Physical Simulation for Probabilistic Motion Tracking

Marek Vondrak
Brown University
marek@cs.brown.edu

Leonid Sigal
University of Toronto
ls@cs.toronto.edu

Odest Chadwicke Jenkins
Brown Univesity
cjenkins@cs.brown.edu

Abstract

Human motion tracking is an important problem in computer vision. Most prior approaches have concentrated on efficient inference algorithms and prior motion models; however, few can explicitly account for physical plausibility of recovered motion. The primary purpose of this work is to enforce physical plausibility in the tracking of a single articulated human subject. Towards this end, we propose a full-body 3D physical simulation-based prior that explicitly incorporates motion control and dynamics into the Bayesian filtering framework. We consider the human's motion to be generated by a "control loop". In this control loop, Newtonian physics approximates the rigid-body motion dynamics of the human and the environment through the application and integration of forces. Collisions generate interaction forces to prevent physically impossible hypotheses. This allows us to properly model human motion dynamics, ground contact and environment interactions. For efficient inference in the resulting high-dimensional state space, we introduce exemplar-based control strategy to reduce the effective search space. As a result we are able to recover the physically-plausible kinematic and dynamic state of the body from monocular and multi-view imagery. We show, both quantitatively and qualitatively, that our approach performs favorably with respect to standard Bayesian filtering methods.

1. Introduction

Physics plays an important role in characterizing, describing and predicting motion. Dynamical simulation allows one to computationally account for various physical factors, *e.g.*, a person's mass, interaction with the ground plane, friction, self-collisions or physical disturbances. A tracking system can take advantage of physical prediction to cope with incomplete information and reduce uncertainty. For example, ambiguities due to self-occlusions in monocular sequences could potentially be resolved by incorporating a passive dynamics-based (rag-doll) prediction. Pose changes that are unlikely or which violate physical constraints can be given lower weights, constraining the space of poses to search over and boosting performance. We claim that proper utilization of dynamics-based prediction will significantly improve the quality of motion tracking.

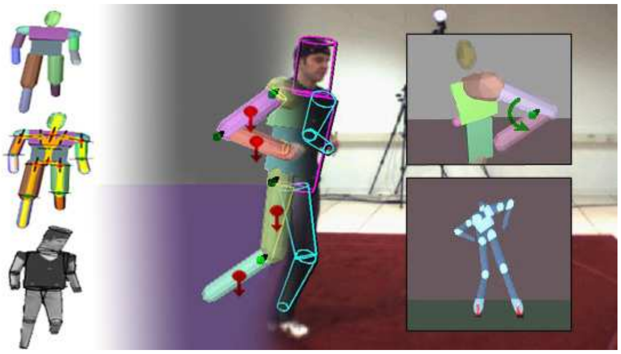

Figure 1. **Incorporating physics-based dynamic simulation with joint actuation and dynamic interaction into Bayesian filtering**. Illustration of the figure model, on the left, shows collision geometries of the figure segments (top-left), the joints and skeletal structure (middle-left), and the visual representation corresponding to an image projection (bottom-left). Most joints have 3 angular degrees of freedom (DOFs), except for the knee and elbow joints (1 angular DOF), spine joint and the clavicle joints (2 angular DOFs) and the root joint (3 linear and 3 angular DOFs). The figure's motion is determined by its dynamics, actuation forces at joints (right-top) and surface interaction at contacts (right-bottom).

We propose a means for incorporating full body physical simulation with probabilistic tracking. The tracked individual is modeled as an actuated articulated structure ("figure") composed of three-dimensional rigid body segments connected by joints. Segments correspond to parts of the figure body, like the torso, head and limbs. The inference process uses Bayesian filtering to estimate the posterior probability distribution over figure states, consisting of recursive parameterizations of figure poses (relative joint DOF values and velocities) and associated information. Posterior distribution is represented by samples corresponding to individual state hypotheses. New state hypotheses are generated from past hypotheses by running motion predictors based on *physical simulation* and interpolation of training joint DOF data that define a prior over valid kinematic poses.

Prediction algorithms can exploit knowledge about the environment and incorporate the intentions (policy, goal) of the tracked individual into the prediction process. We assume that the segment shapes, mass properties, collision geometries and other associated parameters are known and remain constant throughout the sequence.

We present results that demonstrate the utility of using a physics-based prior for tracking, compare the method performance against other commonly used methods and show favorable performance under the effects of dynamic interaction exhibited in monocular and multi-view video.

2. Related Work

There has been a vast amount of work in computer vision in the past 10-15 years on articulated human motion tracking (we refer reader to [5] for more detailed review of the literature). Most approaches [1, 3, 13] have concentrated on development of efficient inference methods that are able to handle the high-dimensionality of a human pose. Generative methods typically propose to either learn a low-dimensional embedding of the high-dimensional kinematic data and then attempt to solve the problem in this more manageable low-dimensional space [15], or alternatively advocate the use of prior models to reduce effective search space in the original high-dimensional space [3]. More recent discriminative methods have attempted to go directly from image features to the 3D articulated pose from either monocular imagery [11, 14] or multiple views.

Producing smooth and accurate tracking remains a challenging problem, especially for monocular imagery. In particular, many of the produced results lack plausible physical realism and often violate the constraints imposed on the body by the world (resulting in out-of-plane rotations and foot skate). Such artifacts can be attributed to the general lack of physically plausible priors [2] (that can account for static and/or dynamic balance and ground-person-object interactions) which provide an untapped and very rich source of information.

