Human Pose Estimation from Video and IMUs

Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn

Abstract—In this work, we present an approach to fuse video with sparse orientation data obtained from inertial sensors to improve and stabilize full-body human motion capture. Even though video data is a strong cue for motion analysis, tracking artifacts occur frequently due to ambiguities in the images, rapid motions, occlusions or noise. As a complementary data source, inertial sensors allow for accurate estimation of limb orientations even under fast motions. However, accurate position information cannot be obtained in continuous operation. Therefore, we propose a hybrid tracker that combines video with a small number of inertial units to compensate for the drawbacks of each sensor type: on the one hand, we obtain drift-free and accurate position information from video data and, on the other hand, we obtain accurate limb orientations and good performance under fast motions from inertial sensors. In several experiments we demonstrate the increased performance and stability of our human motion tracker.

Index Terms—Human pose estimation, motion capture, multisensor fusion, inertial sensors, IMU, animation

1 Introduction

In this paper we deal with the task of human pose tracking, also known as motion capturing (MoCap) [1]. Compared to commercial marker-based systems, video-based marker-less motion capture systems are very appealing because they are inexpensive and non-intrusive. Unfortunately, occlusions, partial observations and image ambiguities make the problem very hard. Hence, there is still a gap between the accuracy and reliability of marker-less systems compared to marker-based solutions.

To this end, we propose a hybrid tracker that combines information coming from video cameras with information coming from a small number of inertial sensors. In particular, we use only five inertial sensors attached at the body extremities of the subject. By combining both sensor types the tracking performance increases in both accuracy and stability. The proposed tracking solution is an inexpensive alternative to commercial marker-based systems to perform motion capture. Although it is more intrusive than pure marker-less systems, five miniature IMU sensors do not hamper the range of motions a subject can perform. This makes it a very appealing and practical solution for applications where high accuracy and realism is required, e.g., for movie production and medical analysis.

Stabilizing pure video-driven MoCap with learned priors is very common in the literature to compensate for inherent ambiguities. Using additional a priori knowledge such as familiar pose configurations learned from motion capture data helps considerably to handle more difficult scenarios like partial occlusions, background clutter, or corrupted image data. There are several ways to employ such a priori knowledge to human tracking. One option is to learn the space of plausible human poses and motions [2], [3], [4], [5], [6], [7], [8]. Another option is to learn a direct mapping from image features to the pose space [7], [9], [10], [11], [12] or to mid-level representations of pose [13] through posebits.

To constrain the high dimensional space of kinematic models, a major theme of recent research on human tracking has been dealing with dimensionality reduction [14], [15]. Here, the idea is that a typical motion pattern like walking should be a rather simple trajectory in a lower dimensional manifold. Therefore, prior distributions are learned in this lower dimensional space. Such methods are believed to generalize well with only little training data.

Inspired by the same ideas of dimensionality reduction, physical and illumination models have been recently proposed to constrain and to represent human motion in a more realistic way [3], [16], [17], [18]. A current trend of research tries to estimate shape deformations from images besides the body pose by either directly deforming the mesh geometry [19] or by a combination of skeleton-based pose estimation with surface deformation [20].

Recently, inertial sensors (e.g., gyroscopes and accelerometers) have become popular for human motion analysis. [21] presents a system to capture full-body motion using only inertial and magnetic sensors. While the system in [21] is very appealing because it does not require cameras for tracking, the subject has to wear a suit with at least 17 inertial sensors, which might hamper the movement of the subject. In addition, long preparation time before recording is needed. Moreover, inertial sensors suffer from severe drift problems and cannot provide accurate position information in continuous operation.

1.1 Contributions

Even using learned priors from MoCap data, obtaining limb orientations from video is a difficult problem. Intuitively, because of the cylindrical shape of human limbs different limb orientations project to very similar appearances in the images. These orientation ambiguities can be easily captured by the inertial sensors but accurate joint positions in

- *T. von Marcard and B. Rosenhahn are with the TNT group at the Leibniz-University of Hannover, Germany.*
  E-mail: {marcard, rosenhahn}@tnt.uni-hannover.de.
- *G. Pons-Moll is with the Perceiving Systems Department of the Max Planck for Intelligent Systems, Tuebingen, Germany.*
  E-mail: gerard.pons.moll@tue.mpg.de.

Manuscript received 8 Mar. 2015; revised 15 Nov. 2015; accepted 21 Dec. 2015. Date of publication 26 Jan. 2016; date of current version 11 July 2016.
Recommended for acceptance by S. Escalero, J. Gonzàlez, X. Baró, and J. Shotton.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2522398

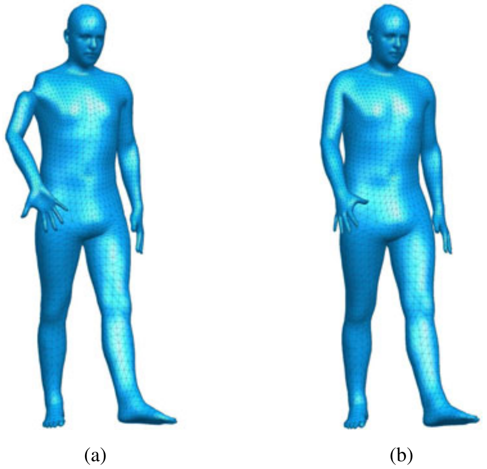

Fig. 1. Tracking result for two selected frames. (a) Video-based tracker. (b) Our proposed hybrid tracker.

the world space cannot be obtained. Therefore, we propose to use a small number of sensors (we use only five) fixed at the body extremities (lower arms, shanks and waist) as a complementary data source to visual information. On the one hand, we obtain stable and drift-free accurate position information from video data and, on the other hand, we obtain accurate limb orientations from the inertial sensors.

The present work is an extension of our preliminary conference paper [22] and improves it in several ways:

- We provide additional details on integrating orientation data to a contour-based video tracker. In Section 6.1 we minimize the squared geodesic distance of estimated and measured sensor orientations. This leads to three independent constraint equations per sensor instead of nine. We also show how to minimize the squared Frobenius norm of orientation differences (chordal distance) in Section 6.2.
- In order to provide a more thorough evaluation of the hybrid approach, we recorded a new data set *TNT15*. We will make it available for research at [23].
- We present a totally new experimental evaluation. In Section 7.1 we describe the experimental setup and introduce two error metrics, which measure the tracker performance in complementary ways. A comparison of the video and hybrid approach is given in Section 7.2.
- Additionally, we investigated different settings of the hybrid tracker. In Section 7.3 we evaluate the influence of the hybrid tracker's weighting parameter λ that balances the IMUs and video energy terms. In Section 7.4 we inspect the tracking error vs. the number of camera views. We also evaluate the sensitivity of the tracker to sensor lag in Section 7.5.
- In Section 7.6 we evaluate the hybrid tracker approach on the *HumanEva* [24] dataset. Ground-truth body poses are used to synthesize the missing IMU data and the average joint position error is reported and compared to state-of-the-art approaches.

2 Related Work

Most works in 3D human pose estimation in computer vision have focused on obtaining a rough estimate of the skeletal configuration by lifting 2D body part detections [25]. Multiple views and temporal consistency have also been exploited. For example, in [26], the authors propose a so-called *temporally consistent 3D Pictorial Structures* model (3DPS) for multiple human pose estimation from multiple cameras views. The model extends multi-view 3D pictorial structures with a temporal consistency between the inferred poses. The focus of these works is to estimate the pose for tasks such as human action recognition or scene understanding. Hence, realism is not a requirement and pose estimates often do not include 3D limb orientations and suffer from errors such as motion jitter, foot skating and unbalance. Higher fidelity body models with an underlying skeleton have also been used for pose estimation [8], [27]. The use of models with higher degrees of freedom also comes with more ambiguities which researchers compensate by using action priors [8] or robust likelihood functions and global optimization schemes [27], [28]. On the other end of the spectrum, performance capture approaches use a large number of cameras to capture the full surface geometry of the body, potentially including clothing [20], [29], [30], [31]. These approaches are very appealing and produce very realistic results. However, it is not trivial to transfer the captured surface motion to new avatars. Furthermore, such approaches could benefit from our proposed hybrid tracker since limb orientations are very hard to estimate when they are occluded by clothing, e.g., legs occluded by a skirt.

The field of human pose estimation has experienced significant advances with the availability of the inexpensive depth sensor kinect. A depth sensor significantly simplifies the problem since many depth ambiguities can be resolved. In the influential paper of [32] the pose estimation problem is turned into a body part classification problem. In [33], [34] they extended the approach of [32] to directly regress correspondences to a model to improve the accuracy of the predictions. Several other approaches have been published to tackle the problem of pose and shape estimation from depth sensors [35], [36], [37], [38], [39]. A survey on pose estimation using depth images has been presented in [40]. Most existing works focus on body part detection and pose estimation. Although depth sensors are very appealing for applications such as gaming, they do not work very well outdoors and the recording volume is limited. Furthermore, for both RGB and depth data, orientation ambiguities are still an issue.

Inertial sensors (IMU) do not suffer from such limitations but they are intrusive by nature: at least 17 units must be attached to the body which poses a problem for biomechanical studies and sports sciences. Additionally, IMUs alone fail to measure accurately translational motion and suffer from drift. Perhaps surprisingly, not many works can be found that combine inertial sensors with visual cues. This is maybe due to the fact that IMUs have been less available in the past. However, inertial sensors are becoming affordable. In fact, most cellphones come with integrated IMUs. IMUs alone have been often used for medical applications, see, e.g., [41] where accelerometer and gyroscope data is fused. However, their application concentrates on the estimation of the lower limb orientation in the sagittal plane. An exception that combines visual and orientation cues is [42], but it is restricted to the tracking of a single limb (the arm). Moreover, only a simple red arm band is used as image feature. In [43], data obtained from few accelerometers is used to retrieve and play back human motions from a database. In [44], the authors fuse information from densely placed inertial sensors with a global position estimate by using a laser range scanner equipped robot accompanying the tracked person.

In terms of full-body motion tracking with visual and inertial cues, the most similar approaches to the current work are probably [38] and [45]. In [38], the authors combine a generative tracker and a discriminative tracker by retrieving closest poses in a database. A visibility model based on depth images (kinect) as well as an inertial database lookup is used. In [45] IMUs are used to derive a manifold of poses that satisfy the sensor orientation constraints. A particle-based optimization scheme is then applied to find the pose in the manifold, which best matches the image information obtained from video cameras. The proposed hybrid tracker in this work is based on fast local optimization, while the approach in [45] relies on global optimization which is computationally too expensive for many applications. Gall et al. [27] have demonstrated, that fusing global and local optimization methods can lead to systems which combine the best of both worlds; similarly [45] could be combined with the hybrid tracker to recover from tracking failures. In contrast to [38], our proposed approach directly optimizes consistency of model and sensor measurements and is not restricted to motions in a prerecorded inertial database.

