Real-time Physics-based Motion Capture with Sparse Sensors

Sheldon Andrews
Disney Research
Edinburgh, UK

Ivan Huerta
Disney Research
Edinburgh, UK

Taku Komura
University of
Edinburgh
Edinburgh, UK

Leonid Sigal
Disney Research
Pittsburgh, USA

Kenny Mitchell*
Disney Research
Edinburgh, UK

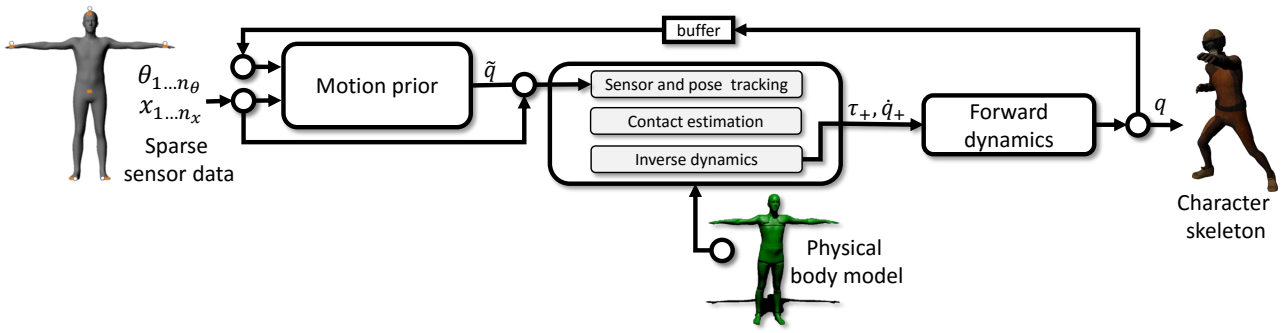

Figure 1: High-level components of the tracking framework. IMU and optical marker data are calibrated to a body model that is generated for the capture subject. An inverse dynamics solver generates motion that satisfies the orientation and position constraints introduced by the sensors, as well as pose constraints from the motion prior. Finally, the pose of a character rig is updated using a forward dynamics simulation.

ABSTRACT

We propose a framework for real-time tracking of humans using sparse multi-modal sensor sets, including data obtained from optical markers and inertial measurement units. A small number of sensors leaves the performer unencumbered by not requiring dense coverage of the body. An inverse dynamics solver and physics-based body model are used, ensuring physical plausibility by computing joint torques and contact forces. A prior model is also used to give an improved estimate of motion of internal joints. The behaviour of our tracker is evaluated using several black box motion priors. We show that our system can track and simulate a wide range of dynamic movements including bipedal gait, ballistic movements such as jumping, and interaction with the environment. The reconstructed motion has low error and appears natural. As both the internal forces and contacts are obtained with high credibility, it is also useful for human movement analysis.

Keywords

motion capture; character animation; inverse dynamics

1. INTRODUCTION

There is an increasing demand to capture human motion in natural and conventional settings. For example, with actors wearing costumes, animation pre-visualization using an adhoc or lightweight setup, and virtual reality gaming create interesting engineering and research challenges for motion capture since traditional approaches are often unsuitable. Optical tracking systems [2, 4] perform poorly when there are occlusions, causing problems when there are multiple actors or obstacles blocking marker visibility. Costumes and close interactions may likewise restrict the placement and visibility of optical markers. Other tracking solutions use inertial measurement units (IMUs) [3, 5] for motion capture without concern for occlusions and visibility obstruction. But they suffer from drift resulting in inaccurate position and orientation measurements over time. Some of these problems can be overcome by using a large number of sensors. However, this can limit the range of movement of the capture subject, or results in an unwieldy amount of capture equipment. Tracking with a minimal, light weight configuration of sensors is therefore desirable.

The goal of our work is to develop a tracking framework capable of high quality motion capture with only a small number of sensors. In this paper, we propose a multimodal sensor configuration for tracking human motion. This combines the benefits of marker based and markerless tracking systems. An additional objective is to reconstruct the motion in real-time, allowing our framework to be useful for a number of interactive applications such as cinematic pre-visualization, gaming, and virtual reality.

Data from a sparse set of optical markers and inertia-based sensors is fused using a physics-based body model, ensuring that the resulting motion is physically plausible. Furthermore the lack of tracking information, due to sensor sparsity, is compensated by combining a pose estimate from a black box motion prior within the same physics-based tracking framework. This helps to estimate the movements of body parts which are not actively tracked.

Our solver simultaneously computes a plausible motion, the internal torques of the body, and the contact forces between

*Email: kenny.mitchell@disneyresearch.com

CVMP '16, December 12 - 13, 2016, London, United Kingdom

ISBN 978-1-4503-4744-0/16/12...$15.00

DOI: http://dx.doi.org/10.1145/2998559.2998564

the body and the environment. The human body inertia is estimated using a biomechanical model and scanned geometry. The tracking method imposes physical constraints such as conservation of momentum, ground reaction force and maximum torque at the joints to estimate plausible movements of the body. The preservation of momentum plays an important role in ballistic movements, such as running, jumping, cartwheeling and flipping. Limitation of joint torques and position also prevents the body from moving beyond the capability of maximum muscle forces and kinematic limits.

Our system can track and simulate a wide range of dynamic movements including bipedal gait and acrobatic movements such as jumping and high-kicking in real-time, which is useful for live performance capture and animation synthesis. As both the internal forces at the joints and contacts can be obtained with high credibility, our system may also be useful for human movement analysis or animation retargeting.

The rest of the paper is organized as follows: in the remainder of this section, we introduce the related work and give an overview of our system. In Section 2, we provide details about the physically-based motion tracking system. In Section 3, we describe the motion priors used to evaluate our system. In Section 4, we present the experimental results and evaluate the system using several motion sequences. Finally, the paper concludes in Section 5 with a discussion of future work directions.

1.1 Related Work

Existing techniques for physically based tracking and motion capture with heterogeneous sensor combinations are reviewed in this section. As tracking body movements using motion priors is part of our work, previous work that incorporates a motion prior for body tracking is also reviewed.

Body Tracking by Inertial Sensors Slyper and Hodgins [27] use acceleration data obtained from an array of low-cost accelerometers to reconstruct motion based on similarity to samples in a large motion capture database. We evaluate our motion tracking framework with a comparable motion prior. Kruger et al. [19] conduct a nearest neighbour search using KD-trees for motions that corresponds to sensor input. Tautges et al. [28] extend this approach and propose a data structure for efficient lookup of movements from accelerometer readings. These methods require a significant amount of data preprocessing for classifying the data and constructing a data structure for successfully tracking the motion on-the-fly. Also, it requires a significant amount of memory usage for saving various types of movements. Liu et al. [22] use a small number of IMUs and Bayes estimator trained on a motion capture database to reconstruct full body motion. Their approach gives an improvement over IK based tracking and PCA based reconstruction methods [11].

Sparse and Multi-modal Sensor tracking As inertial sensors do not provide absolute position data, they are often augmented by other sensors for accurate motion reconstruction without positional drift. Vlasic et al. [29] use ultrasonic sensors that provide absolute distance between the sensors to significantly reduce drift compared to a purely inertial capture setup. Von Marcard et al. [30] use video data to augment the IMUs to estimate the full body motion. We combine IMUs with optical markers, capturing with only a small number of each type of sensor. Furthermore, motion is reconstructed at real-time frame rates. Schwarz et al. [25] use activity specific motions priors to constrain the tracking of inertial sensors, whereas our work uses a physical body model to ensure physical plausiblity. Tracking of sparse sensor data has also been used for reconstruction of hand motions [16,17].

