated monocular video sequences such as sports video footage.

Keywords: Video-based motion capture, performance animation, physics-based animation, data-driven animation, interactive 3D visual tracking, vision for graphics

1 Introduction

One of the most popular and successful approaches for creating natural-looking human characters is to use motion capture data. Although we have made great strides in using data to model and synthesize human motion in the past decade, current motion capture technologies are often restrictive, cumbersome, and expensive. Optical and magnetic motion capture systems must be operated in carefully calibrated, restrictive lab settings, inhibiting the possibility of acquiring outdoor activities. Inertial or mechanical systems, on the other hand, are not constrained by a fixed capture space, but require the subject to wear cumbersome sensors or confined exoskeletons, reducing the naturalness and quality in the performance.

One way to address these limitations is to use standard video cameras to capture live performances in 3D. The minimal requirement of a single video camera is particularly appealing, as it offers the lowest cost, a simplified setup, and the potential use of legacy sources such as film footage. Graphics and vision researchers have been actively exploring the problem of video-based motion capture for many years, and have made great advances. However, these results are often vulnerable to ambiguities in the video data (*e.g.*, occlusions, cloth deformation, and illumination changes), degeneracies in camera motion, and a lack of discernible features on a human body.

In this paper, we present a video-based motion capture technique for modeling physically realistic 3D human motion from uncalibrated monocular video sequences such as sports video footage (Figure 1). Our technique combines the power of automatic computer vision techniques and physics-based motion modeling techniques to generate 3D human motion with a high degree of physical realism. The use of physics-based dynamics models for video-based motion modeling produces three benefits. First and foremost, it significantly reduces ambiguities in video-based motion modeling, producing more accurate motions that naturally obey the laws of physics. Second, it allows us to properly model interactions with the environment (*e.g.*, ground contact) as well as balance during locomotion. Third, it enables us to compute joint torques and contact forces from input video sequences, a capacity which has not been demonstrated in previous video-based motion modeling work.

We formulate the video-based motion modeling process in an image-based keyframe animation framework. Our system consists of two main steps: interactive 3D keyframe modeling and image-based 3D keyframe interpolation. The user first selects a small set of keyframe images from the input video and annotates each of

*e-mail: xwei@cse.tamu.edu
†e-mail: jchai@cse.tamu.edu

ACM Reference Format
Wei, X., Chai, J. 2010. VideoMocap: Modeling Physically Realistic Human Motion from Monocular Video Sequences. *ACM Trans. Graph. 29*, 4, Article 42 (July 2010), 10 pages. DOI = 10.1145/1778765.1778779 http://doi.acm.org/10.1145/1778765.1778779.

them with a small number of 2D constraints. The system then automatically computes human skeletal size and 3D key poses from the annotated 2D constraints. In the second step, the system tracks 2D image features at intermediate frames and uses them to interpolate 3D motion between the key frames. During reconstruction, we leverage Newtonian physics and 2D image measurement to simultaneously reconstruct full-body poses, joint torques, and contact forces. In addition, our system allows the user to briefly review the result by playing back the interpolated motion and incrementally edit the result at any frame if the reconstructed motion does not precisely match the input video sequence.

We demonstrate the power and effectiveness of this system by modeling a wide variety of human actions from monocular video sequences such as Internet videos and sports footage. We show that our system can model physically realistic motion for highly dynamic motion such as gymnastics, low energy motion such as walking, interaction with environments such as sitting and standing up, and multiple actor interaction such as fencing (Figure 1). We assess the quality of the reconstructed motion by comparing with high quality motion data captured with a full marker set in a twelve-camera optical motion capture system.

2 Background

Our system combines 2D image data and physics-based dynamics models to capture physically realistic human motions from monocular video sequences. We therefore discuss related work in modeling 3D human motion from monocular video sequences as well as physics-based motion modeling.

One way to model 3D human motion from monocular video sequences is model-based motion tracking [Bregler et al. 2004], which initializes a 3D human pose at the starting frame and sequentially updates 3D poses by minimizing the image differences between two consecutive frames. The approach, however, has many restrictions because it often assumes known human skeletons and requires manual initialization of the 3D pose at the first frame. More importantly, the approach is often vulnerable to occlusions, cloth deformation, illumination changes, and a lack of discernible features on human body because 2D image measurements are often noisy and insufficient to determine high-dimensional 3D human movement.

An efficient way to reduce the modeling ambiguities is to utilize kinematic motion priors embedded in prerecorded motion data. Thus far, two different approaches have been taken, including generative approaches [Howe et al. 1999; Pavlović et al. 2000; Sidenbladh et al. 2002; Sminchisescu and Jepson 2004; Chai and Hodgins 2005; Urtasun et al. 2005; Chen and Chai 2009] and discriminative models [Rosales and Sclaroff 2000; Elgammal and Lee 2004; Agarwal and Triggs 2006; Kanaujia and Metaxas 2007]. However, data-driven approaches can only model motions that are similar to training datasets. This significantly limits their application to video-based motion capture. Another limitation is that these approaches do not consider the dynamics that cause motion. When motion data is generalized to achieve new goals, the results are often physically implausible, displaying noticeable visual artifacts such as unbalanced motions, foot sliding, and motion jerkiness.

Several researchers have recently started to employ physics-based models of human motion for video-based motion tracking [Brubaker and Fleet 2008; Vondrak et al. 2008]. For example, Brubaker and Fleet [2008] introduced a low-dimensional biomechanically-inspired model that accounts for human lower-body walking dynamics and used it to track human motion in a recursive Bayesian framework. However, the model, while powerful, is inherently limited to walking motions in 2D. Vondrak and his colleagues [2008] adopted a full-body 3D dynamics model and combined it with motion capture data to constrain the search space for Bayesian motion tracking. Their approach has been shown to be effective for tracking walking and jogging motion. However, it is not clear how this approach can be extended to model the wide variety of human actions reported in our paper because they also used motion capture data to reduce the solution space. Furthermore, their methods are specifically tailored towards online vision applications such as visual surveillance, and thus do not address the range of applications in computer graphics targeted in this work.

Our work builds on the success of physics-based optimization techniques for human motion modeling. Physics-based motion optimization, first introduced to the graphics community by Witkin and Kass [1988], provides a powerful framework for generating human motion from user constraints, physics constraints, and a performance objective that measures the performance of a generated motion. These methods have recently been extended to 3D full-body animation with the help of simplified physical models [Popović and Witkin 1999], reduced dimension subspaces [Safonova et al. 2004], initializations derived from motion capture data [Sulejmanpasic and Popović 2005], and performance objectives optimized from reference motion data [Liu et al. 2005]. Our work shares the use of optimization procedures to model physically realistic motions. It differs in that we utilize physics-based dynamics models to match human motion in video data. The use of image data for physics-based motion modeling ensures that the generated motion is not only physically plausible but also natural-looking.