3 Exponential Maps for Rigid Body Motion

To model human joint motion, it is often needed to specify the axis of rotation of the joint. For example we might want to specify the motion of the knee joint as a rotation about an axis perpendicular to the leg and parallel to the hips. Therefore, for our purpose the axis-angle representation is optimal because rotations are described as an angle θ and an axis in space $\omega \in \mathbb{R}^3$ where θ determines the amount of rotation about ω. Unlike quaternions the axis-angle, requires only three parameters $\theta\omega$ to describe a rotation. The axis angle representation does not suffer from gimbal lock and their singularities occur in a region of parameter space that can be easily avoided. For a more detailed description we refer the reader to [46], [47].

3.1 *The Exponential Formula*

Every rotation $\mathbf{R}$ can be written in exponential form in terms of the axis of rotation $\omega \in \mathbb{R}^3$, s.t. $\|\omega\| = 1$ and the angle of rotation θ as

$$\mathbf{R} = \exp(\theta\widehat{\omega}), \tag{1}$$

where $\widehat{\omega} \in so(3)$ is the skew symmetric matrix constructed from ω. The elements of $so(3)$ are skew symmetric matrices i.e., matrices that verify $\{\mathbf{A} \in \mathbb{R}^{3\times3} | \mathbf{A} = -\mathbf{A}^T\}$. Given the vector $\theta\omega = \theta[\omega_1, \omega_2, \omega_3]^T$ the skew symmetric matrix is constructed with the wedge operator $\wedge$ as follows:

$$\theta\widehat{\omega} = \theta \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{2}$$

By definition, the multiplication of the matrix $\widehat{\omega}$ with a point p is equivalent to the cross-product of the vector ω with the point. Hence, the tangential direction of a rotating point is obtained as $\dot{\mathbf{p}}(t) = \omega \times \mathbf{p}(t) = \widehat{\omega}\mathbf{p}(t)$, which is a differential equation that can be integrated to obtain the the exponential formula in Eq. (1).

The exponential map of a matrix $\mathbf{A} \in \mathbb{R}^{3\times3}$ is analogous to the exponential used for real numbers $a \in \mathbb{R}$. In particular the Taylor expansion of the exponential has the same form:

$$\exp(\theta\widehat{\omega}) = e^{(\theta\widehat{\omega})} = I + \theta\widehat{\omega} + \frac{\theta^2}{2!}\widehat{\omega}^2 + \frac{\theta^3}{3!}\widehat{\omega}^3 + \cdots \tag{3}$$

Exploiting the fact that $(\theta\widehat{\omega})$ is screw symmetric, we can easily compute the exponential of the matrix $\widehat{\omega}$ in closed form using the *Rodriguez formula*:

$$\exp(\theta\widehat{\omega}) = I + \widehat{\omega}\sin(\theta) + \widehat{\omega}^2(1 - \cos(\theta)), \tag{4}$$

where only the square of the matrix $\widehat{\omega}$ and sine and cosine of real numbers have to be computed. Note that this formula allows us to reconstruct the rotation matrix from the angle θ and the axis of rotation ω by simple operations and this is probably the main justification of using the axis-angle representation at all. The exponential map formulation can be extended to represent rigid body motions, namely any motion composed by a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. This is done by extending the parameters $\theta\omega$ with $\theta v \in \mathbb{R}^3$ which is related to the translation along the axis of rotation and the location of the axis. These six parameters form the *twist coordinates* $\theta\xi = \theta(v_1, v_2, v_3, \omega_1, \omega_2, \omega_3)$ of a twist. Analogous to Eq. (1), any rigid motion $\mathbf{G} \in \mathbb{R}^{4\times4}$ can be written in exponential form as:

$$\mathbf{G}(\theta, \omega) = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \exp(\theta\widehat{\xi}), \tag{5}$$

where the 4×4 matrix $\theta\widehat{\xi} \in se(3)$ is the *twist action* and is a generalization of the screw symmetric matrix $\theta\widehat{\omega}$ of Eq. (2). The twist action is constructed from the twist coordinates $\theta\xi \in \mathbb{R}^6$ using the wedge operator $\wedge$

$$[\theta\xi]^\wedge = \theta\widehat{\xi} = \theta \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

and its exponential can be computed using the following formula

$$\exp(\theta\widehat{\xi}) = \begin{bmatrix} \exp(\theta\widehat{\omega}) & (I - \exp(\theta\widehat{\omega}))(\omega \times v + \omega\omega^T v\theta) \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{7}$$

with $\exp(\theta\widehat{\omega})$ computed by using the Rodriguez formula Eq. (4) as explained before.

3.2 Pose Parameterization

The dynamics of the subject are modelled by a *kinematic chain* $\mathcal{F}$, which describes the motion constraints of an articulated rigid body such as the human skeleton. A kinematic chain models the motion of a body segment as the motion of the previous body segment in the chain and an angular rotation around a joint axis. Specifically, the kinematic chain

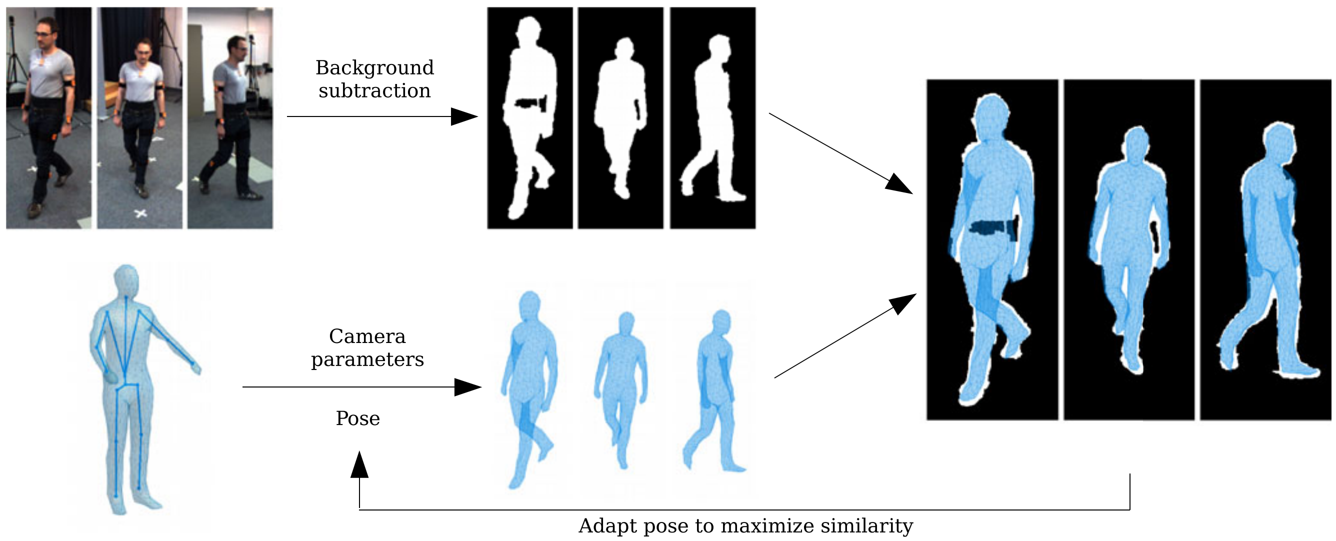

Fig. 2. General tracking procedure of the video tracker, using silhouettes as image features. Multi-view silhouettes are obtained by background subtraction. In parallel, the mesh model is adapted to the current pose and projected to the respective camera views. One seeks for the pose parameters that best explain the image evidence.

is defined with a 6 DoF (degree of freedom) root joint representing the global rigid body motion and a set of 1 DoF revolute joints describing the angular motion of the limbs. Higher DoF joints like hips or shoulders are represented by concatenating two or three 1 DoF revolute joints; for a comparison of balljoint parameterizations see [48].

The root joint is expressed as a twist of the form $\theta\xi$ with the rotation axis orientation, location, and angle as free parameters. Revolute joints are expressed as special twists with no pitch of the from $\theta_j\xi_j$ with known ξ_j (the location and orientation of the rotation axis as part of the model representation). Therefore, the full configuration of the kinematic chain is completely defined by a $(6+n)$ vector of free parameters

$$\mathbf{x} := (\theta\xi, \theta_1, \ldots, \theta_n) \tag{8}$$

similar to [49]. Now, for a given point $\mathbf{p} \in \mathbb{R}^3$ on the kinematic chain, we define $\mathcal{J}(\mathbf{p}) \subseteq \{1, \ldots, n\}$ to be the ordered set that encodes the joint transformations influencing $\mathbf{p}$. Let $\bar{\mathbf{p}}_s = \frac{\mathbf{p}}{1}$ be the homogeneous coordinate of $\mathbf{p}$ and denote $\mathcal{P}_c()$ as the associated projection with $\mathcal{P}_c(\bar{\mathbf{p}}) = \mathbf{p}$. Then, the transformation of a point $\mathbf{p}$ using the kinematic chain $\mathcal{F}(\mathbf{x};\mathbf{p})$ and a parameter vector $\mathbf{x}$ is defined by

$$\begin{aligned}\mathcal{F}(\mathbf{x};\mathbf{p}) &= \mathcal{P}_c\big(\mathbf{G}^{TB}(\mathbf{x})\bar{\mathbf{p}}_s(0)\big)\\ &= \mathcal{P}_c\left(\left(\exp(\theta\hat{\xi})\prod_{j\in\mathcal{J}(x)}\exp(\theta_j\hat{\xi}_j)\right)\bar{\mathbf{p}}_s(0)\right).\end{aligned} \tag{9}$$

Here, $\mathcal{F}(\mathbf{x};\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the function representing the total rigid body motion $\mathbf{G}^{TB}(\mathbf{x})$ of the segment in the chain where $\mathbf{p}$ belongs. Eq. (9) is commonly known as the *product of exponentials formula* [46], denoted as $\mathcal{F}(\mathbf{x};\mathbf{p})$.

