Spatio-temporal Motion Tracking with Unsynchronized Cameras

A. Elhayek, C. Stoll, N. Hasler, K. I. Kim, H.-P. Seidel, C. Theobalt

MPI Informatik

{elhayek, stoll, hasler, kkim, hpseidel, theobalt}@mpi-inf.mpg.de

Abstract

We present a new spatio-temporal method for markerless motion capture. We reconstruct the pose and motion of a character from a multi-view video sequence without requiring the cameras to be synchronized and without aligning captured frames in time. By formulating the model-to-image similarity measure as a temporally continuous functional, we are also able to reconstruct motion in much higher temporal detail than was possible with previous synchronized approaches. By purposefully running cameras unsynchronized we can capture even very fast motion at speeds that off-the-shelf but high quality cameras provide.

1. Introduction

Human pose estimation from videos is one of the fundamental problems in computer vision and has been researched extensively in the past decades. Applications for these methods can be found in a wide range of industries, from entertainment (movies and games) to biomechanics, in sports, and medical sciences. Real-time capture methods made possible through new sensors such as the Microsoft Kinect have opened up new possibilities for human-computer interaction. However, even with all the developments in the past years, for accurate motion capture industry and academia alike still rely on marker-based optical systems that require complex and expensive setups of cameras and markers.

A significant amount of research has thus been devoted to simplifying the setup and accuracy of markerless methods [18, 19, 22]. However, these methods often rely on recording videos with synchronized cameras. These setups require special hardware, and cannot make use of commodity camera hardware with limited frame rates. They are also often expensive and difficult to set up. Hasler *et al*. [11] have introduced a method that performs markerless capture with unsynchronized commodity cameras. Their approach does not make use of sub-frame timing information and instead aligns all frames to the nearest discrete time step. The motion tracking is then performed in the same way as if the cameras were synchronized. This in turn leads to inaccuracies and a reduction of quality in the final results.

Another limitation of markerless methods is that modern video cameras still have a limited frame rate. Marker-based systems often capture motion with over 120 frames per second, allowing them to accurately capture fast and subtle motions alike. In contrast, most commodity video camera systems usually capture images with 30 Hz, with specialized vision systems capturing up to 60 frames per second at reasonable resolutions. This means that fast motion is harder to capture accurately with a markerless setup. If the cameras are run without enforcing synchronization, more samples would be captured in the temporal domain, but spatial coherence will be lost, as in general no two cameras capture at the same time instance.

To address these problems, we introduce a new spatio-temporal marker-less motion capture algorithm that can capture continuous human motion from unsynchronized video streams. Our method allows cameras to capture videos with different sub-frame time offsets and even varying frame-rates. At the same time, we are able to capture faster motion more accurately as the time domain may be sampled much more densely. The new formulation preserves spatial and temporal coherency of the model.

Our main contribution is the introduction of a continuous spatio-temporal energy functional that measures model-to-image alignment at any point in time: Rather than estimating discrete pose parameters at each time step, we estimate continuous temporal parameter curves that define the motion of the actor. By design, the energy functional is *smooth* and accordingly the derivatives of any order can be computed analytically, allowing effective optimization. Similar to [23], we represent both the actors body as well as the input images as Sums-of-Gaussians (SoG). We also present a method to enforce joint limits in the continuous pose-curve space. In the experiments we show that our approach can simplify the capture setup in comparison to previous marker-less approaches and that it enables reconstruction of much higher temporal detail than synchronized capture

methods. Because of this, slow cameras can be used to capture very fast motion with only little aliasing.

2. Related Work

Human motion capture has been extensively studied in the computer vision community. We refer the reader to the surveys [18, 19, 22] for a detailed overview of the field. The approaches can be roughly divided into methods that rely on multi-view input and methods that try to infer pose from a single view. Single view methods, such as [1, 13], have gained more attention in the past few years. However, the results do not reach the accuracy of multi-view methods and usually do not use character models with many degrees of freedom. Almost all multi-view methods to date rely on synchronized multi-view input.

The majority of multi-view tracking approaches combine a body model, usually represented as a triangle mesh or simple primitives, with silhouette and image features, such as SIFT [16], for tracking. The methods differ in the type of features used and the way optimization is performed. The multi-layer framework proposed in [8] uses a particle-based optimization related to [7] to estimate the pose from silhouette and color data in the first layer. The second layer refines the pose and extracted silhouettes by local optimization. The approaches in [15, 14, 3] require training data to learn either motion models or a mapping from image features to the 3D pose. The accuracy of these models is usually measured on the HumanEVA benchmark [22].