The computer graphics and robotics community, on the other hand, has been very successful in developing realistic physical models of human motion. These models for the most part have only been developed and tested in the context of synthesis (*i.e.* animation [6, 10, 19, 17]) and humanoid robotics [18]. Here, we introduce a method that uses a full body physics-based dynamical model as a prior for articulated human motion tracking. This prior accounts for physically plausible human motion dynamics and environmental interactions, such as disallowing foot-ground penetration.

Earliest work on integrating physical models with vision-based tracking can be attributed to influential work by Metaxas *at el* [9] and Wren *at el* [16]. In both [9] and [16] a Lagrangian formulation of the dynamics was employed, within the context of a Kalman filter, for tracking of simple upper body motions using segmented 3D marker [9] or stereo [16] observations. In contrast, we incorporate full body human dynamical simulation into a Particle Filter, suited for multi-modal posteriors that commonly arise from ambiguities in monocular imagery. More recently, Brubaker *at el* [2] introduced a low-dimensional biomechanically-inspired model that accounts for human lower-body walking dynamics. The low-dimensional nature of the model [2] facilated the tractable inference; however, the model, while powerful, is inherently limited to walking motions in 2D.

In this work, we introduce a more general full-body model that can potentially model a large variety of human motions. However, the high-dimensionality of our model makes inference using standard techniques (*e.g.* particle filtering) challenging. To this end, we also introduce an exemplar-based prior for the dynamics to limit the effective search space and allow tractable inference in this high-dimensional space. Exemplar based methods similar to ours have been successfully used for articulated pose estimation in [11, 15], dynamically adaptive animation [20], and humanoid robot imitation [7]. Here, we extend the prior exemplar methods [11] to deal with exemplars that account for single-frame kinematics and dynamics of human motion.

3. Tracking with Dynamical Simulation

Tracking, including human motion tracking, is most often formulated as Bayesian filtering [4], which in computer vision literature is often implemented in the form of a Particle Filter (PF). In PF the *posterior*, $p(\mathbf{x}_f|\mathbf{y}_{1:f})$, where $\mathbf{x}_f$ is the state of the body at time instant f and $\mathbf{y}_{1:f}$ is the set of observations up to the time instant f, is approximated using a set of (typically) weighted samples/particles and is computed recursively, $p(\mathbf{x}_{f+1}|\mathbf{y}_{1:f}) \propto p(\mathbf{y}_{f+1}|\mathbf{x}_{f+1}) \int p(\mathbf{x}_{f+1}|\mathbf{x}_f) p(\mathbf{x}_f|\mathbf{y}_{1:f})\, \mathrm{d}\mathbf{x}_f$. In this formulation, $p(\mathbf{x}_f|\mathbf{y}_{1:f})$ is the posterior from the previous frame and $p(\mathbf{y}_{f+1}|\mathbf{x}_{f+1})$ is the *likelihood* that measures how well a hypothesis at time instant $f+1$ explains the observations; the $p(\mathbf{x}_{f+1}|\mathbf{x}_f)$ is often referred to as the *temporal prior* and is the main focus of this paper.

The temporal prior is often modeled as a first or second order linear dynamical system with Gaussian noise. For example, in [1, 3] the non-informative smooth prior $p(\mathbf{x}_{f+1}|\mathbf{x}_f) = \mathcal{N}(\mathbf{x}_f, \Sigma)$, which facilitates continuity in the recovered motions, was used; alternatively, constant velocity temporal priors of the form $p(\mathbf{x}_{f+1}|\mathbf{x}_f) = \mathcal{N}(\mathbf{x}_f + \gamma_f, \Sigma)$ (where γ_f is scaled velocity learned or inferred), have also been proposed [13] and shown to have favorable properties when it comes to monocular imagery. However, human motion, in general, is non-linear and non-stationary.

Physical Newtonian simulation is better suited as the basis for a temporal prior that addresses these issues. For simulation, our world abstraction consists of a known static environment and a loop-free articulated structure ("figure") representing the individual to be tracked. We assume "physical properties" (mass, inertial properties, and collision geometries) are known for each rigid body segment. Given these properties and a state hypothesis at frame f, we use constrained dynamics simulator within the "control loop" to predict the state at the next frame. Constraints are used to model various physical phenomena like interaction with the environment and to control the figure motion. Motion planning and control procedures incorporate training motion capture data in order to estimate the human's next intended pose and produce corresponding motion constraints that would drive the figure towards its intended pose. Similar to earlier methods, we add Gaussian noise (with diagonal covariance) to the dynamics to account for observation noise and minor physical disturbances.

3.1. Body Model and State Space

Our *figure* (body) consists of 13 rigid body segments and has a total of 31 degrees of freedom (DOFs), as illustrated in Figure 1. Segments are linked to parent segments by either 1-DOF (hinge), 2-DOF (saddle) or 3-DOF (ball and socket) rotational joints to ensure that only relevant rotations about specific joint axes are possible. The root segment is "connected" to the world space origin by a 6-DOF global "joint" whose DOF values define the global figure orientation and position. The values of rotational joint DOFs are encoded using Euler angles. Collision geometries attached to individual segments affect physical aspects of the motion. Segment shapes define visual appearance of the segments.

Joint DOF values concatenated along the kinematic tree define the *kinematic pose*, $\mathbf{q}$, of the figure. Joint DOF velocities, $\dot{\mathbf{q}}$, defined as the time derivatives, together with the kinematic pose $\mathbf{q}$ determine the figure's *dynamic pose* $[\mathbf{q}, \dot{\mathbf{q}}]$. The pose is considered *invalid* if it causes self-penetration of body parts and/or penetration with the environment, or if the joint DOF values are out of the valid ranges that are learned from the training motion capture data. These constraints on the kinematic pose allow us to reject invalid samples early in the filtering process.