4 Video-Based Tracker

The input of our video tracker consists of a rigid surface mesh of the actor obtained from a laser scanner and multiview images obtained by a set of calibrated and sychronized RGB-cameras. In order to relate the surface model to the human's images we find correspondences between the 3D surface vertices and the 2D image contours obtained with background subtraction, see Fig. 2. We first collect 2D-2D correspondences by matching the projected surface silhouette with the background subtracted image contour. Thereby, we obtain a collection of 2D-3D correspondences since we know the 3D counterparts of the projected 2D points of the silhouette. In the experiments we only use the silhouettes as image features. We then minimize the distance $\mathbf{e}_i$ between the transformed 3D points $\mathcal{F}(\mathbf{x};\mathbf{p}_i(0))$ of the model and the projection rays defined by the 2D contour points $\mathbf{p}_i$ and the respective camera center. This gives us a point-to-line constraint for each correspondence. Defining $L_i = (\mathbf{n}_i, \mathbf{m}_i)$ as the 3D Plücker line with unit direction $\mathbf{n}_i$ and moment $\mathbf{m}_i$ of the corresponding 2D point $\mathbf{r}_i = [x_i, y_i]$, the point to line distance residual $\mathbf{e}_i \in \mathbb{R}^3$ can be expressed as

$$\mathbf{e}_i = \mathcal{F}(\mathbf{x};\mathbf{p}_i) \times \mathbf{n}_i - \mathbf{m}_i\,. \tag{10}$$

Similar to Bregler et al. [50] we now linearize the equation by using $\exp(\theta\widehat{\xi}) = \sum_{k=0}^{\infty}\frac{(\widehat{\theta\xi})^k}{k!}$. With $\mathbf{I}$ as identity matrix, this results in

$$\left(\mathbf{I} + \Delta\widehat{\xi} + \sum_{j\in\mathcal{J}(x)}\Delta\theta_j\widehat{\xi}'_j\right)\mathbf{p}_i(\mathbf{x}) \times \mathbf{n}_i - \mathbf{m}_i = \mathbf{0}\,. \tag{11}$$

where $\widehat{\xi}'_j$ is the jth twist in the chain transformed to the current pose configuration. Having N correspondences, the energy we minimize E_{video} is the sum of squared point-to-line distances $\mathbf{e}_i$

$$\arg\min_{\mathbf{x}} E_{\text{video}}(\mathbf{x}) = \sum_{i=1}^{N}\|\mathbf{e_i}\|^2 \tag{12}$$

$$= \sum_{i=1}^{N}\left\|\mathcal{F}(\mathbf{x};\mathbf{p}_i) \times \mathbf{n}_i - \mathbf{m}_i\right\|^2 \tag{13}$$

which can be locally optimized. After linearization, Eq. (13) can be re-ordered into an equation of the form

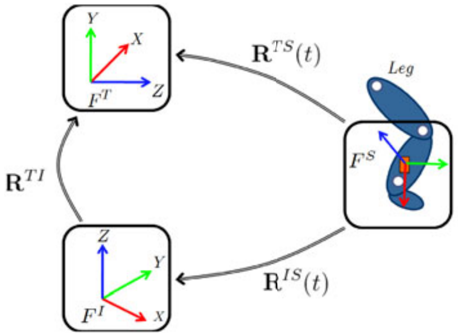

Fig. 3. Global frames: tracking frame F^T and inertial frame F^I. Local frame: sensor frame F^S.

$\mathbf{J}_{\text{video}}(\mathbf{x})\Delta\mathbf{x} = \mathbf{e}_{\text{video}}$. Collecting a set of such equations leads to an over-determined system of equations, which can be solved using numerical methods like the Householder algorithm. The pose parameters are then updated as $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$. The Rodriguez formula can be applied to reconstruct the group action $\mathbf{g}$ from the estimated twists $\theta_j\xi_j$. Then, the 3D points can be transformed and the process is iterated until convergence.

5 IMUs

An Inertial Measurement Unit or IMU is an electronic device that measures orientation and acceleration, using a combination of accelerometers and gyroscopes, sometimes also magnetometers. IMUs are very appealing because they provide a direct 3D measurement in contrast to images where the 3D information needs to be hallucinated. Furthermore, IMUs do not suffer from occlusions/self-occlusions and are not restricted to a designated recording volume. However, they have some limitations:

- Wearing many IMU is intrusive for the subject.
- IMUs suffer from drift in continuous operation. Sensor biases are constantly estimated and this is difficult, especially if the local magnetic field is distorted by ferromagnetic material in the surrounding.
- No positional measurement is directly available from the IMU. One could in principle derive position from the acceleration measurements but we have found this to be numerically unstable.
- The orientation data provided by IMUs suffers from lag. That is due to the fact that orientation data is obtained as the output of a Kalman filter that integrates acceleration, magnetometer and gyroscope information together. This is lag is specially problematic during fast motions.

Hence, we introduce a hybrid tracker that fuses information coming from a small set of IMUs (we use 5) and information coming from video cameras to compensate for the drawbacks of each sensor type.

The IMU measurements are taken with respect to a global inertial coordinate frame F^I, which is commonly defined by gravity and magnetic north direction. The video tracking coordinate frame F^T is defined by a calibration cube placed in the recording volume and usually differs from the inertial frame. Therefore, in order to be able to integrate the orientation data from the inertial sensors into our tracking system, we must determine the rotational offset $\mathbf{R}^{TI} : F^I \rightarrow F^T$ between both coordinate systems, see Fig. 3. Then, we can easily transform the IMU data according to

$$\mathbf{R}^{TS}(t) = \mathbf{R}^{TI}\mathbf{R}^{IS}(t)\,, \tag{14}$$

such that they define a map from the local sensor frame F^S to the tracking frame F^T.

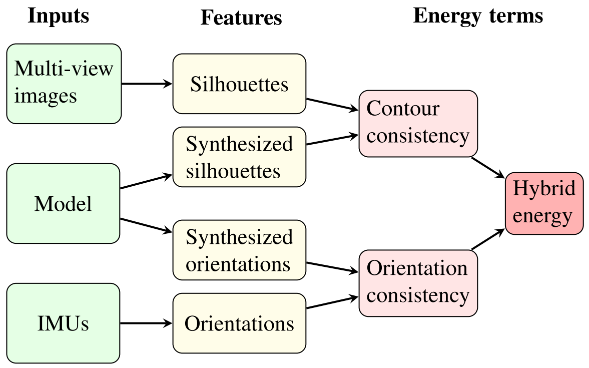

Fig. 4. Sketch of the hybrid tracker pipeline. Silhouette and orientation features are obtained from the inputs and their consistencies are combined in a hybrid energy term. We search for the model pose, which results in the minimal hybrid energy.

6 Hybrid Tracker

The input of our hybrid tracker is identical to the video tracker, but extended with global orientation measurements of the IMUs. We define a joint energy E_{hybrid} that measures the consistency between pose estimates with measurements coming from video and inertial sensors:

$$\arg\min_{\mathbf{x}} E_{\text{hybrid}}(\mathbf{x}) = E_{\text{video}}(\mathbf{x}) + \lambda E_{\text{sens}}(\mathbf{x}), \tag{15}$$

where $E_{\text{video}}(\mathbf{x})$ is the energy cost corresponding to the video measurements defined in Eq. 13 and $\lambda E_{\text{sens}}(\mathbf{x})$ is the cost associated with the IMU orientation measurements. To have a balanced energy we normalize the individual terms in the range of $[0, 1]$. As we will see in Section 6.1, $E_{\text{sens}}(\mathbf{x})$ can also be expressed as a sum of squared errors. This allows us to use numerical optimization techniques such Newton-Raphson or Levenberg-Marquardt. Let $\mathbf{e}_{\text{video}} : \mathbb{R}^D \mapsto \mathbb{R}^{3N}$ be the vector valued function of residuals of image correspondences and $\mathbf{e}_{\text{sens}} : \mathbb{R}^D \mapsto \mathbb{R}^{3N_s}$ the function of orientation residuals, where N_s is the number of available sensors. Now, we can express the energy in Eq. (15) as

$$\begin{aligned}\arg\min_{\mathbf{x}} \mathbf{e}^T_{\text{hybrid}}(\mathbf{x})\mathbf{e}_{\text{hybrid}}(\mathbf{x}) &= \\ \mathbf{e}^T_{\text{video}}(\mathbf{x})\mathbf{e}_{\text{video}}(\mathbf{x}) &+ \sqrt{\lambda}\mathbf{e}^T_{\text{sens}}(\mathbf{x})\sqrt{\lambda}\mathbf{e}_{\text{sens}}(\mathbf{x}).\end{aligned} \tag{16}$$

Eq. (16) is then iteratively linearized and the step $\Delta\mathbf{x}$ is found by solving the following linear system

$$\begin{bmatrix} \mathbf{J}_{\text{video}}(\mathbf{x}) \\ \sqrt{\lambda}\,\mathbf{J}_{\text{sens}}(\mathbf{x}) \end{bmatrix} \Delta\mathbf{x} = \begin{bmatrix} \mathbf{e}_{\text{video}}(\mathbf{x}) \\ \sqrt{\lambda}\,\mathbf{e}_{\text{sens}}(\mathbf{x}) \end{bmatrix}. \tag{17}$$

The pose parameters are then updated as $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \Delta\mathbf{x}$. The term corresponding to the video data is explained in the previous section. The term for the inertial sensors is explained in Section 6.1. In Fig. 4 we summarize the main ingredients of the hybrid tracker.

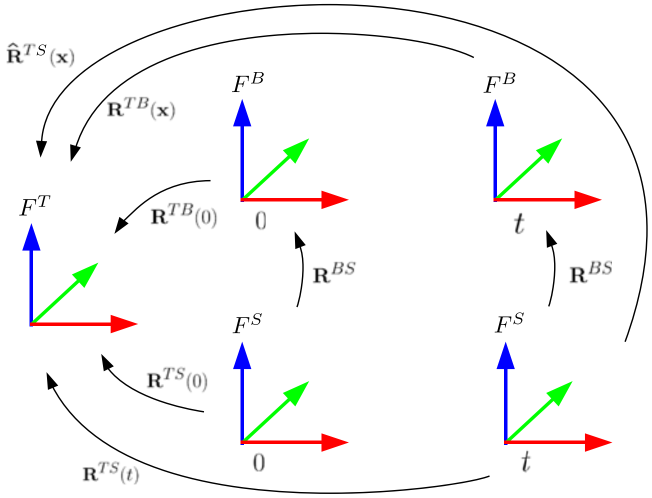

Fig. 5. Integration of orientation data into the video-based tracker. *Ground-truth orientation*: clockwise down path from F^S at time t to F^T. *Tracking orientation*: anti-clockwise upper path from F^S at time t to F^T.

6.1 Geodesic Distance Minimization

In this section we explain how to integrate the orientation data into the video-based tracker described earlier. In particular, we derive the linearization of a cost function that accounts for orientation consistency. After linearization this can be integrated into a big linear system according to Eq. (17).

In order to relate the orientation data to the differential twist parameters $\mathbf{x}_t$ of our model, we will compare the *ground-truth orientations* $\mathbf{R}^{TS}(t)$ of each of the sensors with the estimated sensor orientations from the tracking procedure $\hat{\mathbf{R}}^{TS}(\mathbf{x}_t)$, which we will denote as *tracking orientation*. For the sake of clarity we will drop the time subindex $\mathbf{x}_t$ and just write $\hat{\mathbf{R}}^{TS}(\mathbf{x})$, and will consider an energy for a single sensor. We define the estimation error $\mathbf{e}_{\text{sens}}$ in terms of the screw coordinates $\omega_{\text{rel}}(\mathbf{x}) \in \mathbb{R}^3$ of the relative rotation between *tracking* and *ground-truth orientation*

$$\mathbf{e}_{\text{sens}}(\mathbf{x}) = \omega_{\text{rel}}(\mathbf{x}) = \log\left(\mathbf{R}^{TS}(t)\hat{\mathbf{R}}^{TS}(\mathbf{x})^{-1}\right), \tag{18}$$

see Section 3. The energy cost related to orientation consistency E_{sens} can now be expressed as

$$\arg\min_{\mathbf{x}} E_{\text{sens}}(\mathbf{x}) = \|\mathbf{e}_{\text{sens}}(\mathbf{x})\|^2. \tag{19}$$

Note that E_{sens} corresponds to the squared geodesic distance between $\mathbf{R}^{TS}(t)$ and $\hat{\mathbf{R}}^{TS}(\mathbf{x})$.

