Fast Articulated Motion Tracking using a Sums of Gaussians Body Model

Carsten Stoll, MPI Informatik
Nils Hasler, MPI Informatik
Juergen Gall, ETH Zurich
Hans-Peter Seidel, MPI Informatik
Christian Theobalt, MPI Informatik

Abstract

We present an approach for modeling the human body by Sums of spatial Gaussians (SoG), allowing us to perform fast and high-quality markerless motion capture from multi-view video sequences. The SoG model is equipped with a color model to represent the shape and appearance of the human and can be reconstructed from a sparse set of images. Similar to the human body, we also represent the image domain as SoG that models color consistent image blobs. Based on the SoG models of the image and the human body, we introduce a novel continuous and differentiable model-to-image similarity measure that can be used to estimate the skeletal motion of a human at 5-15 *frames per second even for many camera views. In our experiments, we show that our method, which does not rely on silhouettes or training data, offers an good balance between accuracy and computational cost.*

1. Introduction

One of the fundamental problems in computer vision is estimating the 3D motion of humans. Motion capture is an essential part in a wide range of modern industries, ranging from sport science, over biomechanics to animation for games and movies. The state-of-the-art in industrial applications are still marker-based optical capture systems, which enable accurate capture at the cost of requiring a complex setup of cameras and markers. A lot of research has been devoted to developing marker-less methods which can track the motion of characters without interfering with the scene geometry [20, 22, 26].

Marker-less approaches that have been proposed address several aspects like the human model [21, 1, 2, 14], the optimization approach [7, 9, 10], the image features [3, 27], or motion priors [25, 29]. In this work, we revisit the human model that is used for tracking. While recent methods have focused on realistic 3D models of humans that can nowadays be easily derived from full body 3D scans [1, 14], early works like Pfinder [33] relied on simple spatial 2D blob models due to computationally efficiency and achieved real-time performance nearly 15 years ago. Although that approach estimates the articulated pose only in 2D, it does not rely on silhouettes obtained by background subtraction as many current methods [26].

We investigate the idea of representing the human model by a set of spatial Gaussians instead of making use of an explicit surface modeled by geometric primitives or a detailed triangle mesh. To this end, we create a person-specific model from a sparse set of images. The model comprises a kinematic skeleton that defines the degrees-of-freedom (DoF) of the human model and a statistical model that represents the shape and appearance of the human. Since we are interested in real-time performance for pose estimation from multiple views, we use a *Sums of 3D Gaussians* (SoG) for the statistical model. Furthermore, we also represent the image as a sum of spatial 2D Gaussians that cover image blobs that are consistent in color. Based on the representation of the model and the images as Sums of Gaussians, we introduce a novel formulation of the model-to-image similarity and derive an analytical solution that can be solved very efficiently with a gradient ascent algorithm.

In our experiments, we demonstrate that our method, which does not assume an a-priori background estimate, works even in relatively uncontrolled settings. We are able to track the motion in some challenging situations including characters closely interacting with each other, and occlusion situations, such as a person sitting on a chair at a desk. We provide a quantitative evaluation of the tracking performance and demonstrate the reliability of our approach on 17 sequences with over 12000 frames of multi-view video. Our unoptimized implementation processes a multi-view sequence recorded with 12 cameras at a resolution of 1296×972 pixels at 5-15 frames per second on a standard computer.

2. Related Work

Human motion capture has been extensively studied and for a detailed overview we refer to the surveys [20, 22, 26]. Recent approaches [10, 18, 17, 5] that report good results on the HumanEva dataset [26] rely on silhouette data, require training data and/or do not achieve real-time performance. The multi-layer framework proposed in [10] uses a particle-based optimization related to [9] to estimate the pose from

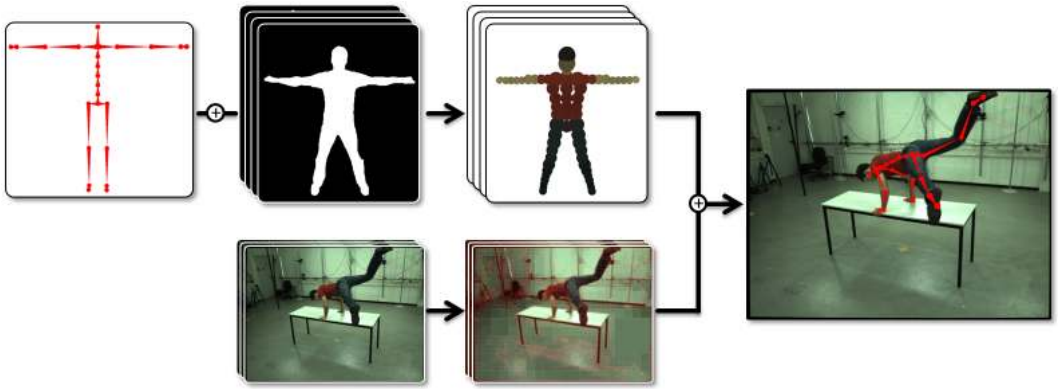

Figure 1. Method overview: We construct an actor-specific human 3D body model based on SoG from a sparse set of multi-view input images in a pre-processing step (top, Section 5.2). We convert our input video streams into a 2D SoG using a quad-tree (bottom, Section 4.1), and use the 3D human body model to estimate the skeletal pose of the actor in the frames (right, Section 5.3).

silhouette and color data in the first layer. The second layer refines the pose and extracted silhouettes by local optimization. The full system requires about 2 minutes per frame. The approaches in [18, 17, 5] require training data to learn either restrictive motion models or a mapping from image features to the 3D pose. These approaches do not generalize to motions that are not part of the training data. By contrast, we propose a nearly real-time approach that does not rely on extracted silhouettes or training data.