The *control policy* information comprises of the identifier π of the policy type and the frame index υ the policy became effective. The policy type can either be *active* motion-capture-based (π^A) or *passive* (π^P). When the *passive* policy is in effect, no motion control takes place. The final *figure* state $\mathbf{x}$ is defined as a tuple $[\mathbf{q}, \dot{\mathbf{q}}, \pi, \upsilon]$, where $\mathbf{q} \in \mathbb{R}^{31}$, $\dot{\mathbf{q}} \in \mathbb{R}^{31}$, $\pi \in \{\pi^A, \pi^P\}$, $\upsilon \in \mathbb{N}^1$.

3.2. Likelihood

The likelihood function measures how well a particular hypothesis explains image observations $\mathbf{I}_f$. We employ a relatively generic likelihood model that accounts for silhouette and edge information in images [1]. We combine these two different feature types and across views (for multi-view sequences) using independence assumptions. Resulting likelihood, $p(\mathbf{I}_f|\mathbf{q}_f)$, of the kinematic pose, $\mathbf{q}_f$, at frame f can be written as,

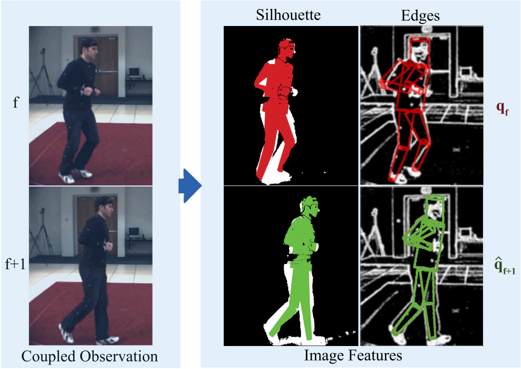

Figure 2. **Image Likelihood.** The *coupled observation* $\mathbf{y}_f$, consisting of two consecutive frames $\mathbf{I}_f$ (upper left) and $\mathbf{I}_{f+1}$ (lower left), matches the dynamic pose $[\mathbf{q}_f, \dot{\mathbf{q}}_f]$ well if features (silhouette and edges) at frame f fit the kinematic pose $\mathbf{q}_f$ (red pose) and features at frame $f+1$ fit the kinematic pose $\hat{\mathbf{q}}_{f+1}$ (green pose).

$$p(\mathbf{I}_f|\mathbf{q}_f) \propto \prod_{views} [p_{sh}(\mathbf{I}_f|\mathbf{q}_f)]^{w_{sh}} [p_{edge}(\mathbf{I}_f|\mathbf{q}_f)]^{w_{edge}}, \quad (1)$$

where $p_{sh}(\mathbf{I}_f|\mathbf{q}_f)$ and $p_{edge}(\mathbf{I}_f|\mathbf{q}_f)$ are the silhouette and edge likelihood measures defined as in [1], and w_{sh} and $w_{edge} = 1 - w_{sh}$ are *a priori* weighting parameters[1] for the two terms which account for the relative reliability between these two features.

Because our state carries both kinematic and velocity information, we model the likelihood of *dynamic pose* $[\mathbf{q}_f, \dot{\mathbf{q}}_f]$ using information extracted from both the current and the next frame; we refer to this as the *coupled observation* $\mathbf{y}_f = [\mathbf{I}_f, \mathbf{I}_{f+1}]$. We define the likelihood of the *coupled observation* as a weighted product of two kinematic likelihoods from above:

$$p(\mathbf{y}_f|\mathbf{x}_f) \propto p(\mathbf{I}_f|\mathbf{q}_f) p(\mathbf{I}_{f+1}|\hat{\mathbf{q}}_{f+1}), \quad (2)$$

where $\hat{\mathbf{q}}_{f+1} = \mathbf{q}_f + \Delta t \cdot \dot{\mathbf{q}}_f$ is the estimate of the kinematic state/pose at the next frame, assuming the Δt is the time between the two consecutive frames (see Figure 2).

This likelihood implicitly measures the velocity level information. Alternatively, one can formulate a likelihood measure that explicitly computes the velocity information [2] (*e.g.* using optical flow) and compares it to the corresponding velocity components of the state vector. Notice that portions of our state, $\mathbf{x}_f$, such as control policy, are inherently unobservable and are assumed to have uniform probability with respect to the likelihood function[2].

---

[1] For all of the experiments in this paper we use $w_{sh} = w_{edge} = 0.5$.

[2] The resulting dual-counting of observations, only makes the unnormalized likelihood more peaked, and can formally be handled as in [2].

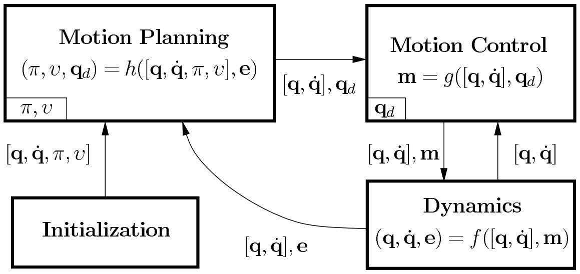
Motion Planning
$(\pi, \upsilon, \mathbf{q}_d) = h([\mathbf{q}, \dot{\mathbf{q}}, \pi, \upsilon], \mathbf{e})$
π, υ
$[\mathbf{q}, \dot{\mathbf{q}}], \mathbf{q}_d$
Motion Control
$\mathbf{m} = g([\mathbf{q}, \dot{\mathbf{q}}], \mathbf{q}_d)$
$\mathbf{q}_d$
$[\mathbf{q}, \dot{\mathbf{q}}, \pi, \upsilon]$
$[\mathbf{q}, \dot{\mathbf{q}}], \mathbf{m}$
$[\mathbf{q}, \dot{\mathbf{q}}]$
Dynamics
$(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{e}) = f([\mathbf{q}, \dot{\mathbf{q}}], \mathbf{m})$
Initialization
$[\mathbf{q}, \dot{\mathbf{q}}], \mathbf{e}$

Figure 3. **Prediction Model: Control Loop.** Components of the control loop and the data flow. Each iteration advances the figure state $[\mathbf{q}, \dot{\mathbf{q}}, \pi, \upsilon]$ by time Δ and records recent events $\mathbf{e}$ so they could be accounted for by the motion planner at the next iteration. The little boxes within the components represent "memory locations" holding component-specific state information preserved across component exits.