We can linearize Eq. (19) and reformulate our objective function in terms of an optimal pose variation $\Delta\mathbf{x}$

$$\arg\min_{\Delta\mathbf{x}} \|\omega_{\text{rel}}(\mathbf{x}) + \frac{\Delta\omega_{\text{rel}}(\mathbf{x})}{\Delta\mathbf{x}}\Delta\mathbf{x}\|^2. \tag{20}$$

The expression $\frac{\Delta\omega_{\text{rel}}(\mathbf{x})}{\Delta\mathbf{x}}$ maps an increment in parameter space to the equivalent screw of the associated rigid motion. It corresponds to the Jacobian $\mathbf{J}_{\text{ori}} : \mathbb{R}^D \mapsto so(3)$ of the orientation forward kinematics map $\mathcal{F} : \mathbb{R}^D \mapsto SO(3)$. Since Eq. (20) is essentially a least squares problem, the optimal step can be found by solving the following linear equations

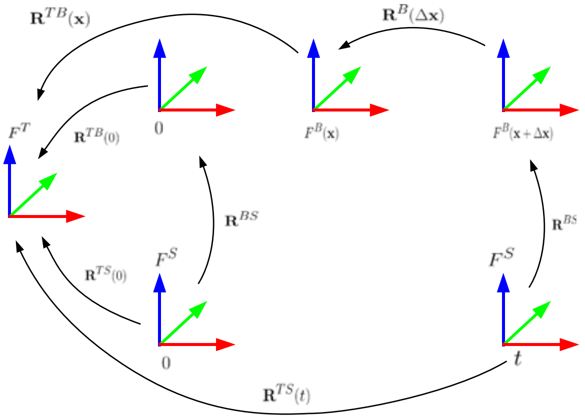

Fig. 6. Alternative interpretation of integrating orientation data into the video-based tracker. In comparison to Fig. 5 the *tracking orientation* path is extended by a local infinitesimal rotation $\mathbf{R}^B(\Delta\mathbf{x})$ of the body frame F^B.

$$\mathbf{J}_{\text{ori}}\Delta\text{x} \;= -\omega_{\text{rel}}(\mathbf{x}). \tag{21}$$

Thus, our sensor estimation error Jacobian $\mathbf{J}_{\text{sens}}$ is simply the Jacobian of the forward kinematic map $\mathbf{J}_{\text{ori}}$. We can now setup those equations for all orientation sensors and plug them into the linear system defined in Eq. (17).

However, we need to define the *ground-truth orientations* $\mathbf{R}^{TS}(t)$ and the estimated sensor orientations $\hat{\mathbf{R}}^{TS}(\mathbf{x})$. Recall from Section 5 that the sensor orientation data is given as a rotation matrix $\mathbf{R}^{TS}(t) : F^S \rightarrow F^T$ defining the transformation from the local sensor frame F^S to the global tracking frame F^T. In order to derive an expression for $\hat{\mathbf{R}}^{TS}(\mathbf{x})$, we introduce the body frame F^B (the local frame of a segment in the chain, e.g., the leg). As depicted in Fig. 5, the tracking orientation can be constructed by the concatenated transformation $\mathbf{R}^{BS}(t)$ and $\mathbf{R}^{TB}(\mathbf{x})$, i.e., $\hat{\mathbf{R}}^{TS}(\mathbf{x}) = \mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{BS}$. The first transformation defines the mapping from the sensor frame to the body frame $\mathbf{R}^{BS}(t) : F^S \rightarrow F^B$. The second term describes the total accumulated motion of a body segment at time t, i.e., $\mathbf{R}(\mathbf{x}) : F^B \rightarrow F^T$. As the sensor is rigidly attached to the body segment, this mapping remains constant during tracking and we can compute it in the first frame

$$\mathbf{R}^{BS} = \mathbf{R}^{TB}(0)^{-1}\mathbf{R}^{TS}(0), \tag{22}$$

where $\mathbf{R}^{TB}(0)$ is the configuration of the body part B in the first frame where the sensor is attached.

6.2 Chordal Distance Minimization

The previous derivation allows us to integrate linearized equations associated to orientation consistency to the equations corresponding to the cost associated with the image evidence. It is however interesting to derive alternative equations from a more geometric point of view.

Consider the local infinitesimal rotation $\mathbf{R}^B(\Delta\mathbf{x})$ of frame F^B, see Fig. 6. $\mathbf{R}^B(\Delta\mathbf{x})$ is defined in the body frame and represents the transformation from $F^B(\mathbf{x}+\Delta\mathbf{x})$ to $F^B(\mathbf{x})$. The *tracking orientation* $\hat{\mathbf{R}}^{TS}$ is now given by the longer path in Fig. 6, $F^S \overset{\mathbf{R}^{BS}}{\Longrightarrow} F^B(\mathbf{x}+\Delta\mathbf{x}) \overset{\mathbf{R}^{B}}{\Longrightarrow} F^B_{\mathbf{x}+\Delta\mathbf{x}} \overset{\mathbf{R}^{TB}(\mathbf{x})}{\Longrightarrow} F^T$. In We can make this transformation matrix match the *ground-truth orientation* $\mathbf{R}^{TS}$ by minimizing the geodesic distance between them. This leads to exactly the same set of equations as in Eq. (21). However, in [22] the squared chordal distance was minimised:

$$\arg\min_{\Delta \mathbf{x}} \left\| \mathbf{R}^{TB}\ (\mathbf{x})\mathbf{R}^{B}\ (\Delta \mathrm{x})\ \mathbf{R}^{BS} - \mathbf{R}^{TS}\ (t) \right\|_F^2 . \tag{23}$$

The rotation $\mathbf{R}^B(\mathbf{x})$ defined in the body frame is related to the rotation $\mathbf{R}(\mathbf{x})$ defined in the tracking frame by the *adjoint transformation* $Ad_{\mathbf{R}^{-1}(\mathbf{x})}$,

$$\mathbf{R}^B(\Delta\mathbf{x}) = \mathbf{R}^{TB}(\mathbf{x})^{-1}\mathbf{R}(\Delta\mathbf{x})\mathbf{R}^{TB}(\mathbf{x}) . \tag{24}$$

Substituting $\mathbf{R}^B(\mathbf{x})$ by its expression in (24) it simplifies to

$$\arg\min_{\Delta \mathbf{x}} \left\| \mathbf{R}(\Delta\mathbf{x})\mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{BS} - \mathbf{R}^{TS}(t) \right\|_F^2 . \tag{25}$$

In [22], we have shown how to linearize Eq. (25) and integrated it into the linear system defined in Eq. (17). Nonetheless, it is interesting to take a closer look at the left term of Eq. (25). Substituting the rotational displacement R^{BS} in Eq. (25) by its expression in Eq. (22) $\mathbf{R}^{TB}(0)^{-1}\mathbf{R}^{TS}(0)$, and writing $\mathbf{R}^{TB}(\mathbf{x}) = \prod_{j=t-1}^{1}\mathbf{R}(j)\mathbf{R}^{TB}(0)$ in terms of instantaneous rotations we obtain

$$\mathbf{R}(\Delta\mathbf{x})\left(\prod_{j=t-1}^{1}\mathbf{R}(j)\right)\mathbf{R}^{TS}(0) . \tag{26}$$

This last equation has a very nice interpretation because the columns of the matrix $\prod_{j=t-1}^{1}\mathbf{R}(j))\mathbf{R}^{TS}(0)$ are simply the coordinates of the sensor axis in the first frame (columns of $\mathbf{R}^{TS}(0)$), rotated by the accumulated tracking motion from the first frame forward (i.e., not including the initialization motion in frame 0). This last result was very much expected and the interpretation is the following: if we have our rotation matrices defined in a reference frame F^T, we can just take the sensor axes in global coordinates in the first frame (columns of $\mathbf{R}^{TS}(0)$) and rotate them at every frame by the instantaneous rotational motions of the tracking. This will result in the estimated sensor axes in world coordinates, which is exactly the *tracking orientation* defined earlier in this section. Therefore, the problem can be simplified to additional *3D-vector to 3D-vector* equations which can be very conveniently integrated in our twist formulation. Being $\hat{\mathbf{r}}_1^{TS}(\mathbf{x}), \hat{\mathbf{r}}_2^{TS}(\mathbf{x}), \hat{\mathbf{r}}_3^{TS}(\mathbf{x})$ the *tracking orientation* basis axes at configuration $\mathbf{x}$, and $\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t)$*ground-truth orientation* basis axes in the current frame t, the constraint equations are

$$\mathbf{R}(\Delta\mathrm{x})\hat{\mathbf{r}}_i^{TS}(\mathrm{x}) = \mathbf{r}_i^{TS}, \quad i = 1\ldots3 \tag{27}$$

which can be linearized similarly as we did in the video-based tracker with image points to mesh points correspondences (*2D-point to 3D-point*). The difference now is that since we rotate vectors, only the rotational component of the twists is needed. Each additional sensor results in an additional nine equations in the linear system

$$\left(\mathbf{I} + \Delta\widehat{\omega} + \sum_{j\in\mathcal{J}(\mathbf{x})}\Delta\theta_j\widehat{\omega}'_j\right)\hat{\mathbf{r}}_i^{TS}(\mathbf{x}) = \mathbf{r}_i^{TS}(t), \quad i = 1\ldots3 \tag{28}$$

which depends only on $\theta_j\widehat{\omega}_j$. The last equations can more conveniently be expressed in matrix form as $\mathbf{J}_{\text{vec}}\ (\mathbf{x};\hat{\mathbf{r}}^{TS})\Delta\mathbf{x} = \mathbf{e}_{\text{vec},i}$ for $i = \{1\ldots3\}$. Here $\mathbf{J}_{\text{vec}} : \mathbb{R}^D \mapsto \mathbb{R}^3$ has almost the same structure as the positional pose Jacobian $\mathbf{J}_{\text{video}}$ of the video tracker except that it does not depend on the translational motion nor the location of the joints. This implies that we can integrate the sensor information into the tracking system independently of the initial sensor orientation and location at the body limb. Note that $\mathbf{J}_{\text{vec}}$ takes a vector $\mathbf{r}$ as input as opposed to a point in $\mathbf{J}_{\text{video}}$. Also, note the difference between $\mathbf{J}_{\text{vec}}$ which are the derivatives of a rotating vector $\mathbf{r}$ and is therefore local, and $\mathbf{J}_{\text{ori}}$ which maps to the tangential space $so(3)$.

Minimizing the chordal distance Eq. (28) leads to nine equations for each sensor. In the previous section we minimized the geodesic distance, where the rotational error is defined by screw coordinates producing only three equations. Hence, there is a discrepancy of six equations. According to *Euler's rotation theorem* an orientation $\mathbf{R} \in SO(3)$ can be expressed by a minimum of three real parameters. Thus, the geodesic error term operates on the minimal representation, while minimizing the Frobenius norm produces some additional dependent equations. Indeed, the three *3D-vector to 3D-vector* correspondences are related to the respective coordinate axes, meaning they have to be orthogonal. This relationship is covered by the six dependent equations. In the end, both methods minimize a distance metric of a relative rotation and lead to equivalent results but we minimize geodesic distance because it is more compact and efficient.