Tracking without silhouette information is typically approached by combining segmentation with a shape prior and pose estimation. While [4] use graph-cut segmentation, [5, 9] rely on level set segmentation together with motion features or an analysis-by-synthesis approach. 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 [20]. The approach in [23] introduced an analytic formulation for calculating model to image similarity based on a Sums-of-Gaussians model. Both body model and images are represented as collection of Gaussians with associated colors. The energy functional is continuous in parameter space and allows for near real-time tracking of complex scenes.

The only work addressing the necessity for complex and expensive synchronized multi-view camera setups for tracking is [11]. There, the input sequences are recorded with handheld video cameras. Multi-view calibration is performed using a structure-from-motion approach, and sub-frame accurate synchronization is achieved by optimizing correlation of the audio channels of each video. However, during the human pose estimation stage the sub-frame information is discarded and the videos are treated as synchronized with one-frame accuracy (i.e. all images taken at the same time instant) for further processing. The estimation step creates silhouettes using a level-set segmentation and uses these for pose optimization. As we show in the following this approximation is not valid for fast motion.

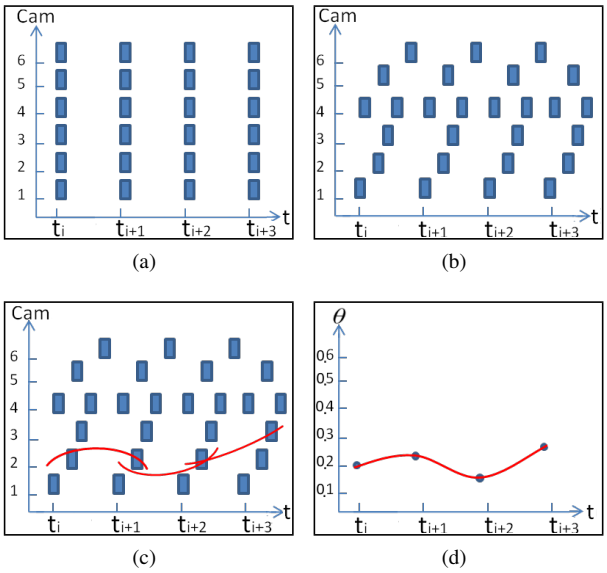

Figure 1. Basic concept: *(a)* Synchronized cameras images distribution. *(b)* Image distribution of unsynchronized cameras after mapping to a single time line. *(c)* The interval functions for a single pose parameter. *(d)* After blending, we have reconstructed a continuous pose function for the entire domain.

3. Overview

Multi-view tracking methods usually capture the performance of an actor with n_{cam} synchronized video cameras (Fig. 1a). The human body is modeled using a kinematic skeleton and an approximation of the body geometry, using, for example, a triangle mesh from a scan [10], a statistical model [2], simple primitives like cylinders [21], or a continuous function [12]. For each frame i at time t_i of the synchronized input video streams the parameters of the kinematic skeleton Θ_{t_i} are optimized to maximize similarity of the pose with the input images. This can be measured with an energy functional $E_{t_i}(\Theta_{t_i})$ that is minimized.

Our approach instead considers unsynchronized video streams where each image is taken at a different time t (Fig. 1b). Note that all cameras can run at different frame rates as well. We assume that timestamps t_i for each image are given. These could be obtained using for example the audio-synchronization method from [11] or by image based methods such as [6, 17].

When recording unsynchronized video, it is possible to sample more densely in time compared to synchronized video. This comes at the cost of losing spatial information

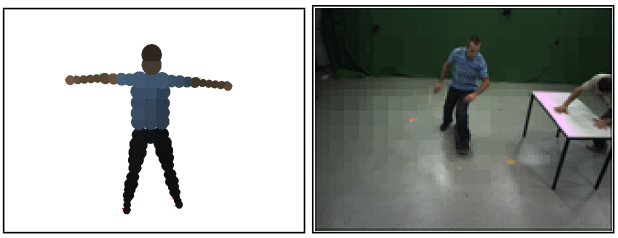

Figure 2. SoG model overview. *Left:* Body model generated from example input images. *Right:* Image SoG approximation generated from a quad-tree (each cell represents one Gaussian).

at each time instant (Fig. 1b). This poses a new challenge, as for a given time step, only a single view will be available. Exclusively fitting pose parameters to a single image at each time step would lead to unstable tracking since the problem is underdetermined due to ambiguities and occlusions. Instead of estimating the pose parameters Θ for each discrete time step, we estimate a smooth function $\Theta(X(t))$, which for each given time instance t, represents the corresponding vector of pose parameters. This representation enables us to aggregate information collected from nearby images in time, such that for each time step, the determination of pose parameter becomes well-posed. Effectively, we are trading spatial resolution for higher temporal resolution but we will show that we only lose a little spatial resolution and gain a lot in temporal accuracy.