3.3. Prediction

Prediction takes a potential figure state and estimates what its value at the next frame would be if the state's evolution followed a certain motion model. We assume that human motion is governed by dynamics and by a thought process that tasks the figure "muscles" so that desired motion would be performed. Our motion model idealizes this process and models the state evolution by executing the "control loop" outlined in Figure 3.

Given a figure state $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}, \pi, \upsilon]$ and a vector of simulation events[3] $\mathbf{e}$ that occured during the previous loop iteration, the *motion planner* decides what the next control policy π will be and, depending on the policy, proposes next desired kinematic pose $\mathbf{q}_d$ that the figure should follow. This desired pose is then processed by the *motion controller* to set up a set of motion constraints[4], $\mathbf{m}$, that need to be honored by the *dynamics simulator* when updating the dynamic pose $[\mathbf{q}, \dot{\mathbf{q}}]$. Motion constraints implicitly generate motor forces to actuate the figure. As a simpler alternative to constraints, the motion controller could generate motor forces directly by a *proportional-derivative servo* [17].

The actual prediction consists of initializing the model from the given initial state $\mathbf{x}$, looping through the control loop for the time duration of the frame, Δt, (this might take several iterations of size $\Delta \ll \Delta t$) and returning the state $\mathbf{x}$ at the end of the frame.

3.3.1 Motion Planning

The motion planner, denoted by the function h in Figure 3, allows the incorporation of different motion priors into the prediction process. It is responsible for picking a control policy π (using the information about the figure state $\mathbf{x}$ and the feedback $\mathbf{e}$), updating the frame index υ since the policy was in effect and generating a desired kinematic pose $\mathbf{q}_d$ for the motion controller using an algorithm specific to the policy, if applicable. New policies π_{f+1} are sampled from simple distributions $p(\pi_{f+1}|\pi_f, \mathbf{e}_f)$ that can depend on the duration of time the current policy π_f has been in effect; for each potential value of $\mathbf{e}_f$ and π_f there is one such distribution[5]. Two control policies have been implemented so far, the *active* motion-capture based policy and the *passive* motion policy.

Passive motion. This policy lets the figure move passively as if it was unconscious, and as a result no $\mathbf{q}_d$ is generated when in effect. Its purpose is to account for unmodeled dynamics in the motion-capture based policy and it should typically be activated for short periods of time.

Active motion. Our motion capture based policy actuates the figure so that it would perform a motion similar to the one seen in training motion capture data. We take an exemplar based approach similar to that of [7, 11, 20]. To that end, we first form a database of observed input-output pairs (from training motion capture data) between a dynamic pose at frame f and a kinematic pose at frame $f+1$, $\mathcal{D} = \{[\mathbf{q}^*_f, \dot{\mathbf{q}}^*_f], \mathbf{q}^*_{f+1}\}_{f=1}^{n}$. For pose invariance to absolute global position and heading, corresponding degrees of freedom are removed from $\mathbf{q}^*_f$ and $\dot{\mathbf{q}}^*_f$. Given this database, that can span training data from multiple subjects and activities, our objective is to determine the intended kinematic pose $\mathbf{q}_d$ given a new dynamic pose $[\mathbf{q}, \dot{\mathbf{q}}]$. We formulate this as in [11] using a K nearest neighbors (k-NN) regression method, where a set of similar prototypes/exemplars to the query point $[\mathbf{q}, \dot{\mathbf{q}}]$ are first found in the database and then the $\mathbf{q}_d$ is obtained by weighted averaging over their corresponding outputs; the weights are set proportional to the similarity of the prototype/exemplar to the query point. This can be formally written as,

$$\mathbf{q}_d = \sum_{[\mathbf{q}^*_f, \dot{\mathbf{q}}^*_f] \in neighborhood[\mathbf{q}, \dot{\mathbf{q}}]} \mathcal{K}(d_f([\mathbf{q}^*_f, \dot{\mathbf{q}}^*_f], [\mathbf{q}, \dot{\mathbf{q}}])) \cdot \mathbf{q}^*_{f+1},$$

where $d_f([\mathbf{q}^*_f, \dot{\mathbf{q}}^*_f], [\mathbf{q}, \dot{\mathbf{q}}])$ is the similarity measure and $\mathcal{K}$ is the *kernel* function that determines the weight falloff as a function of distance from the query point.

We use a similarity measure that is a linear combination of positional and velocity information,

$$d_f([\mathbf{q}^*_f, \dot{\mathbf{q}}^*_f], [\mathbf{q}, \dot{\mathbf{q}}]) = w_\alpha \cdot d_M(\mathbf{q}, \mathbf{q}^*_f) + w_\beta \cdot d_M(\dot{\mathbf{q}}, \dot{\mathbf{q}}^*_f),$$

where $d_M(\cdot)$ denotes a Mahalanobis distance between $\mathbf{q}$ and $\mathbf{q}^*_f$, and $\dot{\mathbf{q}}$ and $\dot{\mathbf{q}}^*_f$, respectively with covariance matrices learned from the training data, $\{\mathbf{q}^*_f\}_{f=1}^{n}$ and $\{\dot{\mathbf{q}}^*_f\}_{f=1}^{n}$; the w_α and w_β are positive constants that account for the relative weighting of the two terms. For the kernel function we use a simple Gaussian, $\mathcal{K} = \mathcal{N}(0, \sigma)$, with empirically determined variance σ^2.

