Real-time Full-Body Motion Capture from Video and IMUs

Charles Malleson Marco Volino Andrew Gilbert Matthew Trumble
John Collomosse Adrian Hilton
CVSSP, University of Surrey, Guildford, U.K.
charles.malleson@surrey.ac.uk, a.hilton@surrey.ac.uk

Abstract

A real-time full-body motion capture system is presented which uses input from a sparse set of inertial measurement units (IMUs) along with images from two or more standard video cameras and requires no optical markers or specialized infra-red cameras. A real-time optimization-based framework is proposed which incorporates constraints from the IMUs, cameras and a prior pose model. The combination of video and IMU data allows the full 6-DOF motion to be recovered including axial rotation of limbs and drift-free global position. The approach was tested using both indoor and outdoor captured data. The results demonstrate the effectiveness of the approach for tracking a wide range of human motion in real time in unconstrained indoor/outdoor scenes.

1. Introduction

Real-time capture of human motion is of considerable interest in various domains including entertainment and the life sciences. Recent advances in computer vision and the availability of commodity wireless inertial sensors [13, 3] are beginning to take motion capture from constrained studio settings to more natural, outdoor environments, and with less encumbrance of the performer from specialized costumes and optical marker setups traditionally required (*e.g.* [4, 2]), while still retaining a high level of capture fidelity.

In this work, a novel optimization-based approach is proposed which combines multi-modal input from inertial sensors and cameras to produce an estimate of the full-body pose in real time without requiring optical markers or a complex hardware setup. The solver optimizes the kinematic pose of the subject based on a cost function comprising orientation, acceleration, position and pose prior terms. In our setup, the orientation and acceleration constraints are provided by a sparse set of 6-13 Xsens inertial measurement units (IMUs) attached to body segments, and positional constraints are obtained from 2D joint detections [7] from two or more synchronised video camera views. In principle, the solver is agnostic as to the source of the inputs.

Combining video and IMU data improves the tracking performance compared to one or the other. The IMUs provide full rotational information for body segments, while the video information provides drift-free global position information.

2. Related work

IMUs and multi-view video data were combined by von Marcard *et al*. [17] to exploit the complementary properties of the data sources, *i.e*. drift free position from video and 3D limb orientation from IMUs. However no comparison is performed against commercial reference-quality motion capture (instead the results are compared with respect to consistency with silhouettes and IMU measurements), and processing time is not specified.

The 'Sparse Inertial Poser' (SIP) system proposed by von Marcard *et al*. [18] uses orientation and acceleration from 6 IMUs as input and is assisted by a prior pose model in the form of the SMPL body model [11]. However, SIP processes sequences as a batch and is thus not suitable for real-time, online operation. Furthermore, it is susceptible to drift in global position since it does not use visual information. Our system requires cameras in addition to sparse IMUs, but processes sequences online in real-time and without accumulating drift in global position.

Andrews *et al*. [6] perform real-time body tracking using a sparse set of labelled optical markers, IMUs, and a motion prior in an inverse dynamics formulation. In contrast, our method is markerless and does not require setting up a physics-rig of the subject.

Convolutional Pose Machines (CPMs) [19, 7] use deep neural networks to estimate 2D pose (joint locations) for multiple people from a single image, with video rate detection possible using GPU acceleration. In Tome *et al*. [14], CPMs are extended to detect 3D pose from a single RGB image by incorporating knowledge of plausible human poses in the training. In VNect [12], 3D pose is estimated in realtime from a single camera using CNNs and kinematic

fitting, while Zhou *et al*. [22] use CNNs for 2D joint detection and offline Expectation-Maximization over an entire sequence for 3D pose. Due to the monocular input, these methods are subject to depth ambiguity.

Trumble *et al*. [15] use convolutional neural networks on multiview video data to perform real-time motion capture. However, this requires extensive training from multi-view video data and the axial rotation of the limbs cannot be recovered since the input is based on visual hulls. Furthermore, controlled capture conditions are required for background segmentation. In contrast, our method requires minimal, simple training of the pose prior, while using a pretrained CPM detector for 2D detections. By incorporating IMU data, our method is able to recover axial rotation of the limbs while handling dynamic backgrounds and occlusions. In subsequent work, Trumble *et al*. [16] combined video and IMU input in a deep learning framework, including using LSTM (long short term memory, [9]) for temporal prediction to reduce noise, but still require at least four cameras and relatively controlled capture conditions for visual hull estimation.

Other recent approaches to realtime body tracking use other types of capture hardware for example Kinect (RGBD) cameras [20, 10] Kinect plus IMUs [8], or HTC Vive infra-red VR controllers strapped to the limbs [1].

Our work performs real-time, online, full-body markerless tracking in unconstrained environments using multiple-view video with as few as two cameras and 6 IMUs as input, recovering the full DoFs including axial rotation and drift-free global position.