Current real-time approaches rely on pose detection based on some image features [28, 4, 31]. These approaches, however, assume that the poses have been previously observed during training. Most related to our approach is the real-time tracking system called Pfister proposed by Wren *et al*. [33]. It models the human by 2D Gaussians in the image domain and represents the appearance of each blob by an uncorrelated Gaussian in the color space. The background is modeled by Gaussians in the color space for each image pixel. Pose estimation is finally formulated as 2D blob detection, *i.e*., each image pixel is assigned to the background or to one of the human blobs. The final 2D pose is obtained by iterative morphological growing operations and 2D Markov priors. The approach has been extended to the multi-view case in [34] where the blobs are detected in each image and the 3D position of the blobs is then reconstructed using inverse kinematics. Our approach does not rely on a statistical background model to detect 2D blobs in the image, but uses 3D spatial Gaussians for the human model and 2D spatial Gaussians for the images to introduce an objective function for the model-to-image similarity that can be very efficiently maximized.

Human pose estimation without silhouette information has been addressed in [6, 13, 8, 11, 24]. These approaches combine segmentation with a shape prior and pose estimation. While [6] use graph-cut segmentation, [8, 11] rely on level set segmentation together with motion features or an analysis-by-synthesis approach. In [13], handheld video cameras and a structure-from-motion approach is used to calibrate the moving cameras. While these approaches iterate over segmentation and pose estimation, the energy functional commonly used for level-set segmentation can be directly integrated in the pose estimation scheme to speed-up the computation [24]. The approach, however, does not achieve real-time performance and requires 15 seconds per frame for a multi-view sequence recorded with 4 cameras at resolution of 656×490 pixels.

Our proposed human model is further related to modeling humans with implicit surfaces. These models have been used for 3D surface reconstruction, *e.g*., [16]. In [21], a human model comprising a skeleton, implicit surfaces to simulate muscles and fat tissue, and a polygonal surface for the skin are used for multi-view shape reconstruction from dynamic 3D point clouds and silhouette data. The implicit surfaces are modeled by Gaussians. Since the estimation of the shape and pose parameters is performed by several processes, the whole approach is very time consuming and not suitable for real-time application. To increase the reliability, motion priors can be used to improve model fitting [30]. In [15], an implicit surface model of a human is matched to 3D points with known normals. By contrast, our approach does not rely on 3D data clouds or silhouette data for tracking, but directly models the similarity between the model and the unsegmented image data.

3. Overview

We capture the performance of an actor with n_{cam} synchronized and calibrated video cameras. The human body model comprises a kinematic skeleton and an attached body approximation modeled as a Sum of Gaussians; see Fig. 3. The skeleton consists of 58 joints, modeling a detailed spine and clavicles. Each joint is defined by an offset to its parent joint and a rotation represented in axis-angle form. It also features an allowable joint limit range l_l to l_h. The

model features a total of 61 parameters Λ, 58 rotational and an additional 3 translational. The skeleton features further a separate degree of freedom (DoF) hierarchy, consisting of n_{DoF} pose parameters Θ. For all the results in this paper we used a DoF hierarchy consisting of $n_{DoF} = 43$ pose parameters.

A linear mapping between Θ and Λ is modeled as an $61 \times n_{DoF}$ matrix $\mathcal{M}$:

$$\Lambda = \mathcal{M}\Theta \tag{1}$$

where each entry of $\mathcal{M}$ defines the influence weight the parameters of Θ have on the joint angles Λ. This construction allows the model to reproduce natural deformation of the spine, as a single DoF can model smooth bending. It also allows straight-forward creation of several different levels of detail without having to edit the kinematic joint hierarchy itself.

An outline of the processing pipeline is given in Fig. 1. In a pre-processing step, we use a low number of manually segmented multi-view images showing example poses to estimate an actor specific body model (Section 4.2). This model is then used for tracking the articulated motion of the actor from multi-view input videos. Each input image is converted into a SoG representation (Section 4.1). Tracking starts with the estimated pose of the model in the previous frame, and optimizes the parameters such that the overlap similarity between the model and image SoG at the current frame is maximized (Section 5.3).