---

[3] Currently, corresponding to a binary indicator variable determining whether a collision with environment has occured.

[4] In case no desired kinematic pose was proposed, $\mathbf{m} = \emptyset$.

[5] These discrete conditional distributions are defined empirically.

3.3.2 Motion Control

The motion controller g in Figure 3 conceptually approximates the human's muscle actuation to move the current pose hypothesis $[\mathbf{q}, \dot{\mathbf{q}}]$ towards the intended kinematic pose $\mathbf{q}_d$ when the figure state is updated by dynamics. We formulate motion control as a set of soft constraints on $\mathbf{q}$ and $\dot{\mathbf{q}}$. Each constraint is defined as an equality or inequality with a softness constant determining what portion of the constraint force should actually be applied to the constrained bodies. Constraints can also limit force magnitude to account for biomechanical properties of the human motion, like muscle power limits or joint resistance.

Unlike traditional constraint-based controllers [8], we *do not directly control (constrain) the position of the figure root* so that global translation will result only from the figure's interaction with the environment (contact)[6]. This introduces several problems that require a new approach to motion control. Consider the case where the desired kinematic pose $\mathbf{q}_d$ is infeasible (*e.g.* causing penetration with the environment). Leaving the linear DOFs unconstrained, in this case, often leads to unexpected contacts/impacts with an environment during simulation which can affect the motion adversely[7]. To address these problems, we propose a new kind of hybrid constraint-based controller (see Figure 4) that aims to follow desired joint angles as well as trajectories of selected markers (points) defined on the figure segment geometries. The controller takes as input dynamic pose $[\mathbf{q}, \dot{\mathbf{q}}]$ and desired kinematic pose $\mathbf{q}_d$ and outputs a set of desired angular velocities $\dot{\mathbf{q}}_d$ obtained using inverse dynamics.

Given the desired kinematic pose $\mathbf{q}_d$ and positions $\mathbf{z}^j$ of markers on selected figure segments (toes), the controller first computes the marker positions with respect to the desired pose (using forward kinematics), $\mathbf{z}^j_d$. These positions are then adjusted so that they do not penetrate the environment. The adjusted positions $\mathbf{z}^j_d$ produce requests on desired positions of markers $\mathbf{z}^j$, which are subsequently combined with requests on desired values of joint angles $\mathbf{q}^k$ at other figure segments (with no associated markers). Finally, these requests are converted to constraints $\mathbf{m} = \{\dot{\mathbf{q}}^i = \dot{\mathbf{q}}^i_d\}$ on angular velocities that are passed to the simulator.

This process is implemented using first order inverse dynamics on a helper figure, where position and orientation requests serve as inverse dynamics goals; we fix the root segment in the helper figure to ensure that these goals can not be solved by simple translation or rotation of the root segment. The process consists of the following steps. First, the pose of the helper figure is set up to mirror the current pose $[\mathbf{q}, \dot{\mathbf{q}}]$ of the original figure. Next, given the value of $c_\alpha > 0$ determining how fast the controller should approach the desired values, the requests on desired positions of markers are converted to soft constraints on desired marker velocities $\dot{\mathbf{z}}^j = -c_\alpha \cdot (\mathbf{z}^j - \mathbf{z}^j_d)$, and the requests on desired joint angles at other segments are converted to soft constraints on desired joint angle velocities $\dot{\mathbf{q}}^k = -c_\alpha \cdot (\mathbf{q}^k - \mathbf{q}^k_d)$. These constraints are finally combined with additional constraints on joint angle limits $\mathbf{q}^i \geq \mathbf{q}^i_{min}$ and $\mathbf{q}^i \leq \mathbf{q}^i_{max}$; the constraints are solved and final desired angular velocities, $\dot{\mathbf{q}}^i_d$, are obtained. The last step is implemented by using the facilities of the physics engine.

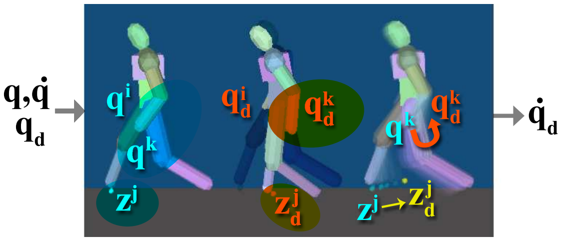

Figure 4. **Motion Controller.** Input kinematic pose $\mathbf{q}$ determines the positions $\mathbf{z}^j$ of markers on the feet (left), the desired kinematic pose $\mathbf{q}_d$ their desired positions $\mathbf{z}^j_d$ (middle). Desired positions are adjusted to prevent penetration with the ground and constraints on the marker velocities $\dot{\mathbf{z}}^j$ and joint DOF derivatives $\dot{\mathbf{q}}^k$ of the helper figure are formed (right). Superscripts ^{i} index the figure's angular DOFs, superscripts ^{j} the markers and superscripts ^{k} the angular DOFs of the figure segments that have no markers ^{j} attached.

3.3.3 Dynamical Simulation

The dynamical simulator, denoted by (with slight abuse of notation) function f in the control loop, numerically integrates an input dynamic pose forward in time based on Newtonian equations of motion and specified constraints. We use the Crisis physics engine [21] which provides facilities for constraint-based motion control and implements certain features suitable for motion tracking. The simulator's collision detection library is used to validate poses[8].