Physically-based Motion Tracking. Another prior that can be applied for augmenting incomplete inertial data is the physical prior, that can be applied to examine if the synthesized motion follows physical laws. Simulating the tracked motion data requires exerting torques at the body joints such that every body part follows the marker data or the kinematics of the captured data. Ha et al. [15] solve an optimization problem that computes the body motion such that it satisfies physical laws while tracking the sensor data. They use pressure sensors to evaluate the ground reaction force made between between the feet and the ground. Zhang et al. [32] put pressure sensors in the sole of the shoes to allow the body to freely move around in the environment. Usually obtaining the pressure between the body and the environment at arbitrary location on the body is difficult. Our approach estimates contacts based on motion and a geometry from a body model and the results are comparable. Lee et al. [20] modulates the motion data continuously and seamlessly to track the motion that is captured by accurate optical markers. Liu et al. [23] propose a sampling-based approach that selects the optimal series of movements for tracking the optical markers. We wish to avoid fully relying on optical markers that suffer from occlusion problems, and thus use inertia sensors to also track the body movements while imposing physical plausibility. Vondrak et al. [31] perform monocular 3D pose estimation by combining video based tracking with a dynamical simulation. Like our work, their approach is able to track a variety of motion styles and estimate joint torques and contact forces. However the single camera sensor setup is prone to occlusion problems, and their method does not achieve real-time frame rates.

It is worth mentioning the recent system proposed by Dou et al. [13] that uses RGBD images from multiple camera for real-time motion capture that is robust to dynamic motion. However, the capture distance is limited due to the use of depth sensors, and requires good visibility of the full body.

1.2 System Outline

Fig. 1 shows the pipeline used by our real-time body solver. Briefly, a combination of IMU and optical marker sensors are calibrated to a physics-based body rig which is generated for the capture subject. An inverse dynamics solver is then used to solve for motion that satisfies the orientation and position constraints introduced by the sensors, as well as pose constraints from a black box motion prior. Finally, the pose of a skeletal character rig is updated using a forward dynamics simulation.

Terminology. We use $x \in \mathbb{R}^3$ to refer to positions of individual markers and $\theta \in \mathbb{R}^4$ to refer to quaternion orientations of individual IMUs. The skeletal DOF vector $\mathbf{q} \in \mathbb{R}^m$ gives the pose of the tracking skeleton (or body model) and in our case consists collectively of the 3D position and orientation at the skeleton's root and three Euler angles for each non-root bone. The pose estimate from the motion prior is $\tilde{\mathbf{q}}$, and body torques used to actuate the body model are τ. Unless otherwise noted, all terms refer to values at the current frame.

2. PHYSICS-BASED BODY TRACKING

A key aspect of our solver is that a physics-based framework is used to track the motion of the capture subject. By using

inverse dynamic techniques to solve for full body motion, we not only ensure that motions remain physically plausible, but additional details may be extracted such as contact information and body forces. In this section we provide details on how the sensor data is fused and then tracked using inverse dynamics and a physics-based body model.

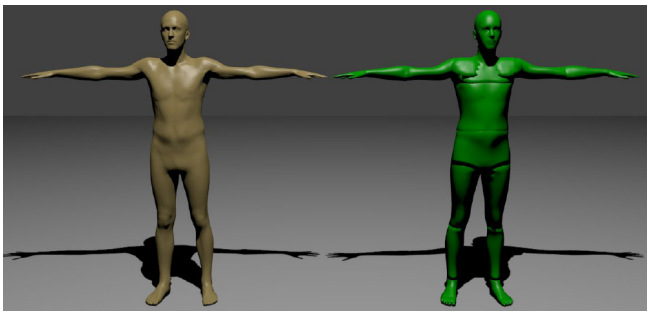

Figure 2: An unsegmented body mesh generated using depth and RGB images (left); the segmented and rigged body model used by our solver (right). The physics-based body model provides mass and geometry distribution of the tracked user.

2.1 Body Model

Fig. 2 shows an example body model used in our experiments. The model provides a skeleton of the capture subject, in addition to mass and geometry distribution of bone segments. Except for the root segment, each bone is articulated by a rotational joint with three angles, for a total of 96 DOFs in the skeleton. A mesh associated with the bone approximates the surface of the corresponding body part. Furthermore, a mass and inertia for the body segment is computed using the volume of this mesh and the total mass of the capture subject; a uniform density is assumed. The mass is distributed across body segments according to statistics found in the biomechanics literature [7]. By using a body model that more accurately represents the mass and geometry of the capture subject, we gain a better physical representation, which can be leveraged by an inverse dynamics tracking technique. This is an important distinction when comparing to inverse kinematics based approaches. Massive bodies are more difficult to accelerate, and this fundamental behaviour is captured by our approach.

Acquiring body geometry using consumer level hardware is becoming increasingly common. The body mesh in Fig. 2 was created using depth and RGB images and generated using a third party application[1]. However, the body model may be generated by a variety of other methods, such as photogammetric and depth sensor reconstruction [26], employing statistical models [6], or by manual artistic effort.

For the model shown on the right in Fig. 2, a post-processing step was used to segment the mesh. As the skeleton moves, the segmented geometry is updated using a single rigid transform, which is obtained readily from the skeleton bodies. Alternatively, a skinning algorithm could be used to smoothly interpolate the unsegmented geometry based on the skeleton's bone transforms. However, we have not found this to have any significant effect on the contact estimation where geometric detail is most important, and thus rig the body model to update the segmented geometry from a single bone.

[1]Body{SNAP}. http://www.bodysnapapp.com/

Limits for body torques and joint angles are also estimated during body model construction. A dense collection of optical markers is placed on the capture subject and their motion is tracked using inverse dynamics and a skeleton with geometry and mass as described above. Motions that explore a large range of poses, as well as explosive and ballistic motions that produce large joint torques, are captured and used to compute $[\mathbf{q}_{lo}, \mathbf{q}_{hi}]$ and $[\tau_{lo}, \tau_{hi}]$ which are the lower and upper limits on the joint positions and torques, respectively, for the body model. This calibration step is performed once and stored with the body mass and geometry.

2.2 Inverse Dynamics Tracking

The motion of the capture subject is reconstructed at each frame by using the body model to track the IMU and optical marker data. For this purpose we use an inverse dynamics framework, where body forces and acceleration are found which meet the kinematic constraints introduced by sensor data. Specifically, the orientation of IMUs and positions of optical markers. Since our objective is to use a sparse sensor set, the problem of reconstructing full body motion from sensor data is underconstrained. This is because the skeleton degrees of freedom outnumber the constraints introduced by the sensors. Pose predictions from a motion prior component are therefore combined with sensor data to track the full body motion.

At each frame, the joint velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ are determined, integrated, and used to update all skeletal DOFS such that

$$\begin{aligned} \dot{\mathbf{q}}_+ &= \dot{\mathbf{q}} + h\ddot{\mathbf{q}} \\ \mathbf{q}_+ &= \mathbf{q} + h\dot{\mathbf{q}}_+, \end{aligned}$$

where h is the integration time step which equals the period of the sensor update loop (running at 60 Hz for the experiments in this paper).

