

Outdoor Human Motion Capture by Simultaneous Optimization of Pose and Camera Parameters

A. Elhayek C. Stoll K. I. Kim and C. Theobalt

Max-Planck-Institute for Informatics, Saarbrücken, Germany

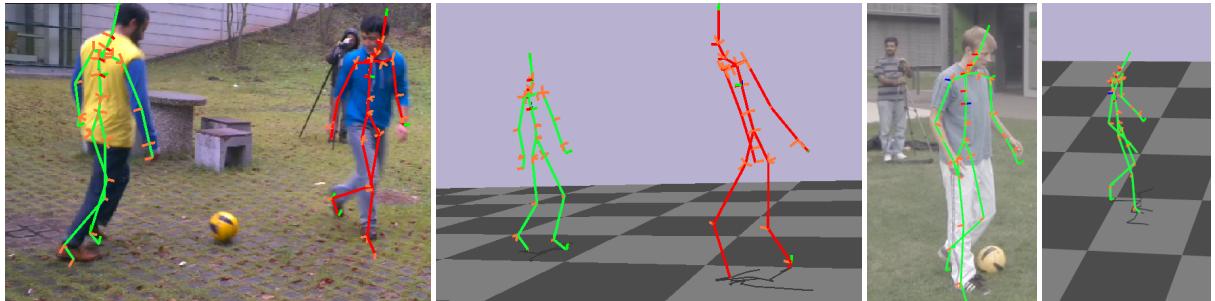


Figure 1: Examples of multi-person tracking with moving cameras. (Left two images) two actors, and two moving and 3 static cameras (Soccer1). (Right two images) One actor, and three moving and two static cameras (Walk2).

Abstract

We present a method for capturing the skeletal motions of humans using a sparse set of potentially moving cameras in an uncontrolled environment. Our approach is able to track multiple people even in front of cluttered and non-static backgrounds, and unsynchronized cameras with varying image quality and frame rate. We completely rely on optical information and do not make use of additional sensor information (e.g. depth images or inertial sensors). Our algorithm simultaneously reconstructs the skeletal pose parameters of multiple performers and the motion of each camera. This is facilitated by a new energy functional that captures the alignment of the model and the camera positions with the input videos in an analytic way. The approach can be adopted in many practical applications to replace the complex and expensive motion capture studios with few consumer-grade cameras even in uncontrolled outdoor scenes. We demonstrate this based on challenging multi-view video sequences that are captured with unsynchronized and moving (e.g. mobile-phone or GoPro) cameras.

Categories and Subject Descriptors (according to ACM CCS):

1. Introduction

In computer graphics, motion capture is a widely used way to animate virtual human characters. Unfortunately, traditional marker-based motion capture systems are expensive and cumbersome to use.

Recent years have seen a significant improvement of *marker-less skeletal human motion capture* algorithms [MHK06, Pop07, SBB10]. Many state-of-the-art methods come close to the performance of marker-based al-

gorithms, but only when recording in highly controlled *studio setups*, where 1) there are sufficiently many exactly synchronized high-quality cameras; 2) each camera is static and scene motion is due to foreground objects only; 3) the background is not cluttered; 4) lighting is controlled; 5) the main foreground actor is seldom occluded.

While relative to marker-based systems, this yields an easier apparatus with a reduced setup time, the hurdles towards practical application are still large and the costs are still no-

table. By being constrained to a controlled studio, marker-less methods fail to fully play out their advantage of being able to capture scenes without actively modifying them. Many practical computer graphics applications require motions to be captured on site, i.e. the camera system needs to be brought to the set location, because the motion itself cannot be relocated to a studio. Examples are capturing drivers in cars, motion capture on outdoor film sets, recordings of street performances, or the reconstruction of athletes in the field. In such situations, scenes are often cluttered and foreground and backgrounds may be dynamic. Further on, placement and number of cameras may be starkly constrained, cameras can often not be synchronized, and they may (have to) move during recording. Some methods succeed in uncontrolled recording scenarios and allow certain camera motion (also outdoors [HRT^{*}09]), but have limited accuracy and would fail in case of 1) cluttered scenes and with unconstrained sparse camera sets; 2) small camera translation or pure rotational motion; 3) motion blur due to hand-held camera shaking.

We therefore present a method for marker-less 3D skeletal human motion capture that succeeds in uncontrolled environments and uses only a sparse, heterogeneous and weakly constrained camera setup. The algorithm reliably captures even complex 3D skeletal body motion 1) with potentially as little as five cameras (e.g. mobile-phone cameras); 2) with camera setups that are unsynchronized and of differing makes, resolutions and frame rates; 3) in cluttered indoor and outdoor scenes where backgrounds are dynamic and the actor can be occluded; 4) without using specialized auxiliary sensors, such as 3D cameras; 5) with any type of camera motion even including notable shaking.

The core algorithmic contributions are a new generative skeletal pose tracker that minimizes a single model-to-image consistency measure simultaneously in the skeletal actor poses *and* the poses of moving cameras (Sec. 4). We demonstrate that this strategy is essential to deal with scenes where cameras, foreground, and background can move and image-based pre-calibration (e.g. via structure-from-motion (SfM), e.g. [PVGV^{*}04, THWpS08]), fails. 3D model and 2D image data compared during consistency measurement are based on a Sums-of-Gaussian model used previously for indoor tracking with static cameras [SHG^{*}11]. However, the energy function and the minimization strategy have been profoundly extended to match this more challenging scenario. The smooth nature of our energy functional with analytic derivatives enables continuous optimization. It also enables the automatic detection of the occlusion of body parts either caused by the same person (self-occlusion) or by the other people in the same scene (Sec. 5). This is properly taken into account in the optimization.

While this is not the first method for outdoor motion capture, to the best of our knowledge, our algorithm is the first that aims for motion reconstruction with moving cameras,

unsynchronized video streams, uncontrolled environment, uncontrolled cameras motion, and multiple characters, all at once. In summary, our algorithm augments earlier work on markerless motion capture [SHG^{*}11, ESH^{*}12] with static cameras that does not succeed under the aforementioned challenging conditions. The novel algorithmic contributions over previous work, that enable this, are:

1. A new pose fitting energy function extended to estimate each camera's motion together with actor pose; see (Sec. 4.2). In particular, the following extensions were done to the measurement of model-to-image consistency
 - a. Support for multi-person/multi-camera tracking
 - b. A two-sided similarity term [†]
 - c. Weighting in HSV color space
 - d. Prior on camera motion (smoothness)
2. The pose estimation scheme is using a new and improved occlusion handling.

In our experiments, we show qualitatively and quantitatively against ground truth that our algorithm can capture even complex and fast body motion in cluttered outdoor scenes, and that it succeeds with a wide range of heterogeneous, unsynchronized and moving camera systems with varying resolution, also just a few mobile phone or *GoPro* cameras. We also contribute with a comprehensive evaluation data set for quantitative comparison. It comprises multi-view video footage recorded with static and moving cameras, ground truth camera motion data, as well as reference data from a marker-based motion capture system.

2. Related Work

Algorithms for tracking human motion from videos have seen great progress in recent years [BM98, DR05, BSB^{*}07]. Detailed overviews of this field and discussions on established algorithms can be found in recent survey articles [MHK06, Pop07, SBB10].

In recent years, methods that try to infer poses [ARS10, ILS11] from single-view images, or motions from monocular video [WC10], have gained more attention. However, these methods do not yet reach the accuracy of multi-view methods, and only work on very simplified models with few degrees of freedom. Most multi-view approaches combine a body model of the actor to be tracked with data extracted from images, such as silhouettes, for pose estimation (e.g. [GRBS10, LSG^{*}11]). Other works make use of additional sensors, such as inertial sensors [PMBG^{*}11], or depth sensors [BMB^{*}11].