3. Method

3.1. Notation and skeleton parametrization

The kinematic skeleton consists of a pre-defined hierarchy of n_b rigid bones, b attached at joints. The root bone $b = 1$ (*i.e*. the hips) has a global position, $\mathbf{t}_1$ and orientation, $\mathbf{R}_1$. Each child bone, $b \in [2, n_b]$ is attached to its parent with a fixed translational offset, $\mathbf{t}_b$, and pose-varying rotation, $\mathbf{R}_b$, w.r.t. the parent bone coordinates. In this work, $n_b = 21$ bones are used. The total degrees of freedom (DoF) are $d = 3 + 3 \times 21 = 66$, consisting of the root translation and 3 rotational degrees of freedom per joint. We encode the pose of the skeleton as a single 66-dimensional vector $\boldsymbol{\theta}$ containing the 3D global translation of the root, followed the stacked local joint rotations of each bone (including the root), represented as 3D angle-axis vectors (*i.e*. the axis of rotation multiplied by the angle of rotation in radians). This parameter vector is the variable which is optimized, with the root translation $\mathbf{t}_1$ and joint rotations $\mathbf{R}_b$ being extracted and used in calculations as applicable.

For each bone, b, the global rigid body transform $\mathbf{T}_b^g$ is computed by concatenating bone offset and joint rotation transforms along the kinematic chain as follows:

$$\mathbf{T}_b^g(\boldsymbol{\theta}) = \prod_{b' \in \mathcal{P}(b)} \left[\begin{array}{c|c} \mathbf{R}_{b'} & \mathbf{t}_{b'} \\ \hline 0 & 1 \end{array}\right] \tag{1}$$

where $\mathcal{P}(b)$ is the ordered set of parent joints of bone b.

We define a set of n_i IMU track targets, i, each attached to a bone b_i. The rotational and translational offsets of the IMU w.r.t. the bone are denoted $\mathbf{R}_{ib}$ and $\mathbf{t}_{ib}$, respectively. The rotational transform between each IMU reference frame and the global coordinates is denoted $\mathbf{R}_{ig}$. IMU orientation measurements (w.r.t. the IMU inertial reference frame) and acceleration measurements (w.r.t. the IMU device frame) are denoted $\mathbf{R}_i$ and $\mathbf{a}_i$, respectively. Likewise, we define a set of n_p positional track targets, p, each attached to a bone b_p with translational offset $\mathbf{t}_{pb}$ w.r.t. the bone. Note that here we use the term 'track target' to refer to a specific point on the body for which motion is estimated, not a physical optical marker. In our approach 2D joint positions are estimated using natural images and no visual markers are required.

Finally, we define a set of n_c cameras, c with calibrated 3×4 projection matrices $\mathbf{P}_c$ and let $\mathbf{t}_p^c$ denote the 2D position measurement for track target p in the local coordinates of camera c.

3.2. Pose optimization

The following pose optimization energy is used:

$$E(\boldsymbol{\theta}) = \overbrace{E_R(\boldsymbol{\theta}) + E_P(\boldsymbol{\theta}) + E_A(\boldsymbol{\theta})}^{Data} + \overbrace{E_{PP}(\boldsymbol{\theta}) + E_{PD}(\boldsymbol{\theta})}^{Prior} \tag{2}$$

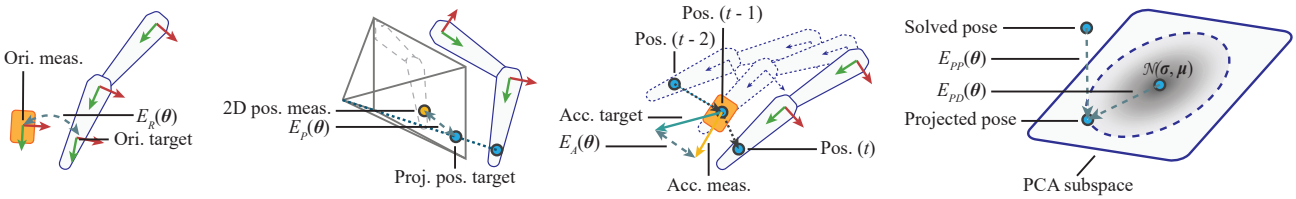
where $E_R(\boldsymbol{\theta})$, $E_P(\boldsymbol{\theta})$ and $E_A(\boldsymbol{\theta})$ contain orientation, position and acceleration constraints, respectively and $E_{PP}(\boldsymbol{\theta})$ and $E_{PD}(\boldsymbol{\theta})$ are pose projection and pose deviation priors, respectively. The data and prior constraints are visualized in Figure 1. Each term is described in the following subsections, where solved values have a '^' circumflex and their dependence on $\boldsymbol{\theta}$ is omitted for clarity. Unless otherwise specified, values are for the current frame, t.

3.2.1 Orientation term

For each IMU, i, an orientation constraint is added which seeks to minimize the relative orientation between the measured and solved global bone orientation (Figure 1).

The measured global bone orientation, $\mathbf{R}_{b_i}^g$ is obtained from the IMU measurement $\mathbf{R}_i$ using the IMU-bone offset $\mathbf{R}_{ib}$ and IMU reference frame-global offset as follows:

$$\mathbf{R}_{b_i}^g = \mathbf{R}_{ig} \cdot \mathbf{R}_i \cdot (\mathbf{R}_{ib})^{-1}. \tag{3}$$



(a) Orientation term (b) Position term (c) Acceleration term (d) PCA prior (figure adapted from [10])

Figure 1: Visualization of data and prior terms in the cost function (Equation 2).