As fitting a single continuous function to the whole sequence at once would require a very complex function and be difficult to optimize, we instead divide the sequence into overlapping segments $\mathcal{S}_j$ of length l_{seg} and fit a set of simple polynomial functions to each segment (Fig. 1c). A globally continuous function is then computed by blending the segments with a partition of unity method (Fig. 1d).

4. Spatio-Temporal Tracking

The proposed tracking algorithm adopts an energy-minimization approach. We use an energy functional which measures the dissimilarity between a human body model and the input sequence. As described shortly, the energy functional is continuous both in space and in time such that the evaluation of the model (*i.e.* measuring the disagreement from the input) is possible at any given time (*c.f.* Sec. 4.2). To facilitate this, we represent the model based on continuous functions. Specifically, we adopt the Sums-of-Gaussians (SoG) representation as proposed by Stoll *et al.* [23]. Human articulation is modeled by a kinematic skeleton and its shape is represented using a 3D SoG, where each 3D Gaussian is attached to exactly one bone in the articulation hierarchy. The model is generated by fitting it to a set of example images (Fig. 2 left). To reduce the computational cost, the input images are also approximated based on 2D SoG using a fast quad-tree based clustering method (Fig. 2 right). Each single Gaussian in the SoG sets is associated with a color $\mathbf{c}$ that can be used to measure color similarity between two blobs. For each time step, measuring the similarity between a 3D SoG and a 2D SoG is facilitated by projecting the 3D SoG of the body model into the corresponding image plane and performing the comparison in 2D (Sec. 4.1). Using this SoG-based formulation as a basis has the advantage that the original formulation is already smooth in space. It does not rely on calculating and updating any image features or silhouette correspondences. As a result, extending the approach to the temporal domain comes naturally. It can also handle tracking highly complex articulated models.

The skeleton we use in our approach consists of 58 joints. Each joint is defined by an offset to its parent joint and a rotation represented in axis-angle form. In total, the model has 61 parameters Λ (58 rotational and 3 translational). The skeleton further features a separate degree of freedom (DoF) hierarchy, consisting of the n_{DoF} pose parameters Θ. The degrees of freedom are mapped to the joint parameters using a $61 \times n_{DoF}$ matrix $\mathcal{M}$, where $\Lambda = \mathcal{M}\Theta$. For all the results in this paper we used a DoF hierarchy consisting of $n_{DoF} = 43$ pose parameters. We also model an allowable parameter range l_l to l_h for each DoF that prevents anatomically implausible pose configurations.

4.1. Model to Image Similarity Measure

For two given $2D$ SoGs $\mathcal{K}_a$ and $\mathcal{K}_b$ provided with colors $\mathbf{c}$ for each Gaussian blob, respectively, their similarity is defined as [23]

$$\begin{aligned} &E(\mathcal{K}_a, \mathcal{K}_b) \\ &= \int_\Omega \sum_{i \in \mathcal{K}_a} \sum_{j \in \mathcal{K}_b} d(\mathbf{c}_i, \mathbf{c}_j) \mathcal{B}_i(\mathbf{x}) \mathcal{B}_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= \sum_{i \in \mathcal{K}_a} \sum_{j \in \mathcal{K}_b} E_{ij}, \end{aligned} \tag{1}$$

where $\mathcal{B}(\mathbf{x})$ is a Gaussian basis function

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

E_{ij} is the similarity between a pair of Gaussians $\mathcal{B}_i$ and $\mathcal{B}_j$ given their colors $\mathbf{c}_i$ and $\mathbf{c}_j$:

$$\begin{aligned} E_{ij} &= d(\mathbf{c}_i, \mathbf{c}_j) \int_\Omega \mathcal{B}_i(\mathbf{x}) \mathcal{B}_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= d(\mathbf{c}_i, \mathbf{c}_j) 2\pi \frac{{\sigma_i}^2 {\sigma_j}^2}{{\sigma_i}^2 + {\sigma_j}^2} \exp\left(-\frac{\|\mu_i - \mu_j\|^2}{{\sigma_i}^2 + {\sigma_j}^2} \right). \end{aligned} \tag{3}$$

The color similarity function $d(\mathbf{c}_i, \mathbf{c}_j)$ measures the Euclidean distance between $\mathbf{c}_i$ and $\mathbf{c}_j$ in the HSV color space

and feeds the result into a Wendland function [24]. This renders d a smooth function bounded in $[0, 1]$ (0 for dissimilar input and 1 for similar input).