Only few works exist that deal with tracking human motion from moving cameras. Hasler *et al.* [HRT^{*}09] proposed

[†] The concept of symmetric similarity was presented in [ST02]. However, our novel continuous and differentiable two-sided term is essential in case of moving cameras and allows for fast tracking.

an algorithm for motion tracking with moving and unsynchronized cameras. In their approach, camera synchronization and calibration problems were decoupled from pose estimation by explicitly solving these problems before pose estimation: frame accurate synchronization is performed based on audio. The camera parameters for each set of (synchronized) video frames are estimated by performing SfM. Shiratori *et al.* [SPH11] mount outwards facing cameras to the limbs of an actor and estimating the skeletal pose based on SfM of the actor’s environment. These approaches have several limitations: SfM fails in case of cluttered scenes with dense moving background (e.g., crowds of people), motion blur due to hand-held camera shaking, and small camera translation or pure rotational motion. Furthermore, frame-level synchronization might be insufficient for heterogeneous cameras as demonstrated in [ESH*12] (i.e., sub-frame accurate synchronization leads to a significant improvement), and body-mounted cameras mean unwanted active modification of the scene. Recently, Elhayek *et al.* [ESH*12] proposed an algorithm that broke the limitation of frame-level synchronization in [HRT*09]. By representing the pose parameters as an analytic function of time, they enabled tracking with heterogeneous and unsynchronized cameras in sub-frame accuracy. However, they showed results with static cameras only.

Ye *et al.* [YLH*12] presented an algorithm that tracks human motion with multiple consumer depth sensors (i.e. Kinects). They simultaneously optimize skeletal pose and sensor position based on image correspondences from feature tracking and geometric correspondences between the point clouds and the performer’s surface. However, due to the use of depth sensors, the method cannot be applied in outdoor scenarios, and fails if no stable image features can be found in the background. To enable rendered fly-arounds in virtual replays, Germann *et al.* [GHK*10] tracked articulated billboard models of soccer players from TV cameras in a soccer stadium. However, their algorithm is not fully automatic and tailored to soccer pitches where foreground separation is easier.

3. Overview

Input to our algorithm is a set of video streams of the same scene, yielding a set of frames $\mathcal{I} = \{I_1, \dots, I_n\}$ obtained from several cameras (camera indices omitted for readability). These cameras can be of varying make, resolution and frame rate, and they can move during recording. Video streams are not expected to be temporally synchronized (see Sec. 6) and the global time stamps are explicitly estimated, as discussed shortly. We assume that intrinsic cameras parameters are known (e.g. through calibration before recording).

As opposed to studio-based methods, we assume that lighting can vary mildly during recording, large part of the background can be dynamic, and the tracked person can be occluded for the duration of a few frames. The output of our

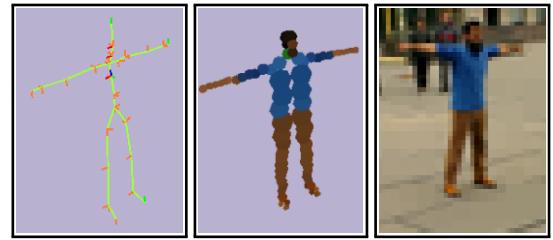


Figure 2: Left: skeletal representation of a performer. Center: a 3D SoG representation approximating the body shape of the performer. Right: image SoG representation (each box represents one 2D Gaussian G^2).

algorithm is a continuous *motion function* $X(t) : \mathbb{R} \rightarrow \mathbb{R}^n$ that returns an n -dimensional pose vector for a given time stamp t . Here, n is given as $n = 6c + m$, where $m = 43$ is the number of degrees of freedom of the skeletal model (pose and joint angles, thus describing the *pose*; see [SHG*11] for more details) and c is the number of moving cameras in the scene. The 6 parameters for each moving camera describe translation and rotation. Due to the ambiguity between camera and performer motion in a single camera view, we can represent camera motion as an additional rigid transformation to the pose of the actor in a specific single view. This simplifies the optimization, as camera parameter optimization can be handled in the same way as actor motion. Note that in our setting we represent joint parameters as continuous temporal curves, thus they can be calculated for every sub-frame time instant of the motion (see Sec. 4).

For each tracked actor, the template body model must be shape-adjusted, which we do in a semi-automatic way from a set of calibration poses prior to motion recording using the algorithm of Elhayek *et al.* [ESH*12]. It could also be done manually in case one has no control over the footage and actor motion.

For tracking multiple people in a scene, we initialize the pose of each actor independently. Then, our algorithm estimates a single combined motion function X that concatenates the motion functions of individual actors. This is different from running a single-person tracker for each actor where the occlusions caused by different actors would not be taken into account. By estimating a single large motion function, we handle multiple people tracking exactly in the same way as the self-occlusion (see Sec. 5). Accordingly, the remainder of this and next sections focus only on the single actor case without loss of generality for multiple actors.

Before tracking commences, we first synchronize video streams up to frame-level accuracy by using the audio stream [HRT*09]. We refine this initial result by the global multi-view image-based space-time alignment method of Elhayek *et al.* [ESH*12] which yields frame rates and offsets at sub-frame precision.

In the beginning, we also expect a small amount of user interaction to obtain an extrinsic camera calibration C_{ext} for

one multi-view frame of each camera at the nearest time stamps (after temporal alignment). We employ a bundle-adjustment with manually marked features in the scene background [HZ04]. Note that this is only needed for one set of frames.

The core of our algorithm is a new energy minimization approach where a model-to-image consistency energy functional is jointly optimized with respect to camera pose and skeletal pose parameters (Sec. 5). The energy functional is based on the Sums-of-Gaussians scene representation of [ESH^{*}12] which we profoundly extended to deal with our more general scene conditions, such as moderate appearance variations, occlusion, and dynamic background, as well as the sparse visual evidence from only few cameras (Sec. 4.2). In this paper, we briefly restate important concepts from prior work that we build on, but focus on the newly developed extensions. Employing a space-time optimization strategy is essential (Sec. 4.1) to deal with the lack of exact frame synchronization, and to be able to benefit from larger temporal baselines to regularize tracking with few cameras. With few cameras, occlusions of the actor, even for a short period, can lead to catastrophic failure of joint angle and camera optimization (see Sec. 6 for examples). We explicitly detect occlusions by monitoring the energy variation in time. Once an occlusion is detected, the corresponding camera is disabled for optimization and does not contribute to the energy anymore. In case it is a moving camera, its pose parameters are re-initialized based on corresponding linearly interpolated parameter values (Sec. 5).

4. Tracking with moving and unsynchronized cameras

One of the most important aspects of motion tracking with casually captured videos is that the cameras may move, and accordingly, the camera parameters have to be estimated for each frame in the video. In existing approaches for this scenario, estimation of these two sets of parameters is decoupled by pre-estimating camera parameters (e.g. performing SfM) and subsequently optimizing the pose (of the actors) given these known cameras. Unfortunately, this strategy cannot be exercised when the background is cluttered, the camera translation motion is not sufficient or the videos are blurry due to shaking cameras unless these conditions, SfM fail (see Fig. 7 and supplementary video). Our video streams (e.g. with as little as five cameras) are sparse and frame capture is not synchronized. Furthermore, there are many inherent ambiguities in model-image-matching which aggravate finding an optimal solution: the free motion of the body cannot be decoupled from the ego-motion of the cameras. For instance, it is often impossible to distinguish between an actor moving towards a static camera and a moving camera approaching a static actor.