The simulation state is advanced by time Δ by following standard Newton-Euler equations of motion, while obeying a set of constraints — the explicit motion control constraints $\mathbf{m}$, soft position constraints $\mathbf{q}^i \geq \mathbf{q}^i_{min}$ and $\mathbf{q}^i \leq \mathbf{q}^i_{max}$ due to angular DOFs i implementing joint angle limits, and implicit velocity or acceleration constraints enforcing non-penetration and modeling friction. Because constraints, $\mathbf{m}$, are valid only with respect to a specific dynamic pose, the constraints have to be reformulated each time the state is internally updated by the simulator. As a result, motion controller can be called back throughout the simulation process.

---

[6] However, the orientation of the root segment is constrained, which implements balancing. Although this is not physically correct, because the orientation can change regardless of the support from the rest of the body, it serves our purpose well.

[7] For example, unwanted impacts at the end of the walking cycle will force the figure to step back instead of forward.

[8] When noise is added to a kinematic pose, it has to be determined whether the proposed pose is *valid* according to the metrics discussed in Section 3.1.

This is illustrated by the corresponding arrows in Figure 3. Once the simulation completes, the dynamic pose $[\mathbf{q}, \dot{\mathbf{q}}]$ matching the resulting state of the physical representation is returned. In order to provide feedback about events in the simulated world for the motion planner ("perception"), recent simulation events (see footnote 3) are recorded into $\mathbf{e}$, which is returned together with the updated pose.

4. Experiments

Datasets. In our experiments we make use of the two publicly available datasets that contain synchronized motion capture (MoCap) and video data from multiple cameras (@60 Htz). The use of this data allows us to (1) quantitatively analyze the performance (by treating MoCap as ground truth), and (2) obtain reasonable initial poses for the first frame of the sequence from which tracking can be initiated. The first dataset, used in [1], contains a single subject (L1) performing a walking motion with stopping, imaged with 4 grayscale cameras (see Figure 8). The second, HUMANEVA dataset [12] (see Figure 7), contains three subjects (S1 to S3) performing a variety of motions (*e.g.* walking, jogging, boxing) imaged with 7 cameras (we, however, make use of the data from at most 3 color cameras for our experiments). Each dataset contains disjoint training and testing data, that we use accordingly.

Error. To quantitatively evaluate the performance we make use of the metric employed in [1] and [12], where pose error is computed as an average distance between a set of 15 markers defined at the key joints and end points of the limbs. Hence, in 3D this error has an intuitive interpretation of the average joint distance, in (mm), between the ground truth and recovered pose. In our monocular experiments, we use an adaptation of this error, that measures the average joint distance with respect to the position of the pelvis to avoid biases that may arise due to depth ambiguities. For tracking experiment, we report the error of the expected pose[9].

Prediction. The key aspect of our physics-based prior is the ability to perform accurate physically-plausible predictions of the future state based on the current state estimates. First, we set out to test how our prediction model compares, quantitatively, with the standard prediction models based on stationary linear dynamics described in Section 3.

Figure 6 (right) shows performance of the smooth prior (No Prediction), constant velocity prior, and individual predictions based on the two control strategies implemented within our physics-based prediction module. For all 4 methods we use 200 frames of motion capture data from the L1 sequence to predict poses from 0.05 to 0.5 seconds ahead. We then compare our predictions to the poses observed by motion capture data at corresponding times.

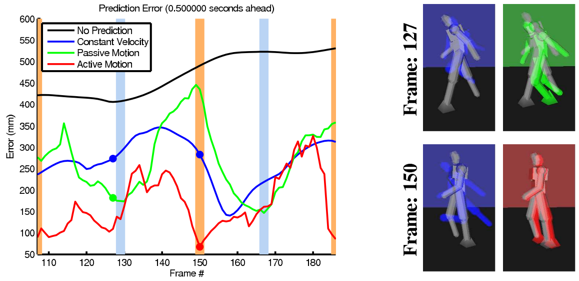

Figure 5. **Prediction Error.** Error in predictions (0.5 seconds ahead) are analyzed as a function of one walking cycle. Vertical bars illustrate different phases of walking motion: light blue – foot hits the ground, light orange – change in the direction of the arm swing. Notice that passive and dynamic predictions have complementary behavior during different motion phases (right).

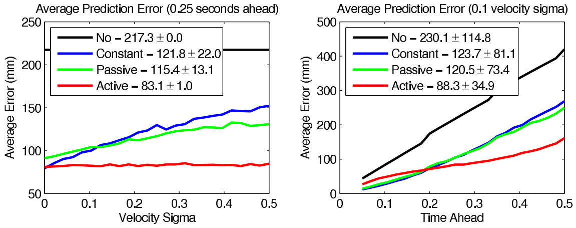

Figure 6. **Average Prediction Error.** Illustrated, on the right, is the quantitative evaluation of 4 different dynamical priors for human motion: smooth prior (No Prediction), constant velocity prior and (separately) active and passive physics-based priors implemented here. On the left, performance in the presence of noise is explored. See text for further details.

For short temporal predictions all methods perform well; however, once the predictions are made further into the future, our *active motion* control strategy, based on exemplar-based MoCap method, significantly outperforms the competitors. Overall, the active motion control strategy achieves 29% lower performance error over the constant velocity prior (averaged over the range of prediction times from 0.05 to 0.5 seconds).

Figure 6 (left) shows the effect of noise on the predictions. For a fixed prediction time of 0.25 seconds, a zero mean Gaussian noise is added to each of the ground truth dynamic poses before the prediction is made. The performance is then measured as a function of the noise variance. While performance of the constant velocity prior and passive motion prior degrade with noise, the performance of our active motion prediction stays low and flat.