4. SoG-based Similarity

We represent both the image domain $\Omega \in \mathbb{R}^2$ and our 3D human model as Sums of un-normalized Gaussians (SoG). A single Gaussian $\mathcal{B}$ has the form

$$\mathcal{B}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mu\|^2}{2\sigma^2}\right) \tag{2}$$

where σ^2 is the variance and $\mu \in \mathbb{R}^d$ the mean. For the image domain, we have $d = 2$, and $d = 3$ for the human model. Note that the Gaussians model only spatial statistics but not any appearance information like color. Since we are interested in real-time performance, we currently do not model the full covariance matrix. To cover the full image domain or the 3D model, we combine several spatial Gaussians into a Sum of Gaussians $\mathcal{K}$:

$$\mathcal{K}(\mathbf{x}) = \sum_{i=1}^{n} \mathcal{B}_i(\mathbf{x}). \tag{3}$$

In the image domain (Section 4.1), Eq. (3) describes the spatial extent of super-pixels that cluster pixels with similar colors as shown in Fig. 2. For the human body model (Section 4.2), Eq. (3) describes the spatial extent of the human model as illustrated in Fig. 3. Note that our model has in contrast to explicit surfaces infinite spatial support, but the influence decreases with the spatial distance to the mean.

Figure 2. SoG image approximation. Left: Input image. Right: Quad-tree structure with average colors used to generate the SoG. Each square is represented by a single Gaussian.

We store an additional color model $C = \{\mathbf{c}_i\}_i$ for each SoG $\mathcal{K}$, where $\mathbf{c}_i$ is the color associated with the respective Gaussian $\mathcal{B}_i$.

4.1. Approximating Images using SoG

Given an image $\mathbf{I}$, we want to find an approximation of the form of an SoG $\mathcal{K}_\mathbf{I}$ that represents consistent pixel regions. The straight-forward approach is to create a single Gaussian $\mathcal{B}_i$ for each image pixel $\mathbf{p}_i$ and assign to each Gaussian the color value $\mathbf{c}_i \in \mathbb{R}^3$ of the pixel. However, this creates an excessive amount of elements and introduces a large performance penalty. To improve this, we use a quad-tree structure to efficiently cluster image pixels with similar color into larger regions and each of these regions is then approximated using a single Gaussian $\mathcal{B}_i$; see Figure 2). For clustering, we use a threshold ϵ_{col} to determine which pixels to cluster together (typically set to 0.15). If the standard deviation of colors on a quad-tree node is larger than ϵ_{col}, we subdivide the node into four sub-nodes, up to a maximum quad-tree depth of typically 8. Each quadratic cluster is then represented by a Gaussian $\mathcal{B}_i$ where μ is the center of the cluster and σ^2 is set to be the square of half the side-length of the node. Furthermore, the average color $\mathbf{c}_i$ of the cluster is assigned to the Gaussian.

4.2. SoG-based Body Model

Our body model consists of a kinematic skeleton to which we attach a 3D SoG approximation of the performer's body. We manually modeled a default human model, consisting of 58 joints with 63 Gaussians attached to it; see Figure 3. Each Gaussian is attached to a single parent joint of the skeleton, resulting in an SoG model $\mathcal{K}_m$ that is parameterized by the pose parameters Θ of the kinematic skeleton. We adapt this model to generate an actor-specific body model that roughly represents the shape and color statistics for each person we want to track. Since the model acquisition is just a special case of our tracking ap-

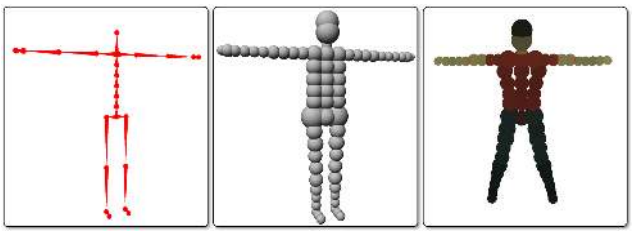

Figure 3. SoG-based body model. From left to right: Default skeleton, default SoG model, actor-specific model. Each Gaussian is illustrated as a sphere with the radius of its variance.

proach, the details will be presented in Section 5.2. From now on, we assume that we already have a human body model with a color value assigned to each Gaussian.

4.3. Projecting 3D SoG to 2D

For comparing the 3D body model $\mathcal{K}_m$ with the image model $\mathcal{K}_\mathbf{I}$, we have to define a projection operator Ψ, which projects a given Gaussian $\tilde{\mathcal{B}}_i : \mathbb{R}^3 \mapsto \mathbb{R}$ of the model to a Gaussian $\mathcal{B}_i : \mathbb{R}^2 \mapsto \mathbb{R}$. Given a camera $\mathcal{C}_l$ with respective 3×4 camera projection matrix P_l and focal length f_l, we define $\mathcal{B} = \Psi_l(\tilde{\mathcal{B}})$ as the following operations :

$$\mu = \begin{pmatrix} [\tilde{\mu}^p]_x / [\tilde{\mu}^p]_z \\ [\tilde{\mu}^p]_y / [\tilde{\mu}^p]_z \end{pmatrix} \qquad s = \tilde{s} f_l / [\tilde{\mu}^p]_z \tag{4}$$

with $\tilde{\mu}^p = P_l \tilde{\mu}$ and $[\tilde{\mu}^p]_{x,y,z}$ being the respective coordinates of the transformed Gaussian mean. Note that this is only an approximation of the true projection due to computational efficiency. The perspective projection of a sphere is usually not a circle as we assume, but rather an ellipsoid. However, the error introduced by this approximation proved to be negligible.