Joint velocities are determined in three steps. First, optimal velocities are determined that track the sensor data and motion prior pose estimates. Second, contact forces that explain root motion are estimated. Since the skeleton root contains unactuated degrees of freedom, only contact with the external environment will affect changes in velocity. Finally, once contact forces have been estimated, a least squares optimization is used to determine body forces that track the optimal motion. The body forces lie in the range of natural human joint torques and a novel filtering scheme is used to help enforce this.

2.2.1 *Multi-body dynamics*

The Newton-Euler equations of motion are used to solve for the joint velocities and body forces that produce a physically plausible motion trajectory. A velocity level formulation gives the linear system

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^* - h\mathbf{J}(\mathbf{q})^T\lambda = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} - hC(\mathbf{q}, \dot{\mathbf{q}}). \quad (1)$$

Here, $M(q)$ is the mass matrix computed for the body model at the current pose, $C(q, \dot{q})$ gives the gravitational, centrifugal and Coriolis forces of the system, and $\mathbf{J}(q)$ and λ are the Jacobian and constraint forces introduced by the kinematic tracking constraints for the sensors and motion prior. The mass, Coriolis, and Jacobian matrices are evaluated using the current state of the skeleton, and we refer to them succinctly as $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{J}$ in the remaining text.

Solving Eq. (1) for $\dot{\mathbf{q}}^*$ and λ gives the optimal velocities and sensor tracking forces, respectively, that are used in subsequent stages of the tracker. The following sections discuss the kinematic constraints used to track the sensor data and how this is combined with pose estimates from the motion prior.

2.2.2 Orientation constraints from IMUs

The IMUs provide global orientation information about the body parts to which they are attached. A body part corresponds to a bone in the skeleton, which is assumed to be rigid. Therefore, the angular difference between the sensor and its registered bone location is computed as

$$\phi_{\theta_i} = \log(\bar{R}_i^T R_{imu,i} R_b),$$

where $\bar{R}_i$ and R_b are rotation matrices giving the orientation of the ith IMU sensor and bone b in the world, respectively. The rotation matrix $R_{imu,i}$ is defined locally and registers the IMU to the bone by a relative orientation. The log function provides a mapping between $SO(3) \rightarrow so(3)$ meaning that ϕ_θ is angular screw motion. Formulating orientation tracking as a velocity level constraint gives

$$h\mathbf{J}_\theta \dot{q} = \phi_\theta, \tag{2}$$

where $\mathbf{J}_\theta \in \mathbb{R}^{3n_\theta \times m}$ is the Jacobian matrix mapping an instantaneous change in the skeletal DOFs to a global orientation change for all bodies with IMUs attached. Note that R_b and $\mathbf{J}_\theta$ are dependent on the current pose $\mathbf{q}$ of the skeleton and therefore are recomputed at each frame.

Gyroscopic rate of turn and linear acceleration information are also measured by IMU sensors. However, in early experiments tracking these quantities directly with an inverse dynamics solver produced volatile and unstable motion. Their inclusion in a physical tracking framework needs further investigation.

2.2.3 Position constraints from markers

The optical markers provide position information about a small number of bodies on the capture subject. The difference between a marker position $\bar{x}$ and its corresponding location on the body model is computed as

$$\phi_{x_i} = \bar{x}_i - (p_b + R_b r_i),$$

where p_b is the position of the bone in a global coordinate frame, and r_i is the position of the ith optical marker in local coordinates. The positional constraint used to track all markers is

$$h\mathbf{J}_x \dot{\mathbf{q}} = \phi_x, \tag{3}$$

where $\mathbf{J}_x \in \mathbb{R}^{3n_x \times m}$ is the Jacobian matrix mapping a change in the skeletal DOFs to a global position change at the marker location. The matrix $\mathbf{J}_x$ is also dependent on the current pose of the skeleton and updated at each frame.

2.2.4 Joint angle constraints from the motion prior

The motion prior gives the estimated pose $\tilde{\mathbf{q}}$ based on current values from the sensors and the previous state of the skeleton. We treat the motion prior as a black box function, such that $\tilde{\mathbf{q}} = g(\theta, \mathbf{x}, \mathbf{q})$, where θ and $\mathbf{x}$ contain the orientation and positions of all IMUs and optical markers, respectively.

Since the output of $g(\theta, \mathbf{x}, \mathbf{q})$ is an estimate of the skeletal DOFs, the Jacobian constraint matrix takes the form of an identity matrix. However, the motion prior does not predict the global pose of the skeleton, and rows corresponding to the root DOFs are removed. This gives the constraint equation

$$h\boldsymbol{\Upsilon}\dot{\mathbf{q}} = \phi_{\tilde{q}}, \tag{4}$$

where the difference between the estimated pose and current pose for non-root DOFs is computed as $\phi_{\tilde{q}} = (\tilde{\mathbf{q}} - \mathbf{q})_{6\ldots m}$, and $\Upsilon \in \mathbb{R}^{(m-6)\times m}$ is the truncated identity matrix.

Null space projection. The kinematic constraints introduced by Eq. (4) may interfere with those of the sensor tracking constraints. To avoid these conflicts, the null space of the sensor constraint equations is computed and the prior pose estimate is tracked only in directions that don't compromise sensor tracking. The intuition here is that sensor data is given the highest priority, and the motion prior estimate is only used when real world observations are unavailable. Therefore the motion prior constraint becomes

$$h\mathbf{N}_s\dot{\mathbf{q}} = \phi_{\tilde{\mathbf{q}}}, \tag{5}$$

where $\mathbf{N}_s = \boldsymbol{\Upsilon}(\mathbf{I} - \mathbf{J}_s^\dagger \mathbf{J}_s) \in \mathbb{R}^{(m-6)\times m}$ is the null space matrix of the the sensor Jacobian $\mathbf{J}_s = \begin{pmatrix}\mathbf{J}_\theta^T & \mathbf{J}_x^T\end{pmatrix}^T$.

2.2.5 Constraint relaxation

A compliant formulation is used for the sensor and motion prior constraints. A stiffness and damping parameter is applied to each constraint, transforming them into a spring-damper system. This has the benefit that the tracker remains stable even when singular skeleton configurations are encountered or discontinuities occur in the skeletal motion.

An additional benefit of this formulation is that it provides intuitive parameters for tuning constraint behaviour, which may be done per sensor or per degree of freedom. For instance, we found it useful to increase the damping parameter when data from the motion prior or sensors is noisy, and in Section 2.3 this aspect is leveraged in the development of filters that remove non-physical behaviour by effectively eliminating spurious and high frequency motion in the tracker output. The system of constrained equations becomes

$$\begin{pmatrix} \mathbf{M} & -h\mathbf{J}_\theta^T & -h\mathbf{J}_x^T & -h\mathbf{N}_s^T \\ \mathbf{J}_\theta & \boldsymbol{\Sigma}_\theta & 0 & 0 \\ \mathbf{J}_x & 0 & \boldsymbol{\Sigma}_x & 0 \\ \mathbf{N}_s & 0 & 0 & \boldsymbol{\Sigma}_{\tilde{q}} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}^* \\ h\lambda_\theta \\ h\lambda_x \\ h\lambda_{\tilde{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{M}\dot{\mathbf{q}} - hC(\mathbf{q}, \dot{\mathbf{q}}) \\ \Gamma_\theta \phi_\theta \\ \Gamma_x \phi_x \\ \Gamma_{\tilde{q}} \phi_{\tilde{q}} \end{pmatrix}. \tag{6}$$

where $\boldsymbol{\Sigma}_x$,$\boldsymbol{\Sigma}_\theta$,$\boldsymbol{\Sigma}_{\tilde{q}}$ and $\boldsymbol{\Gamma}_x$,$\boldsymbol{\Gamma}_\theta$,$\boldsymbol{\Gamma}_{\tilde{q}}$ are diagonal matrices encoding the stiffness and damping for the marker position, IMU orientation, and motion prior constraints, respectively. Diagonal entries are computed by the user specified k_i stiffness and damping d_i coefficients for each constraint row as

$$\boldsymbol{\Gamma}_{i,i} = \frac{1}{(1 + h^{-1}k_i^{-1})d_i}$$
$$\boldsymbol{\Sigma}_{i,i} = \frac{h^{-2}k^{-1}}{(1 + h^{-1}k_i^{-1})d_i}.$$

We note the similarities of this formulation with the *constraint force mixing* used by rigid body physics engines [1] and soft constraints [10].