A core innovation of our tracking algorithm is the simultaneous optimization of skeletal pose parameters and the pose parameters of every moving camera. Both camera and skele-

tal pose parameters are separately retained, but the effect of changing one set of parameters could still be compensated by the change of the other. Capturing the scene with one or more static cameras resolves this ambiguity. However, when there are no static cameras, the final results can only represent relative motion to the cameras and will not have a fixed global coordinate space. Our system uses the body model as common reference point to optimize skeletal pose and the pose of moving cameras. We are thus not forced to rely on unstable background features.

Our approach is instantiated as an energy minimization algorithm:

$$E(X) = -L(\mathcal{S}|X) + \lambda_1 E_{Lim}(X) + \lambda_2 E_{Smooth}(X), \quad (1)$$

where $L(\mathcal{S}|X)$ is a *likelihood* term that measures the similarity of model parameter X to data \mathcal{S} and E_{Lim} and E_{Smooth} are *prior terms* that enforce limits and smoothness on joint angles and camera poses, respectively. The hyper-parameters λ_1 and λ_2 are set to 0.1 and 0.01, respectively (see Sec. 6 for discussion on tuning hyper-parameters).

The individual components of the energy are detailed in the following. We will also detail the continuous pose parametrization and specific representation of image and shape data we employ. Optimizing continuous curves for skeletal and camera poses is essential since our data are not frame-synchronized and are spatially sparse. We can stabilize optimization by considering image data from larger temporal baselines.

4.1. Continuous parameterization and scene representation

We extend concepts from [ESH^{*}12] and instead of identifying parameter vectors for each discrete time stamp (corresponding to synchronized frame indices), we construct a continuous, parameter vector-valued *motion function* $X(t)$ of time t representing both skeletal and camera pose parameters. In this setting, the likelihood of a specific motion function X with respect to a set of images is evaluated by sampling X at the time stamp t_i of each image $I_i \in \mathcal{I}$.

For representing the 3D spatial extent of the body model, as well as the 2D input images, we employ the Sums-of-Gaussians (SoG) model of [SHG^{*}11] where human body is approximated with a kinematic skeleton, to the bones of which 72 3D isotropic Gaussian functions G^3 are attached. The parameters of a Gaussian function, including the location (mean) μ , the variance σ_k^2 , and the corresponding representative color value \mathbf{c} , are estimated based on a set of example images; see [SHG^{*}11] for more details. This yields a smooth (i.e., continuously differentiable as many times as desired) representation of human body model as exemplified in Fig. 2. An analogous model is used to describe images. This 2D Gaussian representation G^2 is used to represent each uniformly colored region in every image after quad-tree clustering of similarly colored regions (Fig. 2 right).

Since estimating a continuous motion function X for a long sequence is computationally intractable, we divide the sequence into overlapping time segments $\{\mathcal{S}_1, \dots, \mathcal{S}_{n_s}\}$ (of length 2/30 sec. for each, with an overlap interval of 0.6/30 sec.) and estimate the local motion functions X_i as quadratic polynomials for each \mathcal{S}_i independently. This results in $3 \times n$ parameters for each X_i (3 being the number of parameters for a quadratic polynomial; there is no coupling between different parameters). A globally smooth motion function X is implicitly reconstructed by blending the X_i at overlap using partition-of-unity [ESH*12]. Accordingly, the variables to be optimized are the coefficients of the polynomial for each segment.

4.2. Likelihood: model-to-image similarity

We now explain the likelihood of a motion function X_i corresponding to a given segment \mathcal{S}_i of an input sequence which is an extension of [ESH*12]. At a given time instance, the motion function and a video frame are represented as a 3D and a 2D SoG, respectively. Then, the likelihood is calculated by projecting the 3D Gaussian functions in the 3D SoG into the corresponding image plane and measuring the overlap to all 2D Gaussians. The positions of each 3D Gaussian is a function of both the skeleton and camera parameters. However, it should be noted that the skeletal pose parameters are optimized based on the data of every cameras, while the parameters of each moving camera are optimized based on the data of that camera only (i.e. by maximizing the similarity between this camera's 2D SoG and the projected 3D SoG). For simplicity of notation, we will omit the segment index i when there is no risk of confusion.

We represent an image I_j with time stamp t_j as the 2D SoG $\mathcal{K}_I(t_j)$ and the respective model $X(t_j)$ as 3D SoG $\mathcal{K}_M(t_j)$. Then we can define the likelihood of the motion function X as the sum of similarities of $\mathcal{K}_M(t_j)$ and $\mathcal{K}_I(t_j)$ for $t_j \in \mathcal{S}_i$:

$$L(X|\mathcal{S}) = \frac{1}{n(i)} \sum_{t_j \in \mathcal{S}} \frac{S_3(\mathcal{K}_M(t_j), \mathcal{K}_I(t_j))}{S_2(\mathcal{K}_I(t_j), \mathcal{K}_I(t_j))}, \quad (2)$$

where $n(i)$ is the total number of images in \mathcal{S}_i , $S_3(A, B)$ is the similarity of a 3D SoG, A and a 2D SoG, B as will be defined shortly. Since every $\mathcal{K}_I(t_j)$ for $t_j \in \mathcal{S}_i$ consists of a different number of 2D Gaussians, we normalize $S_3(\mathcal{K}_M(t_j), \mathcal{K}_I(t_j))$ by the similarity of $\mathcal{K}_I(t_j)$ with itself. The general similarity of two 2D SoGs \mathcal{K}_a and \mathcal{K}_b is defined as

$$\begin{aligned} S_2(\mathcal{K}_a, \mathcal{K}_b) &= \int_{\mathbb{R}^2} \sum_{i \in \mathcal{K}_a} \sum_{l \in \mathcal{K}_b} d(\mathbf{c}_i, \mathbf{c}_l) G_i^2(\mathbf{x}) G_l^2(\mathbf{x}) d\mathbf{x} \\ &= \sum_{i \in \mathcal{K}_a} \sum_{l \in \mathcal{K}_b} E_{il}, \end{aligned} \quad (3)$$

where G_k^2 is a Gaussian function parameterized by the color \mathbf{c}_k , center pixel location μ_k , and blob size σ_k . E_{il} is the similarity of two image Gaussians G_i^2 and G_l^2 . The color similarity function $d(\mathbf{c}_i, \mathbf{c}_j)$ measures the similarity between the

colors of two Gaussians and is defined at the end of this subsection. This integral has an analytic solution and analytic derivatives with respect to μ_k and σ_k , enabling efficient optimization (Sec. 5).

Based on S_2 , the similarity S_3 is calculated by projecting each 3D Gaussian G_k^2 in $\mathcal{K}_M(t_j)$ into the respective camera image plane using the projection operator $\Psi(t_j)$ and by computing the corresponding 2D similarity S_2 therein:[‡]

$$S_3(\mathcal{K}_M(t_j), \mathcal{K}_I(t_j)) = \sum_{i \in \mathcal{K}_I} \min \left(\sum_{l \in \Psi(\mathcal{K}_M)} E_{il}, E_{ii} \right), \quad (4)$$

where the min (with E_{ii}) is taken as an efficient approximate self-occlusion test, in order to prevent the overlapping projected 3D SoGs from erroneously contributing multiple times in the above sum. The role of the denominator in Eq. 2 is to normalize the contributions of Gaussians of differing sizes across different images.