To measure the similarity between a given pose Θ of our body model $\mathcal{K}_m(\Theta)$ and a given input image SoG $\mathcal{K}_I$, we first need to project the body model into the respective camera image plane using the projection operator Ψ. Given a camera $\mathcal{C}_l$ with respective 3×4 camera projection matrix P_l and focal length f_l, we define the *projected* 2D Gaussian $\mathcal{B} = \Psi_l(\tilde{\mathcal{B}})$ corresponding 3D Gaussian $\tilde{\mathcal{B}}$ based on the following operations:

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

with $\tilde{\mu}^p = P_l \tilde{\mu}$ being the perspective-transformed 3D Gaussian mean.

Using this projection operator we define the model to image similarity as

$$E_{sim}(\mathcal{K}_I, \mathcal{K}_m(\Theta)) = \sum_{i \in \mathcal{K}_I} \min \left(\left(\sum_{j \in \Psi(\mathcal{K}_m)} E_{ij} \right), E_{ii} \right). \tag{5}$$

To prevent overlapping projected 3D SoGs from contributing multiple times in the above sum and distorting the similarity function accordingly, we clamp the similarity to be at most E_{ii}, which is the similarity of the image Gaussian with itself. This can be seen as a simple approximation of an occlusion term (*c.f.* [23] for more details).

4.2. Spatio-Temporal Similarity Measure

As estimating a single continuous function for a whole sequence quickly becomes intractable, we first divide the sequence into overlapping time segments $\mathcal{S}_j$ of length l_{seg}. We represent each of the n_{DoF} parameters of the kinematic skeleton for each segment using a polynomial $X(t, \psi_j)$ of degree n_{deg}, where $\psi_j = \left[\chi_l^k\right]$ with $k \in 1 \dots n_{DoF}$ and $l \in 1 \dots n_{deg}$ are the coefficients of the polynomial. We call the function $X(t, \psi_j)$ the *motion function* for time segment j (see Fig. 1c). Choosing a low degree polynomial as local motion function presents a good compromise between function smoothness and function complexity.

Given an input image SoG $\mathcal{K}_I^i$ with its respective timestamp t_i and coefficients ψ_j of the current motion function we can estimate the similarity between the two using equation 5 as

$$E_{sim}(\mathcal{K}_m(X(t_i, \psi_j)), \mathcal{K}_I^i). \tag{6}$$

We can now sum up the similarity of all n_{img} image SoGs $\mathcal{K}_I$ which belong to the segment $\mathcal{S}_j$ to get a spatio-temporal similarity measure over the the entire segment:

$$E_{sim}(\psi_j) = \frac{1}{n_{img}} \sum_{t_l \in \mathcal{S}_j} \frac{1}{E_{sim}(\mathcal{K}_I^{t_l}, \mathcal{K}_I^{t_l})} E_{sim}(\mathcal{K}_m(X(t_i, \psi_j)), \mathcal{K}_I^{t_l}). \tag{7}$$

It should be noted that this similarity measure is smooth in space and time and accordingly the analytical derivatives of any order can be computed easily with respect to the coefficients ψ_j of the model's motion functions.

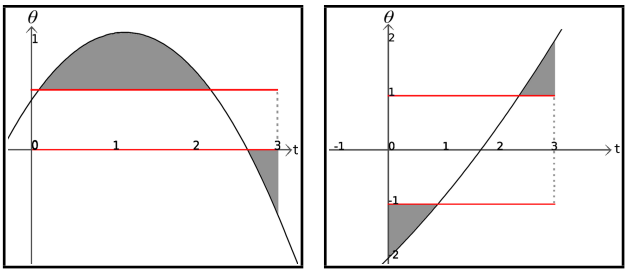

Figure 3. Two limit violation examples. The red lines are the DoF limits boundaries. We compute the DoF function in the interval $\mathcal{S}$=[0,3]. The proposed error measure is the integral of the gray areas.