Forming the Schur complement, the linear system in Eq. (6) is solved using its reduced form

$$A\dot{\mathbf{q}}^* = b$$
$$\text{s.t. } \dot{\mathbf{q}}_{lo} \leq \dot{\mathbf{q}}^* \leq \dot{\mathbf{q}}_{hi}$$

where $A = \mathbf{M} + \mathbf{J}^T\boldsymbol{\Sigma}^{-1}\mathbf{J}$ and $b = \mathbf{M}\dot{\mathbf{q}} - hC(\mathbf{q}, \dot{\mathbf{q}}) + h^{-1}\mathbf{J}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}\phi$.

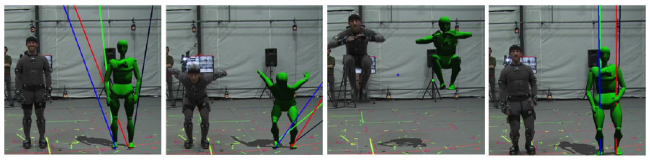

Figure 3: Visualizing contact forces from a jump sequence. The direction of the contact forces is shown by the colored lines, and their length is proportional to the magnitude of the force. Contacts are estimated automatically from the spatial velocity of individual body parts and the surface geometry.

2.2.6 *Contact Estimation*

The solution of Eq. (6) gives optimal velocities $\dot{\mathbf{q}}^*$ that track the sensors and pose estimate. However, this solution neglects the fact that the root degrees of freedom are unactuated, and in order for the computed motion to be physically plausible, the root motion should be generated through contact interactions. Therefore, contact forces are estimated using an approach that treats the six root DOFs as a floating base and solves for strict contact force constraints [33]. Since our physics tracking framework does not use geometry information about the capture environment, contact is instead estimated from $\dot{\mathbf{q}}^*$ and the body geometry.

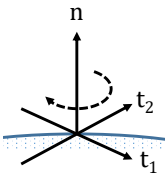

Figure 4: The contact basis.

We make two assumptions in estimating contact with the external environment. One is that bodies in contact with the environment have zero linear velocity at the point of contact. As a result, only the case of contact with static friction is considered, since for cases involving sliding or kinetic friction this assumption does not hold. However, this is a reasonable assumption for many types of human motion. Another assumption is that the point of contact lies on the surface of the body. The body model geometry provides a good representation of the shape of the person being tracked, and this is leveraged during contact estimation; there are no assumptions about the geometry of the external environment, e.g. collision detection with a single level planar floor.

Based on these two assumptions, potential points of contact are identified for each body part in the tracking skeleton. Each body part can have at most a single contact point.

Velocity criteria. The linear velocity at a location on the surface of a body is computed as $(\omega \times r) + v$, where ω and v are the angular and linear velocities of the body, respectively, and r is a vector from the body center of mass to a position on the surface. The values of ω and v are computed using forward kinematics and the joint velocities $\dot{\mathbf{q}}^*$. The first step in contact estimation is to solve for r as

$$(\eta \mathbf{I} - \hat{\omega}) r = v$$

which includes a regularization term η and the 3×3 skew symmetric cross product matrix $\hat{\omega}$. The resulting vector is considered a contact point if:

- The point $p_b + R_b r$ is near the surface of the body, and is determined by checking that the minimum distance from the point to triangles in the geometry mesh is less than threshold δ_1;
- The linear velocity of the body at r is small, such that $\|\omega \times r + v\| < \delta_2$, where δ_2 is the maximum allowable velocity.

A contact basis is constructed for each of the n_c viable contacts. For contact j, this includes the surface normal $\mathbf{n}_j$, which is estimated from body geometry, and two surface tangential directions $\mathbf{t}_{1,j}$,$\mathbf{t}_{2,j}$. An example basis is shown in Fig. 4. The basis is encoded as matrix $B_j \in \mathbb{R}^{6\times 4}$ where

$$B_j = \begin{pmatrix} 0 & 0 & 0 & \mathbf{n}_j \\ \mathbf{n}_j & \mathbf{t}_{1,j} & \mathbf{t}_{2,j} & 0 \end{pmatrix}.$$

Note that in addition to linear forces, B_j allows torsion about the contact normal. Torsional contact forces are necessary since, if only a single contact point is used per body, without this term motions such as quick turns involving foot pivoting could not be adequately explained by our method. Linear forces are constrained to the friction cone F_{c_j}, such that

$$\lambda_{n_j} \geq 0 \tag{7}$$

$$\|\lambda_{t_{1,j}}\| + \|\lambda_{t_{2,j}}\| < \mu \|\lambda_{n_j}\| \tag{8}$$

$$\|\lambda_{\tau_j}\| < s\,\mu \|\lambda_{n_j}\|. \tag{9}$$

A friction coefficient $\mu = 0.8$ is used for all of our experiments, and this is multiplied by the positive scalar s to compute bounds for the angular force about the normal λ_{τ_j}. In our experiments, $s = 5.0$.

The combined force and torque generated at a contact, or the *c*ontact wrench, is computed directly as

$$\begin{pmatrix} \tau_{c_j} \\ f_{c_j} \end{pmatrix} = B_j \begin{pmatrix} \lambda_{n_j} \\ \lambda_{t_{1,j}} \\ \lambda_{t_{2,j}} \\ \lambda_{\tau_j} \end{pmatrix}$$

subject to the constraints imposed by Eq. (7)-Eq. (9).

Finally the solution of a constrained minimization problem is used to compute contact forces that explain the trajectory of root DOFs. That is,

$$\min_{\lambda_c} \|\mathbf{M}_1 \ddot{\mathbf{q}}^* - \mathbf{J}_c^T \mathbf{B} \lambda_c\|$$
$$\text{s.t.} \, \forall j : B_j \begin{pmatrix} \lambda_{n_j} \\ \lambda_{t_{1,j}} \\ \lambda_{t_{2,j}} \\ \lambda_{\tau_j} \end{pmatrix} \in F_{c_j}. \tag{10}$$

Here $\mathbf{M}_1 \in \mathbb{R}^{6\times m}$ is a truncated mass matrix containing only rows corresponding to the root, and $\mathbf{J}_c \in \mathbb{R}^{(6n_c \times 6)}$ is the contact Jacobian for all contact positions affecting only the root DOFs. $\mathbf{B}$ is a block diagonal matrix containing the basis matrix B_j for each viable contact j and $\lambda_c \in \mathbb{R}^{4n_c}$ contains forces and torques for all contacts.