Weighting in HSV color space. As illumination in outdoor scenes can vary more strongly than in studio setups, we use a new color similarity that is more resistant to intensity changes. The similarity d for two HSV values \mathbf{a} and \mathbf{b} is defined as

$$d(\mathbf{a}, \mathbf{b}) = 2\varphi_{3,1}(\|\mathbf{a} - \mathbf{b}\|_W) - 1, \quad (5)$$

where $\varphi(\cdot)_{3,1} : \mathbb{R} \rightarrow [0, 1]$ is the smooth, compactly supported Wendland function [Wen95] and $\|\mathbf{a} - \mathbf{b}\|_W^2 := (\mathbf{a} - \mathbf{b})^\top W(\mathbf{a} - \mathbf{b})$ with $W = \text{diag}([1, 1, 0.2]^\top)$, and $\text{diag}(\mathbf{v})$ builds a diagonal matrix consisting of the entries of \mathbf{v} . The down-weighting of the *value* component (using W) in the HSV model is decided based on preliminary experiments: In outdoor scenes, the value component was most severely affected by changes of illumination (e.g., shading, highlight, and specularity) and we experimentally determined a factor of 0.2, i.e. 20% to be a good choice.

Two-sided color similarity term. In contrast to the one-sided color similarity term used in [SHG*11, ESH*12] where $d \geq 0$, the *two-sided* color similarity term (Eq. 5) can be negative when the two input colors are distinct (see Fig. 3). This is important in case of moving cameras, where each camera has its own pose (i.e. translation and rotation) parameters which are optimized based on its own data only (in contrast, the human pose parameters are constrained with data from several views). Figure 4 shows that with the one-sided similarity term, for a given skeletal pose and camera parameters, one can erroneously increase the likelihood (specifically, the similarity S_3 between the model and the image; Eq. 2) by moving the camera towards the object. In this case, the corresponding projected 3D Gaussians become larger and accordingly they lead to higher similarities in (3)

[‡] It should be noted that the projection operator is a function of the camera parameter which is encoded in $X(t_j)$.

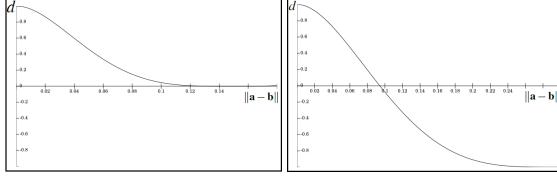


Figure 3: Two-sided vs. one-sided color similarity term. Left: The one-sided similarity term $d(\mathbf{a}, \mathbf{b}) = \varphi_{3,1}(|\mathbf{a} - \mathbf{b}|)$ only adds positive values to the energy if the color differences $|\mathbf{a} - \mathbf{b}|$ are less than a Wendland support of 0.15. Right: The two-sided similarity term (Eq. 5) additionally penalize dissimilar colors.

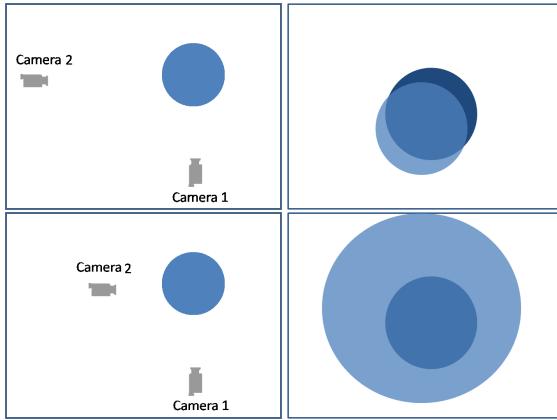


Figure 4: Tracking an object with two cameras. Left: locations of cameras with respect to the object and Right: the constant input image from camera 2 (dark blue circle) overlaid with the projection of the 3D object (light blue) which is a function of the camera and pose parameters. If we move camera 2 closer to the object (bottom row), the virtual object is larger than the input which increase the overlapping between them. This can erroneously increase the likelihood if we do not penalize the dissimilar with the white background, which is achieved with the two-sided similarity term (Sec. 4.2).

in general. In contrast, the two-sided term solves this ambiguity by penalizing the dissimilarity between the projected object and the background. In addition, the two-sided term is also important to enable reliable tracking with only very few static cameras, as we show later.

4.3. Prior on camera motion

As motions of camera and actor are inherently ambiguous (see earlier discussion), estimation of the camera and pose parameters is inherently ill-posed. During the optimization, the effect of a change in camera position may be canceled out in the energy by opposite global pose change of the body. Temporal drifting over subsequent segments and irrevocable convergence to an erroneous local minimum could thus easily happen (see Fig. 9d for an example).

We approach this problem by enforcing first-order temporal smoothness over the parameters. This prevents any rapid change in parameters and the above-mentioned problem can be prevented as our experiments confirm (see Sec. 6):

$$E_{Smooth}(X) = \sum_{i=1}^{n_s} \sum_{j: S_j \cap S_i \neq \emptyset} \|X(l_i) - X(l_j)\|^2, \quad (6)$$

where l_i is the middle point of a segment S_i in time and n_s is the number of segments. E_{Smooth} requests similar values for model parameters at midpoints of overlapping segments from all camera sources.

The other prior E_{Lim} constrains joint angles to an anatomically plausible range as in [ESH*12].

5. Combined camera and pose optimization

We optimize the energy functional (1) using the conditioned gradient descent approach presented in [SHG*11]. At the beginning of a motion sequence, we expect that the body model is shape initialized, and a rough manual initialization of the pose \mathcal{S}_0 in which the actor stands is given. This shape initialization is performed as described in [SHG*11].

Occlusion handling. As explained earlier, detecting and handling the case that a person is occluded from a camera view is crucial for our method. By design, the contribution of each camera to the likelihood is clearly separated from the other cameras (2) and is smooth over time. This enables occlusion detection by monitoring the variation of each model Gaussian's energy component in time.

During the optimization, we inspect the similarities (3) between the projected 3D Gaussians of the model \mathcal{K}_M and the 2D image Gaussians of each camera \mathcal{K}_I : For a given image SoG $\mathcal{K}_{I(k)}$ (corresponding to a camera C_k), the model Gaussian $G_i \in \mathcal{K}_M$ is marked as *false-projected* when $\max_{G_j \in \mathcal{K}_{I(k)}} E_{ij} < T_o$ for a given threshold T_o , where the similarity E_{ij} is calculated for the pair $\Psi(G_i)(G_i \in \mathcal{K}_M)$ and $G_j \in \mathcal{K}_{I(k)}$. When the number of false-projected Gaussians is larger than a threshold T_n , we decide that an occlusion has occurred in the image I_k . In this case, we exclude this camera from the optimization, as the occlusion may otherwise negatively influence the pose estimation of the other cameras. If the occluded camera is non-static, we also do not optimize the corresponding camera parameters. The pose optimization is then continued with the remaining cameras, and the parameters of the camera in which the skeleton is occluded are linearly extrapolated. The parameters T_o and T_n were held fixed at 0.6 and 30 during the experiments.

During the occlusion, the extrapolated parameters of the cameras are compared with the corresponding projected SoGs of the human model (as estimated based on unoccluded cameras). In this way, the end of the occlusion can be detected (i.e. When the number of false-projected Gaussians is less than a threshold T_n). In case the occluded camera is moving, once the occlusion is finished, the camera tracking

starts again with the extrapolated parameters as initialization. This strategy succeeds in most of our test cases where occlusions are short and camera motion smooth. In all other cases (e.g. Fig. 13), more time-consuming global pose optimization would be needed after the occlusion ends.