Our work draws inspiration from systems that utilize user assistance for video-based human motion modeling [Cowley and Taylor 2001; DiFranco et al. 2001; Loy et al. 2004]. Taylor [2000] presented an interactive system to model 3D key poses from 2D images; later the system was extended to video-based motion interpolation [Cowley and Taylor 2001]. However, this interpolation does not utilize image measurements at intermediate frames and therefore requires intensive user interaction to achieve good results. The approach was recently extended by Loy and his colleagues [2004]. They interpolated the motion by minimizing the image projection error while keeping limb lengths constant. DiFranco and his colleagues [2001] used a similar batch-based optimization process to interpolate the motion based on 2D joint trajectories defined at intermediate frames; however, they assumed a known human skeletal model, predefined 3D key poses, and predetermined joint trajectories at intermediate frames.

Our approach is different in that we leverage Newtonian dynamics and image measurements to interpolate the motion. The use of physical constraints for motion modeling not only reduces the modeling ambiguity but also ensures that the reconstructed motion is physically correct. Our system is also more flexible because it assumes unknown skeleton sizes as well as uncalibrated cameras. In addition, the system does not require predefined 2D joint trajectories, which significantly reduces the amount of user intervention needed for video-based motion modeling. Another distinction is that our work incorporates environmental contacts into the video-based motion modeling process and thereby removes noticeable visual artifacts such as foot sliding and ground penetration, which are commonly seen in previous video-based motion modeling systems.

3 Overview

Our system creates physically realistic human motion from uncalibrated monocular video sequences with minimal user interaction. We formulate the video-based motion modeling process in an image-based keyframe animation framework. The system first estimates a small set of 3D key poses and human skeletal sizes with

minimal user interaction, and then interpolates them automatically using physical constraints as well as image measurements at intermediate frames. Here we highlight the issues that are critical for the success of this endeavor and summarize our approach for addressing them.

Interactive 3D keyframe modeling. Our system models 3D keyframe poses using 2D image data, rather than the talents of artists/animators. The first challenge of our system is therefore how to estimate a small set of 3D key poses from 2D image sequences. The problem is challenging because we are dealing with a single moving camera. In addition, neither camera parameters nor human skeletal sizes are known. To address this challenge, we have introduced an efficient algorithm to simultaneously computing 3D key poses, human skeletal sizes, and camera parameters from a number of 2D image constraints annotated by the user.

Image-based 3D keyframe interpolation. Another challenge for our system is how to utilize image measurements at intermediate frames to interpolate 3D key frames. We have developed an efficient algorithm to automatically track 2D image constraints at intermediate frames. In addition, we have introduced a physics-based optimization algorithm to generate in-between motions from 2D tracking results. Another nice feature of the proposed system is motion refinement. The user can briefly review the interpolated motion, edit the interpolated motion at any point in time, and continue refining the result until the interpolated motion precisely matches the input video.

We describe these components in more detail in the next sections.

4 Interactive 3D Keyframe Modeling

Our dynamics models approximate human motion with 17 rigid body segments, which include head, neck, back, left and right clavicle, humerus, radius, hip, femur, tibia, and metatarsal. We describe a full-body pose using a set of independent joint coordinates $\mathbf{q} \in R^{37}$, including absolute root position and orientation as well as the relative joint angles of individual joints. We represent the skeletal size of a human figure using a long vector $\mathbf{l} = [l_1, ..., l_{17}]^T$, where $l_b, b = 1, ..., 17$ is the length of the bth bone segment.

The goal of our interactive 3D keyframe modeling step is to estimate a small number of 3D key poses $(\mathbf{q}_1, ..., \mathbf{q}_K)$ as well as the skeletal size ($\mathbf{l}$) from video with minimal user interaction. Our idea for achieving this goal is to allow the user to annotate 2D joint locations at keyframe images and use them to automatically compute 3D key frames and human skeletal size (Figure 2). We choose to use 2D joint positions for 3D keyframe modeling because they could be easily annotated by a novice user from input image sequences. In our system, the key frames often correspond to the instants when contact state changes occur and/or instants with the highest visual content change.

4.1 Camera Parameter Estimation

Our system works for both static cameras and moving cameras. For moving cameras, structure and motion analysis is carried out on the input video sequence before any interactive human motion modeling takes place. Structure and motion analysis is a computer vision technique that automatically reconstructs the camera parameters which describe the relationship between the camera and the scene. We use MatchMover [2008] to estimate both intrinsic and extrinsic camera parameters $\vec{\rho} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, f)$, where the parameters (t_x, t_y, t_z), $(\theta_x, \theta_y, \theta_z)$, and f describe the position, orientation and focal length of the camera, respectively. The MatchMover system works well for both pan-tilt-zoom cameras and rotating and translating cameras with unknown and varying focal lengths.

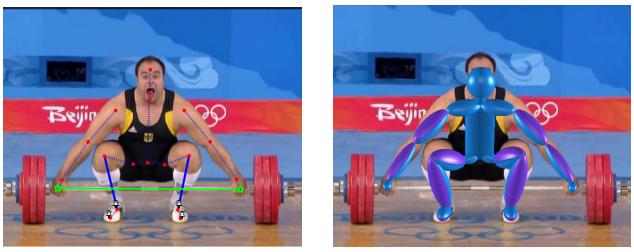


Figure 2: *Three types of 2D image constraints are annotated for 3D keyframe modeling. The user specifies 2D joint positions, bone direction constraints, and environmental contact constraints. The positional contact constraints are enforced at ankle and toe joints, and distance contact constraints are enforced at two hands. For bone directional constraints, the blue line indicates the parent joint is closer to the camera, and the white line indicates the bone segment is parallel to the image plane.*

When an input video is taken by a static camera with a constant focal length, structure from motion analysis techniques cannot be used to estimate camera parameters. Our solution is to automatically estimate the focal length of the camera along with 3D key poses and skeletal size in the 3D keyframe modeling step (Section 4.2). For simplicity of discussion, we focus our discussion on capturing human motion using static cameras. However, the basic reconstruction scheme that will be proposed in this section can easily be extended to moving cameras with camera parameters estimated by MatchMover.

4.2 Interactive 3D Keyframe Modeling

We now discuss how to model 3D key poses and human skeleton sizes from a number of 2D joint constraints specified by the user. The problem is challenging because 2D joint constraints are often not sufficient to determine 3D poses of an articulated object with an unknown skeletal size [Taylor 2000]. Our proposed reconstruction algorithm builds upon our previous work on structure from motion for articulated objects [Wei and Chai 2009]. More specifically, we define an energy function as a combination of cost functions described in the previous work and a new cost function measuring how well contact constraints are satisfied. We also extend the previous system to a full perspective camera model because a weak perspective camera model was adopted in the previous system.

Mathematically, we compute the human skeletal size $\mathbf{l}$ and 3D key poses $\mathbf{q}_1, ..., \mathbf{q}_K$ as well as the focal length of the camera f by minimizing the following energy function:

$$\begin{array}{c} \arg\min_{\mathbf{q}_1,...,\mathbf{q}_K,\mathbf{l},f} \; E_p + \lambda_s E_s + \lambda_r E_r + \lambda_c E_c \\ \text{subject to } \; E_d \leq 0 \end{array} \quad (1)$$

where the bone projection constraints (E_p) consider the relationship between 3D end points of bone segments and their 2D projections in image space (*i.e.*, 2D joint locations). The bone symmetry constraints (E_s) ensure the reconstructed human skeleton is symmetric; symmetry is imposed on seven bones, including clavicle, humerus, radius, hip, femur, tibia, and metatarsal. The rigid body constraints (E_r) preserve the distances between any two points located on the same rigid body regardless of the movement of a human body. The aforementioned constraints, however, are often insufficient to reduce reconstruction ambiguity. This is because there are two possible solutions for the relative depths of each bone segment at every

key frame, representing the pose ambiguity that has been previously discussed in the work of Taylor [2000]. The system allows the user to specify bone directional constraints E_d to remove this ambiguity.