4.4. 2D-2D SoG Similarity

Given two SoG models $\mathcal{K}_a : \mathbb{R}^2 \mapsto \mathbb{R}$ and $\mathcal{K}_b : \mathbb{R}^2 \mapsto \mathbb{R}$ and the associated color models C_a and C_b, where $C = \{\mathbf{c}_i\}_i$ contains the color values assigned to each Gaussian, we can define a function measuring the similarity of the two models. This similarity is defined as the integral of the product of $\mathcal{K}_a$ and $\mathcal{K}_b$ and a similarity measure between the color models, $d(\mathbf{c}_i, \mathbf{c}_j)$:

$$\begin{aligned} &E(\mathcal{K}_a, \mathcal{K}_b, C_a, C_b) \\ &= \int_\Omega \sum_{i \in \mathcal{K}_a} \sum_{j \in \mathcal{K}_b} d(\mathbf{c}_i, \mathbf{c}_j) \mathcal{B}_i(\mathbf{x}) \mathcal{B}_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= \sum_{i \in \mathcal{K}_a} \sum_{j \in \mathcal{K}_b} E_{ij}, \end{aligned} \tag{5}$$

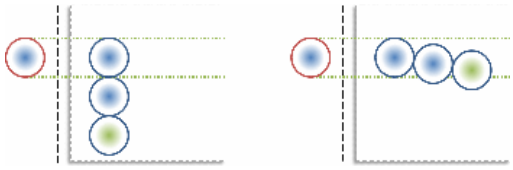

Figure 4. Self-occlusion handling. Inside boxes: Top view of 3D model SoG. Left of dotted line: Image plane with 2D Gaussian. Left column: As long as no occlusions happen, Eq. (5) calculates a correct overlap of a single element. In this example, the color (blue) and the shape are identical, yielding the similarity E_{ii}. Right column: If several 3D model Gaussians project to the same screen space coordinate, their contribution is cumulative, yielding a similarity larger than E_{ii}, even though two of the model Gaussians should be occluded. Using Eq. (8) correctly limits the contribution of a single 2D image Gaussians, yielding the same similarity E_{ii} for both cases.

where

$$\begin{aligned} E_{ij} &= d(\mathbf{c}_i, \mathbf{c}_j) \int_\Omega \mathcal{B}_i(x) \mathcal{B}_j(x) \mathrm{d}\mathbf{x} \\ &= d(\mathbf{c}_i, \mathbf{c}_j) 2\pi \frac{{\sigma_i}^2 {\sigma_j}^2}{{\sigma_i}^2 + {\sigma_j}^2} \exp\left(-\frac{\|\mu_i - \mu_j\|^2}{{\sigma_i}^2 + {\sigma_j}^2} \right). \end{aligned} \tag{6}$$

The similarity $d(\mathbf{c}_i, \mathbf{c}_j)$ is modeled by

$$d(\mathbf{c}_i, \mathbf{c}_j) = \begin{cases} 0 & \text{if } \|\mathbf{c}_i - \mathbf{c}_j\| \geq \epsilon_{sim}, \\ \varphi_{3,1}\left(\frac{\|\mathbf{c}_i - \mathbf{c}_j\|}{\epsilon_{sim}} \right) & \text{if } \|\mathbf{c}_i - \mathbf{c}_j\| < \epsilon_{sim}, \end{cases} \tag{7}$$

with $\varphi_{3,1}(x)$ being the $\mathbf{C}^2$ smooth Wendland radial basis function with $\varphi_{3,1}(0) = 1$ and $\varphi_{3,1}(1) = 0$ [32]. This functional results in a smooth similarity measure that allows for a certain degree of variation in color matching and is guaranteed to be 0 if the difference between the colors is larger than ϵ_{sim}, which was typically chosen to be 0.15.

4.5. 3D-2D SoG similarity

When considering the similarity between the projected SoG model $\Psi(\mathcal{K}_m)$ and the image model $\mathcal{K}_\mathbf{I}$, we have to take the special properties of the projection operation into account. The projection function ignores possible self-occlusions that may happen when projecting the 3D model onto the 2D image plane. Several Gaussians may be projected onto overlapping 2D positions and thereby contribute several times to the energy function. This issue can be resolved implicitly by modifying Eq. (5) to limit the energy that a single image Gaussian can contribute to the total energy of the model:

$$\begin{aligned} &E(\mathcal{K}_\mathbf{I}, \Psi(\mathcal{K}_m), C_\mathbf{I}, C_m) \\ &= \sum_{i \in \mathcal{K}_\mathbf{I}} \min\left(\left(\sum_{j \in \Psi(\mathcal{K}_m)} E_{ij} \right), E_{ii} \right) \end{aligned} \tag{8}$$

with $E_{ii} = \pi {\sigma_i}^2$ being the overlap of an image Gaussian with itself, defining the maximal possible energy it can contribute. This approach is intuitively motivated in Figure 4. If an image Gaussian and a projected 3D Gaussian coincide completely the overlap should be maximal. When projecting a second 3D model Gaussian to a nearby location, it cannot contribute more to the already perfect overlap. While this only partially approximates the effects of occlusion, we found that it is sufficient to resolve most ambiguities introduced by occlusions, while still allowing us to calculate analytic derivatives of the similarity function.