To conclude, we have derived the linearized equations for orientation consistency in terms of the geodesic distance (Section 6.1) and motivated the sensor integration from a more geometrical point of view within this section.

7 Experiments

In this section we evaluate our sensor fusion approach by comparing the video-based tracker with our proposed hybrid tracker. Learning-based stabilization methods or joint angle limits can also be integrated into the video-based tracker. However, we did not include further constraints to clearly demonstrate the influence of incorporating inertial data.

For our experimental evaluation we need inertial sensor data, which is missing in publicly available benchmarks for video-based trackers (e.g., *HumanEva* [24], *Human3.6M* [51]). In the preliminary work of this paper [22], the *MPI08* dataset [52] was used to evaluate the hybrid trackers performance. This dataset provides inertial data of 5 IMUs along with video data. In order to expand our experimental evaluation and provide enhanced error metrics, we have recorded a new dataset, *TNT15* [23], which includes data of 10 IMUs and eight synchronized RGB-cameras. Similar to [45], 5 IMU sensors were used for tracking and the residual five sensors were utilized for an independent validation measure. Additionally, we refrained from using a monochrome background cover as in [52] and recorded in a normal office room situation. This generates noisier silhouettes, as it becomes more difficult to clearly separate foreground from background. For the video-based tracker, noisy silhouettes are very demanding, since the local optimization scheme gets stuck in local minima more often. However, this

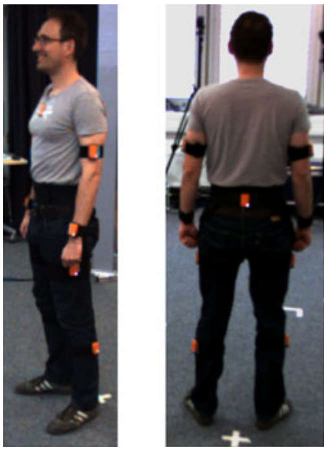

Fig. 7. Sensor placement: 10 sensors are strapped to body extremities (shank, thigh, forearm, upper arm), chest and waist.

semi-controlled scenarios where we think incorporating sparse inertial sensor data is ideal to enable high-quality, marker-less motion tracking with a fast, local optimization scheme.

7.1 Experimental Setup

7.1.1 TNT15 Dataset

The *TNT15* dataset consists of synchronized data streams from eight RGB-cameras and 10 IMUs. Four subjects perform five activities, namely *walking*, *running on the spot*, *rotating arms*, *jumping* and *punching*. The *walking* sequence consists of simple locomotion along a path with a 180 degree turn on the spot. In *running on the spot* the actors were asked to run on the spot at three different velocities. More complex motions are executed in the residual sequences. The *rotating arms* sequence contains forward, backward, synchronized and unsynchronized arm rotations, while *jumping* covers jumping jacks and skiing exercises. The *punching* sequence includes some dynamic boxing motions. In total, the dataset contains more than 4:30 minutes of video data, which amounts to almost 13 thousand frames at a frame rate of 50 Hz.

Multi-view video data was captured by a set of eight synchronized RGB-video cameras at a resolution of 800×600 px. In order to generate silhouettes we used a background subtraction method based on a pixel-wise Gaussian model, similar to [53].

The orientation data was recorded by 10 IMUs, which have been strapped to shanks, thighs, upper arms, lower arms, waist and chest. Fig. 7 illustrates the sensor placement. As stated above, the set of sensors is divided into tracking and validation sensors. Sensors at shanks, lower arms and waist have been selected for tracking, while the orientation measurements at thigh, upper arms and chest are utilized for validation.

7.1.2 Technical and Recording Setup

We have used the wireless MTw system provided by XSens [54] to capture inertial sensor data. The MTw system consists of a receiver and multiple MTw motion trackers. Each MTw motion tracker contains a three-axis gyroscope, accelerometer and magnetometer and has the dimensions 34.5×57.8×14.5 mm at a weight of 27 g. The sensory output is transmitted to the receiver and then fused using a proprietary algorithm to provide a 3D orientation. Orientation accuracy is specified to be smaller than 1 degree with an angular resolution of 0.05°[1]. In our experiments we are using 10 MTw units and record at a frame-rate of 50 Hz. The MTw units provide orientation data relative to a static global inertial frame F^I, which is computed internally in each of the sensor units at the initial static position. It is defined as follows: the Z-axis is the negative direction of gravity measured by the internal accelerometer. The X-axis is the direction of the magnetic north pole measured by the magnetometer. Finally, the Y-axis is defined by the cross product $Z \times X$. For each sensor the absolute orientation data is provided by a stream of quaternions that define, at every frame, the map or coordinate transformation from the local sensor coordinate system to the global one $\mathbf{R}^{IS}(t) : F^S \Rightarrow F^I$, see Section 5.

In order to integrate the sensor orientation measurements in our tracking system, we have to determine the mapping between the inertial and tracking coordinate systems. Since the Y-axis of the calibration cube for the tracking frame is perpendicular to the ground, the Y-axis of the tracking frame and the Z-axis of the inertial frame are aligned. Therefore, $\mathbf{R}^{TI}$ is a one parametric planar rotation that can be estimated beforehand using a calibration sequence [55]. This calibration step can be avoided if the tracking frame coincides with the inertial frame, which is easily achieved by aligning the sensors with the tracking frame and performing a heading reset. This action basically rotates the inertial frame such that its X-axis is adjusted to the MTw units X-axis. To synchronize the cameras with the IMU measurements, the actors were asked to perform a foot stamp at the beginning of every sequence which is easily detected in the camera images and IMU acceleration data.

7.1.3 Methodology

In order to evaluate our hybrid tracker performance we consider two frame-wise error metrics. First we investigate the angular error d_{ang} of our five validation IMUs w.r.t the corresponding bone orientations. We define d_{ang} as the geodesic distance between the *ground-truth* and *tracking* orientations, $\mathbf{R}^{TS}$ and $\hat{\mathbf{R}}^{TS}$, according to

$$d_{ang}(\mathbf{R}^{TS}, \hat{\mathbf{R}}^{TS}) = \|\log(\mathbf{R}^{TS}(\hat{\mathbf{R}}^{TS})^{-1})\|. \tag{29}$$

However, as the validation sensor error only measures orientation consistency, it is not sensitive to erroneous limb positions. Thus, we consider a second error metric and based on silhouette overlap between our projected model estimate and the image silhouette. It measures how well the estimate explains the video observations. Specifically, we define d_{xor} as the ratio of pixels in the XORed image to the number of pixels in the disjunct image

$$d_{xor}(S^{video}, S^{model}) = \frac{1}{K}\sum_{j=1}^{K} \frac{S_j^{video} \oplus S_j^{model}}{S_j^{video} \vee S_j^{model}}, \tag{30}$$

where S_j^{video} and S_j^{model} are the binary silhouette images for every camera view j. It is defined in the range of $[0, 1]$, i.e., a

[1] Specifications provided by the manufacturer.

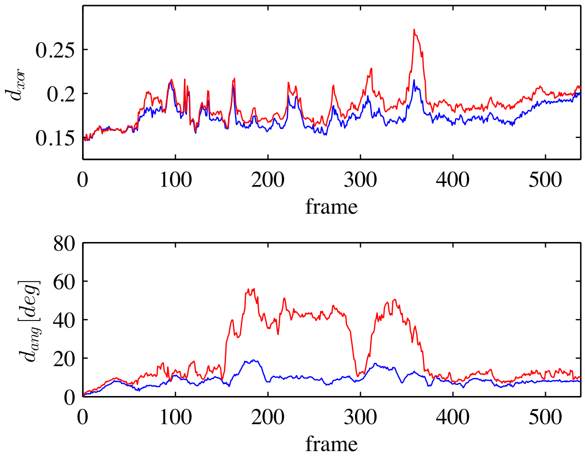

Fig. 8. Frame-wise XOR and orientation error for a walking sequence. The hybrid tracker (blue) performs well for the entire sequence. The video tracker (red) shows some large orientation errors between frames 160 and 370. Interestingly, this is almost invisible in the XOR error curve.

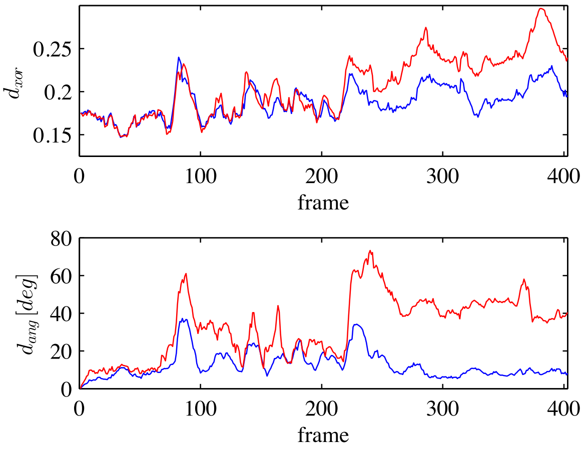

Fig. 9. Frame-wise XOR and orientation error for a dynamic punching sequence. The video tracker (red) struggles to track the complex motion and cannot recover from frame 210 on. The hybrid tracker (blue) performs better with respect to both error metrics.

XOR error of $d_{xor} = 0$ means the silhouettes are identical and $d_{xor} = 1$ indicates no overlap at all.

Our error metrics are different from the commonly used joint position error in motion capture experiments. If MoCap data is available one typically evaluates the euclidean distances of virtual markers corresponding to joint positions and the estimated joint positions of the motion tracker. However, this defines the state of a bone by two points in space, thus a rotational degree of freedom is not captured. Ideally, a metric for evaluating the human body pose should consider both joint position and joint orientation, i.e., the rigid motion of each bone of the human skeleton. MoCap data is not available for our experiments, thus we alternatively evaluate the estimated pose by measuring the bone orientations corresponding to the validation IMUs and the silhouette overlap error. Several experiments were carried out to investigate the tracker's performance. For every experiment we carefully analyse both error terms, the angular error d_{ang} and XOR error d_{xor}, and define the total tracking error as their concurrent combination.

7.2 Tracking Error Analysis

In this section we investigate the tracking error of the video and hybrid tracker for a fixed parameter setting. The hybrid tracker weighting parameter λ was set to 1.0, which implies equal weighting of silhouette and sensor terms of the objective function, see Eq. (15). First, we present the outcome of two exemplary sequences and then show results for the complete database.

Fig. 8 shows the frame-wise tracking error for a walking sequence. The lower graph shows the orientation error curve, containing the average d_{ang} for all validation sensors. For the hybrid tracker, the orientation error stays below 20 degree for the whole sequence. In contrast, the video tracker shows some large deviations from ground-truth between frames 160 and 370. A manual inspection revealed that the right upper arm was partially flipped about 180 degrees, see Fig. 1a. Interestingly, this is almost invisible in the XOR error curve, shown in the upper graph of Fig. 8. This gross error would also be almost imperceptible in standard joint error metrics used for example in HumanEva. This further demonstrates that the error metrics used in human pose estimation are incomplete and that orientation flips result in similar visual cues. Nevertheless, the hybrid tracker performs better even on this video based metric.