In practice, we find that reconstructed 3D key poses often violate environmental contact constraints because 2D joint locations specified by the user are often noisy. This is always undesirable for graphics applications since this directly leads to noticeable visual artifacts such as foot-sliding in output animation. To address this issue, the system allows the user to specify the environmental contact constraints E_c. The user can define two types of environmental contact constraints: *positional* contact constraints and *distance* constraints (Figure 2). The *positional* contact constraints fix a specific point on the actor to a stationary location at multiple key frames. For example, when an actor sits on a chair, both of his feet stick on the ground for a short period of time. The *distance* constraints preserve the 3D distance between two points. For example, when an olympic athlete is weightlifting, the distance between his two hands should be maintained.

We have observed that direct optimization of the constrained objective function in Equation (1) often produces poor results. The optimization is prone to get trapped at local minima, due to there being a highly nonlinear optimization function related to joint angle representation as well as the use of inequality constraints. The performance of the optimization algorithm strongly depends on the initialization of the optimization. Similar to Chai and Wei [2009], we choose to represent 3D key poses with 3D root positions $\mathbf{p}_k$ and relative depth values $\mathbf{dZ}_k$ of inboard joints and outboard joints of every bone segment.

Initialization. To obtain a good initial guess for unknowns, we remove the bone directional constraints and environmental contact constraints from the objective function, and optimize the objective function with respect to focal length (f), skeletal size ($\mathbf{l}$), and key poses ($\mathbf{p}_k, \mathbf{dZ}_k^2, k = 1, ..., K$). Note that dropping off the bone directional constraints will not affect the reconstruction accuracy because the bone directional constraints are used to eliminate the ambiguity caused by the signs of relative depth values $\mathbf{dZ}_k$. Meanwhile, removal of the contact constraints does not significantly affect the reconstruction accuracy because the contact constraints are mainly used for eliminating environmental contact artifacts (*e.g.*, foot-sliding) caused by noisy joint locations. This allows us to transform a notoriously difficult constrained optimization problem into a well-behaved unconstrained optimization problem. We analytically derive the Jacobian terms of the object function and then run the optimization with the Levenberg-Marquardt algorithm in the Levmar library [Lourakis 2009].

Optimization. We now can set very good initial values for the constrained optimization problem in Equation (1). More specifically, we initialize $f, \mathbf{l}, \mathbf{p}_k, \mathbf{dZ}_k, k = 1, ..., K$ with the corresponding values estimated from the initialization step. The values of $\mathbf{dZ}_k$ are initialized by the root square of the estimated $\mathbf{dZ}_k^2$ with appropriate signs determined by the bone directional constraints. The optimization typically converges in less than ten iterations due to a very good initial guess.

The last step of 3D keyframe modeling process is to use inverse kinematics techniques to transform key poses from 3D position space to 3D joint angle space. The 3D keyframe modeling process takes three to eight seconds for all the videos reported in this paper. Figure 2 visualizes three types of 2D image constraints (joint locations, bone directions, and contact constraints) specified at a key frame. We also show the reconstructed 3D key pose and human skeleton model in the same figure.

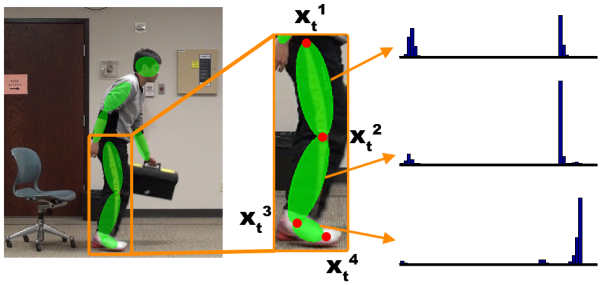


Figure 3: *2D multi-joint tracking: (left) tracked bone segments superimposed on images; (middle) the state vector for tracking one leg is represented as* $\boldsymbol{e}_t = \{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \boldsymbol{x}_t^3, \boldsymbol{x}_t^4\}$*; (right) The histogram feature vector for each bone segment. The resulting feature vector for the leg stacks the feature vector of each individual bone segment (upper leg, lower leg, foot).*

5 Image-based 3D Keyframe Interpolation

This section discusses how to interpolate 3D key frames $\mathbf{q}_1, ..., \mathbf{q}_K$ using 2D image measurements at intermediate frames. Our key idea is to use both image measurements and physical constraints to interpolate in-between poses. Briefly, our system interactively tracks a small set of 2D joint points at intermediate frames (Section 5.1) and uses the tracked 2D joints as well as physical constraints to interpolate the 3D key frames (Section 5.2). The user can briefly review the result by playing back the interpolated motion superimposed on the input video sequence, and incrementally edit the result at any frame until the interpolated motion precisely matches the input video sequence (Section 5.3).

5.1 Keyframe based 2D Multi-joint Tracking

The first component of video-based motion interpolation is to use 2D joint positions annotated at key frames to track 2D joint locations at intermediate frames. In practice, it is almost impossible to track every joint accurately due to various ambiguities (*e.g.*, significant occlusions) present in monocular video sequences. We, instead, allow the user to interactively select which joints to track. The user could choose to either track a single bone segment (*e.g.*, head), or simultaneously track multiple connected bone segments from the same limb (*e.g.*, upper leg, lower leg, and foot).

We represent the state of a single bone segment as $\mathbf{e}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2\}$, where $\mathbf{x}_t^1$ and $\mathbf{x}_t^2$ are the 2D image coordinates of the inboard and outboard joints, respectively. Similarly, the state of multiple connected bone segments can be represented by a long vector sequentially stacking the 2D coordinates of all end points. For example, consider a limb consisting of three bone segments: upper leg, lower leg and foot. The state of the limb is represented by the 2D positions of four end points $\mathbf{e}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \mathbf{x}_t^3, \mathbf{x}_t^4\}$ (Figure 3).

Let $\mathbf{e}_1$ and $\mathbf{e}_T$ represent the states of the start and end frames, respectively. Our goal here is to track the object state $\mathbf{e}_t, t = 2, ..., T-1$ at intermediate frames.