4.3. Spatio-Temporal Joint Limits

An important component of articulated motion tracking systems is enforcing anatomically correct joint motion. All joints in the human body only have a limited amount of articulation. To prevent anatomically implausible poses, tracking systems usually penalize poses that exceed certain joint limits. This happens either by adding a penalty to the energy that is being optimized or by limiting the admissible range of DoF parameters through box constraints. Modeling these limits in the discrete case is straightforward, but becomes more involved in the spatio-temporal formulation from Section 4.2.

We want to penalize motion functions $X(t, \psi_j)$ where parts of the functions lie outside an admissible limit range $[l_l, l_h]$ for $t \in \mathcal{S}_j$ (see Fig. 3 for examples). We can define a penalty function $E_{lim}(j)$ that measures the area of the functions that exceeds the limits within the segment as

$$E_{lim}(\psi_j) = \left(\int_{t \in \mathcal{S}_j \wedge X(t,\psi_j) < l_l} l_l - X(t, \psi_j) \, \mathrm{d}t + \int_{t \in \mathcal{S}_j \wedge X(t,\psi_j) > l_h} X(t, \psi_j) - l_h \, \mathrm{d}t \right)^2. \tag{8}$$

As can be seen in Figure 3, this penalty function has to handle 10 different cases depending on the position of the curve with respect to the limits and the segment boundaries. However, each case has a compact analytical solution and deriva-

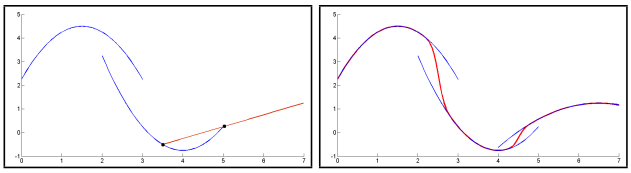

Figure 4. Motion functions. *Left:* Initialization of new segment (red) from previous segments functions (blue) . *Right:* Blended global motion function (red) generated from three local motion functions (blue).

tives with respect to the curve coefficients ψ_j (*c.f.* supplementary material).

4.4. Segment Tracking

We combine the spatio-temporal similarity measure E_{sim} and the limit penalty term E_{lim} into a single energy functional

$$E(\psi_j) = -E_{sim}(\psi_j) + \alpha E_{lim}(\psi_j), \tag{9}$$

where α is a weight factor which determines how strongly we want to penalize non-anatomical pose configurations during tracking. As we can calculate analytical derivatives of both energy terms, we can calculate the gradient $\nabla E(\psi_j)$ efficiently. We find the minimum of $E(\psi_j)$ using a simple conditioned gradient descent method similar to [23] :

$$\psi_j^{i+1} = \psi_j^i + diag(\sigma_i)\nabla E(\psi_j^i). \tag{10}$$

The conditioner σ_i is updated after every iteration according to the rules:

$$\sigma_{i+1}^{(l)} = \begin{cases} \sigma_i^{(l)}\mu^+ & \text{if } \nabla E(\psi_j^i)\nabla E(\psi_j^{i-1}) > 0 \\ \sigma_i^{(l)}\mu^- & \text{if } \nabla E(\psi_j^i)\nabla E(\psi_j^{i-1}) \leq 0. \end{cases} \tag{11}$$

Using the conditioner increases the convergence rate of the gradient descent method in long and narrow valleys of the objective function, as it effectively dampens oscillations and increases step-size in the direction of the valley. We found that this simple approach needs more iterations to converge than higher order optimization schemes, but is still faster in many cases as each iteration is simpler to calculate.

We assume that the actor in each sequence starts in a known pose (for example T-Pose) and is not moving for a brief moment. We find the parameters for the first segment $\mathcal{S}_0$ by initializing the body model pose to the known pose and only optimizing the constant coefficients χ_0^k of the motion function (Fig. 4). We ignore all linear and higher order coefficients and set them to 0. This essentially optimizes for a constant pose without any motion in the current segment.

Each following segment $\mathcal{S}_j$ is placed so that it overlaps with the previous segment by $l_{overlap}$, which is given as percentage of the segment length (Fig. 4). We initialize the coefficients of our current segment to be a linear extrapolation of the motion in the previous segment (Fig. 4). We then run the optimization for all parameters χ_l^k until convergence.