Another example sequence is shown in Fig. 9. It depicts the tracking error for a dynamic punching sequence. The angular error curve indicates that the video tracker struggles to track the complex motion which starts at frame 80. From frame 210 on, the average orientation error increases to approximately 60 degree and remains in this region for the rest of the sequence. The tracking failure is also visible in the XOR error curve. The hybrid tracker in contrast performs better with respect to both error metrics.

In both example sequences the hybrid approach performed better and successfully resolved visual ambiguities, which caused the video tracker to get stuck in undesired local minima or loose track completely.

To evaluate the performance of our hybrid tracker on more sequences, we computed the tracking error for all sequences of the data set. We denote the mean and standard deviation of the XOR error as μ_{xor} and σ_{xor} and the angular error μ_{ang} and σ_{ang}, respectively. As depicted in Table 1, the mean angular error got almost halved to 15.71 degree. Additionally, the XOR error has been reduced from 0.209 to 0.192, which shows that the orientation cues at the extremities propagate up the skeleton resulting in better pose estimates. The improved tracking results are also supported by the respective standard deviations.

We conclude this section with the remark, that the mean tracking errors might be higher as one expects. For the XOR error this is due to the imperfect silhouettes. As we have recorded the data in a normal office room situation,

TABLE 1
Mean Tracking Error Values μ and Standard Deviations σ for Video-Based and Hybrid Tracker for All Sequences of the Database

approach	μ_{xor}	σ_{xor}	μ_{ang}[deg]	σ_{ang}[deg]
video	0.209	0.056	30.17	42.38
hybrid	0.192	0.044	15.71	19.19

TABLE 2
Mean Angular Error $\mu_{ang}[deg]$ of the Validation Sensors Attached to Thighs, Chest and Upper Arms for the Video-Based and Hybrid Tracker for All Sequences of the Database

	lThigh	rThigh	chest	lUArm	rUArm
video tracker	19.12	12.36	11.97	61.03	46.28
hybrid tracker	8.64	6.75	6.88	27.30	28.96

background subtraction generates artifacts in the silhouettes, i.e., holes in the foreground regions and background pixels, which have been incorrectly labeled as foreground. Such an artifact is visible in the left silhouette in Fig. 2. In order to explore the main components of the mean angular error, we depict the respective terms for each validation sensor in Table 2. We immediately see that the validation sensors placed at the upper arms obtain a much higher error than the ones placed at thighs and chest. The reason for this imbalance is twofold. First, the data set contains very difficult arm motions, which are difficult to track. Second, given the tracking orientation at the lower arms, some joint angles are simply not observable. If we assume an extended elbow joint, a rotation of the lower arm can be caused by a rotation in the elbow or in the shoulder. Besides, the skeletal structure of the shoulder and arms is very complex and thus difficult to model accurately. However, in comparison to the video tracker, the hybrid tracker reduced the angular error of the upper arm validation sensors by approximately one half.

7.3 Tracking Error versus Feature Weighting

In this section we investigate how the weighting parameter λ influences the tracking error. Several weighting factors have been tested and their tracking error statistics are summarized in Table 3. Additionally, the respective mean XOR errors and mean angular errors are visualized in Fig. 10.

In comparison to the video tracker, even a small weighting parameter of $\lambda = 0.1$ improves the tracking and reduces the average XOR error by 0.04 and mean angular error by 7.32 degree. In terms of the XOR error, the best values are obtained for moderate weighting values between 0.5 and 2.0, reaching its minimum of 0.192 at $\lambda = 1.0$. For larger weighting factors the XOR error increases almost proportionally. Thus, forcing higher penalties on orientation deviations does not necessarily help to resolve visual ambiguities. The objective is altered in a way that the tracker finds a local minimum which might not be in accordance to the visual cues. This illustrates why it is important to consider both error metrics to really judge the tracker's performance.

The angular error shows a different behavior. For small weighting parameters, it drops from 30.17 degree for the video tracker ($\lambda = 0.0$) to 16.57 degree for the hybrid tracker with a weighting parameter $\lambda = 0.5$. Then the angular error decreases at a slower rate to its minimum of 15.41 degree at $\lambda = 3.0$. A further increase of λ results in slightly growing angular errors. At first this might be counter-intuitive, but we only penalize orientation deviations of the tracking sensors. A too large weight makes the tracker ignore the video cues producing un-plausible poses to match the sensor orientations.

TABLE 3
Mean Tracking Error for Varying Weighting of Sensor Cues, Computed over All Sequences of the Database

approach	μ_{xor}	σ_{xor}	μ_{ang}[deg]	σ_{ang}[deg]
video ($\lambda = 0.0$)	0.209	0.056	30.17	42.38
hybrid ($\lambda = 0.1$)	0.205	0.052	22.85	32.54
hybrid ($\lambda = 0.5$)	0.197	0.047	16.57	21.23
hybrid ($\lambda = 1.0$)	0.192	0.044	15.71	19.19
hybrid ($\lambda = 2.0$)	0.196	0.046	15.46	19.53
hybrid ($\lambda = 3.0$)	0.202	0.047	15.41	19.54
hybrid ($\lambda = 5.0$)	0.211	0.051	15.98	21.50
hybrid ($\lambda = 10.0$)	0.223	0.057	16.42	22.15

We defined the optimal weighting parameter to be the one that minimizes the XOR error, which happens for a weighting factor of $\lambda = 1.0$. This value has a slightly higher angular error compared to higher weighting factors, but we think the difference is reasonably small.

In general, the value of the weighting parameter depends heavily on how well the IMU readings represent the truth limb orientations. Sensor readings are corrupted by noise and rigidly attaching them to the bones is simply not possible. Thus, the ideal weighting parameter varies depending on the motion to be tracked.

7.4 Tracking Error versus Number of Views

So far we investigated how the trackers perform using eight cameras and five IMUs. In this section, we compare tracking results for reduced sets of camera views. Using only a subset of camera views has the effect that ambiguities in the silhouettes increase and local minima in the video-based energy term become more prevalent. Especially within those scenarios we expect adding inertial cues improves the tracking.

In order to compare the results to the previous experiments we compute the XOR error on the full set of available camera views and use only a subset for tracking. The weighting parameter λ of the hybrid tracker is fixed to 1.0 for all experiments in this section.

In a first experiment we evaluate the tracking performance for three distinct camera setups with six, four and two cameras. To distinguish the different setups, we denote the video and hybrid tracker as $video_{xc}$ and $hybrid_{xc}$, where x will be replaced by the quantity of cameras used for tracking. Fig. 11 shows the rough camera placements around the recording volume, where each camera is denoted with a circle, filled with its associated ID. For the six camera setup we removed cameras with ID 0 and 4 from the set of available camera views. Excluding those cameras removes an entire viewing direction on the scene, as they are arranged at

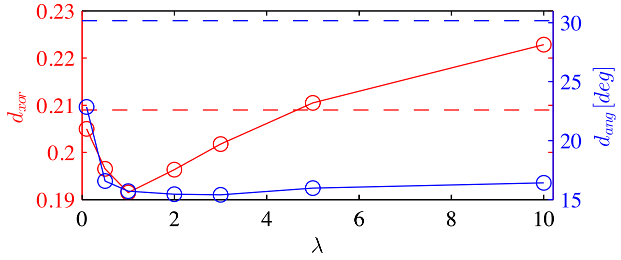

Fig. 10. Mean XOR error (red) and mean angular error (blue) for different weighting parameter λ of the hybrid tracker. The dashed lines show the respective error levels of the video tracker.

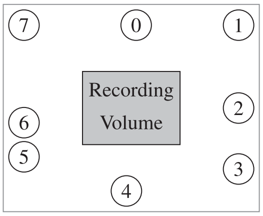

Fig. 11. Sketch of the camera setup. Eight cameras (IDs 0 to 7) are positioned around the recording volume.

opposite positions. For the four camera setup we remove half of the cameras and used only cameras with ID 0, 1, 2 and 3 for tracking. This setup would have the advantage of requiring less space than the eight camera setup. We also evaluated the tracker performance using only two cameras for tracking. We have chosen camera 0 and 2 for this scenario, as they have orthogonal viewing directions to the scene, which provides most information for a stereo setup. See Table 4, for a summary of which cameras are used for tracking in the respective setups.

In Table 5 the mean tracking errors and associated standard deviations are summarized for the respective camera setups. The tracking error of the video tracker deteriorates by reducing the number of available camera views. In contrast, the hybrid tracker is more robust to escalating visual ambiguities and the tracking error increases at a far slower rate.

Especially $video_{6c}$ shows inferior tracking results, though only two cameras have been removed from the tracking subset. Compared to the full camera setup, the mean XOR error rises by 0.099 and the mean angular error almost doubles. Obviously, the missing camera views provide vital information to the video tracker. In fact, inferring limb positions and orientations orthogonal to the viewing direction of the missing cameras is hampered. As a result it gets more difficult to determine, whether limbs are oriented forward or away from the residual camera views. However, the inertial sensor cues of the hybrid tracker provide accurate limb orientations and this suffices to correctly track the full body pose. The mean tracking errors of $hybrid_{6c}$ merely increased by 0.022 and 2.91 degree in comparison to the respective eight camera setup.

The previous camera setups were designed to investigate a certain scenario, such as a missing viewing direction or spatial restrictions for camera positioning. In order to investigate how the trackers perform with respect to the number of tracking cameras in general we now vary the tracking camera count. Thus, we incrementally reduce the number of tracking cameras and run the trackers for all camera permutations.

We computed the tracking errors of this experiment for a single actor of the database, which adds up to evaluating 2,460 recording sequences, each of it containing 621 frames on average. As can be seen in Fig. 12 the hybrid tracker clearly outperforms the video tracker. For both trackers the average XOR error and mean angular error increase by reducing the number of tracking cameras, but the average error terms of the hybrid tracker increase at a slower rate. The mean tracking error and standard deviations are summarized in Table 6.

TABLE 4
Tracking Camera List for Three Experiments with a Designated Scenario

Camera ID	0	1	2	3	4	5	6	7
6 cameras		X	X	X		X	X	X
4 cameras	X	X	X	X				
2 cameras	X		X					

Cameras that have been used for tracking during the scenarios are marked with a X.

TABLE 5
Tracking Error Statistics for Camera Setups Shown in Table 4

approach	μ_{xor}	σ_{xor}	μ_{ang}[deg]	σ_{ang}[deg]
$video_{8c}$	0.209	0.056	30.17	42.38
$hybrid_{8c}$	0.192	0.044	15.71	19.19
$video_{6c}$	0.308	0.139	54.64	56.16
$hybrid_{6c}$	0.214	0.059	18.62	26.31
$video_{4c}$	0.269	0.103	41.58	49.14
$hybrid_{4c}$	0.219	0.057	17.56	22.70
$video_{2c}$	0.360	0.138	58.68	54.76
$hybrid_{2c}$	0.258	0.079	20.63	26.44

So far, the mean XOR error has been averaged over all camera views, independent of which cameras have been selected for tracking. As the XOR evaluation metric and objective functions of the trackers both operate on silhouette data, we carried out a leave-one-camera-out experiment to prove validity of the XOR metric. For this experiment we only consider the XOR error of the camera that has not been used for tracking and reevaluate the tracking results of the previous seven tracking camera setup. We denote the leave-one-out XOR error as XOR_{loo} in the following. For the hybrid tracker the average XOR_{loo} results in 0.207 and 0.222 for the video tracker. In comparison, the XOR error μ_{xor} computed over all camera views is 0.197 and 0.210 respectively, see Table 6. Thus there is a minor offset in the values, but the relative difference for both trackers is almost identical. In fact, with respect to XOR_{loo} the hybrid tracker performs even better than for μ_{xor}, as the difference of mean values is slightly higher.