Feature vector. We approximate the 2D shape of a bone segment with an elliptic region. We choose to represent the appearance of the bone segment as a regularized histogram distribution of all pixels within the elliptic region in Hue-Saturation-Value (HSV) color space because the histogram distribution is robust to image noise, cloth deformation, and occlusions. The feature vector of a target region at the frame t can then be represented as a vector-valued

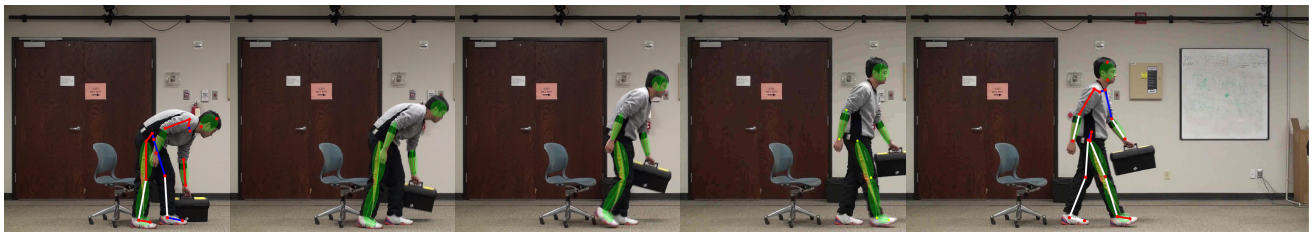


Figure 4: *Keyframe based 2D multi-joint tracking: the first and last frames are the annotated keyframe images. The user interactively selects which joints to track and the system automatically tracks their locations at intermediate frames.*

function of the state $\mathbf{e}_t$ (for details, see Appendix A):

$$\mathbf{h}(\mathbf{e}_t) = (h_1(\mathbf{e}_t), \ldots, h_M(\mathbf{e}_t))^T, \quad \sum_{m=1}^{M} h_m(\mathbf{e}_t) = 1, \qquad (2)$$

where $\mathbf{h}(\mathbf{e}_t)$ represents the current target model in the feature space and M is the total number of bins used in the HSV color space. The function $h_m(\mathbf{e}_t)$ is the density of the mth bin. For multiple connected bone segments, we represent the feature vector of multiple connected regions as a long vector that sequentially stacks the feature vector of each individual bone $\mathbf{h}^s(\mathbf{e}_t), s = 1, ..., S$, where S is the number of connected bone segments (Figure 3).

Parameterized template model. We assume that the template model for any in-between frame can be represented as a weighted interpolation of the feature vectors at the start and end frames:

$$H_m(\beta_t) = \beta_t h_m(\mathbf{e}_1) + (1 - \beta_t) h_m(\mathbf{e}_T), \;\; m = 1, ..., M \qquad (3)$$

where the parameter β_t ranges between 0 and 1 and is assumed to be unknown. One nice feature of the parameterized template models is to model possible appearance changes between two key frames with the time varying weight β_t [Wei and Chai 2008].

Matching distance. We estimate the states of the bone segments at intermediate frames by matching the parameterized template model $\mathbf{H}(\beta_t)$ with the target region $\mathbf{h}(\mathbf{e}_t)$ in the feature space. We use the Bhattacharyya distance to measure the matching distance between the target region and the parameterized template model:

$$d(\mathbf{e}_t, \beta_t)^2 = 1 - \sum_{m=1}^{M} \sqrt{h_m(\mathbf{e}_t)\, H_m(\beta_t)}. \qquad (4)$$

where $d(\mathbf{e}_t, \beta_t)^2$ represents the matching cost between the template model and the target region. To deal with occlusions and image noise, we apply robust statistics [Huber 1981; Hampel et al. 1986] to measure the residual distance. In our experiment, we choose the Lorentzian robust estimator to define the matching cost term:

$$\rho(d(\mathbf{e}_t, \beta_t)^2) = \log(1 + \frac{d(\mathbf{e}_t, \beta_t)^2}{2\sigma^2}) \qquad (5)$$

where the scalar σ is a parameter for the robust estimator. For all examples reported in the paper, σ is set to 0.25.

Objective function. We now can formulate the keyframe based 2D joint tracking process in a batch-based optimization framework. Our system computes the "best" state trajectory $\mathbf{e}_2$,...,$\mathbf{e}_{T-1}$ as well as the unknown template parameters $\beta_t, t = 2, ..., T-1$ by optimizing over all intermediate frames at once:

$$\arg\ min_{\{\mathbf{e}_t\},\{\beta_t\}} \sum_{t=2}^{T-1} \rho(d(\mathbf{e}_t, \beta_t)^2) + \lambda_e \sum_{t=2}^{T} \|\mathbf{e}_t - \mathbf{e}_{t-1}\|^2 + \lambda_\beta \sum_{t=2}^{T} (\beta_t - \beta_{t-1})^2 \qquad (6)$$

where the first term minimizes the matching cost between the template model and the target region. The second and third terms penalize the sudden changes of the state $\mathbf{e}_t$ and appearance β_t of the target region. The weights λ_e and λ_β are set to 0.0015 and 0.5 respectively.

Real-time optimization. We initialize the states of in-between frames by a linear interpolation of the first and last frames. We optimize the objective function with trust-region reflective Newton methods. The gradient of the energy function is analytically evaluated at each iteration. We found that the optimization procedure often converges quickly (usually less than 20 iterations). The current multi-joint tracking system can track 2D joint locations at interactive frame rates (less than one second). The interface appears very responsive because the user can immediately see the tracking results.

User interaction. The tracking system is fairly robust to occlusions and illumination changes as well as noise caused by cloth deformation and motion blurring. However, due to the complexity of a real world, it is almost impossible to build a fully automatic system that can accurately track 2D joint locations at intermediate frames. When the tracker fails, user interaction must be used to correct tracking errors. The realtime batch optimization process provides an efficient way to combine user interactions with an automatic vision process. The user can refine the tracking result at any frame, include new constraints as a part of the object function, and restart the optimization. Figure 4 shows sample images of our tracking result.

5.2 3D Motion Interpolation

The second component of video-based motion interpolation is to use 2D tracking joints to interpolate 3D key frames. This problem is challenging because the number of constraints derived from the 2D joint tracking system is often not sufficient to determine a unique solution for in-between motion. In practice, real video sequences often contain significant occlusions, which make it almost impossible to track end points of every bone segment. So there will be a space of possible solutions that meet the image constraints derived from video. We eliminate this ambiguity with physics-based dynamics constraints.

Full-body dynamics. The Newtonian dynamics equations for full-body movement can be defined as follows:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + h(\mathbf{q}) = \mathbf{u} + \mathbf{J}_c^T \mathbf{f}_c \qquad (7)$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ represent the joint angle poses, joint velocities, and joint accelerations, respectively. The quantities $M(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $h(\mathbf{q})$ are the joint space inertia matrix, centrifugal/Coriolis and gravitational forces, respectively. The vectors $\mathbf{u}$ and $\mathbf{f}_c$ are joint torques and contact forces respectively. The contact force Jacobian matrix $\mathbf{J}_c$ maps joint velocities to world space cartesian velocities at

Sequences	No. of frames	Camera types	No. of key frames	No. of tracking joints per frame	Refinement A	Refinement B
Uneven bar	150	pan-tilt-zoom	10	10	0	7
Acting	585	static	9	10	2	0
Weightlifting	310	pan-tilt-zoom	13	11	3	4
Fencer A	92	pan-tilt-zoom	6	12	0	0
Fencer B	92	pan-tilt-zoom	6	14	0	0
Jumping	100	static	6	12	0	4
Frisbee	114	static	6	10	0	0
All examples	1443		56	11	5	15

Table 1: *Details of our experiments. Refinement A counts the total number of secondary keyframes used in motion refinement, and Refinement B counts the total number of 2D joint constraints used for motion refinement.*

contact points. Human muscles generate torques about each joint, leaving global position and orientation of the body as unactuated joint coordinates. The movement of global position and orientation is controlled by contact forces $\mathbf{f}_c$. Modifying those coordinates requires contact forces $\mathbf{f}_c$ from the environment.