6. Experiments

We evaluated our algorithm on seven real world sequences, which we recorded in an uncontrolled outdoor scenario with varying complexity: The sequences vary in the numbers and identities of actors to track, the existence and number of moving objects in the background, and the number of moving and static cameras. Sequences also differ in the makes, the frame resolutions and, and the frame rates of cameras. By quad-tree decomposition, all images are effectively down-sampled to a resolution of 160×90 before tracking. Moreover, We recorded two additional sequences in studio for marker based quantitative evaluation of both skeletal motion, as well as camera motion reconstruction accuracy. Table 1 summarizes the specification of the sequences used in experiments. Since cameras are not sub-frame-level synchronized, it is unlikely that frames from all cameras are available for a given time stamp. Accordingly, the time complexity can only be measured based on an average over a time span. Due to the more elaborate energy (in particular the two-sided term), and a larger parameter space, runtime of our algorithm is slightly lower than the approaches by [ESH*12, SHG*11]. Further, the run-time of our algorithm depends on the number of cameras and actors, and the complexity of the scene, e.g. the number of Gaussians needed in 2D. For a single actor and five cameras, our algorithm takes around a minute for processing a single segment S (Sec. 4.1) that contains around two frames captured from each camera. Using the discrete (non-space-time) optimization algorithm of [SHG*11] (see also Section 6.3) on a similar sequence recorded in studio (i.e. lower scene complexity) with 8 cameras, our method performs at 13 seconds per frame. Apart from body model initialization which requires the user to apply a few strokes to background segment the images of four actor initialization poses (see [SHG*11]), tracking is fully-automatic.

Figures 1, 5, 6, and 7 show example poses tracked from sequences *Walk1*, *Walk2*, *Walk4*, *Run*, *Soccer1*, and *Soccer2* (see also the accompanying video). Our algorithm successfully estimated the pose parameters of actors as well as the positions of the moving cameras in these sequences. In particular, our algorithm successfully tracked the two actors in *Soccer2* who often occlude each other (Fig. 9) and the actors in highly cluttered scenes (*Walk2*, *Walk4*, and *Run* each of which contains 9 moving background people; see video). When tracking the moving camera from the *Soccer1* sequence, SfM failed to successfully estimate the camera motion due to motion blur as shown in Fig. 7: Since the hand-held camera is shaking, motion blur occurred across several frames which causes feature tracking to fail. In contrast, our

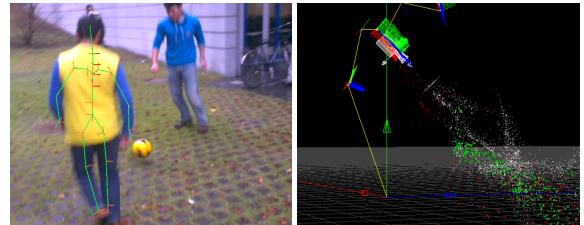


Figure 7: Left: Pose tracking on *Soccer1* sequence viewed from a moving camera . Right: Tracking of the same camera using SfM. The estimated trajectory of the camera is displayed as a yellow line which is far from being smooth indicating the failure.



Figure 8: Left: Tracking *Soccer1* fails with only 3 static cameras. Right: With 2 additional moving cameras it succeeds.

method was able to successfully track both camera and actor pose, even with the challenging background (see video). Only on some isolated frames with stark occlusion, the arm or head are incorrectly tracked, as expected.

To evaluate whether using moving cameras in addition to static cameras, actually improves the quality of the pose reconstruction, we tracked sequence *Soccer1* once with only 3 static cameras, and compared the results to the full tracking with 3 static and 2 moving cameras (Fig. 8). While moving cameras add unknowns to the optimization problem, the additional images provide enough information to increase the tracking accuracy and estimate the camera dynamics.

6.1. Evaluation of system design choices

We qualitatively evaluated the importance of the various components of our algorithm (see Secs. 3 and 4) on the sequences.

Figure 9a compares tracking with our new (non-positive definite) color similarity measure (5) (top row) against tracking with the old color measure of [ESH*12, SHG*11] (bottom row). The new color measure is crucial for successfully estimating the motion of moving cameras (see Sec. 4.2).

In outdoor recordings, the observed brightness of the objects and the background can change. By making our color similarity measure more resistant to changes in brightness (Fig. 9b, top row), tracking becomes more stable compared to the original brightness-sensitive color measurement (Fig. 9b, bottom row). When the Euclidean distance is used instead (bottom), the color of the body model is not distinc-

Table 1: Specification of the sequences used in the experiments.

Sequence	Soccer1	Soccer2	Walk1	Walk2	Walk3	Walk4	Run	Walk5	Walk6
# moving cams.	2	1	1	3	8	2	1	0	3
# static cams.	3	4	4	2	0	6	6	8	5
frame rates	23.8			120			25		
camera types	mobile-phone (<i>HTC One X</i>)			GoPro			studio cameras		
frame resol.	1280 × 720 (original); 160 × 90 (operating resol.; see text)						256 × 256		
# tracked objs.	2				1				
# moving background objs.	0			9			0		

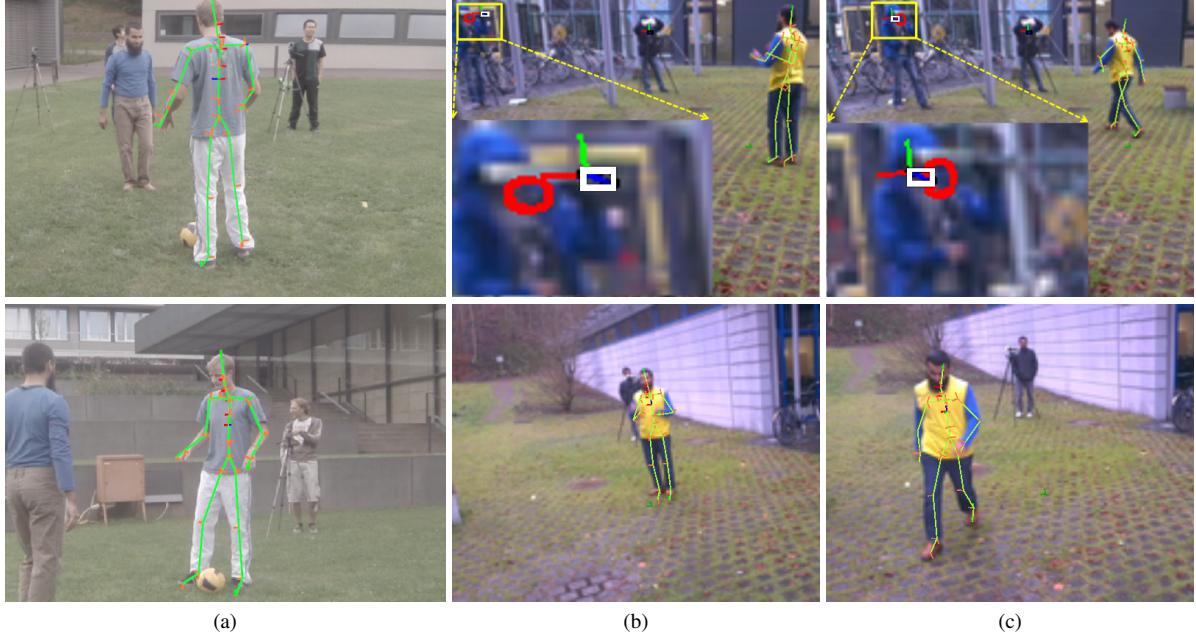


Figure 5: Qualitative analysis of tracking results. The tracking results are displayed as skeletons overlaid over two of the input images. (a) Walk2: The second row shows the accurately tracked skeleton in a camera view that is not used by the algorithm. (b) and (c) Walk1: The first row shows two frames from a static camera which captures the motion of another camera. The second row is the view of the moving camera. The tracked location of the moving camera (white rectangle) is overlaid on the static camera view. The green and red lines depict x and y axes of the camera orientation in the image plane. The moving camera location in the static view is highlighted with the red circles; see the video.

tive enough from the background color as the change of incident illumination (due to shadows; see supplementary video) leads to a large variation in the value component, that causes a tracking error.