Eq. (10) is solved using a projected Gauss-Seidel (PGS) method. Fig. 3 shows a jumping motion reconstructed with our solver and overlaid with visual representation of the estimated contact forces.

2.2.7 *Final motion trajectory*

The final stage solves for updated joint velocities that account for contact forces. The optimal joint motion $\dot{\mathbf{q}}^*$ is tracked by minimizing $\|\dot{\mathbf{q}}^* - \dot{\mathbf{q}}_+\|_{W_q}$. The scaling matrix W_q has diagonal elements equal to $10\mathbf{M}_{i,i}$, which is the scaled mass rooted at each joint. Likewise, the torques used to drive the body model are minimized by $\|\tau\|_{W_\tau}$. The scaling matrix has values $W_{\tau_{i,i}} = \frac{200}{\mathcal{M}}$ for the root degrees of freedom where

$\mathcal{M}$ is the total mass of the body model, and $W_{\tau_{i,i}} = \frac{1}{M_{i,i}}$. This scaling scheme for the joint velocities and torques is similar to the one used in [21], and provides intuitive controls for adjustment of how strongly the optimal motion estimate is tracked.

The final motion should also be physically plausible, and the final joint velocities and body torques are found by solving the linear system:

$$\begin{pmatrix} \mathbf{M} & -h\mathbf{I} \\ \mathbf{W}_{\dot{q}} & 0 \\ 0 & \mathbf{W}_{\tau} \end{pmatrix} \begin{pmatrix} \dot{q}_+ \\ \tau_+ \end{pmatrix} = \begin{pmatrix} \mathbf{M}\dot{q} - h\left(C(q,\dot{q}) + \mathbf{J}_c^T\lambda_c\right) \\ \mathbf{W}_{\dot{q}}\dot{q}^* \\ 0 \end{pmatrix}. \quad (11)$$

The PGS algorithm is used to solve Eq. (11) by forming the normal equations and applying the box constraints $\tau_{lo} \leq \tau_+ \leq \tau_{hi}$. The joint accelerations may be recovered using finite differences as $\ddot{\mathbf{q}} = h^{-1}(\dot{\mathbf{q}}_+ - \dot{\mathbf{q}})$.

Note that the lower and upper bound for $\tau_{+,1\ldots6}$ is set to $-\infty$ and ∞ respectively. In the event that contact estimation fails, this still allows the motion to be tracked although non-physical forces at the root may be used. However, our experiments indicate these forces are typically small, indicating plausibility of the motion.

2.3 Physical motion filters

Although the inverse dynamics tracker ensures motions are physical, it may still produce motions that are not possible due to limitations of human muscle forces. Specifically, high frequency motion may occur due to tracking noise in the sensor data or training errors in the motion prior. For this purpose we devise a simple but effective method to reduce such artefacts in the final output.

Upon solving Eq. (6), the body torques for each constraint are recovered by

$$\tau = \mathbf{J}^T\lambda.$$

If τ_i for a particular DOF lies outside the range $[\tau_{lo_i}, \tau_{hi_i}]$, we compute the violation as

$$\Delta\tau_i = min(\tau_i - \tau_{hi_i}, \tau_{lo_i} - \tau_i).$$

and assemble a torque violation vector for all joints, or $\Delta\tau$.

This vector is mapped into constraint space using the constraint Jacobian and then used to increase the damping coefficient for each tracking constraint by

$$d \leftarrow d(1.0 + h\alpha J\mathbf{M}^{-1}\Delta\tau).$$

This has the effect of damping high frequency motion or noise, but still tracks the overall motion. Eq. (6) is solved again and the process is repeated until all torque are within the boundaries, or a maximum iteration count is reached. Usually 4 or 5 iterations is sufficient to produce smooth, natural looking motion and this step is done before contact estimation. The motivation for this approach is that "jerky" motion corresponds to relatively large torques.

The pseudoinverse $J^\dagger$ could also be used here, but we opt to use the more efficient approach of J and providing the user with a gain parameter α.

3. MOTION PRIORS

The motion prior estimates the pose of the skeleton based on sensor data and the previous pose $\mathbf{q}$. Since the estimate $\tilde{\mathbf{q}}$ should be invariant to global position and orientation, the root DOFs are excluded from this estimate. Several different motion priors are used in our experiments.

Reference pose. This is the simplest prior used in our evaluation. It consists of a single reference pose, which in our case is a T-pose for the tracking skeleton. In other words, if a DOF is not actively tracking sensor constraints, it tracks the reference pose.

Perturbed ground truth. We consider a "gold standard" motion prior to be the skeletal pose reconstructed from a dense set of optical markers and cameras. Our system is evaluated using a motion prior built with just such a system. However, we perturb the data using a Gaussian noise function applied to each joint angle. This represents a class of motion priors which has learned the mapping from sensor data to skeleton posture, but is prone to high frequency noise or random errors. Unless otherwise noted, Gaussian parameters of mean $\mu = 0$ and standard deviation $\sigma = 0.12$ are used in all of our experiments, with units in radians.

Clustered mocap database. As an offline step the spectral algorithm proposed by Chen and Cai [12] is used to cluster samples from a large motion capture database and then decimate it. By storing only representative poses, the size of the database is significantly reduced. This makes it more efficient to store and search. Sensor data for synthetic IMUs and optical markers is computed and stored alongside each pose in the clustered database. At run time, a k-nearest neighbour algorithm is used to find examples with similar pose and sensor data, which are interpolated using an inverse quartic weighting scheme. That is, the weight w of each nearby sample $\hat{y}$ is computed as

$$w = \frac{1}{dist(y,\hat{y})^4},$$

where $y = (\theta, \mathbf{x}, \mathbf{q})$ contains the sensor data and skeleton pose for the current frame and the function $dist(y,\hat{y})$ returns the distance between y and the database sample $\hat{y}$. The distance metric is a weighted sum of the Euclidean distance between marker positions, the difference of quaternions for each IMU, and the Euclidean norm of the difference of non-root DOFs, or

$$w_\theta \log(\hat{\theta}^{-1}\theta) + w_x\|\hat{\mathbf{x}} - \mathbf{x}\| + w_q\|\hat{\mathbf{q}} - \mathbf{q}\|.$$

4. RESULTS

Here we present some results of tracking various motion sequences with our framework. The accompanying videos demonstrates many of the experiments discussed in this section. An early version of our tracking framework was also recently used for a VR demonstration [18].

Performance. Solving for a single frame of motion requires approximately 17 ms of computation time on a 3.3 GHz Intel i7 processor, meeting real-time requirements for a 60 Hz sensor update rate. Our C++ implementation uses DART[2] to perform the forward dynamics simulation and compute Jacobian matrices.

Sensor config and solver parameters. A combination of six IMUs and five optical markers are used for the results in this section. The sensors and their placement on the body are shown in Fig. 5. Placing sensors at the end of kinematic chains helps to reduce overall tracking error, and so markers and IMUs are located at the head, hands, and feet. An additional IMU is placed near the lower back since we found this to improve the overall quality of motion, particularly for the hips. Stiffness values of 5×10^8, 1.2×10^8, and 2×10^6 are used

[2]DART. http://dartsim.github.io/

for tracking marker, IMU, and prior kinematic constraints, respectively; all tracking constraints use an initial damping value of 2.0.