Friction limit constraints. During ground contact, the feet can only push, not pull on the ground, contact forces should not require an unreasonable amount of friction, and the center of pressure must fall within the support polygon of the feet. We use Coulomb's friction model to compute the forces caused by the friction between the character and environment. A friction cone is defined to be the range of possible forces satisfying Coulomb's function model for an object at rest. We ensure the contact forces stay within a basis that approximates the cones with nonnegative basis coefficients. We model the contact between two surfaces with multiple contact points $m = 1, ..., M$. This allows us to represent the contact forces $\mathbf{f}_g$ as a linear function of nonnegative basis coefficients [Pollard and Reitsma 2001; Liu et al. 2005]:

$$\mathbf{f}_g(\mathbf{w}_1, ..., \mathbf{w}_M) = \sum_{m=1}^{M} \mathbf{B}_m \mathbf{w}_m \quad \text{subject to} \quad \mathbf{w}_m \geq \mathbf{0} \tag{8}$$

where the matrix $\mathbf{B}_m$ is a 3×4 matrix consisting of 4 basis vectors that approximately span the friction cone for the m-th contact force. The 4×1 vector $\mathbf{w}_m$ represents nonnegative basis weights for the m-th contact force. Note that we do not enforce friction limit constraints on other types of environmental contact forces $\mathbf{f}_e$, *e.g.*, when the actor is swinging on a high bar or monkey bars or when the actor is carrying a bag.

Objective function. We now can formulate the motion interpolation problem in a space-time motion optimization framework [Witkin and Kass 1988; Cohen 1992]. Given the start and end poses, contact constraints, and 2D joint positions at intermediate frames, the optimization simultaneously computes the joint poses $\mathbf{q}$, joint torques $\mathbf{u}$, and contact forces $\mathbf{f}_g(\mathbf{w})$ and $\mathbf{f}_e$ that maximize the performance of the following multiobjective function:

$$\begin{gathered} \arg\min_{\mathbf{q},\mathbf{u},\mathbf{w},\mathbf{f}_e} E_{image}(\mathbf{q}) + \lambda_1 E_{torque}(\mathbf{u}) + \lambda_2 E_{smooth}(\ddot{\mathbf{q}}) \\ \text{subject to} \quad M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + h(\mathbf{q}) = \mathbf{u} + \mathbf{J}_c^T[\mathbf{f}_g(\mathbf{w})^T, \mathbf{f}_e^T]^T \\ \mathbf{w} \geq \mathbf{0} \\ \mathbf{G}_c = \mathbf{0} \end{gathered} \tag{9}$$

where the first term E_{image} measures how well the interpolated motion matches the 2D position constraints from tracking joints. The second term E_{torque} minimizes the sum of squared torques at intermediate frames. The third term E_{smooth} ensures smoothness of the joint angle trajectories and root trajectory over time by minimizing the sum of squared joint accelerations and sum of squared root accelerations. The optimization is also subject to the discretization of Newtonian dynamics equations determined by a finite difference scheme, friction limit constraints $\mathbf{w} \geq \mathbf{0}$, and contact constraints $\mathbf{G}_c = \mathbf{0}$.

Motion optimization. In our implementation, we drop off Newtonian dynamics equations in the objective function by replacing joint torques $\mathbf{u}$ with $M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + h(\mathbf{q}) - \mathbf{J}_c^T \mathbf{f}_c(\mathbf{w})$. We use backward difference to compute joint velocities and use central difference to compute joint accelerations. This allows us to optimize the entire motion in terms of joint poses $\mathbf{q}_t, t = 2, ..., T-1$ and contact forces $\mathbf{w}$. We follow a standard approach of representing $\mathbf{q}_t$ using cubic B-splines. We solve the optimization problem using sequential quadratic programming (SQP) [Bazaraa et al. 1993], where each iteration solves a quadratic programming subproblem. We initialize the joint poses by a linear interpolation of the start and end poses.

The 3D motion interpolation process typically takes about five seconds to converge for a 20-frame motion segment. For example, the "jumping", "uneven bar", and "acting" examples took 20, 27 and 83 seconds to interpolate the entire video sequences, respectively.

5.3 3D Motion Refinement

3D motion interpolation might not always produce a motion sequence that precisely matches the input video sequence because the 2D joint tracking process cannot track 2D locations of all joints at intermediate frames. When this happens, the user can briefly review the result by playing back the interpolated motion sequence superimposed on the input video sequence, and incrementally edit the result at any frame until the interpolated motion precisely matches the input video sequence. The current system supports two types of user interactions for motion refinement:

Secondary keyframes. When interpolated motions do not closely match 2D image measurements at intermediate frames, the user can select an intermediate image as a secondary key frame, which often corresponds to the poorest reconstruction result, and manually adjust 2D joint locations at the secondary keyframes. This allows us to reconstruct a 3D secondary key pose in the same way as the interactive keyframe modeling process. We use the secondary key poses to divide the original subsequence into two smaller subsequences and run the motion interpolation process for each subsequence again.

Interactive 2D joint dragging. When the interpolated motion sequence is already close to the input video sequence, the user can fine tune the result with 2D direct manipulation interfaces. More specifically, the user picks an unsatisfactory frame, pulls on a single joint which is the furthest from the actual image location, and then restarts the optimization. This human guidance is often enough to achieve a desired solution. If not, the user may pull on additional points, and iterate. In each refinement step, we incrementally add 2D joint location constraints, initialize the current motion with motion from the previous step, and rerun motion interpolation again.

The refinement process runs in real time because the interpolated motion is already very close to the final motion.

6 Results

We demonstrate the performance of our system by modeling a wide variety of human actions from monocular video sequences, which include locomotion ("walking"), acrobatic motion ("uneven bar"), highly dynamic motions ("jumping'), sudden burst activities ("weightlifting"), interactions with environments ("sitting", "standing up", and "picking up an object"), and multiple actor interactions ("fencing"). Our system works for both static and moving cameras. Our results are best seen in the accompanying video although we show sample frames of a few motions in the paper. Table 1 summarizes the experimental details of all testing sequences.

Uneven bar. This video sequence was downloaded from the Internet. The video was taken by a pan-tilt-zoom camera. We used the MatchMover to estimate full-perspective camera parameters and reconstructed 3D key frames and human skeleton size based on 2D image constraints annotated at *ten* keyframe images, along with the estimated camera parameters. The system then interactively tracked 2D positions of *ten* joints at intermediate frames and used them to interpolate 3D key frames. In the motion refinement step, we dragged *seven* points to improve the result. Figure 5 shows some sample frames of the reconstruction motion from the original viewpoint and a new viewpoint (first and second rows). Note that the green lines visualize the magnitudes and directions of the estimated contact forces.