Figure 9c demonstrates the importance of our occlusion handling. When the number of cameras is limited, the erroneous contribution of a camera under occlusion to the likelihood can mislead tracking (bottom). This is avoided by actively detecting the occlusion and excluding the corresponding camera from likelihood computation (top).

Finally, our first-order smoothness prior (E_{Smooth} ; Eq. 6) prevents the camera or pose parameters from drifting quickly to implausible values, as observed in the second row of Fig. 9d (see Sec. 4.3).

6.2. Quantitative evaluation

As it is difficult to obtain ground-truth values for real actor motion in an outdoor scenario, we performed a quantitative analysis of our tracking algorithm on synthetic and studio data that were jointly recorded with a multi-view video and marker-based motion capture system.

Synthetic data. We rendered multiple sequences containing a single actor with several combinations of static and moving cameras. The synthetic datasets represent perfect conditions for our algorithm, i.e. it is free of noise and motion blur and clearly separates background and foreground. This allows us to exactly evaluate the accuracy of the optimization (Fig. 10). The motions of each camera are

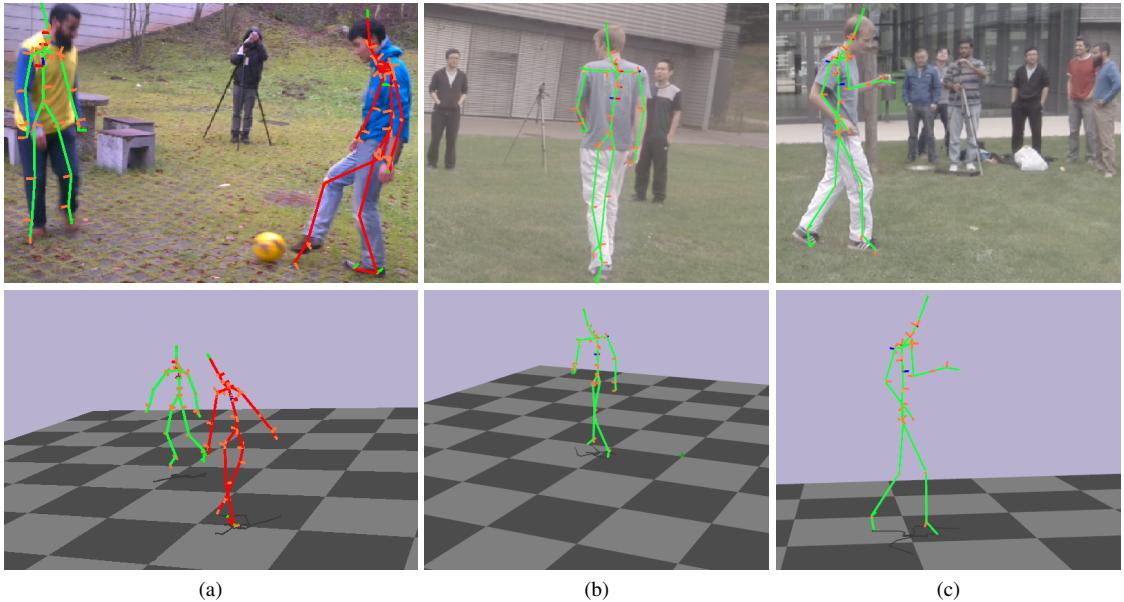


Figure 6: Examples of tracking. The tracking results are displayed as skeletons overlaid over the input images in the first and the third rows. (a) Soccer2, (b) Walk4, and (c) Run: The second row shows the tracked skeletons in views that do not correspond to any camera views

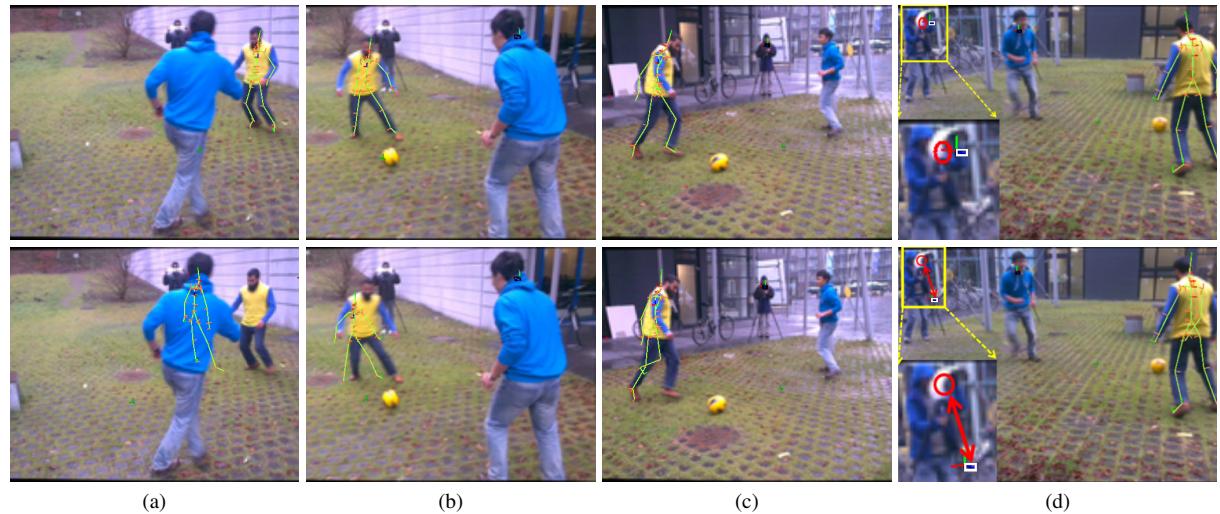


Figure 9: Importance of algorithmic components (Soccer2). The results of our algorithm (top) and alternatives constructed by replacing or removing a certain component, respectively: (a) the two-sided color similarity (5) is replaced by a one-sided similarity [ESH*12, SHG*11], (b) the weighting in HSV color scheme is disabled (i.e., $W = I$ in Eq. 5), (c) the occlusion handling is disabled, (d) the smoothness term in the prior (6) is removed, see text for details.

generated by combining different translations and rotations around the capture volume.

Visual inspection shows that our algorithm manages to correctly track the skeletal and camera poses in all synthetic sequences. As expected, numerical evaluation indicates that a higher number of moving cameras (from a fixed number of total cameras) increases the error, since the optimization

becomes more difficult (Table 2). In this particular setup, at least 2 static cameras fix the global position of the actor accurately. Therefore, decreasing the number of static cameras from 5 to 3 does not affect the skeletal joint position accuracy on an absolute scale, however, it decreases the moving camera tracking accuracy. In general, one static camera is not sufficient to localize the absolute coordinates of the ac-

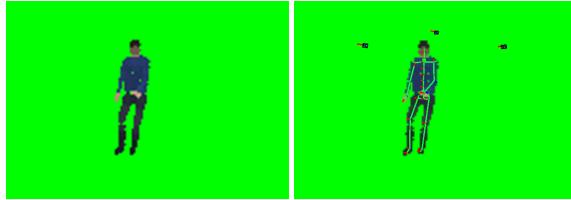


Figure 10: Left: an example frame from the synthetic sequence. Right: tracking result with estimated locations and orientations of three moving cameras overlaid on the frame.

tor and the cameras completely. This leads to an unknown global transformation between our and the ground truth coordinate frames which make the absolute 3D coordinate error not meaningful. Thus, we use 2D joint position error.