The solved global bone orientation, $\hat{\mathbf{R}}_b^g$ is obtained using the kinematic chain, ignoring translations:

$$\hat{\mathbf{R}}_b^g = \prod_{b' \in \mathcal{P}(b_i)} \mathbf{R}_{b'}. \tag{4}$$

and the orientation cost is

$$E_R(\boldsymbol{\theta}) = \sum_{i \in [1,n_i]} \rho_R \left(\lambda_R \left\| \boldsymbol{\psi} \left((\hat{\mathbf{R}}_{b_i}^g)^{-1} \mathbf{R}_{b_i}^g \right) \right\|_2^2 \right) \tag{5}$$

where $\boldsymbol{\psi}(\cdot)$ extracts the vector part of the quaternion representation of the rotation matrix, λ_R is orientation constraint weighting factor, $\rho_R(\cdot)$ is a loss function. Discussion of the weightings and loss functions are deferred to Section 3.2.5.

3.2.2 Position term

For each positional measurement from each camera, a constraint is added which seeks to minimize the Euclidean distance between the measured 2D location in camera coordinates and the solved global track target location projected into the camera (Figure 1).

The solved global track target location, $\hat{\mathbf{t}}_p^g$ is determined by applying the translational offset to the global bone transform $\mathbf{T}_{b_p}^g$ calculated according to Equation 1:

$$\hat{\mathbf{t}}_p^g = \boldsymbol{\tau}_t \left(\boldsymbol{\tau}_T(\mathbf{t}_{pb}) \cdot \mathbf{T}_{b_p}^g \right) \tag{6}$$

where the operators $\boldsymbol{\tau}_T(\cdot)$ and $\boldsymbol{\tau}_t(\cdot)$ are shorthand for creating a transform matrix from a translation vector and extracting the translation vector from a transform matrix, respectively. This global target position is projected into each camera to obtain 2D solved targets $\hat{\mathbf{t}}_p^c$ in camera coordinates:

$$\hat{\mathbf{t}}_p^c = \mathrm{dh}(\mathbf{P}_c \hat{\mathbf{t}}_p^g) \tag{7}$$

where the operator $\mathrm{dh}(\cdot)$ performs de-homogenization of a homogeneous vector.

The position cost is defined as

$$E_P(\boldsymbol{\theta}) = \sum_{c \in [1,n_c]} \sum_{p \in [1,n_p]} \rho_P \left(\lambda_P c_p^c \| \hat{\mathbf{t}}_p^c - \mathbf{t}_p^c \|_2^2 \right) \tag{8}$$

where $c_p^c \in [0,1]$ is a confidence weighting for constraint p obtained from the image-based position measurement mechanism (Section 3.3), λ_P is a position constraint weighting factor, $\rho_P(\cdot)$ is a loss function (see Section 3.2.5). The confidence weighting c_p and loss function enable robust output pose estimates in spite of persistently high levels of noise and frequent outliers in input position detections.

In these experiments, the track targets are located on a subset of the joints and thus have zero offset w.r.t. the bone ($\mathbf{t}_{pb} = \mathbf{0}$). In general positional targets could be offset from the joint locations (this would be the case if the positions were to come from optical markers attached to the surface of the body, rather than markerless joint detections).

3.2.3 Acceleration term

In addition to orientation, the IMUs provide acceleration measurements (in the local IMU coordinates). In order to include an acceleration term, it is necessary to consider a window of three frames, t (current frame), and previous two frames $t-1$ and $t-2$. For each IMU, a constraint is added which seeks to minimize the difference between the measured and solved acceleration of the track target site (Figure 1). The solved acceleration $\hat{\mathbf{a}}_i^g$ is computed using central finite differences using the solved pose from previous two frames along with the current frame being solved:

$$\hat{\mathbf{a}}_i^g(t-1) = \left(\hat{\mathbf{t}}_i^g(t) - 2\hat{\mathbf{t}}_i^g(t-1) + \hat{\mathbf{t}}_i^g(t-2) \right) / (\Delta t)^2. \tag{9}$$

where the solved IMU positions $\hat{\mathbf{t}}_i^g$ are computed analogously with Equation 6 (replacing subscripts p with i) and Δt is the frame period (in our case, 16.7 ms).

The measured local accelerations from the previous frame of IMU data[1] are converted to global coordinates as follows:

$$\mathbf{a}_i^g(t-1) = \mathbf{R}_{ig} \cdot \mathbf{R}_i(t-1) \cdot \mathbf{a}_i(t-1) - \mathbf{a}_g \tag{10}$$

where $\mathbf{a}_g = [0, 9.8707, 0]^T$ is the acceleration of gravity, which needs to be subtracted. The acceleration cost is then

[1] The previous frame is used because central differences are used to estimate the solved acceleration.

simply defined as

$$E_A(\boldsymbol{\theta}) = \sum_{i \in [1,n_i]} \rho_A \left(\lambda_A \left\| \hat{\mathbf{a}}_i^g - \mathbf{a}_i^g \right\|_2^2 \right) \quad (11)$$

where once again λ_A is a constraint weighting factor, $\rho_A(\cdot)$ is a loss function (see Section 3.2.5).

Note that the orientation constraints only require the orientation offset of the IMU w.r.t. the bone to be known, whereas the acceleration constraints require the translational offset to be known as well.

It is well known that double integrating acceleration to obtain position is prone to drift, thus these acceleration terms alone would not be sufficient to locate the body in global coordinates over any length of time. The evaluation section considers the merits of including the acceleration term in the optimization.