4.4.1 Motion Function Blending

The estimated continuous functions for each segment $\mathcal{S}_j$ may not agree with each other in the overlapping regions (Fig. 4b in blue). To generate a globally smooth motion function we therefore blend all local motion functions together using a partition-of-unity approach (Fig. 4b in red). We define a weight function $w_j(t)$ for each segment that is 1 at the center and falls off smoothly to 0 at the segments boundaries, and is 0 everywhere else. Using the $\mathbf{C}^2$ smooth Wendland radial basis function $\varphi_{3,1}(x)$ [24] the final global motion function is defined as

$$X_{global}(t) = \frac{\sum_{\forall \mathcal{S}_j} w_j(t) X(t, \psi_j)}{\sum_{\forall \mathcal{S}_j} w_j(t)}. \tag{12}$$

Blending the motion function is a post-processing step and is performed after all segments have been optimized. The resulting motion function $X_{global}(t)$ is $\mathbf{C}^2$ smooth in t and represents the tracking result of our algorithm.

5. Experiments

We evaluated our method on 9 sequences recorded with 11 unsynchronized cameras at a resolution of 162×121 pixels with varying frame-rates between 45 and 70 frames per second with a total of about $\sim$ 6000 frames of video. The camera setup used for our experiments provides us with accurate timestamps for each image. When using setups without this possibility, we could estimate timestamps using methods such as [11] or [6, 17]. This was not the focus of our work however. We estimated kinematic skeletons and Gaussian body models for 3 actors and used the quad-tree based image conversion from [23] to convert the input images to SoG models.

The recorded scenes cover a wide range of different motions, from simple walking/running, over fast acrobatic motions, to scenes with as many as 6 people featuring strong occlusions. The tracking approach does not rely on an explicit background subtraction and implicitly separates actors from background using the colors of the SoG body models. The green screen visible in part of the background is not used for explicit segmentation.

Our non-optimized, single-threaded implementation of the spatio-temporal tracker requires on average between 1 and 5 seconds to find the optimal parameters for each segment per actor. This depends mainly on the motion complexity, *i.e.*, fast motions take longer to track.

Figure 5 shows pose estimation results of our algorithm for some of the sequences from different camera views. Our

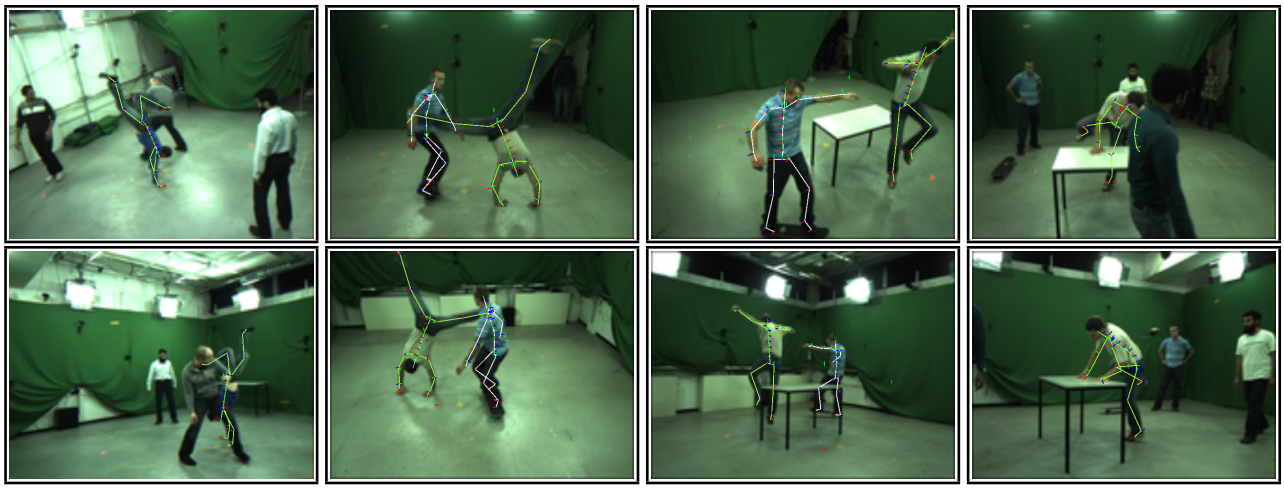

Figure 5. Complex motion tracking. Tracking results of the proposed method on unsynchronized sequences shown as skeleton overlay over the input images. We successfully tracked actors in several challenging scenarios, including sequences with multiple people interacting closely, heavy occlusions, and fast motion from acrobatics and skateboarding.

method tracked all sequences successfully with the same settings used for segment size $l_{seg} = 2.0$ frames of the slowest framerate, overlap of $l_{overlap} = 0.6$, and joint limit weight of $\alpha = 0.1$. The figure also shows results for tracking multiple people in the same sequence. Here, we tracked each actor separately without specifically modeling character interactions (such as contact) or segmenting the input images.