5. A SoG-based Tracking Framework

Our goal is to estimate the pose-parameters Θ of the kinematic skeleton from the set of input images $\mathbf{I}$. To achieve this, we define an energy function $\mathcal{E}(\Theta)$ that evaluates how accurately the model described by the parameters Θ represents what we see in the images. The function is based on the proposed SoG models and can be used for estimating the initial actor-specific body model (Section 5.2) as well as for tracking of articulated motion (Section 5.3).

5.1. Objective Function

The most important part of our energy function is measuring the similarity of the body model in the pose defined by Θ with all input images. Given n_{cam} cameras $\mathcal{C}_l$ with respective SoG images $(\mathcal{K}_l, C_l)$ and the 3D body model $(\mathcal{K}_m, C_m)$ parameterized by the pose vector Θ, we define the similarity function $E(\Theta)$ as

$$E(\Theta) = \frac{1}{n_{cam}} \sum_{l=1}^{n_{cam}} \frac{1}{E(\mathcal{K}_l, \mathcal{K}_l)} E(\mathcal{K}_l, \Psi_l(\mathcal{K}_m(\Theta)), C_l, C_m). \tag{9}$$

For the final energy function $\mathcal{E}(\Theta)$, we add a skeleton and motion-specific term:

$$\mathcal{E}(\Theta) = E(\Theta) - w_l E_{lim}(\mathcal{M}\Theta) - w_a E_{acc}(\Theta). \tag{10}$$

$E_{lim}(\Lambda)$, with $\Lambda = \mathcal{M}\Theta$ (1), is a soft constraint on the joint limits and $E_{acc}(\Theta)$ is a smoothness term that penalizes high acceleration in parameter space:

$$E_{lim}(\Lambda) = \sum_{l \in \Lambda} \begin{cases} 0 & \text{if } l_l^{(l)} \leq \Lambda^{(l)} \leq l_h^{(l)} \\ \|l_l^{(l)} - \Lambda^{(l)}\|^2 & \text{if } \Lambda^{(l)} < l_l^{(l)} \\ \|\Lambda^{(l)} - l_h^{(l)}\|^2 & \text{if } \Lambda^{(l)} > l_h^{(l)} \end{cases}$$

$$E_{acc}(\Theta_t) = \sum_{l \in \Theta_t} \left(\frac{1}{2} \left(\Theta_{t-2}^{(l)} + \Theta_t^{(l)} \right) - \Theta_{t-1}^{(l)} \right)^2$$

where l_l and l_h are lower and upper joint limits and Θ_{t-1} and Θ_{t-2} the poses of the previous frames.

The weights w_l and w_a influence the strength of the constraints and were set to $w_l = 1$ and $w_a = 0.05$ for the majority of our experiments. The impact of the constraints is evaluated in the experimental section.

Figure 5. Estimating an actor specific model from example pose images. Left: Single segmented input image of the multi-view sets for each pose. Right: Resulting actor-specific body model after optimization and color estimation.

5.2. Estimating an Actor Specific Body Model

We use our default skeleton and body model to estimate an actor specific model in a pre-processing step; see Fig. 3. We use a low number of temporally not subsequent, manually segmented multi-view images of example poses as shown in Fig. 5. The four example poses are chosen to articulate a wide range of skeletal joints and therefore allow a relatively accurate estimation of the bone lengths of the skeleton. For each pose represented through a set of multi-view images, we estimate the pose parameters Θ. Additionally, we optimize a common set of shape parameters Θ_{shape} that defines bone lengths as well as the positions and variances of the Gaussian model for a total of 216 degrees of freedom.

The pose parameters Θ are roughly initialized to correspond to the initial poses manually, and the similarity measure (9) based on the binary color values c_i of the silhouette is maximized using the gradient-ascent approach explained in Section 5.3. After the joint optimization, we back-project the color images of each pose onto the 3D Gaussian models and calculate the mean color $\mathbf{c}_i$ for each Gaussian blob taking occlusions into account. Figure 5 shows an actor-specific model that has been acquired.

5.3. Articulated Motion Tracking

Given an image sequence with m frames, we want to estimate the pose parameters Θ^t for each time-step. In each time-step, the parameters Θ^t are initialized by linear extrapolation of the motion in the previous time-steps, *i.e.*, $\Theta_0^t = \Theta^{t-1} + \alpha(\Theta^{t-1} - \Theta^{t-2})$ with α set to 0.5 for all sequences. We now optimize the parameters to maximize the energy (10). Because of the analytic formulation of our overlap measure, we can calculate the analytic gradient $\nabla\mathcal{E}(\Theta)$ efficiently and use it in our optimization procedure.