3.2.4 Pose prior terms

In practice, not all the body segments are observed in the input - the kinematic skeleton has more degrees of freedom than are constrained by the IMUs and positional measurements. For instance, the spine has several segments, but only one or two IMUs attached to it. A pose prior is therefore required to constrain all degrees of freedom and produce plausible poses in spite of sparse or noisy sensor input.

In these experiments, two prior terms were incorporated based on a principal component analysis (PCA) of a corpus of motion capture data. The pose prior should be invariant to the global position and heading of the subject. We therefore use $\bar{\boldsymbol{\theta}}$, denoting the $d_p = d - 6$ pose vector excluding the first six elements, in the pose prior formulation.

A subset of ground-truth motion sequences from the *Total Capture* dataset [16], covering a wide variety of poses were used as training of the PCA pose model. In order to obtain a representative sample of poses without overemphasis on commonly recurring poses for standing and walking, for instance, we perform k-means clustering on the full set of $n_f = 126,000$ training frames, with $k = n_f/100 = 1,260$. The cluster centres are concatenated to form a $k \times d_p$ data matrix $\mathbf{D}$ and PCA is performed on the mean-centered data. The dimensionality is reduced to $d_r = 23$ (chosen so as to keep 95% of the variance in the data) and the resulting PCA model is a $d_p \times d_r$ coefficient matrix, $\mathbf{M}$, a d_p-dimensional mean vector, $\boldsymbol{\mu}$ and a d_r-dimensional vector of standard deviations, $\boldsymbol{\sigma}$ (the square-roots of the principal component eigenvalues).

We use two priors based on the PCA of the pose: PCA projection and PCA deviation. The projection prior encourages the solved body pose to lie close to the reduced dimensionality subspace of prior poses (soft reduction in the degrees of freedom of the joints), while the deviation prior discourages deviation from the prior observed pose variation (soft joint rotation limits) [10]. The pose projection cost is

$$E_{PP}(\boldsymbol{\theta}) = \rho_{PP} \left(\lambda_{PP} \left\| (\bar{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \mathbf{M}\mathbf{M}^T(\bar{\boldsymbol{\theta}} - \boldsymbol{\mu}) \right\|_2^2 \right) \quad (12)$$

and the pose deviation cost is

$$E_{PD}(\boldsymbol{\theta}) = \rho_{PD} \left(\lambda_{PD} \left\| \operatorname{diag}(\boldsymbol{\sigma})^{-1} \mathbf{M}^T(\bar{\boldsymbol{\theta}} - \boldsymbol{\mu}) \right\|_2^2 \right) \quad (13)$$

where, as with the data terms, weighting factors λ and loss functions ρ are used (see Section 3.2.5). A geometric interpretation of these constraints is shown in Figure 1. Together these terms produce soft constraints that yield plausible motion while not strictly enforcing a reduced dimensionality on the solved pose, thus allowing novel motion to be more faithfully reproduced at run time.

3.2.5 Energy minimization

As described in the previous subsections, weightings λ are used to control the contributions of each term to the overall cost in Equation 2. These are required because the different terms compare different physical quantities, and because some sources of data may be more reliable than others - for instance IMU orientations may be more stable than noisy position triangulations from images (refer to Section 4 for the values used).

Furthermore, each term has a loss function, $\rho(\cdot)$ for each residual. The purpose of the loss function is to make the cost robust against outlier data (as well as to allow deviation from the prior, when the measurements support it). For the orientation constraints, a null loss is used (standard L2 distance), since the IMUs tend not to produce outlier measurements. For the position, acceleration, PCA projection prior and PCA deviation prior a robust Cauchy loss function is used, $\rho(x) = \log(1 + x)$. The Cauchy loss function limits the effect of gross outliers by penalizing large residual values proportionally less than small values. Using the robust loss functions was found to be necessary to get good pose estimations in the presence of outlier measurements as well as novel unseen poses.

The pose cost function $E(\boldsymbol{\theta})$ is optimized using the Ceres non-linear least-squares solver [5]. The individual residuals for the data and prior terms are written using templated types in order to use the autodifferentiation functionality of Ceres.

The position, orientation and acceleration constraints are only affected by parameters associated with the bone to which they are attached and its parent bones in the kinematic chain. Therefore, the Jacobian is sparse and its computation can be sped up by using parameter blocks.

The computation is further sped up using multi-threaded Jacobian computation. The solving is performed using Levenberg-Marquardt with a sparse normal Cholesky linear solver. For each frame, the pose vector is initialized with the solved value from the previous frame, yielding full-body 6-DoF pose estimation at real-time video rates.

3.3. Joint detection from multi-view video

The convolutional pose machines (CPMs) detector of Cao *et al*. [7] is used to perform joint detections $\mathbf{t}_p^c$ from each viewpoint in a multi-view video setup. The detector also outputs confidences, c_p^c. These detections are used for the positional constraints in the cost function (Section 3.2.2). Although no explicit triangulation is performed in our formulation, at least two views are required in order for the solver to localize the subject in global coordinates without depth ambiguity.

On our hardware, the CPM detector requires 125 ms per frame on a single video stream, while fully utilizing the GPU. Video-rate detection of all frames in multiple views would thus not be practical. We employ two techniques to improve throughput and achieve video rate solving: ROI (region of interest) packing and temporal sub-sampling of position measurements.