Compared to results created by aligning multiple images to a single time-step and using a discrete tracking approach, our spatio-temporal formulation creates more accurate results. The discrete tracker also fails to correctly track some sequences with complex occlusions and fast motions.

Quantitative Evaluation: To evaluate our method quantitatively we recorded a sequence $\mathcal{S}_{ref}$ with the actor walking with increasing speed with a synchronized camera setup recording at 70 frames per second (Figure 6a). We then created an unsynchronized sequence $\mathcal{S}_{unsync}$ from this scene by temporally subsampling the input video such that only a single camera image is kept at each time instant (Figure 6b) . The downsampled sequence effectively has each camera recording at ~ 7 frames per second, slightly offset to each other. This represents an extreme case, as for all but the slowest motions, the cameras will see vastly different poses for the actor. Finally, we also created a synchronized low-speed sequence $\mathcal{S}_{low}$ which contains only every 11th frame for each camera (Figure 6d) . All three downsampled sequences contain the same number of images.

We used the full sequence $\mathcal{S}_{ref}$ to create a baseline synchronized tracking results $\mathcal{T}_{ref}$ using the method from [23]. We then tracked the actor from the unsynchronized sequence $\mathcal{S}_{unsync}$ with our spatio-temporal approach to generate a result $\mathcal{T}_{cont}$. We also generated tracking results by aligning all 11 cameras of $\mathcal{S}_{unsync}$ to the same time-step (Figure 6c) and using the synchronized tracker to generate $\mathcal{T}_{aligned}$, and tracked sequence $\mathcal{S}_{low}$ to generate $\mathcal{T}_{low}$.

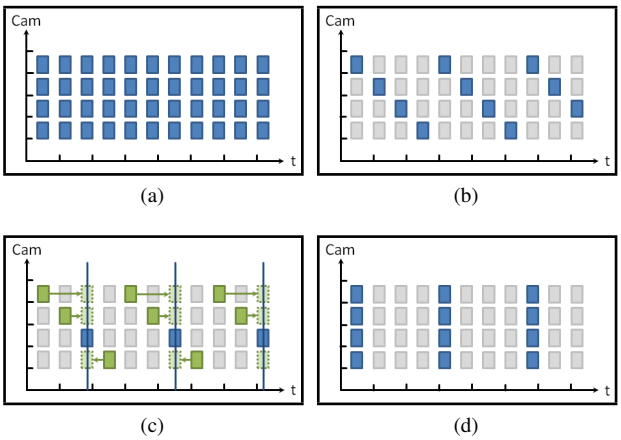

Figure 6. Evaluation sequence: (a) Synchronized baseline sequence. (b) Subsampled unsynchronized sequence. (c) Aligned sequence created from the unsynchronized sequence. (d) Low framerate sequence.

As can be seen in the supplementary video, both $\mathcal{T}_{aligned}$ and $\mathcal{T}_{low}$ fail to track the sequence correctly until the end. On the other hand, our spatio-temporal tracking result $\mathcal{T}_{cont}$ successfully tracks the motion of the actor even when the actor is moving extremely fast towards the end of the sequence. Figure 8 shows the per frame joint position error compared to the baseline result $\mathcal{T}_{ref}$ for the spatio-temporal result (red), the aligned discrete tracker (blue), and the low fps synchronized tracker (green). We used linear interpo-