As one of our main goals is fast performance, we apply an efficient conditioned gradient ascent to optimize our en-

ergy function. Simple gradient ascent tends to be very slow when optimizing energy functions that consist of long narrow valleys in the energy landscape, as it tends to "zig-zag" between opposing walls. To prevent this, we introduce a conditioning vector σ_i into the optimization

$$\Theta_{i+1}^t = \Theta_i^t + \nabla\mathcal{E}(\Theta_i^t) \circ \sigma_i \tag{11}$$

Here, $\circ$ is the component-wise Hadamard product of two vectors. The conditioner σ_i is updated after every iteration according to the following rule:

$$\sigma_{i+1}^{(l)} = \begin{cases} \sigma_i^{(l)}\mu^+ & \text{if } \nabla\mathcal{E}^{(l)}(\Theta_i^t)\nabla\mathcal{E}^{(l)}(\Theta_{i-1}^t) > 0 \\ \sigma_i^{(l)}\mu^- & \text{if } \nabla\mathcal{E}^{(l)}(\Theta_i^t)\nabla\mathcal{E}^{(l)}(\Theta_{i-1}^t) \leq 0 \end{cases} \tag{12}$$

Intuitively this conditioning will increase step-size in directions where the gradient sign is constant, and decrease it if the ascent is "zig-zagging". This is inspired by the resilient back-propagation algorithm [23] used for updating neural networks, and proved to reduce the number of iterations necessary to reach a minimum drastically without having to resort to a more complex and more expensive second order optimizer. We choose $\mu^+ = 1.2$ and $\mu^- = 0.5$ for all our experiments.

We perform at least n_{iter} iterations for each time-step and stop the iterations when $\|\nabla\mathcal{E}(\Theta_i^t) \circ \sigma_i\| < \epsilon$, where $n_{iter} = 10$ and $\epsilon = 0.002$ for most of our experiments.

6. Experiments

For evaluation, we processed 17 sequences with over 12000 frames. The sequences were recorded with 12 cameras at a resolution of 1296×972 pixels at 45 frames per second. Our quad-tree based image conversion (Section 4.1) was set to a maximal depth of 8 nodes, effectively limiting the used resolution to 162×121 2D Gaussians. We found this resolution to be a suitable compromise between tracking accuracy in all our scenes and processing speed. We used the HSV color space for calculating color similarity in all our examples.

The sequences were recorded in a room without any special background and cover a wide range of different motions, including simple walking/running, multi-person interactions, fast acrobatic motions, and scenes having strong occlusions by objects in the scene, such as chairs and tables. The remaining 5 sequences are taken from a different dataset [19]. These sequences were recorded with a green-screen background and show closely interacting characters fighting, dancing, and hugging. Since our approach does not rely on background subtraction, we do not make explicit use of the green-screen, but our approach still benefits from the distinct background color. The segmentation is handled implicitly by our formulation. Please see the accompanying video for more details.

w_l	0.0	1.0	2.0
error (mm)	48.29	44.93	47.78

Table 1. Effect of the joint limit weight w_l on tracking the ground truth dataset. Disregarding this term will lead to physically implausible joint motions, *e.g.*, knees bending backwards.

w_a	0.0	0.015	0.05	0.1	0.5
error (mm)	46.75	46.33	44.93	46.37	51.74

Table 2. Effect of the smoothness weight w_a on tracking the ground truth dataset. Low smoothness leads to jitter in the motion, while high smoothness decreases tracking accuracy.

To speed up the calculation of the energy function in Eq. (10), we remove all image Gaussians B_i whose color similarity is 0 to all of the body model Gaussians B_j, as they will not contribute to the energy function at all. We also remove all elements that are outside of an enlarged bounding box of the pose in the last estimated frame.

Fig. 6 shows some pose estimation results of our algorithm for 6 of the sequences from different camera views. Our method tracked all 17 sequences successfully at $5-15$ frames per second depending on the scene complexity and the actor's motion speed. Even scenes with complex occlusion scenarios and ambiguities can be tracked successfully. For instance, we tracked a performer sitting at a table, and a scene where two actors wearing similarly colored pants interacted closely.

To evaluate the range of image resolution we can reliably deal with, and thus how robust the algorithm is to camera resolution, we also down-sampled the images of several sequences to 81×60 pixels and created a single Gaussian for each pixel. In this experiment, we could not observe a significant loss of the tracking quality.

We compared a standard gradient ascent optimization to our conditioned gradient ascent on the cartwheel sequence shown in Fig. 6. The standard gradient ascent method required on average 65.38 iterations per frame to converge and failed to track the pose of the arms correctly in some frames. Our method on the other hand required only 34.56 iterations per frame and tracked the sequence successfully.