Acting. This video sequence was taken by a static and uncalibrated camera. The "acting" sequence contains a variety of everyday actions, including sitting on a chair, standing up, turning around, yawning, picking up a box, and walking with a box. We assumed a full perspective camera model. Our system simultaneously estimated the camera focal length, human skeletal size, and 3D key poses at *nine* key frames and used image measurements and physical constraints to automatically calculate the "in between" poses and camera parameters. After playing back the reconstructed motion, the user refined the motion with *two* secondary key frames. Some sample frames of the reconstructed motion are shown in Figure 5 (third and forth rows).

Jumping. We also experimented with a jumping sequence of a subject seen from an oblique view. Modeling human motion from this sequence is difficult due to the significant depth change and the perspective effects of the person jumping closer to the camera. The accompanying video shows that our algorithm can successfully estimate the focal length as well as the 3D motion and skeletal size.

Weightlifting. This is another video sequence downloaded from the Internet. It was taken by a full perspective pan-tilt-zoom camera. Reconstructing human movement from this video is challenging due to a sudden burst of movement. Some sample frames of the reconstruction results are shown in Figure 5 (fifth and six rows).

Multiple actor interaction. The video was taken by a full perspective pan-tilt-zoom camera. The white fencing clothing makes it extremely difficult to capture the movements of two fencers. We captured the motion data of two fencers separately. We used six key frames to capture the movement of each fencer. No refinements were needed. The final motion is shown in Figure 5 (seven and eighth rows).

Statistics of user interaction. User interactions are needed for annotating 2D joint locations at key frames as well as interactive 3D motion refinement. The use of physical constraints and 2D image measurements for 3D motion interpolation minimizes the number of key frames required for video-based motion modeling. Based on the testing sequences (1443 frames in total) we reported here, the users annotated 2D joint locations at 56 keyframe images (3.8% of the total frames) for video-based motion modeling. The users also added *five* secondary key frames (0.3% of the total frames) and dragged 15 points to refine the interpolated 3D motions. The total amount of interaction time to create the final 3D motions varied from 5 to 20 minutes depending on the complexity and length of the motions.

Comparison. The accompanying video shows the importance of both physical constraints and image measurements to our motion modeling system. Specifically, we dropped off the terms of the objective function described in Equation (9) and compared our approach with 3D keyframe interpolation using linear interpolations, 3D keyframe interpolation using physics-based optimization, and 3D keyframe interpolation using the image term and the smoothness term. The comparison shows that only our approach, *i.e.*, motion interpolation using both physical constraints and image measurements, can generate a natural-looking animation that matches the input video.

Evaluation. We quantitatively assessed the quality of the captured motion by comparing with ground truth motion data captured with a full marker set in a twelve-camera Vicon system [2009]. The average reconstruction error, which is computed as average 3D joint position discrepancy between the estimated poses and the ground truth mocap poses, was about 4.5 cm per joint per frame. Figure 6 shows a side-by-side comparison between our result and the optical mocap result. As shown in the accompanying video, the quality of our reconstruction result is comparable to motion data recorded by the Vicon system but our system is much cheaper and requires less intrusive capturing devices.

We also evaluated the 3D/2D matching errors between reconstructed motions and input video sequences for all the testing examples. The matching errors were computed by the average pixel distances between the 2D joint locations of reconstructed motions and those tracked from input video sequences, *i.e.*, the first term of the objective function in Equation (9). The errors for different testing sequences are 2.4 pixels ("acting"), 3.4 pixels ("uneven bar"), 4.1 pixels ("weightlifting"), 2.4 pixels ("fencer A"), 2.8 pixels ("fencer B"), 2.8 pixels ("jumping"), and 2.1 pixels ("frisbee"), respectively. Note that the actual matching errors could be much smaller because 2D joint positions tracked from an input video are often very noisy.

7 Conclusion and Discussion

In this paper, we have developed an end-to-end system that models physically realistic human motion from uncalibrated monocular video sequences. The key idea of our system is to utilize physics-based dynamics models and minimal user interaction to remove the ambiguities in video-based motion modeling. Our system is desirable for video-based motion modeling because it does not require known skeletons, it works for static or moving cameras, it does not need any prerecorded motion data, and it is capable of modeling a wide range of human actions from single-camera video streams. With such a system, live performances and important events such as memorable Olympic moments can be documented, analyzed, and animated in 3D.

Our system benefits from the combined power of video-based motion modeling and physics-based motion modeling. Physics-based motion modeling is a mathematically ill-posed problem because there are many ways to adjust a motion so that physical laws are satisfied, and yet only a subset of such motions are natural-looking. By accounting for physical constraints and observed image data simultaneously, we can estimate physically realistic motion that is