3.3.1 ROI packing

The CPM detector is able to detect multiple people within a single image, while maintaining computation time [7]. We propose to increase the detection throughput by packing ROIs from multiple cameras (and optionally, frames) into a single image. The detection is performed on the packed image and the resulting detections are assigned the originating camera and frame (Figure 2). The ROIs for each camera are updated at every frame to an expanded bounding box of the current detections. In the event of missed detections, the corresponding ROI is reverted to the full image. In practice, the subject cannot be too small in the frame or the detector will fail. Packing 8 ROIs was found to be satisfactory (*e.g.* 1 frame from 8 cameras or 2 frames from 4 cameras).

3.3.2 Temporal sub-sampling

To increase the frame-rate of our solver in spite of relatively long CPM detection times, we propose to perform the CPM detections on a subset of the input frames, resulting in temporally sparse position measurements. For the intervals of frames without positional constraints, global motion is still produced because of the acceleration term, which essentially performs 'dead-reckoning'. In Section 4.1.2 different sub-sampling strategies are evaluated.

3.4. Implementation details

For the IMU data, we use Xsens MTw wireless IMUs [13]. These contain gyroscopes, accelerometers and magnetometers and through internal sensor fusion they output an orientation at 60 Hz. The inertial reference frame of each IMU, $\mathbf{R}_{ig}$ is assumed to be consistent between IMUs and in alignment with the world coordinates through the global up direction and magnetic north. The IMU-bone positions $\mathbf{t}_{ib}$ are specified by manual visual alignment and the IMU-bone orientations $\mathbf{R}_{ib}$ are calibrated using the measured orientations with the subject in a known pose (the T-pose, facing the direction of a given axis).

We use a set of 4K video cameras, with intrinsics and extrinsics calibrated using a chart [21], for simplicity of integration with the inertial measurements, the global reference frame of the camera system is chosen to align with the up direction and magnetic north. The *Total Capture* dataset was recorded at HD, 1920×1080, 60 fps, and *Outdoor 1* was recorded at UHD, 3840×2160, 60 fps. For efficiency of processing and display, we use downsampled video (960×540) throughout, since realtime decoding and display of multiple streams of high-resolution video proved a bottleneck.

To temporally align the IMU and video data an initial footstamp was performed by the actor, which is visible in the video and produces a strong peak in acceleration in the IMU data.

4. Results and evaluation

The approach was tested using an existing indoor dataset, *Total Capture* [16] containing ground-truth data from a commercial mo-cap system, as well as on a new outdoor dataset *Outdoor 1*. The solver can easily be configured to take an arbitrary subset of the available IMUs and positional constraints to evaluate the effect of camera and IMU sparsity. Note that in this work, all positional constraint information is obtained from the multiple-view video based on per-view CPM as discussed in Section 3.3 and no optical markers or visible targets are used.

First, quantitative results are presented showing the relative performance with various configurations of IMUs and cameras, sub-sampling configurations of position detections as well as the contribution of each term in the cost function. Next, further quantitative results are presented for multiple sequences of the *Total Capture* dataset. Finally, qualitative results are presented for the *Outdoor 1* dataset, which does not contain ground truth data. Videos of the results are presented in the supplementary material.

Throughout the experiments, the same weightings were used for the cost function terms, namely $\lambda_R = 1$, $\lambda_P = 1 \times 10^{-3}$, $\lambda_A = 7 \times 10^{-4}$, $\lambda_{PP} = 0.9$, $\lambda_{PD} = 0.08$. These values were arrived at by a gradient-based parameter

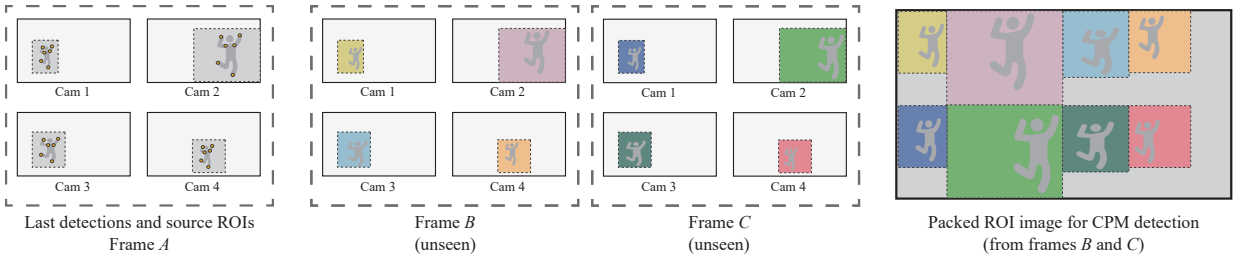


Figure 2: Visualization of the ROI packing process for efficient multi-camera/frame CPM detection.

optimization over 200 frames of one motion sequence.

4.1. Indoor capture results

The *Total Capture* dataset includes five subjects (*S*) performing various motions including range of motion (*ROM*), walking (*W*), acting (*A*), and 'freestyle' (*FS*). The subjects were recorded simultaneously using 13 Xsens MTw IMUs, 8 HD video cameras, and a commercial infra-red motion capture system consisting of 16 cameras and a dense set of retro-reflecting markers worn by the subject. The marker-based input is not used in the runtime solver and is only used in this work as a 'ground truth' reference for evaluation.