Synthetic sensors. Experimental results in this section use synthetic sensor trajectories that are reconstructed from motion generated by a commercial inverse kinematics (IK) solver [4] and dense optical marker coverage, followed by a manual clean-up step. This motion also serves as a ground truth comparison for our solver. Synthetic sensors are registered to locations on the body model by a relative transform. Their global position and orientation is then obtained at each frame using forward kinematics of the skeleton, and compounding the bone transforms with the registration transform.

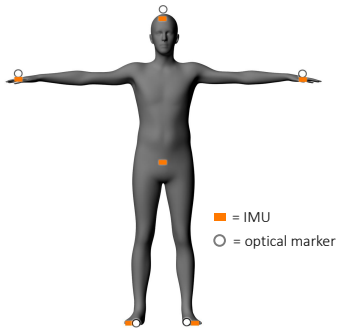

Figure 5: The sensor configuration used in our experiments.

4.1 Tracking error

In this section, the tracker is evaluated with the motion priors described in Section 3: reference pose (REF), ground truth perturbed by noise (NGT), and the nearest neighbour interpolation of a clustered and decimated motion capture database (CD). The CD prior uses 5000 poses found by clustering the CMU motion capture database. Synthetic IMU and marker data is computed for each pose using the experimental sensor configuration

IK comparison. The tracking error is also compared with a baseline IK algorithm [9] that solves for updates to the skeleton pose using the pseudo-inverse method, and at each frame $\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}$. The poses satisfy kinematic constraints from Eq. (6), such that $\mathbf{J}\Delta\mathbf{q} = \phi$. Only the CD prior is used for comparison with the IK solver.

Fig. 6 shows the error in body position and orientation for several different styles of motion, including running, jumping, high kicks, and interaction with hands and feet to climb a staircase. The mean squared error (MSE) of the positions and orientations of all skeleton bodies are computed for each frame of motion. Position error for each body part is computed as the Euclidean distance between the center of mass position of the ground truth skeleton and the model used by our tracker; orientation error is computed similarly using the angular difference of quaternions. The ground truth and reconstructed motions can be seen in the attached video.

Best performance is achieved by using the NGT prior. As indicated by Table 1, when the NGT prior is used the average body position error is less than 0.15 cm and body orientation error is on average less than 2 degrees. This demonstrates that given a motion prior that provides a good prediction of the pose estimate, our system does well to eliminate noise and produce physically plausible motions. Even using the CD prior, the position and orientation error is quite low. Popping artefacts that occur due to continuously updating the set of nearest neighbours are effectively eliminated by the physical filter and torque limits.

The IK solver tends to track the motion without consideration for physical plausibility; jerk and popping artefacts are visible in the output. For example, a side-by-side comparison of the motion reconstructed by our solver versus the baseline IK algorithm is provided in the video. It's clear that the output of our solver is, visually, much better. Fig. 6 and Table 1 also indicate that lower tracking error is possible using our physical tracker.

	REF	CD	NGT	IK
Running	1.47 cm	0.80 cm	0.11 cm	1.26 cm
	7.26 deg	6.41 deg	0.72 deg	6.69 deg
Jump & kicks	0.59 cm	0.20 cm	0.15 cm	0.34 cm
	3.99 deg	2.33 deg	1.39 deg	5.26 deg
Stairs	1.71 cm	0.69 cm	0.08 cm	2.30 cm
	9.08 deg	4.39 deg	0.58 deg	11.67 deg

Table 1: MSE bone position and orientation error averaged over all frames for the sequences and motion priors shown in Fig. 6. Error results for a baseline IK algorithm with the CD prior are also provided (last column).

The null space projection procedure also ensures that sensors are given priority over pose estimates from the motion prior. Fig. 7 demonstrates the benefit of this approach when using various motion priors. The error in the resulting motion is typically lower when using the technique. In particular, if the prior estimates the motion poorly, it can significantly reduce the ability to track the sensors. This is evident as more sensors are used, and can be seen in the error plot where 17 IMUs are used. The motion is only tracked with high accuracy if the null space technique is used.

4.2 Estimation of contact and body torques

Being able to reliably estimate contact position and forces is an interesting and useful feature of our tracking framework. Fig. 8 shows the contact forces generated by our system for several motion sequences. The transitions and contact phases are identifiable for each activity. For example, in the walking sequence is it clear when the left or right foot is planted. The heel strike to toe off transitions are also identified by the contact force profile. In the running sequence, there are frames that contain no contact forces. This indicates ballistic motion, which is expected for running. The stair example is interesting since the capture subject also used their hands to support themselves while doing a "crab walk" style descent. The left and right hands may be considered in the contact estimation process, and the resulting contact force profiles shows a coordination between the hands and feet as the capture subject transitions between supporting themselves with their feet and momentarily shifting support to their hands.

Fig. 9 shows the torques in the left knee reconstructed from walking and running sequences. In the case of the walk sequence, the torque throughout the gait cycle is qualitatively similar to torque profiles collected by the biomechanics community [14, 24]. We also note the similarity of our results to those obtained in experiments conducted by Zhang et al. [32] and Brubaker et al. [8].

4.3 Analysis of physical motion filter

Fig. 10 shows the reconstructed motion for selected DOFs when tracking a running motion using the NGT motion prior. Joint angles for the left femur are compared with the ground truth motion, the NGT prior pose estimate, and solver output. Gaussian noise of $\sigma = 0.12$ and $\sigma = 0.06$ radians