Notice that the constant velocity prior performs similarly to the passive motion; intuitively, this makes sense since the constant velocity prior is an approximation to the passive motion dynamics, that does not account for environment interactions. Since such interactions happen infrequently and we are averaging over 200 frames, the differences between the two methods are not readily observed, but are important at the key instants when they occur (see Figure 5).

---

[9] Other error metrics such as *optimistic* error [1] and error of *maximum a posteriori* (MAP) pose estimate produce very similar results.

Tracking with multiple views. We now test the performance of the Bayesian tracking framework that incorporates the physics-based prior considered above in the context of multi-view tracking using a 200 frame, 4 view, image sequence of L1. We first compare the performance of the proposed physics-based prior method (L1), to two standard Bayesian filtering approaches that employ smooth temporal priors, Particle Filtering[10] (PF) and Annealed Particle Filter[10] with 5 levels of annealing (APF 5). To make the comparison as fair as possible we use the same number of particles[11] (250), same likelihoods, and same interpenetration and joint limit constraints in all cases; joint limit constraints are learned from training data. The quantitative results are illustrated in Figure 9 (left). Our method has 72% lower error then PF and 47% lower error then APF, as well as considerably lower variance. Qualitative visualization of results analyzed in Figure 9 is not shown due to lack of space; typical performance, on HUMANEVA sequence (with error 93.4 ± 24.8), is illustrated in Figure 7.

We have also tested how performance of our method degrades with larger training sets that come from other subjects performing similar (walking) motions (see Physics S1-S3 L1). It can be seen that additional training data does not noticeably degrade the performance of our method, which suggests that our approach is able to scale to large datasets. We also test whether or not our approach can generalize, by training on data of subjects from HUMANEVA dataset and running on a different subject, L1, from the dataset of [1] (Physics S1-S3). The results are encouraging in that we can still achieve reasonable performance that has lower error then PF and APF (noise and joint levels of which were trained using subject specific data of L1). While due to the exemplar-based nature of our active controller it is likely that our method would not be able to generalize to unobserved motions, our experiments tend to indicate that it can generalize within observed classes of motions given sufficient amount of training data.

Monocular Tracking. The most significant benefit of our approach is that it can deal with monocular tracking. Physical constraints embedded in our prior help to properly place the hypotheses and avoid overfitting of image evidence that in the monocular case lack 3D information (see Figure 8 (Physics)); the results from PF and APF on the other hand tend to overfit the image evidence, resulting in physically implausible 3D hypothesis (see Figure 8 (APF 5) bottom) and lead to more severe problems with local optima (see Figure 8 (APF 5) top). Figure 8 (Physics) bottom, illustrates the physical plausibility of the recovered 3D poses using our approach. Quantitatively, on the monocular sequence, our model has 71% lower error then PF and 74% lower error then APF, with once again considerably lower (roughly $\frac{1}{3}$ to $\frac{1}{4}$) variance (see Figure 9 right).

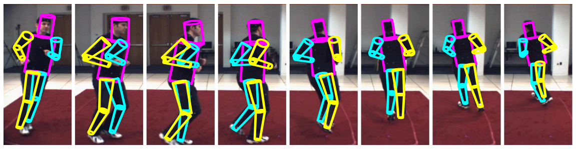

Figure 7. **Multi-view Tracking.** Tracking performance on the Jog sequence of subject S3 form HUMANEVA dataset; 250 particles are used for tracking. Illustrated is the projection of the tracked model into one of the 3 views used for inference.

Analysis of computation time. While the tracking framework was implemented in Matlab, the Physics prediction engine was developed in C++. As a result, the overhead imposed by the physics simulation and motion control is negligible with respect to the likelihood[12] computation. The overhead imposed by the motion planning is a function of the number of training examples; in our experiments corresponding to 11–20%. The sub-linear approximations to k-NN regression [11] can make this more tractable for large datasets. The raw per particle computations in seconds for each of the approaches are: PF – 0.0280, APF 5 – 0.1525, Physics (no motion planning) – 0.0560, Physics L1 – 0.0624, Physics S1, S2, S3, L1 – 0.0672.

5. Discussion and Conclusions

We presented a framework that incorporates the full-body physics-based constrained simulation, as a temporal prior, into the articulated Bayesian tracking. As a result, we are able to account for non-linear non-stationary dynamics of the human body and interactions with the environment (*e.g.* ground contact). To allow tractable inference we also introduce two controllers: a novel hybrid constraint-based controller, which uses motion-capture data to actuate the body, and a passive motion controller. Using these tools, we illustrate that our approach can better model the dynamical process underlying human motion, and achieve physically plausible tracking results using multi-view and monocular imagery. We show both qualitatively and qualitatively that the resulting tracking performance is more accurate and natural (physically plausible) than results obtained using standard Bayesian filtering methods such as Particle Filtering (PF) or Annealed Particle Filtering (APF). In the future, we plan to explore richer physical models and control strategies, which may further loosen the current reliance of our

---

[10] We make use of the public implementation by Balan *et al.* [1] available from `http://www.cs.brown.edu/people/alb/`.

[11] In APF we use 250 particles for each annealing layer.

[12] The likelihood evaluations, however, in our framework involve computing the likelihood over two frames (rather than one in PF) and hence are twice as expensive; the number of likelihood evaluations in APF is a function of the number of layers.