4.1.1 Sparse IMUs and cameras

It is desirable to have a minimal capture hardware setup in order to reduce cost as well as actor setup time. We simulate the effect of reduced capture hardware by excluding selected cameras and IMUs from the input. The 13 IMUs in the full set are placed on the pelvis, sternum, upper and lower limbs, head and feet. The 6 IMUs in the reduced set are positioned on the pelvis, lower limbs and head. The full set of cameras form a ring around the subject and between 2 and 8 of the available cameras are used in these tests.

Figure 3 compares the error using the sparse set of 6 IMUs with the full set of 13 using between 2 and 8 cameras. With the sparse set of IMUs, position and orientation error both decrease as more cameras are added. With the full set of IMUs, the position error is lowest for intermediate numbers of cameras, while the orientation error hardly varies with the number of cameras. An intermediate number of cameras, 4, is used to evaluate on additional sequences in Section 4.1.4.

4.1.2 Temporal sub-sampling of position

We use the following notation for the temporal sub-sampling (SS) of the position detection: N_o/N_p, where position detection is performed on the first N_o frames of every N_p frames in the sequence. For example SS 1/2 - every other frame, SS 2/4 - two out of every four frames. While SS 1/10 and SS 2/20 require the same amount of computation, SS 1/10 provides a shorter interval of detections and shorter time with no detections than SS 2/20. This has an effect on the quality of the solved motion as shown in Figure 4, where a range of sub-sampling rates were used, with $N_o \in \{1, 2, 3\}$.

These results suggest that it is optimal to use $N_o = 2$. Having detections for two successive frames results in a more reliable motion trajectory than having a single frame more frequently. Having three successive frames, $N_o = 3$ means that the interval of no detections is too long and the error increases. With a quarter of the frames detected (SS 2/8) the error is still reasonably low, while the processing time is reduced, increasing the output frame-rate. This decimation rate is used to evaluate on additional sequences in Section 4.1.4.

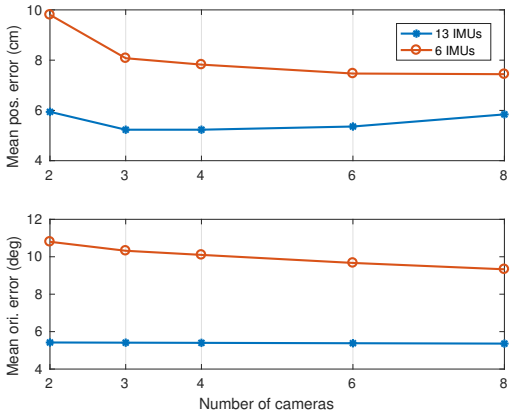


Figure 3: Position and orientation error with different sensor configurations, 13 or 6 IMUs and 2-8 cameras. Sequence: S2 - FS1, SS 1/1.

4.1.3 Contribution of cost terms

Table 1 shows the relative error in solved bone position and orientation with selected terms in the cost function disabled. The results are shown with 4 cameras, with 2/8 detection sub-sampling.

The orientation term from the IMUs has a strong effect on both position and orientation error, while the acceleration term has a limited effect, helping with the position in the 13 IMU case. The position term does not improve

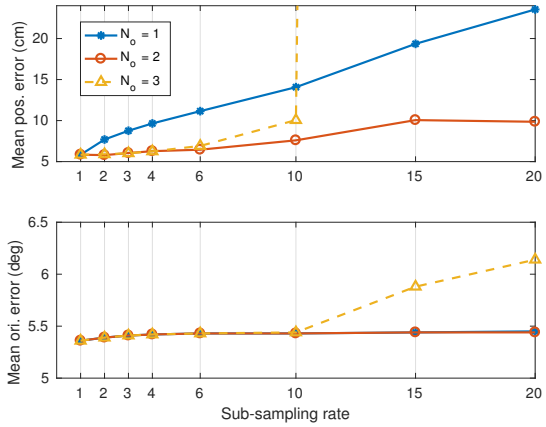


Figure 4: Position error under a range of detection sub-sampling rates using $N_o \in \{1, 2, 3\}$ successive frames. Note that $N_o = 2$ yields the lowest position error across the sub-sampling range. Sequence: S2 - FS1, 13 IMUs, 8 cameras.

	13 IMUs		6 IMUs	
Terms Omitted	Pos.	Ori.	Pos.	Ori.
IMU (E_R, E_A)	1.97	4.82	1.27	2.38
Ori. (E_R)	2.63	6.27	1.54	2.89
Acc. (E_A)	1.11	0.99	1.01	0.97
Pos. (E_P)	188.58	1.00	194.82	1.05
Prior (E_{PP}, E_{PD})	1.50	4.68	1.42	4.33
Prior Proj. (E_{PP})	2.26	6.29	1.63	6.46
Prior Dev. (E_{PD})	1.16	2.86	1.46	3.24

Table 1: Position and orientation error with various terms in the cost function disabled, relative to the error using the full cost function, Equation 2 (Sequence: S2 - FS1, 4 Cam., SS 2/8).

the orientation error, but without it the global position drifts causing gross error in position. Both the prior projection and prior deviation terms improve the position and orientation errors significantly.