Quantitative Evaluation. We evaluated the accuracy of our tracking approach and compared it to the method of Liu *et al*. [19] on the ground truth multi-person data set kindly provided. We associate the 38 marker positions of the marker system with virtual markers on our body model in the first frame by calculating the closest Gaussian and attaching the marker to its parent joint. Due to the black motion-capture suit and the fast motion, the sequence is difficult to track. The average distance between markers and their corresponding virtual markers on our tracked model is $44.93mm$ with a standard deviation of $27.16mm$. While this is not as accurate as the result reported in [19], namely $29.61mm \pm 25.50mm$, their tracking algorithm employs a laser scanned model and requires several minutes per

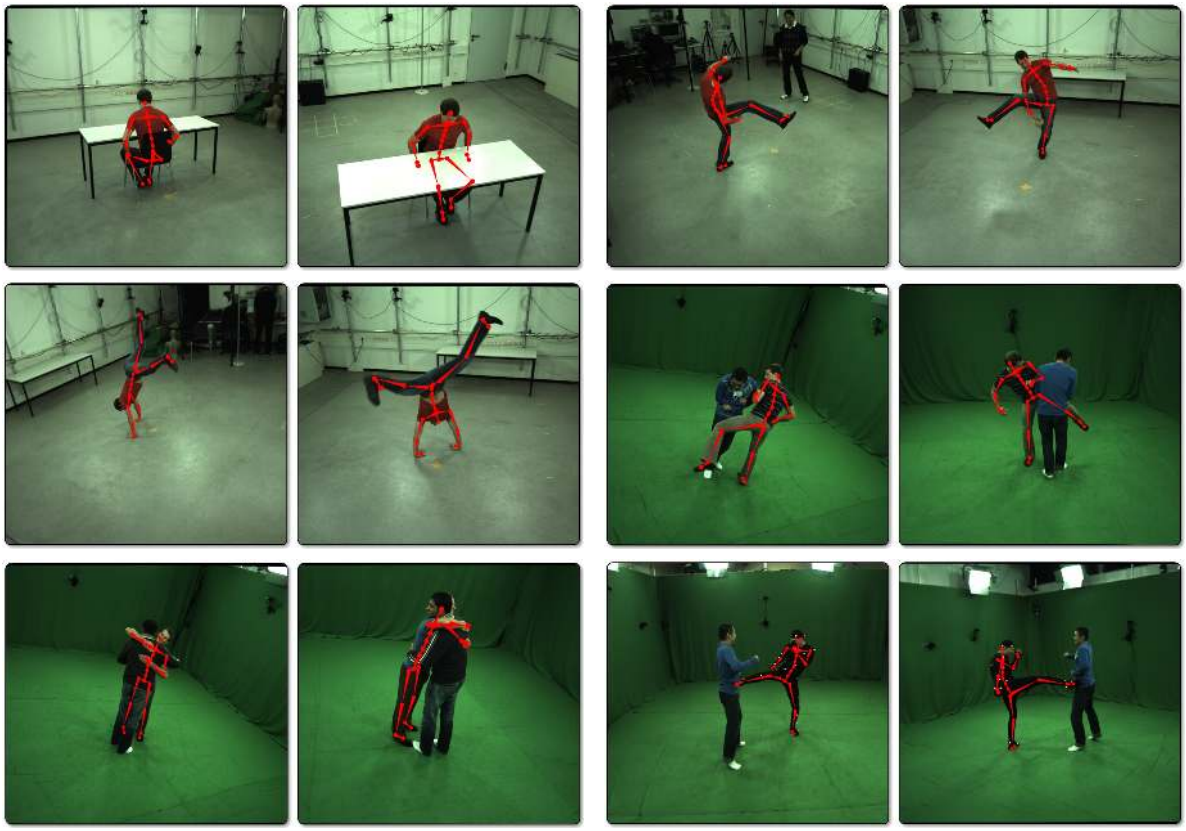

Figure 6. Tracking results of the proposed method shown as skeleton overlay over the input images. Each pair of images shows a single frame of a sequence from two different camera views. From top to bottom, left to right: Sitting at a table, throwing game, cartwheel, fighting, hugging, marker based evaluation scene.

frame for tracking, while our method tracks the sequence at roughly 6 frames per second.

Without compensating for self-occlusions in the overlap term as in Eq. (8), the sequence fails to track correctly at all, leading to an error of $229.09mm \pm 252.04mm$. The impact of the weights w_l and w_a for the joint limits and the smoothness term in Eq. (10) is shown in Tables 1 and 2.

Limitations. Our approach is subject to some limitations. Our constant color model assigned to each Gaussian cannot faithfully model highly textured regions. Although our approach achieves good results at nearly real-time performance, the accuracy could be improved by using a more complex color model at the cost of increased computational expenses. Due to efficiency concerns, the current body model is also only a simplified approximation of a true human body. The spherical shape of the Gaussians make it difficult to accurately track twisting motions, *e.g.*, of outstretched arms or the head. While anatomically correct arm motions can be resolved by inverse kinematics, the head motion could be recovered by an explicit face or head tracker.