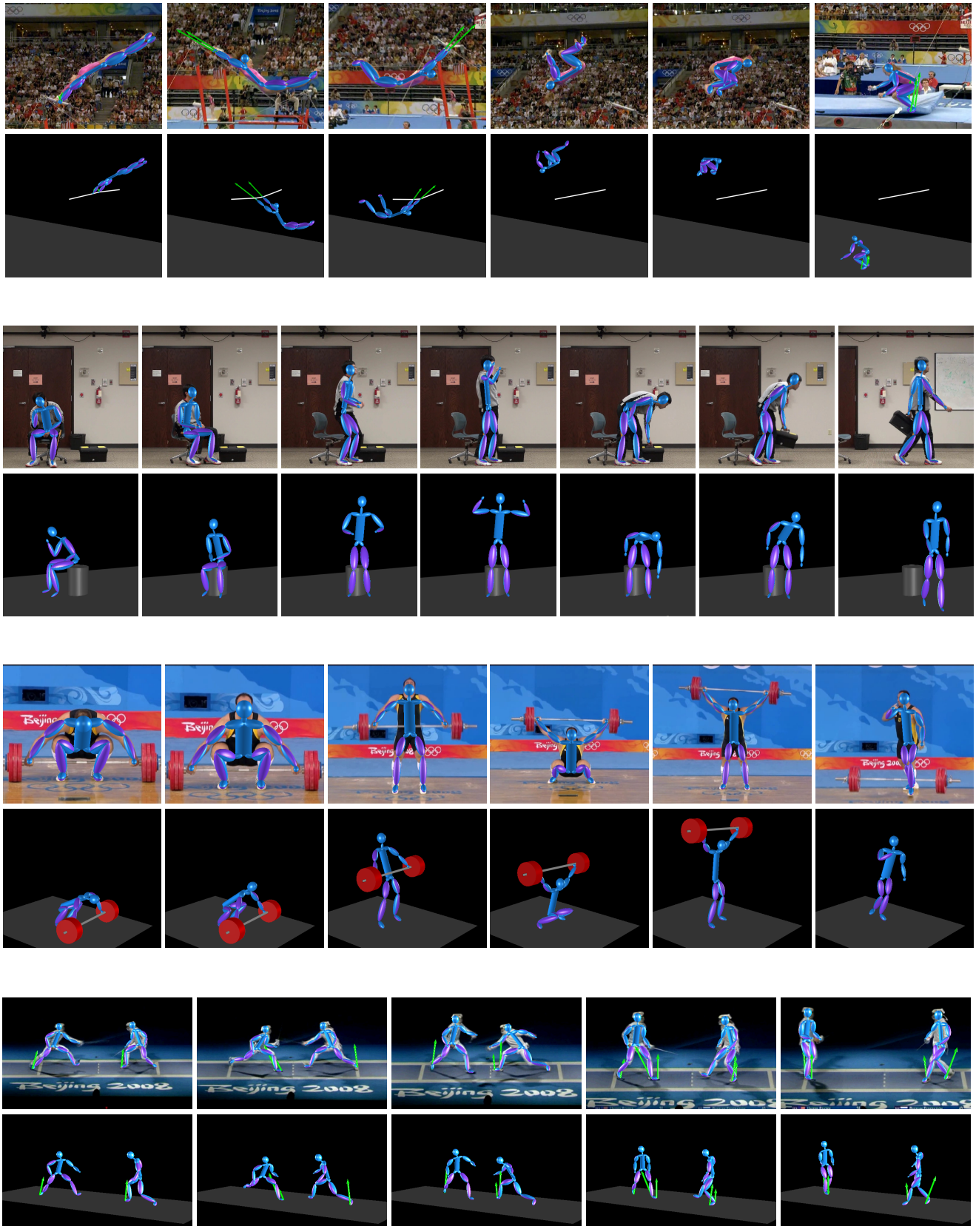
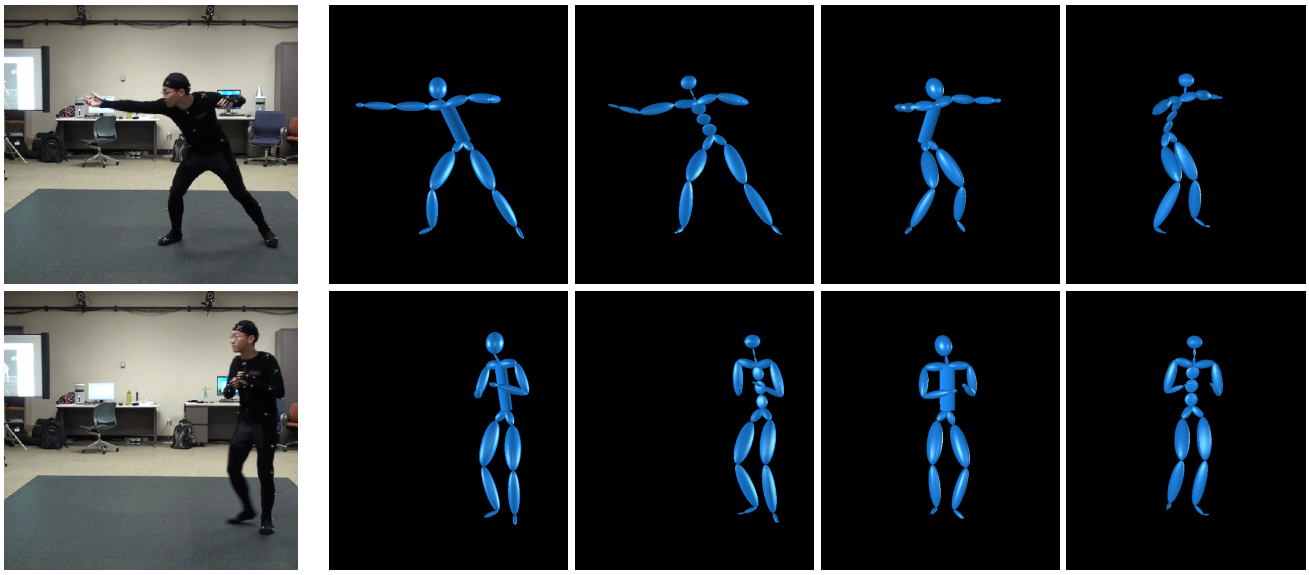


Figure 5: *Modeling a wide variety of human actions from uncalibrated monocular video sequences. The reconstructed motions are rendered from the original viewpoint and a new viewpoint. Note that green lines visualize magnitude and direction of estimated contact forces.*



(a) (b) (c) (d) (e)

Figure 6: *Comparison between our result and ground truth motion data captured by a twelve-camera Vicon mocap system in a full marker set: (a) input image sequences; (b) our result in the original viewpoint; (c) optical mocap result in the original viewpoint; (d) our result in a new viewpoint; (e) optical mocap result in a new viewpoint.*

consistent with the input image data. On the other hand, video-based motion modeling techniques can utilize physical constraints to reduce modeling ambiguities and ensure the reconstructed motion is physically correct.

We have demonstrated the power and effectiveness of our approach by capturing a wide variety of human actions from uncalibrated monocular video sequences (*e.g.*, sports footage). We believe the system could be easily extended to modeling the movement and skeletons of articulated animals such as trotting horses or hopping kangaroos because we assume unknown skeletal sizes and do not require any prerecorded motion capture data. One of the immediate directions for future work is, therefore, to investigate the application of our algorithm to articulated animal movement.

One nice property of our video-based motion modeling system is that we can estimate joint torques and contact forces from 2D image data. However, when there are multiple contact points between human bodies and environments, there is not a unique solution of joint torques and contact forces. When this happens, our system relies on the "minimum torque" principle to remove the ambiguity.

Recent progress in physics-based motion optimization has shown that kinematic motion priors can be used to significantly improve the performance of physics-based motion modeling. In the future, we would like to adapt the algorithm to incorporate kinematic motion priors (*e.g.*, [Chai and Hodgins 2005]) into the video-based motion modeling process in order to improve the robustness and accuracy of our system.

APPENDIX

A Feature Model for 2D Multi-joint Tracking

Let n_t be the number of pixels located inside the target region at frame t and $\mathbf{p}_{t,i}, i = 1, ..., n_t$ be the image coordinates of the i-th pixel. Mathematically, we can define the function $h_m(\mathbf{e}_t)$ as follows:

$$h_m(\mathbf{e}_t) = \sum_{i=1}^{n_t} \delta(f(I(\mathbf{p}_{t,i})) - m) \tag{10}$$

where the function $\delta(\cdot)$ represents the Kronecker delta function. The function $I(\mathbf{p}_{t,i})$ represents the color of the i-th pixel at the location $\mathbf{p}_{t,i}$. The function f maps $I(\mathbf{p}_{t,i})$ to the index of its bin in the quantized feature space.

We regularize the histogram distribution $h_m(\mathbf{e}_t)$ by masking the objects with an isotropic kernel in the spatial domain. When the kernel weights, which carry continuous information, are used in defining the feature space representation, the regularized histogram distribution of target regions becomes a smooth and continuous function of target states, $\mathbf{e}_t$.

An isotropic kernel, with a convex and monotonic decreasing kernel function $k(r)$, is used to assign smaller weights to pixels farther from the center. We choose the Epanechnikov profile as our kernel function [Comaniciu and Meer 2002; Wei and Chai 2008]:

$$k(r) = \begin{cases} 1 - r & 0 \le r \le 1 \\ 0 & r > 1 \end{cases} \tag{11}$$

where $r \ge 0$. This kernel function makes the regularized histogram distribution differentiable everywhere inside the elliptical region. Its gradients can, therefore, be evaluated analytically.