7.5 Tracking Error versus Sensor Lag

In this section we evaluate the robustness of the hybrid tracker to sensor lag. As we fuse sensor information of two

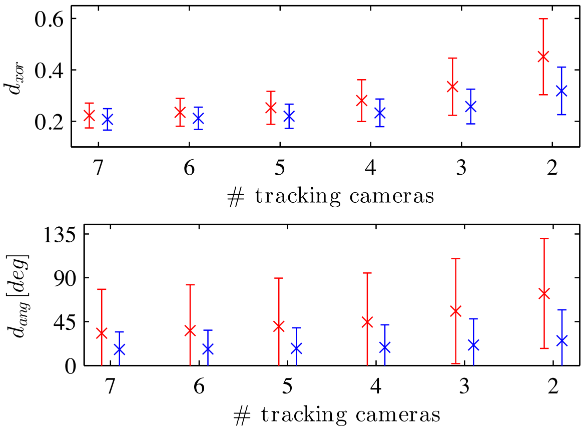

Fig. 12. Average tracking error of the video tracker (red) and hybrid tracker (blue) for varying tracking camera counts. The mean values are marked with a x and the respective bars show one standard deviation.

TABLE 6
Average Tracking Error for Varying Number of Camera Views

approach	μ_{xor}	σ_{xor}	μ_{ang}[deg]	σ_{ang}[deg]	N
$video_{7c}$	0.210	0.045	33.12	45.10	8
$hybrid_{7c}$	0.197	0.040	16.41	17.99	8
$video_{6c}$	0.222	0.051	35.86	46.91	28
$hybrid_{6c}$	0.201	0.042	16.89	19.35	28
$video_{5c}$	0.239	0.061	40.26	49.46	56
$hybrid_{5c}$	0.209	0.046	17.57	21.06	56
$video_{4c}$	0.265	0.079	44.65	50.21	70
$hybrid_{4c}$	0.221	0.052	18.65	22.95	70
$video_{3c}$	0.318	0.111	55.68	53.91	56
$hybrid_{3c}$	0.244	0.066	20.95	27.00	56
$video_{2c}$	0.431	0.152	73.83	56.45	28
$hybrid_{2c}$	0.299	0.094	25.48	31.71	28

The number of tracking cameras is varied from 7 to 2 and all N possible permutations have been considered.

TABLE 7
Tracking Error Statistics for Different Levels of Sensor Lag

offset [frames]	μ_{xor}	σ_{xor}	μ_{ang}[deg]	σ_{ang}[deg]
0	0.192	0.044	15.71	19.19
1	0.196	0.046	16.43	21.40
2	0.197	0.046	16.57	21.72
3	0.198	0.046	16.60	21.42
5	0.203	0.049	18.31	25.89
10	0.213	0.055	20.22	28.39
25	0.232	0.056	27.34	37.48

The orientation data of the IMUs used for tracking have been artificially delayed by constant time offsets of multiple frames.

independent measurement systems, namely video and IMUs, we desire and assume perfect synchronization of the measurement data. However, imperfect manual synchronization during post-processing, sampling rate jitter or time delays due to filtering might lead to asynchronous data streams. The latter refers to the Kalman-filtered orientation estimates of the IMUs, which might lag during high dynamical motions, see Section 5.

In order to evaluate how the hybrid tracker responds to asynchronous measurement data, we added constant time offsets to the orientation data streams of the tracking IMUs. This does not model possibly time-varying characteristics of measurement lag, but applies a constant worst-case delay on each frame. Our experiments comprise the tracking error, where IMU data is artificially delayed by 1, 2, 3, 5, 10 and 25 successive frames. At maximum this corresponds to a delay of 0.5 s at 50 Hz sampling rate. For all experiments in this section we have used the full set of available camera views and a weighting parameter of $\lambda = 1.0$.

As can be seen in Table 7 the mean XOR error and mean angular error increase if the orientation measurements of the tracking sensors are artificially delayed. However, up to a sensor lag of approximately seven frames, the hybrid tracker performs better in both error metrics compared to the video tracker. Thus, even though every orientation measurement is delayed by 0.14 s, the hybrid tracker is able to provide more accurate results than the video tracker.

For small delays such as three frames the hybrid tracker mean XOR error slightly increase by 0.006 and the mean angular error by 0.89 degree. In general, time delays due to jitter, manual synchronization and filtering do not exceed multiple frames. Thus, the preceding experiment shows that the hybrid tracker is robust to moderate asynchronicity in the measurements.

7.6 Evaluation on *HumanEva* Dataset

In order to evaluate the hybrid tracker with respect to ground-truth motion capture data we use the *HumanEva* [24] dataset. This dataset consists of multi-view video and ground-truth 3D body poses obtained from a commercial marker-based motion capture system. As IMU data are not available in *HumanEva* we use ground-truth body part orientations to generate virtual IMU data.

The *HumanEva* dataset consists of four actors performing several activities. Every activity is recorded three times and the data is then divided into disjoint subsets for validation, train and test purposes. For many sequences, including all test subsets, MoCap data is withheld for evaluation purposes. We evaluate the hybrid and video tracker on the first trial of validation sequences of actors 1-3. Because no surface models are provided for those actors we have used the method of [56] to estimate the subject shape from marker data alone. In particular, we used SMPL [57] as our body model which is publicly available. For efficiency, tracking was performed with a skeleton with fewer *DoF* and pose blend-shapes set to zero.

For all experiments on the dataset we proceed as follows. In order to initialize the surface model and align the virtual IMU readings, we use ground-truth body poses of the first frame. For every subsequent frame we estimate the body

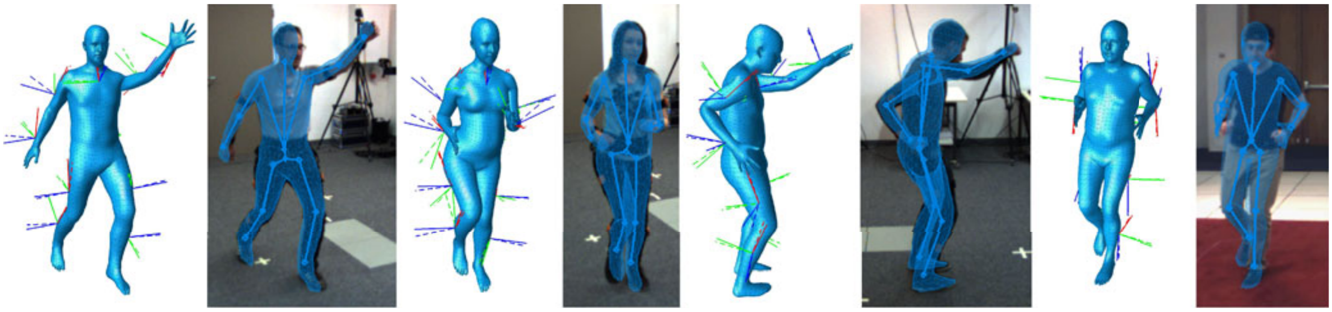

Fig. 13. We show three exemplary frames of the *TNT15* dataset and one of *HumanEva* (right most example). Each frame is is illustrated by two images. The images on the left depict the estimated model pose; ground-truth orientations are shown in solid lines and estimated orientations in dashed lines. The right images show the mesh projected to the respective RGB-image. For the *TNT15* sequences we have used a lower resolution mesh for tracking, which is actually visible in the respective RGB-images.

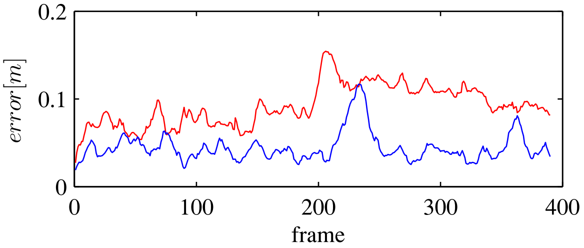

Fig. 14. Joint position error for the Jog sequence of subject 2 in the *HumanEva* dataset. The hybrid tracker (blue) outperforms the video tracker (red). A manual inspection revealed that twisted legs causes the peak of the hybrid tracker around frame 220. The video tracker has problems tracking the legs for the entire sequence.

pose using silhouettes and virtual IMU readings only. Silhouettes are generated by a background subtraction procedure provided along the dataset. All seven cameras are considered for tracking except for Subject 1, where the four monochrome cameras are excluded due to very poor segmentation results. For the hybrid tracker we use ground-truth orientations of lower arms, shanks and torso for tracking and equal weighting of orientation and visual cues.

Different to the previous experiments we now evaluate the tracking performance with respect to the average joint position error, see Section 7.1.3. We use the method proposed within the *HumanEva* dataset and compute the sum of euclidean distances of 15 virtual markers.

In Fig. 14, we show the joint position error for the jogging sequence of subject 2 obtained by the hybrid and video tracker. It clearly demonstrates the superior performance of the hybrid tracker. In Table 8 we show the average tracking results for all sequences that have been evaluated. For the hybrid tracker the mean joint position error is between $3.8 - 5.2$ cm. The video tracker is not capable to track the motions properly as the silhouettes are too ambiguous and achieves a mean joint position error of $7 - 13.8$ cm. Due to corrupted MoCap data, we excluded the Jog sequence of subject 3.

Plenty approaches have been evaluated on the *HumanEva* dataset in the literature. The majority report tracking errors with respect to the test sequences, which we could not use due to missing ground-truth motion capture data. Similar to our experiments, [58] and [11] also evaluated on the validation subsets and report an average joint position error of $5.9 - 7.7$ and $1.9 - 4.8$ cm for walking and jogging activities, respectively. [58] applies a loosely-connected-parts body model and combines an image likelihood function based on silhouette and edge features with body part detectors and uses non-parametric belief propagation for inference. Within the same publication a tracking error of 6.6 to 7.0 cm is reported for an algorithm based on an Annealed Particle Filter, using silhouette and edge features. [11] achieves state-of-the-art results with a discriminative approach based on Twin Gaussian Processes with HoG features generated from the three color camera views only.

To conclude, we have shown that a simple approach based on local optimization and silhouette features achieves competitive tracking errors, when inertial orientation information is incorporated. Thus without using more image features such as edge and color information or adding physical constraints, a sparse set of IMUs can improve the tracking significantly.