6.3. Marker-based quantitative evaluation

An additional important contribution of this paper is a set of validations on sequences that were recorded inside a studio with both a multi-view video system and a frame-synchronized Phase-space marker-based motion capture system. The Phasespace system uses 2 active LED markers attached to the body of the performing actor in the center of the studio. The multi-view video system features cameras of 2048×2048 pixel resolution running at 25 fps. All images are effectively downsampled to a resolution of 256×256 before tracking. The tests in this section are performed using a discrete pose optimization algorithm that estimates a discrete set of pose parameters per time step, rather than our space-time optimizer. For a frame-synchronized video system, this yields better results. Further, this is the only way we can compare against the baseline method of [SHG*11], which also uses this discrete optimization strategy. In the sequences recorded with this setup, the person wears normal street clothing, and markers are attached on top. We will make all these sequences available to the research community. The specifics of each sequence in the set are explained in the following paragraphs that evaluate several facets of our new algorithm.

First, we want to demonstrate that several of our extensions of the pose fitting energy compared to [SHG*11] also lead to improved tracking accuracy over that baseline method when recording with static cameras only. The first sequence was recorded with 8 static video cameras and the marker system in studio lighting, is 150 frames long and shows the actor performing a walking motion. We consider the marker positions measured with the PhaseSpace system as a ground truth for evaluating the tracking accuracy. To this end, we need to identify the positions of the markers w.r.t the skeletal model tracked by our algorithm. We do this by describing each marker with an offset vector in a local frame of the nearest bone. Each such offset between a marker and one skeletal joint is estimated based on observing the offset vector between the marker and the joint on a set of correctly

tracked frames, and keeping the average displacement. First, adding only our weighting in HSV color space to the algorithm of [SHG*11] already decreases the average marker position error from 4.0 cm to 1.9 cm over the baseline method. If, in addition, we add the two-sided color similarity term (which is essential in case of moving cameras) we observe a further reduction in error to 1.4 cm. However, extending the energy from [SHG*11] with the two-sided term alone (i.e. without any weighting in HSV color space), may still lead to errors in bad lighting conditions (e.g. part of the actor is in shadow), because it penalizes dissimilar colors. An adaptive color model would be needed for that which we investigate in future work.. This shows that several of our algorithmic extensions to the baseline fitting energy also benefit the case of static camera tracking and lead to a notable reduction in tracking error; see Fig. 11 and the supplementary video.

We now further show that even in studio conditions, the static algorithm [ESH*12, SHG*11] fails with moving cameras. To confirm this fact and to evaluate the camera tracking accuracy of our algorithm, we recorded a second in studio sequence with 3 moving and 5 static cameras. The sequence is 500 frames long. Our reference for accuracy comparison are the motion capture markers on the body. Our SfM based tracking of the moving cameras may contain errors, and thus yield reprojection errors in the moving cameras. Therefore, the 2D positions of two markers in one moving camera view were annotated manually in a range of 100 frames as ground truth. We use the 2D distance in the image planes of that camera between the respective body markers tracked by our algorithm and the ground truth to assess accuracy. The average error of [SHG*11] is 25.9 pixels which reflects its failure to track this sequence. In contrast, our algorithm achieves an average of 1.8 pixels as it tracked that sequence much more reliably; see Fig. 12 and video.

We further tracked the three moving cameras using a SfM algorithm [THWpS08] and landmarks in the studio background. It failed to track two of the three cameras because their motion consist of only rotation and small translation. This further shows the power of our algorithm which works with any type of motion in the cameras, but also means that we cannot quantitatively compare the tracking accuracy of these two cameras obtained with our algorithm against ground truth. We used the correct SfM tracking of the third camera as a ground truth to evaluate our camera tracking accuracy. The average camera position error is 16.4 (cm) and the average difference in angle in viewing direction between ground truth and our tracked solution is 13.4 degrees. This also shows quantitatively that the camera tracking is of good quality.

Discussion. As the estimation of the camera motion parameters is based only on a small sample of the 3D space (i.e., the position and pose of the actor), resulting camera paths can be less accurate than with SfM approaches. The uncertainty is large in the camera's viewing direction

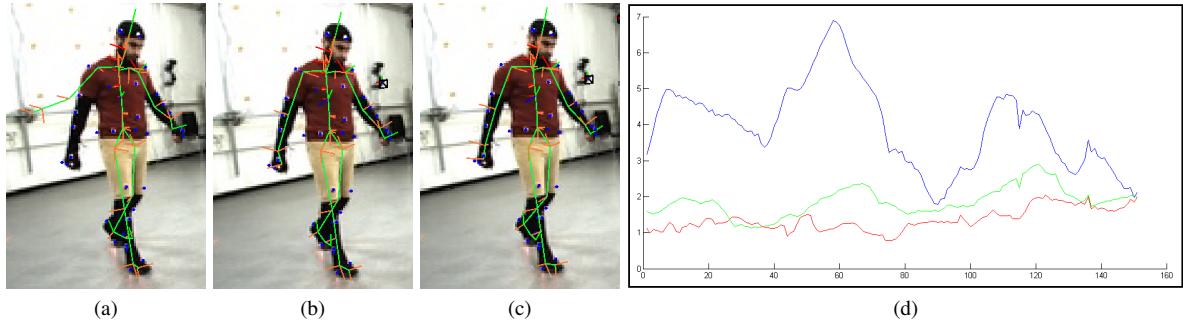


Figure 11: Quantitative evaluation of algorithmic components (Walk5). Tracking result of (a) [SHG*11]; average error 4.0 (cm). The blue dots correspond to the markers positions, (b) our weighting in HSV color scheme with [SHG*11]; average error 1.9 (cm), (c) both weighting in HSV scheme and two-sided similarity with [SHG*11]; average error 1.4 (cm), (d) the plot of the markers positions error per frame where the blue, green and red correspond to (a), (b) and (c); respectively.



Figure 12: Comparison with [SHG*11] on a moving and static cameras sequence (Walk 6). Four sample frames of the tracking result from one moving camera view. Top: [SHG*11]; average error 25.9 (pixel). Bottom: Our algorithm; average error 1.8 (pixel).

(and becomes more pronounced with large focal lengths), as small changes in the distance of the camera to the performer have only little influence on the appearance of the model. But our quantitative evaluation shows that the obtained accuracy is still good under these more challenging conditions. Also, our method successfully tracks both camera and human motion in scenes where traditional SfM methods would fail as

demonstrated in the supplementary video and the quantitative experiments reported earlier. For some scenes, we could include image features as additional evidences into our energy function to increase the stability of the tracking.

Our algorithm requires the tuning of four hyperparameters: λ_1 and λ_2 for controlling the contribution of prior on the final energy and T_o and T_n for occlusion de-

Table 2: Performance of the proposed algorithm for a synthetic scene with varying number of moving and static cameras. The skeletal pose error is measured on average over 65 predetermined skeletal joint position and over the entire frame range in the sequence. The 2D joint position error is measured in a 2D plan of a cameras which is not included in the optimization.

# Moving cams.	1	1	2	3
# Static cams.	5	3	2	1
Average camera position error (cm)	12.44	12.96	16.43	24.17
Average camera view angle error (degree)	2.88	3.09	2.8	2.62
Average skeletal 2D joint position error (pixel)	0.5636	0.5430	0.6532	4.4346



Figure 13: Failure cases. Left: Moving camera does not recover after a long occlusion. Right: Inaccurate arms tracking because of the motion blur.

tection. We chose their values through experiments and kept them fixed for all results.