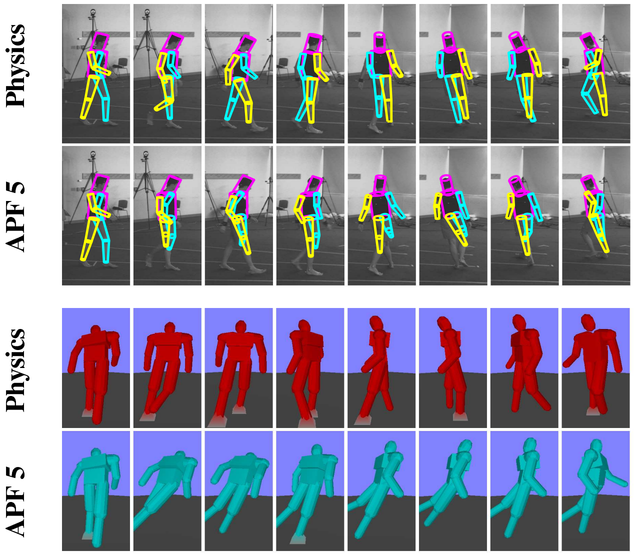

Figure 8. **Monocular Tracking.** Visualization of performance on a monocular walking sequence of subject L1. Illustrated is the performance of the proposed method (Physics) versus the Annealed Particle Filter (APF 5); in both cases with 1000 particles. The top row shows projections (into the view used for inference) of the resulting 3D poses at 20-frame increments; bottom shows the corresponding rendering of the model in 3D along with the ground contacts. Our method, unlike APF, does not suffer from out-of-plane rotations and has consistent ground contact pattern. For quantitative evaluation see Figure 9 (right).

method on motion-capture training data.

Acknowledgments. This work was supported in part by ONR Award N000140710141. We wish to thank Michael J. Black for valuable contributions in the early stages of this project; Alexandru Balan for the PF code; David Fleet, Matt Loper and reviewers for useful feedback on the paper itself; Morgan McGuire and German Gonzalez for useful discussions; Sarah Jenkins for proofreading.

References

[1] A. Balan, L. Sigal and M. J. Black. A Quantitative Evaluation of Video-based 3D Person Tracking, *IEEE VS-PETS Workshop*, pp. 349–356, 2005.

[2] M. Brubaker, D. J. Fleet and A. Hertzmann. Physics-based person tracking using simplified lower-body dynamics, *CVPR*, 2007.

[3] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision (IJCV)*, Vol. 61, No. 2, pp. 185–205, 2004.

[4] A. Doucet, N. de Freitas and N. Gordon. Sequential Monte Carlo methods in practice, *Statistics for Engineering and Infoormation Sciences*, Springer Verlag, 2001.

[5] D. A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien and D. Ramanan. Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis, *ISBN: 1-933019-30-1*, 178pp, July 2006.

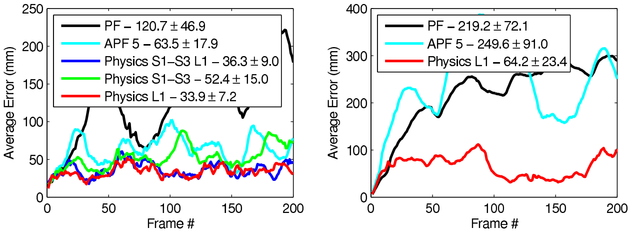

Figure 9. **Quantitative Tracking Performance.** Multi-view tracking performance of the proposed physics-based prior filtering method (Physics) with different training datasets versus standard Particle Filter (PF) and Annealed Particle Filter (APF 5) with 5 layers of annealing is shown on the left. Tracking performance using monocular sequence is analyzed on the right. In both cases L1 walking sequence was used; in the case of multi-view tracking with 4 cameras and 250 particles, in the case of monocular setup with single camera view and 1000 particles. See text for details.

[6] J. Hodgins, W. Wooten, D. Brogan and J. O'Brien. Animating human athletics, *ACM SIGGRAPH*, pp. 71-78, 1995.

[7] O. C. Jenkins and M. J. Mataric. Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion, *International Journal of Humanoid Robotics*, 1(2):237–288, 2004.

[8] E. Kokkevis. Practical Physics for Articulated Characters, *Game Developers Conference*, 2004.

[9] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis, *PAMI*, 15(6), pp. 580–591, June, 1993.

[10] Z. Popovic and A. Witkin. Physically Based Motion Transformation, *ACM SIGGRAPH*, 1999.

[11] G. Shakhnarovich, P. Viola and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing, *ICCV*, Vol. 2, pp. 750–757, 2003.

[12] L. Sigal, M. J. Black HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion. *Techniacl Report CS-06-08*, Brown U., 2006.

[13] H. Sidenbladh and M. J. Black. Learning image statistics for Bayesian tracking. *ICCV*, Vol. 2, pp. 709-716, 2001.

[14] C. Sminchisescu, A. Kanaujia, Z. Li and D. Metaxas. Discriminative Density Propagation for 3D Human Motion Estimation, *CVPR*, Vol. 1, pp. 390–397, 2005.

[15] R. Urtasun, D. J. Fleet, A. Hertzmann, P. Fua. Priors for People Tracking from Small Training Sets, *ICCV*, 2005.

[16] C. R. Wren and A. Pentland. Dynamic Models of Human Motion, *Automatic Face and Gesture Recognition*, 1998.

[17] P. Wrotek, O. Jenkins and M. McGuire. Dynamo: Dynamic Data-driven Character Control with Adjustable Balance, *ACM SIGGRAPH Video Game Symposium*, 2006.

[18] K. Yamane and Y. Nakamura. Robot Kinematics and Dynamics for Modeling the Human Body, *Intl. Symp. on Robotics Research*, 2007.

[19] K. Yin, K. Loken and M. van de Panne. SIMBICON: Simple Biped Locomotion Control, *ACM SIGGRAPH*, 2007.

[20] V. Zordan, A. Majkowska, B. Chiu, M. Fast. Dynamic Response for Motion Capture Animation, *SIGGRAPH*, 2005.

[21] `http://crisis.sourceforge.net/`