TABLE 8
Mean and Standard Deviations of the Joint Position Error for Validation Subsets of the *HumanEva* Dataset

Actor	Action	3D error [cm]	
		Video	Hybrid
S1	Walking	13.81 ± 6.41	4.19 ± 1.31
S1	Jog	11.97 ± 4.64	3.76 ± 1.57
S2	Walking	13.39 ± 2.88	4.88 ± 1.26
S2	Jog	9.21 ± 2.30	4.46 ± 1.66
S3	Walking	7.03 ± 2.56	5.15 ± 2.87

8 Conclusions

In this paper, we presented an approach for stabilizing full-body marker-less human motion capturing using a small number of additional inertial sensors. Reconstructing a 3D pose from 2D video data suffers from inherent ambiguities. We showed that a hybrid approach combining information of multiple sensor types can resolve such ambiguities, significantly improving the tracking quality. In particular, our orientation-based approach could correct tracking errors arising from rotationally symmetric limbs and noisy visual cues. Using only a small number of inertial sensors fixed at outer extremities stabilized the tracking for the entire underlying kinematic chain. Some qualitative results can be seen in Fig. 13.

In contrast to the preliminary work [22], we provide additional derivations and details to integrate orientation data and present an extended evaluation. A thorough evaluation on both orientation and video error metrics have proven the superior performance of the hybrid approach. We have shown that we require less cameras compared to a pure video-based tracker and have evaluated the robustness against sensor lag. Experiments on *HumanEva* dataset show that even using very basic image features we achieve competitive results compared to approaches which rely on learning or expensive inference methods. Another conclusion from our experiments is that commonly used error metrics based only on joint errors are incomplete to asses human pose estimation accuracy. To that end we make the TNT15 dataset including the 10 IMUs publicly available at [23] so that other researchers can use it to validate their human pose estimation methods.

Acknowledgments

The work is funded by the ERC-Starting Grant (DYNAMIC MINVIP). The authors gratefully acknowledge the support.

References

[1] T. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Comput. Vis. Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[2] A. Baak, B. Rosenhahn, M. Müller, and H. Seidel, “Stabilizing motion tracking using retrieved motion priors,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1428–1435.

[3] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker, “Detailed human shape and pose from images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[4] L. Herda, R. Urtasun, and P. Fua, “Implicit surface joint limits to constrain video-based motion capture,” in *Proc. Eur. Conf. Comput. Vis.*, 2004, vol. 3022, pp. 405–418.

[5] S. Ioffe and D. Forsyth, “Human tracking with mixtures of trees,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, vol. 1, pp. 690–695.

[6] H. Sidenbladh, M. Black, and D. Fleet, “Stochastic tracking of 3d human figures using 2d image motion,” in *Proc. Eur. Conf. Comput. Vis.*, 2000, vol. 1843, pp. 702–718.

[7] G. Shakhnarovich, P. Viola, and T. Darrell, “Fast pose estimation with parameter-sensitive hashing,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 750–757.

[8] J. Gall, A. Yao, and L. Van Gool, “2D action recognition serves 3D human pose estimation,” in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 425–438.

[9] A. Elgammal and C. Lee, “Inferring 3D body pose from silhouettes using activity manifoldlearning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 2, pp. II-681–II-688.

[10] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, “Discriminative density propagation for 3D human motion estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, vol. 1, p. 390.

[11] L. Bo and C. Sminchisescu, “Twin Gaussian processes for structured prediction,” *Int. J. Comput. Vis.*, vol. 87, pp. 28–52, 2010.

[12] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, 2014.

[13] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn, “Posebits for monocular human pose estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 2345–2352.

[14] R. Urtasun, D. J. Fleet, and P. Fua, “3D people tracking with Gaussian process dynamical models,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 238–245.

[15] J. Wang, D. Fleet, and A. Hertzmann, “Gaussian process dynamical models for human motion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.

[16] D. Brubaker M. A., Fleet and A. Hertzmann, “Physics-based person tracking using the anthropomorphic walker,” *Int. J. Comput. Vis.*, vol. 87, no. 1-2, pp. 140–155, 2010.

[17] P. Guan, A. Weiss, A. Balan, and M. Black, “Estimating human shape and pose from a single image,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1381–1388.

[18] L. Sigal, M. Vondrak, and O. Jenkins, “Physical simulation for probabilistic motion tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[19] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, “Performance capture from sparse multi-view video,” *ACM Trans. Graph. (TOG)*, vol. 27, no. 3, pp. 1–98, 2008.

[20] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. Seidel, “Motion capture using joint skeleton tracking and surface estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1746–1753.

[21] D. Roetenberg, “Inertial and magnetic sensing of human motion,” Univ. Twente, 2006.

[22] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn, “Multisensor-fusion for 3D full-body human motion capture,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 663–670.

[23] (2016). Multimodal human motion database TNT15 [Online]. Available: http://www.tnt.uni-hannover.de/project/TNT15/

[24] L. Sigal, A. Balan, and M. Black, “Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion,” *Int. J. Comput. Vis.*, vol. 87, no. 1, pp. 4–27, 2010.

[25] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3d pose estimation and tracking by detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 623–630.

[26] V. Belagiannis, X. Wang, B. Schiele, P. Fua, S. Ilic, and N. Navab, “Multiple human pose estimation with temporally consistent 3D pictorial structures,” in *Proc. Eur. Conf. Comput. Vis., ChaLearn Looking People Workshop*, 2014, pp. 742–754.

[27] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel, “Optimization and filtering for human motion capture,” *Int. J. Comput. Vis.*, vol. 87, pp. 75–92, 2010.

[28] W. Zhang, L. Shang, and A. Chan, “A robust likelihood function for 3d human pose tracking,” *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5374–5389, Dec. 2014.

[29] E. De Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel, “Marker-less deformable mesh tracking for human shape and motion capture,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[30] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt, “High-quality shape from multi-view stereo and shading under general illumination,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 969–976.

[31] C.-H. Huang, E. Boyer, N. Navab, and S. Ilic, “Human shape and pose tracking using keyframes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3446–3453.

[32] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1297–1304.

[33] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, “The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 103–110.

[34] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon, “Metric regression forests for human pose estimation,” in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 4.1–4.11.

[35] V. Ganapath, C. Plagemann, S. Thrun, and D. Koller, “Real time motion capture using a time-of-flight camera,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 755–762.

[36] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, *Real-Time Human Pose Tracking from Range Data*. New York, NY,USA: Springer, 2012.

[37] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, “A data-driven approach for real-time full body pose reconstruction from a depth camera,” in *Consumer Depth Cameras for Computer Vision*. New York, NY, USA: Springer, 2013, pp. 71–98.

[38] T. Helten, M. Müller, H.-P. Seidel, and C. Theobalt, “Real-time body tracking with one depth camera and inertial sensors,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1105–1112.

[39] F. Bogo, M. J. Black, M. Loper, and J. Romero, “Detailed full-body reconstructions of moving people from monocular RGB-D sequences,” in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2300–2308.

[40] L. Chen, H. Wei, and J. Ferryman, “A survey of human motion analysis using depth imagery,” *Pattern Recog. Lett.*, vol. 34, no. 15, pp. 1995–2006, Nov. 2013.

[41] H. Dejnabadi, B. Jolles, E. Casanova, P. Fua, and K. Aminian, “Estimation and visualization of sagittal kinematics of lower limbsorientation using body-fixed sensors,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1382–1393, Jul. 2006.

[42] Y. Tao, H. Hu, and H. Zhou, “Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation,” *Int. J. Robot. Res.*, vol. 26, no. 6, p. 607, 2007.

[43] R. Slyper and J. Hodgins, “Action capture with accelerometers,” in *Proc. ACM SIGGRAPH/Eurographics*, 2008, pp. 193–199.

[44] J. Ziegler, H. Kretzschmar, C. Stachniss, G. Grisetti, and W. Burgard, “Accurate human motion capture in large areas by combining IMU- and laser-based people tracking,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 86–91.

[45] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixé, M. Müller, H.-P. Seidel, and B. Rosenhahn, “Outdoor human motion capture using inverse kinematics and von mises-fisher sampling,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1243–1250.

[46] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Baton Rouge, LA, USA: CRC Press, 1994.

[47] G. Pons-Moll and B. Rosenhahn, “Model-based pose estimation,” in *Visual Analysis of Humans: Looking at People*. New York, NY, USA: Springer, 2011, pp. 139–170.

[48] G. Pons-Moll and B. Rosenhahn, “Ball joints for marker-less human motion capture,” in *Proc. Workshop Appl. Comput. Vis.*, 2009, pp. 1–8.

[49] B. Rosenhahn and T. Brox, “Scaled motion dynamics for markerless motion capture,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[50] C. Bregler, J. Malik, and K. Pullen, “Twist based acquisition and tracking of animal and human kinematics,” *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 179–194, 2004.

[51] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[52] (2015). Multimodal human motion database MPI08 [Online]. Available: http://www.tnt.uni-hannover.de/project/MPI08_Database/

[53] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[54] (2015). Xsens Motion Technologies [Online]. Available: http://www.xsens.com/

[55] A. Baak, T. Helten, M. Müller, G. Pons-Moll, B. Rosenhahn, and H. Seidel, "Analyzing and evaluating markerless motion tracking using inertial sensors," in *Proc. 3rd Int. Workshop Human Motion Conjunction with ECCV*, 2010, vol. 6553, pp. 137–150.

[56] M. Loper, N. Mahmood, and M. J. Black, "MoSh: Motion and shape capture from sparse markers," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 220:1–220:13, Nov. 2014.

[57] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[58] L. Sigal, M. Isard, H. Haussecker, and M. J. Black, "Loose-limbed people: Estimating 3d human pose and motion using nonparametric belief propagation," *Int. J. Comput. Vis.*, vol. 98, no. 1, pp. 15–48, 2012.

Timo von Marcard received the Dipl.-Ing.(FH) degree in mechatronics from the University of Applied Sciences Karlsruhe (2008) and the MSc degree in systems, control and mechatronics from the Chalmers University of Technology, Gothenburg in 2010. From 2010 to 2013 he was a developer for embedded software and algorithms at Otto Bock Healthcare GmbH. Since 2013, he is working as a research assistant toward a doctoral degree (Dr.-Ing.) at the Institut für Informationsverarbeitung (TNT) of the Leibniz-University of Hannover. His research interests are marker-less human motion capture and sensor fusion algorithms.

Gerard Pons-Moll received the degree in superior telecommunications engineering from the Technical University of Catalonia (UPC) in 2008 and the PhD degree from the Leibniz University of Hannover in 2014. From 2007 to 2008, he was at Northeastern University in Boston working on medical image analysis. In 2012, he was a visiting researcher at the University of Toronto. In 2012, he was an intern at the computer vision group at Microsoft Research Cambridge. In November 2013 to November 2015, he was a postdoc at the Max Planck Institute for Intelligent Systems in Tuebingen, Germany. Since November 2015, he is a research scientist. His research interests are human motion analysis, 3D modelling and using machine learning and graphics models to solve vision problems.

Bodo Rosenhahn studied computer science (minor subject medicine) at the University of Kiel. He received the Dipl.-Inf. and Dr.-Ing. from the University of Kiel in 1999 and 2003, respectively. From October 2003 to October 2005, he was a PostDoc at the University of Auckland, New Zealand, funded with a scholarship from the German Research Foundation (DFG). From November 2005 to August 2008, he was a senior researcher at the Max-Planck Institute for Computer Science in Saarbruecken. Since September 2008, he is a full professor at the Leibniz-University of Hannover, heading a group on automated image interpretation.