Although our algorithm produced reasonable tracking results even in challenging environments, failure cases remain. Figure 13 exemplifies specific directions where future improvement is desired: Our occlusion handling strategy relies on the linear extrapolation of camera motions during the occlusion. This may fail when the camera motion is highly nonlinear, which is likely for long occlusions as shown in Fig. 13 left. For this case, a more expensive global optimization could be exercised for recovering from the occlusion. Figure 13 right shows an example of tracking failure (in the left arm) due to strong motion blur.

In the future, synergies between motion deblurring and tracking shall be explored. Occlusion of body parts in many camera views can lead to tracking errors. Solutions to this problem deserve further investigation. Our occlusion detection scheme for cameras can also be used in detecting skeletal pose tracking failures (Sec. 5): When there are more than one camera undergoing occlusion, this indicates a likely skeletal pose error, and a global pose optimization, such as particle filtering, could be initiated to recover from it.

7. Conclusion

This paper presents an algorithm for marker-less human motion capture with moving and unsynchronized cameras that requires only minimal user interaction. Unlike existing approaches for skeletal tracking with moving cameras, our algorithm does not require any additional hardware and succeeds on even highly dynamic and cluttered scenes and for a more general range of camera motion where feature-based

camera calibration fails. Furthermore, our algorithm operates with only few cameras and enables accurate full-body outdoor motion tracking of one or several actors who perform non-trivial motion. This is made possible by a new energy functional that simultaneously models camera and skeletal pose parameters in a space-temporally consistent way based on the appearance of tracked actors. We demonstrated the starkly improved performance and application range of our algorithm relative to a baseline method it originated from both quantitatively and qualitatively in an extensive set of experiments. In this context we further contribute with one of the first evaluation datasets for video-based pose tracking with moving cameras that features ground truth marker-based pose data, as well as ground truth motion data of non-stationary cameras. We believe that our technique is a step towards bridging the gap between complex and expensive capture studios and unconstrained outdoor motion capture, such as on-set tracking, which is essential in many computer graphics applications.

References

- [ARS10] ANDRILUKA M., ROTH S., SCHIELE B.: Monocular 3D pose estimation and tracking by detection. In *Proc. CVPR* (2010), pp. 623–630. [2](#)
- [BMB98] BREGLER C., MALIK J.: Tracking people with twists and exponential maps. In *Proc. CVPR* (1998), pp. 8–15. [2](#)
- [BMB*11] BAAK A., MÜLLER M., BHARAJ G., SEIDEL H.-P., THEOBALT C.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. ICCV* (2011), pp. 1092–1099. [2](#)
- [BSB*07] BALAN A., SIGAL L., BLACK M., DAVIS J., HAUSSECKER H.: Detailed human shape and pose from images. In *Proc. CVPR* (2007), pp. 1–8. [2](#)
- [DR05] DEUTSCHER J., REID I.: Articulated body motion capture by stochastic search. *IJCV* 61, 2 (2005), 185–205. [2](#)
- [ESH*12] ELHAYEK A., STOLL C., HASLER N., KIM K. I., SEIDEL H.-P., THEOBALTL C.: Spatio-temporal motion tracking with unsynchronized cameras. In *Proc. CVPR* (2012), pp. 1870–1877. [2, 3, 4, 5, 6, 7, 9, 10](#)
- [GHK*10] GERMAN M., HORNUNG A., KEISER R., ZIEGLER R., WÜRMLIN S., GROSS M.: Articulated billboards for video-based rendering. *CGF (Proc. Eurographics)* 29, 2 (2010), 585–594. [3](#)
- [GRBS10] GALL J., ROSENHAHN B., BROX T., SEIDEL H.-P.: Optimization and filtering for human motion capture – a multi-layer framework. *IJCV* 87 (2010), 75–92. [2](#)

- [HRT^{*}09] HASLER N., ROSENHAHN B., THORMÄHLEN T., WAND M., GALL J., SEIDEL H.-P.: Markerless motion capture with unsynchronized moving cameras. In *Proc. CVPR* (2009), pp. 224–231. [2](#), [3](#)
- [HZ04] HARTLEY R. I., ZISSEMAN A.: *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004. [4](#)
- [ILS11] IONESCU C., LI F., SMINCHISESCU C.: Latent structured models for human pose estimation. In *Proc. ICCV* (2011), pp. 2220–2227. [2](#)
- [LSG^{*}11] LIU Y., STOLL C., GALL J., SEIDEL H.-P., THEOBALT C.: Markerless motion capture of interacting characters using multi-view image segmentation. In *Proc. CVPR* (2011), pp. 1249–1256. [2](#)
- [MHK06] MOESLUND T., HILTON A., KRÜGER V.: A survey of advances in vision-based human motion capture and analysis. *CVIU* 104, 2 (2006), 90–126. [1](#), [2](#)
- [PMBG^{*}11] PONS-MOLL G., BAAK A., GALL J., LEAL-TAIXE L., MUELLER M., SEIDEL H.-P., ROSENHAHN B.: Outdoor human motion capture using inverse kinematics and von Mises-Fisher sampling. In *Proc. ICCV* (2011), pp. 1243–1250. [2](#)
- [Pop07] POPPE R.: Vision-based human motion analysis: An overview. *CVIU* 108, 1-2 (2007), 4–18. [1](#), [2](#)
- [PVGV^{*}04] POLLEFEYS M., VAN GOOL L., VERGAUWEN M., VERBIEST F., CORNELIS K., TOPS J., KOCH R.: Visual modeling with a hand-held camera. *IJCV* 59, 3 (2004), 207–232. doi:[10.1023/B:VISI.0000025798.50602.3a](https://doi.org/10.1023/B:VISI.0000025798.50602.3a). [2](#)
- [SBB10] SIGAL L., BALAN A., BLACK M.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV* 87 (2010), 4–27. [1](#), [2](#)
- [SHG^{*}11] STOLL C., HASLER N., GALL J., SEIDEL H.-P., THEOBALT C.: Fast articulated motion tracking using a sums of Gaussians body model. In *Proc. ICCV* (2011), pp. 951–958. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [9](#), [10](#), [11](#)
- [SPH11] SHIRATORI T., PARK H. S., HODGINS L. S. Y. S. J. K.: Motion capture from body-mounted cameras. *ACM TOG (Proc. SIGGRAPH)* 30, 4 (2011), 31:1–31:10. [3](#)
- [ST02] SMINCHISESCU C., TELEA A.: Human pose estimation from silhouettes. a consistent approach using distance level sets. In *In WSCG International Conference on Computer Graphics, Visualization and Computer Vision* (2002). [2](#)
- [THWpS08] THORMÄHLEN T., HASLER N., W M., PETER SEIDEL H.: Merging of feature tracks for camera motion estimation from video. In *Proc. CVMP* (2008), pp. 43–50. [2](#), [10](#)
- [WC10] WEI X., CHAI J.: VideoMocap: modeling physically realistic human motion from monocular video sequences. *ACM TOG (Proc. SIGGRAPH)* 29, 4 (2010), 42:1–42:10. [2](#)
- [Wen95] WENDLAND H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. In *Adv. in Comput. Math.* (1995), pp. 389–396. [5](#)
- [YLH^{*}12] YE G., LIU Y., HASLER N., JI X., DAI Q., THEOBALT C.: Performance capture of interacting characters with handheld Kinects. In *Proc. ECCV* (2012), pp. 828–841. [3](#)