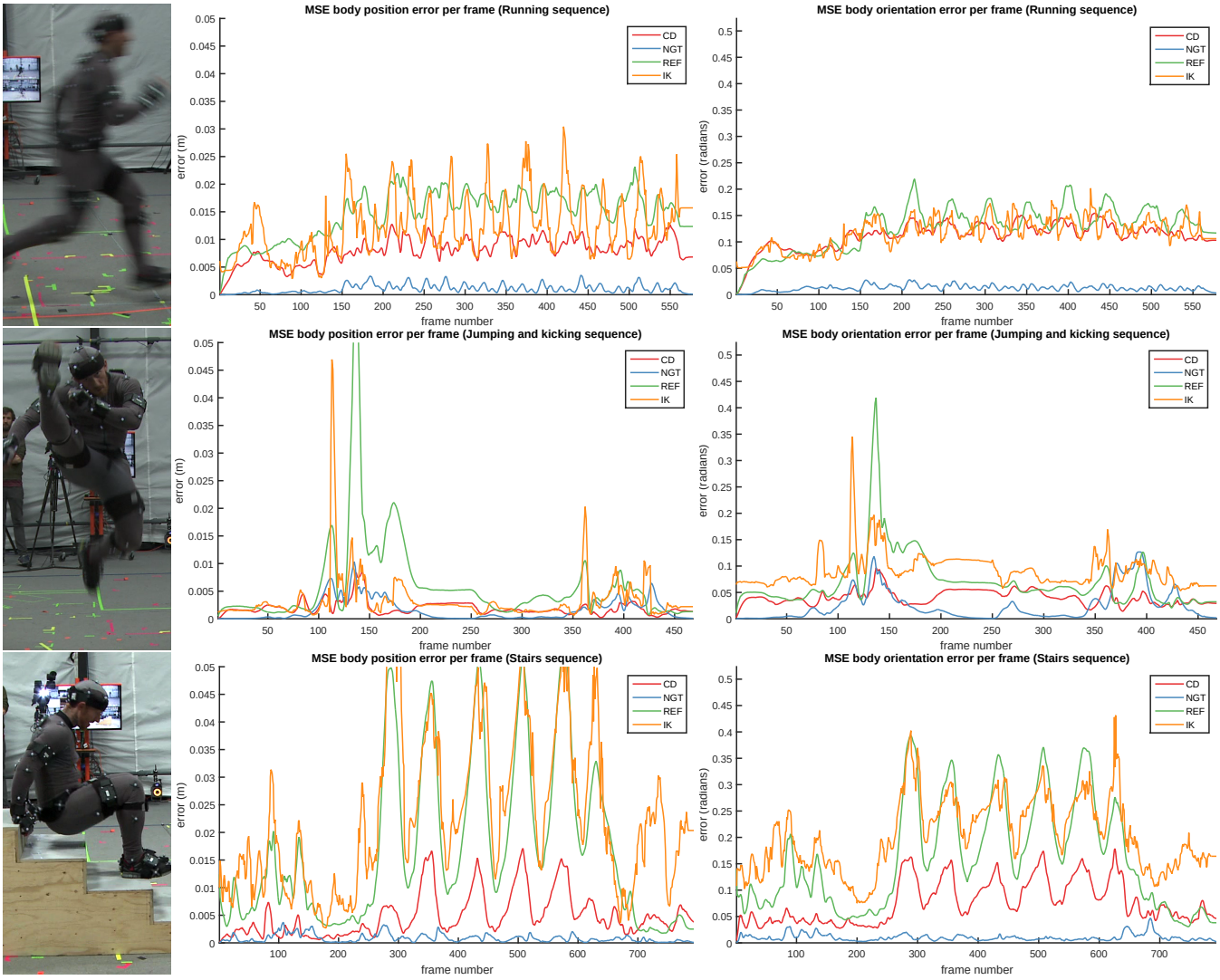

Figure 6: Tracking various motion sequences: running (first row), jump kicking (middle row), and stair walking (bottom row). The position and orientation error per frame, averaged over all skeleton bodies, is shown when different motion priors (REF, CD, NGT) are used with the physics-based tracker. Error produced by the IK algorithm is also shown.

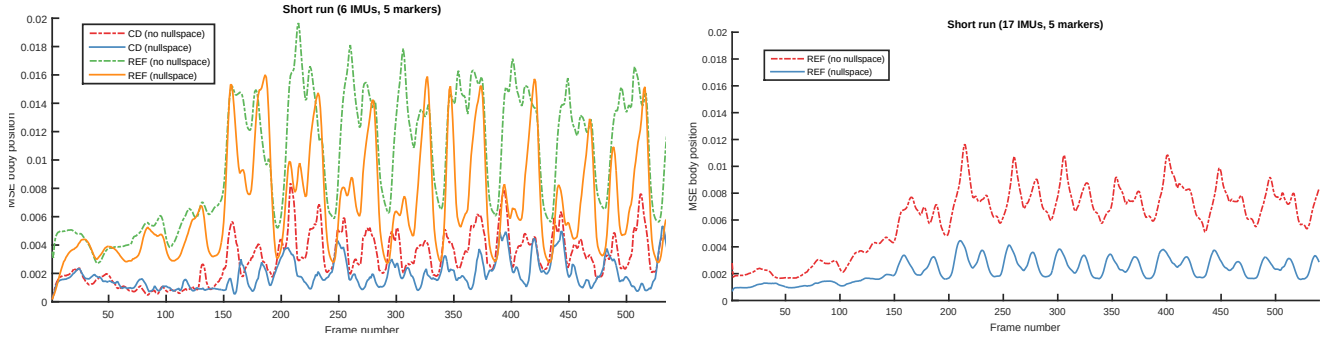

Figure 7: MSE for body positions when tracking a running motion with and without the null space projection step. The overall error is lowered when both the REF and CD priors are used (left). The benefits of using the null space projection are evident when 17 IMUs are used to track the motion with the REF prior(right).

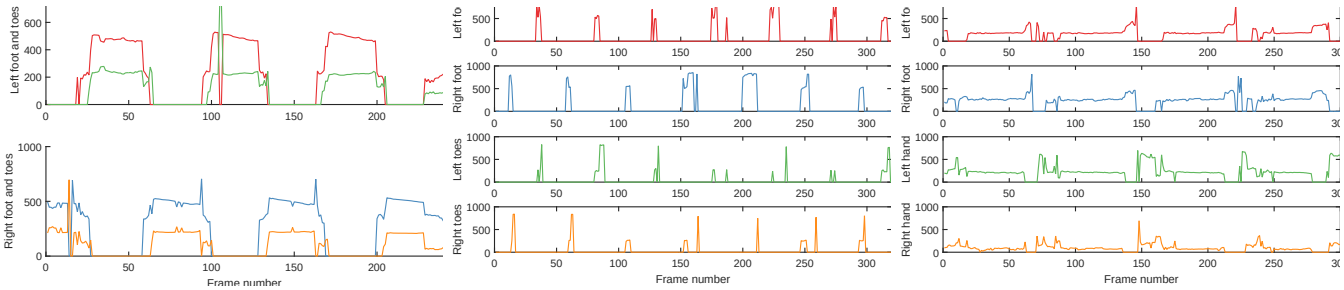

Figure 8: Foot and toe contact forces in Newtons for left and right feet during a walk cycle (left), running sequence (middle), and stair walking (right). In the stair example, the hands were also used to interact with the environment.

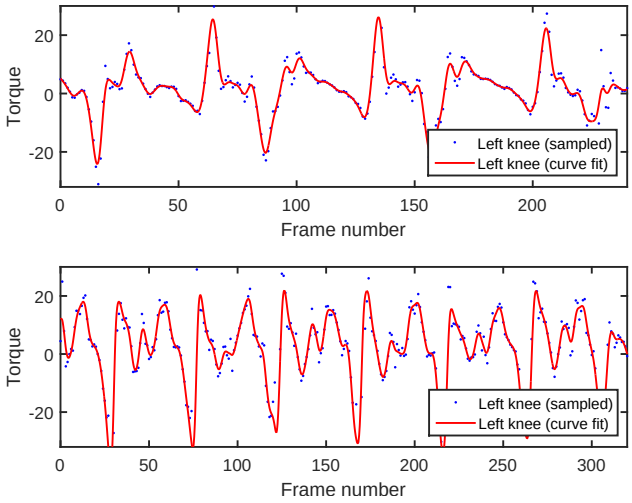

Figure 9: Left knee torques sampled from the inverse dynamics tracker during walking (top) and running (bottom).