Currently, our approach struggles when tracking scenes with less than 5 cameras, where local minima in the energy function are more prevalent. Here, the algorithm may fail to track single limbs correctly and fail to recover. These problems could be overcome by using more complex appearance models for the Gaussians, and by using more sophisticated optimization approaches. Similar to [12], it would be possible to automatically detect tracking errors by inspecting the fitting error and run a global optimization for the misaligned parts.

7. Conclusions

We have introduced a novel model-to-image similarity measure for articulated motion tracking. To this end, we represent both the images and the human body model by Sums of Gaussian where each spatial Gaussian is equipped with a color model. Since the similarity measure is continuous and differentiable, we solve human pose estimation nearly in real-time even for many camera views. Our approach offers a good control over accuracy and computational cost and can be intuitively adapted to the needs of a given application by choosing the resolution of the image and body-model approximations. The accuracy could also be further increased by more complex color models and

the computation time can be further reduced by parallelizing the evaluation of the double sum in Eq. (8) using GPUs or multi-processor systems. Unlike many recent algorithms dealing with markerless motion-capture, our method does not rely on background subtraction or training data and has the capability of running in real-time. This makes the approach practical for real-world applications.

References

[1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. on Graphics*, 24(3):408–416, 2005. 1

[2] A. Balan, L. Sigal, M. Black, J. Davis, and H. Haussecker. Detailed human shape and pose from images. In *CVPR*, 2007. 1

[3] L. Ballan and G. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *3DPVT*, 2008. 1

[4] A. Bissacco, M.-H. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *CVPR*, 2007. 2

[5] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *IJCV*, 87:28–52, 2010. 1, 2

[6] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *ECCV*, pages 642–655, 2006. 2

[7] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *IJCV*, 56(3):179–194, 2004. 1

[8] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. Combined region and motion-based 3d tracking of rigid and articulated objects. *TPAMI*, 32:402–415, 2010. 2

[9] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *IJCV*, 61(2):185–205, 2005. 1

[10] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture – a multi-layer framework. *IJCV*, 87:75–92, 2010. 1

[11] J. Gall, B. Rosenhahn, and H.-P. Seidel. Drift-free tracking of rigid and articulated objects. In *CVPR*, 2008. 2

[12] J. Gall, C. Stoll, E. D. Aguiar, B. Rosenhahn, C. Theobalt, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*, 2009. 7

[13] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR*, 2009. 2

[14] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 2(28), 2009. 1

[15] R. Horaud, M. Niskanen, G. Dewaele, and E. Boyer. Human motion tracking by registering an articulated surface to 3d points and normals. *TPAMI*, 31(1):158–163, 2009. 2

[16] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *TPAMI*, 28:328–333, 2006. 2

This work has been developed within the Max-Planck-Center for Visual Computing and Communication collaboration.

[17] C.-S. Lee and A. Elgammal. Coupled visual and kinematic manifold models for tracking. *IJCV*, 87:118–139, 2010. 1, 2

[18] R. Li, T.-P. Tian, S. Sclaroff, and M.-H. Yang. 3d human motion tracking with a coordinated mixture of factor analyzers. *IJCV*, 87:170–190, 2010. 1, 2

[19] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR*, 2011. 6

[20] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2):90–126, 2006. 1

[21] R. Plankers and P. Fua. Articulated soft objects for multiview shape and motion capture. *TPAMI*, 25(9):1182–1187, 2003. 1, 2

[22] R. Poppe. Vision-based human motion analysis: An overview. *CVIU*, 108(1-2):4–18, 2007. 1

[23] M. Riedmiller and H. Braun. Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. volume IEEE International Conference on Neural Networks, pages 586–591, 1993. 6

[24] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert. Region-based pose tracking with occlusions using 3d models. *Machine Vision and Applications*, pages 1–21, 2011. 2

[25] H. Sidenbladh and M. Black. Learning the statistics of people in images and video. *IJCV*, 54(1-3):183–209, 2003. 1

[26] L. Sigal, A. Balan, and M. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87:4–27, 2010. 1

[27] A. Sundaresan and R. Chellappa. Multicamera tracking of articulated human motion using shape and motion cues. *IEEE Trans. on Image Processing*, 18(9):2114 –2126, 2009. 1

[28] L. Taycher, D. Demirdjian, T. Darrell, and G. Shakhnarovich. Conditional random people: Tracking humans with crfs and grid filters. In *CVPR*, pages 222–229, 2006. 2

[29] R. Urtasun, D. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *CVPR*, pages 238–245, 2006. 1

[30] R. Urtasun and P. Fua. 3d human body tracking using deterministic temporal motion models. In *ECCV*, 2004. 2

[31] M. Van den Bergh, E. Koller-Meier, and L. Van Gool. Real-time body pose recognition using 2d or 3d haarlets. *IJCV*, 83:72–84, June 2009. 2

[32] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. In *Adv. in Comput. Math.*, pages 389–396, 1995. 4

[33] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *TPAMI*, 19:780–785, 1997. 1, 2

[34] S. Yonemoto, D. Arita, and R. Taniguchi. Real-time human motion analysis and ik-based human figure control. In *Workshop on Human Motion*, pages 149–154, 2000. 2