4.1.4 Further results

In Table 2, further quantitative results are provided for several sequences from the *Total Capture* dataset, covering the 5 subjects and a range of motion types from slow ROM motion to challenging sequences including fast motion and unusual poses such as lying on the floor (see Figure 6 and refer to the supplementary video). Figure 5 shows the robustness of our approach to typical misdetections from the CPM joint detector.

Four configurations were used: high quality, 'HQ' (8 camera, SS 1/1) and high speed, 'HS' (4 camera, SS 2/8) each with both the 13 and 6 IMU sets. The average position error using 13 IMUs is 6.2 cm using HQ mode, degrading slightly to 6.8 cm using HS mode, while the orientation error is maintained between HQ and HS (7.8 deg). Our approach outperforms Trumble *et al*. [16] across the test sequences both for HQ and HS modes. The errors for the 6 IMU case are larger at 9.1 cm, 12.5 deg for HQ and 14.2 cm, 15 deg for HS.

	S1 FS3	S2 FS1	S2 RM3	S3 FS1	S3 FS3	S4 FS3	S5 A3	S5 FS1	Mean
Pos. error (cm)									
Ours, 13 IMU, HQ	**7.4**	**5.3**	**3.9**	**6.7**	**6.7**	**6.4**	**6.4**	**7.0**	**6.2**
Trumble [16]	9.4	16.7	9.3	13.6	8.6	11.6	14.0	10.5	11.7
Ours, 13 IMU, HS	8.5	5.4	3.8	7.4	7.3	7.7	6.6	7.5	6.8
Ours, 6 IMU, HQ	9.8	7.1	6.6	10.0	10.7	9.2	9.0	10.0	9.1
Ours, 6 IMU, HS	14.3	9.4	10.8	19.4	17.1	13.9	13.3	16.5	14.3
Ori. error (deg)									
Ours, 13 IMU, HQ	11.2	5.1	5.0	8.3	9.3	8.0	7.6	8.2	7.8
Ours, 13 IMU, HS	11.2	5.1	5.0	8.3	9.3	8.0	7.6	8.2	7.8
Ours, 6 IMU, HQ	16.3	9.2	8.7	13.2	15.7	13.0	11.8	12.1	12.5
Ours, 6 IMU, HS	18.3	10.9	10.6	16.2	19.7	14.8	14.3	15.1	15.0

Table 2: Mean error in position (cm) and orientation (deg) for sequences from the *Total Capture* dataset using high quality (HQ) and high speed (HS) settings, compared to the approach of Trumble *et al*. [16].

4.2. Outdoor capture results

The *Outdoor 1* dataset was recorded outdoors in challenging uncontrolled conditions with a moving background and varying illumination. A set of 6 cameras were placed in a 120° arc around the subject and 13 Xsens IMUs. No ground truth data is available for this dataset. Figure 7 shows a selection of solved frames overlaid on the input image and full sequences are shown in the supplementary video.

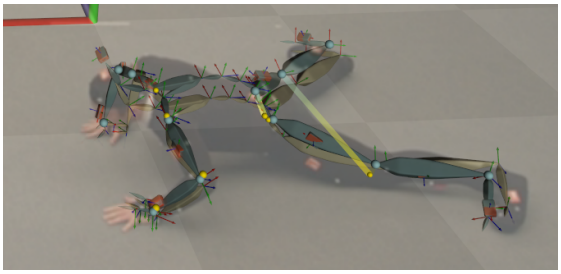
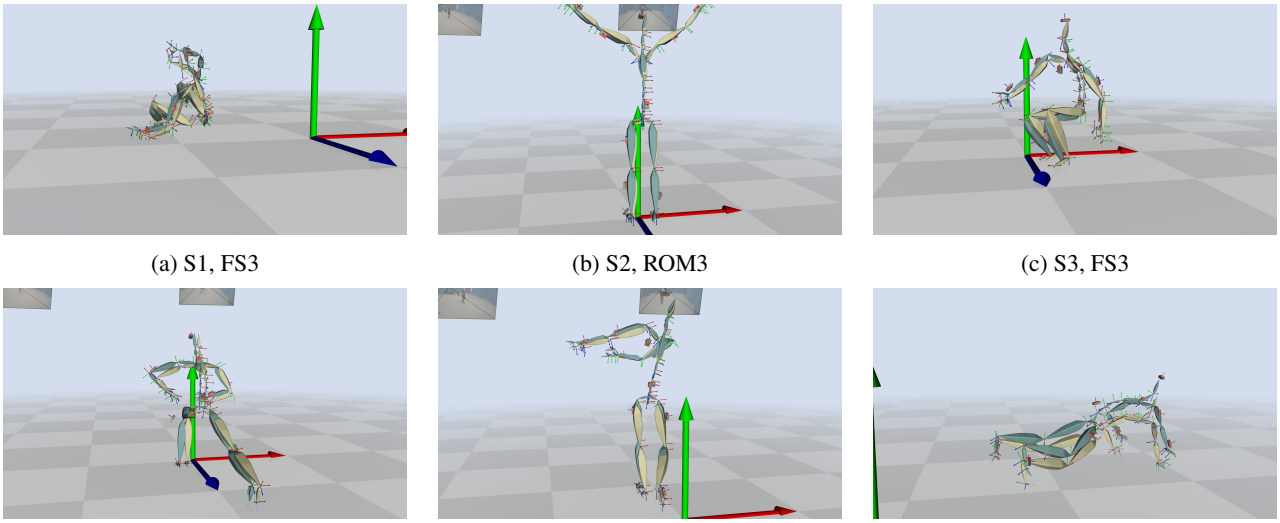


Figure 5: Solved (blue) and ground truth (yellow) skeletons overlaid on an input image showing CPM detections in yellow and the corresponding locations on the solve skeleton in blue. Note the robustness to the outlier detection on the leg. Sequence: S5, FS1.