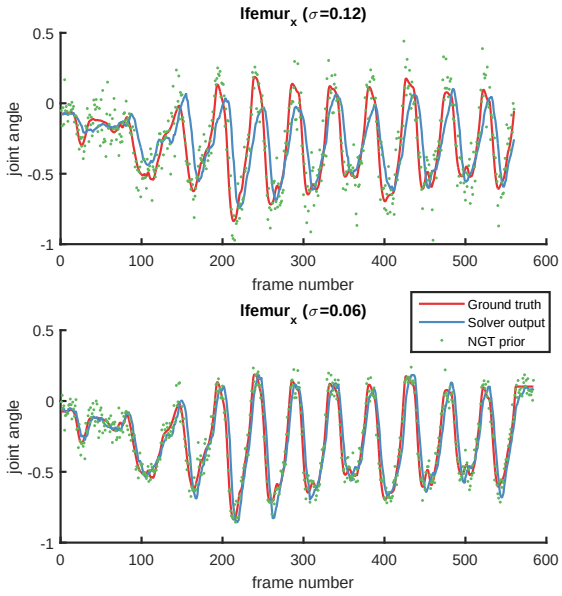

Figure 10: Tracking for various σ values with the NGT prior. Noise is effectively removed by the physical motion filter, and as the noise decreases, the reconstructed motion is closer to the ground truth.

are added to the prior. Although there is significant disturbance in the pose estimate which manifests as high frequency motion, the filtering technique is capable of smoothly tracking ground truth motion. As the noise decreases, the reconstructed motion matches the ground truth, as expected. Sensor noise, i.e. in IMUs, is not explicitly modeled in these tests, but the results indicate that the physical filter may perform similarly well at handling such disturbances.

4.4 Live tracking

The accompanying video shows examples where our framework is used to track a person wearing 17 inertial sensors and 3 optical marker clusters. The reconstructed motion is used to update a skinned character model in real-time. The results show that our multi-modal framework handles occlusion robustly and can deal with scenarios where optical tracking and IMU-based tracking systems would fail.

5. CONCLUSION

A framework for real-time human motion tracking is presented in this paper. Sparse multi-modal sensor configurations is bolstered by physical tracking and a black box motion prior. The method generates motions that are physically plausible and gives estimates of contact forces and body torques. The results shown in Section 4 demonstrate that the reconstructed motions have low error and that body torque and contact forces may also be extracted with high reliability. The physical tracking framework is also robust to significant errors and noise in the motion prior.

5.1 Future work

Online retargeting is an interesting future research direction for our tracking framework. For example, existing physics-based retargeting work [21] could be readily adapted to our framework. A novel application of our work could be to relay to an actor in real-time if their performance is physically feasible given the limitations and parameters of a target body model. This would allow the capture subject to retarget their own performance.

Also, a number of machine learning methods have been applied to the task of human motion prediction. We have begun to examine some of these, and even conducted preliminary experiments to integrate them with our framework. The results are promising, and an open question remains how the contact and torque information extracted by our tracker can be used to bolster the accuracy of these approaches.

Acknowledgments

We appreciate the help of Maggie Kosek and Joanna Jamrozy with rigging and mocap processing tasks, and Babis Koniaris for the Unreal Engine integration. We thank Robin Guiver for his acting performances. This work was funded by the Innovate UK project "Real-time Digital Acting" (# 101857).

6. REFERENCES

[1] Open Dynamics Engine. http://www.ode.org/, 2015.

[2] OptiTrack Motive. http://www.optitrack.com/, 2016.

[3] Perception Neuron. http://www.neuronmocap.com/, 2016.

[4] Vicon Blade. http://www.vicon.com/, 2016.

[5] Xsens MVN. http://www.xsens.com/, 2016.

[6] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. on Graphics*, 24(3):408–416, 2005.

[7] M. A. Brubaker, L. Sigal, and D. Fleet. Physics-based human motion modeling for people tracking: A short tutorial. In *Image*, pages 1–48, 2009.

[8] M. A. Brubaker, L. Sigal, and D. J. Fleet. Estimating contact dynamics. In *IEEE 12th Intl. Conf. on Computer Vision*, pages 2389–2396, 2009.

[9] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.

[10] E. Catto. Soft constraints. In *Game Developers Conference*, 2011.

[11] J. Chai and J. K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. on Graphics*, 24(3):686–696, 2005.

[12] X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, 2011.

[13] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion 4D: real-time performance capture of challenging scenes. *ACM Trans. on Graphics*, 35(4):114, 2016.

[14] K. Endo, D. Paluska, and H. Herr. A quasi-passive model of human leg function in level-ground walking. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 4935–4939, 2006.

[15] S. Ha, Y. Bai, and C. K. Liu. Human motion reconstruction from force sensors. In *Proc. of the 2011 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 129–138, 2011.

[16] L. Hoyet, K. Ryall, R. McDonnell, and C. O'Sullivan. Sleight of hand: perception of finger motion from reduced marker sets. In *Proc. of the ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games*, I3D '12, pages 79–86, New York, NY, USA, 2012. ACM.

[17] C. Kang, N. Wheatland, M. Neff, and V. Zordan. Automatic hand-over animation for free-hand motions from low resolution input. In *Intl. Conf. on Motion in Games*, pages 244–253, 2012.

[18] B. Koniaris, I. Huerta, M. Kosek, K. Darragh, C. Malleson, J. Jamrozy, N. Swafford, J. Guitian, B. Moon, A. Israr, S. Andrews, and K. Mitchell. Iridium: immersive rendered interactive deep media. In *ACM SIGGRAPH 2016 VR Village*. ACM, 2016.

[19] B. Krüger, J. Tautges, A. Weber, and A. Zinke. Fast local and global similarity searches in large motion capture databases. In *Proc. of the 2010 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 1–10, 2010.

[20] Y. Lee, S. Kim, and J. Lee. Data-driven biped control. *ACM Trans. on Graphics*, 29(4):129, 2010.

[21] S. Levine and J. Popović. Physically plausible simulation for character animation. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 2012.

[22] H. Liu, X. Wei, J. Chai, I. Ha, and T. Rhee. Realtime human motion control with a small number of inertial sensors. In *Symp. on Interactive 3D Graphics and Games*, pages 133–140. ACM, 2011.

[23] L. Liu, K. Yin, M. van de Panne, T. Shao, and W. Xu. Sampling-based contact-rich motion control. *ACM Transctions on Graphics*, 29(4):Article 128, 2010.

[24] E. C. Martinez-Villalpando and H. Herr. Agonist-antagonist active knee prosthesis: a preliminary study in level-ground walking. *J. Rehabil. Res. Dev*, 46(3):361–374, 2009.

[25] L. A. Schwarz, D. Mateus, and N. Navab. Multiple-activity human body tracking in unconstrained environments. In *Intl. Conf. on Articulated Motion and Deformable Objects*, pages 192–202, 2010.

[26] A. Shapiro, A. Feng, R. Wang, H. Li, M. Bolas, G. Medioni, and E. Suma. Rapid avatar capture and simulation using commodity depth sensors. *Computer Animation and Virtual Worlds*, 25(3-4):201–211, 2014.

[27] R. Slyper and J. K. Hodgins. Action capture with accelerometers. In *Proc. of the 2008 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 193–199, 2008.

[28] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H.-P. Seidel, and B. Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Trans. on Graphics*, 30(3):18, 2011.

[29] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović. Practical motion capture in everyday surroundings. In *ACM Trans. on Graphics*, volume 26, page 35, 2007.

[30] T. von Marcard, G. Pons-Moll, and B. Rosenhahn. Human pose estimation from video and imus. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(8):1533–1547, 2016.

[31] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical simulation for probabilistic motion tracking. In *In Computer Vision and Pattern Recognition*, 2008.

[32] P. Zhang, K. Siu, J. Zhang, C. K. Liu, and J. Chai. Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture. *ACM Trans. on Graphics*, 33:221, 2014.

[33] Y. Zheng and K. Yamane. Human motion tracking control with strict contact force constraints for floating-base humanoid robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 34–41, 2013.