The regularized histogram distribution of the feature in the target region at frame t is computed as:

$$h_m(\mathbf{e}_t) = \frac{\sum_{i=1}^{n_t} k\left((\frac{u^i}{W})^2 + (\frac{v^i}{H})^2\right) \delta(f(I(\mathbf{p}_{t,i})) - m)}{\sum_{i=1}^{n_t} k\left((\frac{u^i}{W})^2 + (\frac{v^i}{H})^2\right)}, \tag{12}$$

where $W = \|\mathbf{x}_t^1 - \mathbf{x}_t^2\|/2$ and H is assumed to be known and determined by the user. The scalars u^i and v^i are the local coordinates

of the i-th pixel, which can be computed as follows:

$$\begin{pmatrix} u^i \\ v^i \end{pmatrix} = \begin{pmatrix} \cos\theta_t & \sin\theta_t \\ -\sin\theta_t & \cos\theta_t \end{pmatrix} (\mathbf{p}_{t,i} - \mathbf{c}_t). \quad (13)$$

where $\mathbf{c}_t = (\mathbf{x}_t^1 + \mathbf{x}_t^2)/2$ and $\theta_t = atan2([\mathbf{x}_t^2 - \mathbf{c}_t]_v, [\mathbf{x}_t^2 - \mathbf{c}_t]_u)$. Note that $[\mathbf{x}]_u$ and $[\mathbf{x}]_v$ represent the horizontal and vertical coordinates of the vector $\mathbf{x}$ respectively.

References

AGARWAL, A., AND TRIGGS, B. 2006. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. 28(1):44–58.

BAZARAA, M. S., SHERALI, H. D., AND SHETTY, C. M. 1993. *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons Ltd. 2nd Edition.

BREGLER, C., MALIK, J., AND PULLEN, K. 2004. Twist Based Acquisition and Tracking of Animal and Human Kinematics. *International Journal of Computer Vision*. 56(3):179–194.

BRUBAKER, M. A., AND FLEET, D. J. 2008. The Kneed Walker for human pose tracking. In *Proceedings of IEEE CVPR*. 1–8.

CHAI, J., AND HODGINS, J. 2005. Performance Animation from Low-dimensional Control Signals. In *ACM Transactions on Graphics*. 24(3):686–696.

CHEN, Y.-L., AND CHAI, J. 2009. Simultaneous Reconstruction of 3D Human Skeleton and Motion from Monocular Video Sequences. *Proceedings of The Ninth Asian Conference on Computer Vision*.

COHEN, M. F. 1992. Interactive Spacetime Control for Animation. In *Proceedings of ACM SIGGRAPH 1992*. 293–302.

COMANICIU, D., AND MEER, P. 2002. Mean Shift:A Robust Approach Toward Feature Space Analysis. In *IEEE Trans. Pattern Analysis and Machine Intelligence*. 24(5):603-619.

COWLEY, A., AND TAYLOR, C. J., 2001. Videomocap: A video based motion capture system. http://www.cis.upenn.edu/ cjtaylor/RESEARCH/projects/Johansson/VideoMoCap.html.

DIFRANCO, D. E., CHAM, T.-J., AND REHG, J. M. 2001. Reconstruction of 3D figure motion from 2D correspondences. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1:307-314.

ELGAMMAL, A., AND LEE, C. 2004. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2: 681–688.

HAMPEL, F. R., RONCHETTI, E. M., ROUSSEEUW, P. J., AND STAHEL, W. A. 1986. Robust Statistics: The Approach Based on Influence Functions. Wiley.

HOWE, N., LEVENTON, M., AND FREEMAN, W. 1999. Bayesian Reconstruction of 3D Human Motion from Single-camera Video. In *Advances in Neural Information Processing Systems 12*. 820–826.

HUBER, P. J. 1981. Robust Statistics. Wiley.

KANAUJIA, C. S. A., AND METAXAS, D. 2007. BM^3E: Discriminative density propagation for visual tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. 29(11):2030–2044.

LIU, K., HERTZMANN, A., AND POPOVIĆ, Z. 2005. Learning Physics-Based Motion Style with Nonlinear Inverse Optimization. In *ACM Transactions on Graphics*. 23(3):1071–1081.

LOURAKIS, M. I. A. 2009. levmar: Levenberg marquardt nonlinear least squares algorithms in c/c++. In *http://www.ics.forth.gr/ lourakis/levmar/*.

LOY, G., ERIKSSON, M., SULLIVAN, J., AND CARLSSON, S. 2004. Monocular 3D Reconstruction of Human motion in Long Action Sequences. In *European Conference on Computer Vision*. 442–455.

MATCHMOVER, 2008. http://www.realviz.com/.

PAVLOVIĆ, V., REHG, J. M., AND MACCORMICK, J. 2000. Learning Switching Linear Models of Human Motion. In *Advances in Neural Information Processing Systems 13*, 981–987.

POLLARD, N., AND REITSMA, P. 2001. Animation of Human-like Characters: Dynamic Motion Filtering with A Physically Plausible Contact Model. In *In Yale Workshop on Adaptive and Learning Systems*.

POPOVIĆ, Z., AND WITKIN, A. P. 1999. Physically Based Motion Transformation. In *Proceedings of ACM SIGGRAPH 1999*. 11-20.

ROSALES, R., AND SCLAROFF, S. 2000. Specialized Mappings and the Estimation of Human Body Pose from a Single Image. In *Proceedings of the Workshop on Human Motion*. 19-24.

SAFONOVA, A., HODGINS, J., AND POLLARD, N. 2004. Synthesizing Physically Realistic Human Motion in Low-Dimensional, Behavior-Specific Spaces. In *ACM Transactions on Graphics*. 23(3):514–521.

SIDENBLADH, H., BLACK, M. J., AND SIGAL, L. 2002. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*. 784–800.

SMINCHISESCU, C., AND JEPSON, A. 2004. Generative Modeling for Continuous Non-Linearly Embedded Visual Inference. In *ICML*, 759–766.

SULEJMANPASIC, A., AND POPOVIĆ, J. 2005. Adaptation of Performed Ballistic Motion. In *ACM Transactions on Graphics*. 24(1):165–179.

TAYLOR, C. J. 2000. Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. In *Computer Vision and Image Understanding*. 80(3):349-363.

URTASUN, R., FLEET, D. J., HERTZMANN, A., AND FUA., P. 2005. Priors for people tracking from small training sets. In *IEEE International Conference on Computer Vision*, 403–410.

VICON SYSTEMS, 2009. http://www.vicon.com.

VONDRAK, M., SIGAL, L., AND JENKINS, O. C. 2008. Physical simulation for probabilistic motion tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

WEI, X., AND CHAI, J. 2008. Interactive Tracking of 2D Generic Objects with Spacetime Optimization. In *Proceedings of European Conference on Computer Vision*. 1:657-670.

WEI, X., AND CHAI, J. 2009. Modeling 3D Human Poses from Uncalibrated Monocular Images. *Proceedings of IEEE Conference on Computer Vision*.

WITKIN, A., AND KASS, M. 1988. Spacetime Constraints. In *Proceedings of ACM SIGGRAPH 1998*. 159–168.