4.3. Computation time

Figure 8 shows the real-time online frame-rate achieved using the approach as a function of the sub-sampling rate (with the CPM detection running in parallel with the main solver thread). The computing hardware is a standard desktop PC with Intel i7 3.6 GHz CPU and NVIDIA GTX 1080 GPU. A frame rate of 30 fps can be achieved with SS 2/8, while a rate in excess of 60 fps can be can be achieved with



(a) S1, FS3 (b) S2, ROM3 (c) S3, FS3

(d) S3, FS1 (e) S4, FS3 (f) S5, FS1

Figure 6: Selection of frames from the *Total Capture* dataset. The reference (ground truth) skeleton is shown in yellow, while our solved skeleton is shown in blue.

(a) Camera layout (b) Camera views (freestyle) (c) Multiple frames (prop interaction)

Figure 7: Selection of solved frames from the *Outdoor 1* dataset.

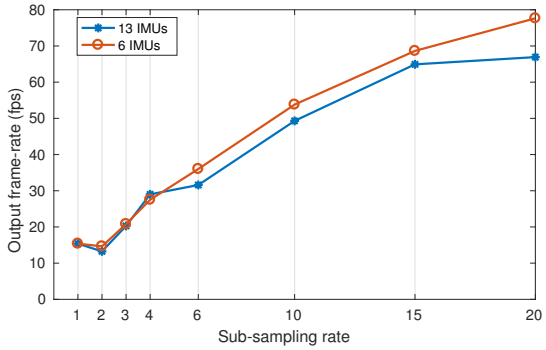


Figure 8: Output frame-rate of our solver (including detections) as a function of sub-sampling level. Sequence: S2-FS1, 4 cameras, $N_o = 2$.

more aggressive sub-sampling (SS 2/40). In practice, 30 fps is sufficient for most applications, and the increase in speed is not worth the increase in error. Although not tested here, it should also be possible to use a lower decimation rate by running two CPM detectors in parallel on two GPUs.

5. Conclusion and further work

The approach presented is flexible in terms of camera and IMU hardware requirements. It is capable of recovering the full 6-DoF pose, without drift in global position and can operate both in constrained studio environments and in unconstrained setups such as outdoor scenes with varying illumination moving backgrounds and occlusion. The solver can handle missing or outlier joint detections and even short periods of complete occlusion because of the inclusion of the IMU input, degrading gracefully as the hardware is reduced. Reducing the number of cameras has less of an effect on quality than reducing the number of IMUs. Future work includes optimizing the code and using multiple GPUs to increase CPM detection throughput. It would also be possible to extend the approach to handle multiple subjects.

Acknowledgements

This work was supported by the Innovate UK *Total Capture* project (grant 102685) and the EU H2020 *Visual Media* project (grant 687800). We wish to thank Anna Korzeniowska, Evren Imre, Joao Regateiro and Armin Mustafa for their help with data capture.

References

[1] IKinemaOrion. https://ikinema.com/orion. 2

[2] OptiTrack Motive. http://www.optitrack.com. 1

[3] Perception Neuron. http://www.neuronmocap.com. 1

[4] Vicon Blade. http://www.vicon.com. 1

[5] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 4

[6] S. Andrews, I. Huerta, T. Komura, L. Sigal, and K. Mitchell. Real-time Physics-based Motion Capture with Sparse Sensors. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016)*, 2016. 1

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 5

[8] T. Helten, M. Muller, H.-P. Seidel, and C. Theobalt. Real-time body tracking with one depth camera and inertial sensors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1105–1112, 2013. 2

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural computation*, volume 9, pages 1735–1780. MIT Press, 1997. 2

[10] A. E. Ichim and F. Tombari. Semantic parametric body shape estimation from noisy depth sequences. *Robotics and Autonomous Systems*, 75:539–549, 2016. 2, 3, 4

[11] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1

[12] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. VNect: Real-time 3D human pose estimation with a single RGB camera. In *ACM Transactions on Graphics*, volume 36, 2017. 1

[13] D. Roetenberg, H. Luinge, and P. Slycke. Xsens MVN : Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors. *Technical report*, pages 1–7, 2013. 1, 5

[14] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[15] M. Trumble, A. Gilbert, A. Hilton, and J. Collomosse. Deep convolutional networks for marker-less human pose estimation from multiple views. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016)*, 2016. 2

[16] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *British Machine Vision Conference (BMVC)*, 2017. 2, 4, 5, 7

[17] T. Von Marcard, G. Pons-Moll, and B. Rosenhahn. Human Pose Estimation from Video and IMUs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1533–1547, aug 2016. 1

[18] T. von Marcard, B. Rosenhahn, M. Black, and G. Pons-Moll. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. In *Eurographics 2017*, volume 36, 2017. 1

[19] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 1

[20] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics*, 31(6):1, 2012. 2

[21] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, volume 00, pages 0–7, 1999. 5

[22] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4966–4975, 2016. 2