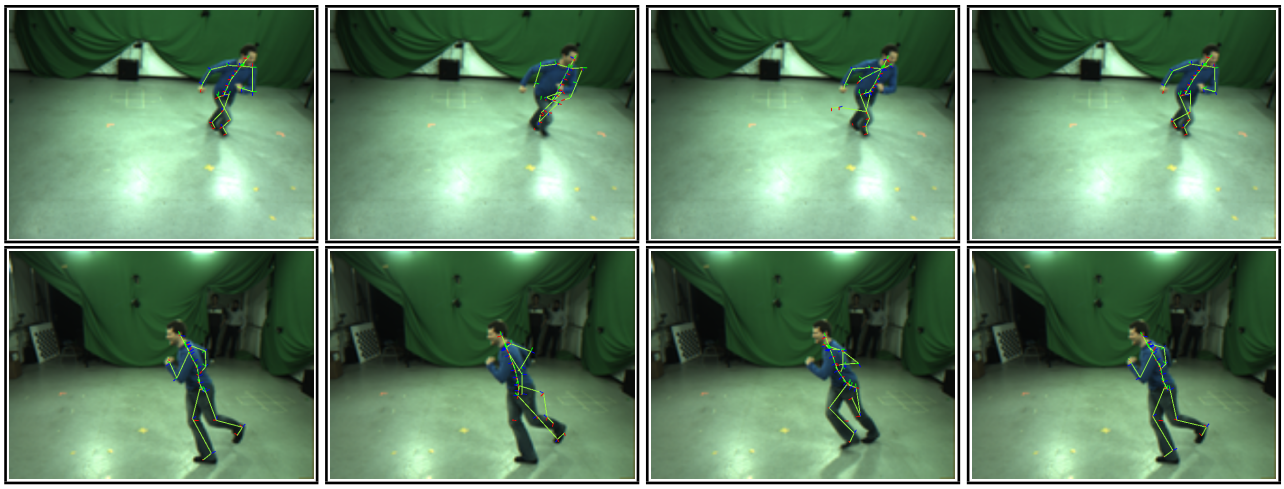

Figure 7. Quantitative evaluation. From left to right: Baseline tracking result $\mathcal{T}_{ref}$, aligned tracking $\mathcal{T}_{aligned}$, subsampled tracking result $\mathcal{T}_{low}$ and our tracking result $\mathcal{T}_{cont}$. Only our spatio-temporal tracking method is able to successfully track the whole sequence.

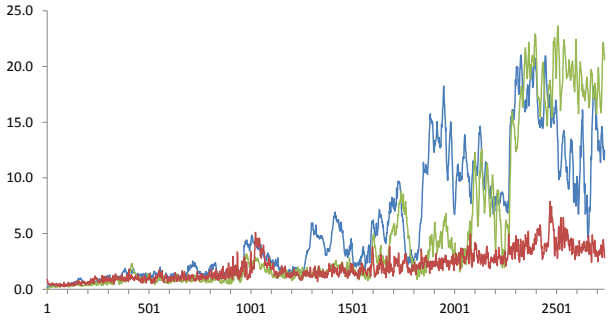

Figure 8. Quantitative comparison between our spatio-temporal tracking approach (red), synchronized tracking with unsynchronized input images (blue), and synchronized tracking with 7fps input (green). The vertical axis shows average joint position error in cm compared to the baseline result in the respective video frame. All tracking approaches use the same number of input images. As the actors motion becomes faster towards the end of the sequence, only our spatio-temporal approach is able to track the sequence correctly.

lation to create parameters for all frames of the sequence for the two discrete tracking approaches. Our approach has a slightly higher joint position error in the beginning of the sequence, where the motion of the actor is slow and aligning all frames to a single time instant is still a good approximation. However, as soon as the motion of the actor becomes faster, the discrete tracker's error increases until it fails to produce correct poses at around frame 1800 (*c.f.* supplementary video).

Discussion: Our approach shows that using unsynchronized cameras not only enables us to use lower frame rate cameras for tracking, but also increases the tracking quality for fast motion as our quantitative evaluation shows. Despite this simpler setup, by running the cameras purposefully out-of-sync, the continuous tracker reconstructs fast motion at much higher quality as Figure 8 shows. In practical situations, for example when capturing with camcorders, it will not be possible to control the sub-frame alignment of the camera shutters. Depending on the alignment the result will have more spatial accuracy (when nearly synchronized) or more temporal resolution (with unaligned input images).

As our method is using a simple local optimization approach, it may fail in complicated cases with many occlusions and few cameras. Although our approach is more reliable than the synchronized implementation in [23] in our experience, we may get stuck in a local minimum and not recover. Using more advanced global optimization schemes such as presented in [10], would enable us to detect these errors and recover. We also rely on the color of the actor being sufficiently different from the background in our error function, which could be improved upon by using more advanced color models for each Gaussians, such as color histograms. Despite these limitations, in most cases our algorithm successfully tracked even complex motions under severe occlusions.

To estimate a globally continuous function representing the motion parameters, we firstly construct local polynomials and then blended them using a partition of unity approach. This leads to a computationally efficient algorithm since the optimization of each local polynomials can be done independently. However, from a theoretical perspective, this approach is sub-optimal in the sense that the optimization does not take advantage of all available observed data (*i.e.* images). In the future, we will explore different possibilities of trading the computational complexity and the optimality of the parameter function in this context.

6. Conclusions

We have introduced a spatio-temporal approach to articulated motion tracking from unsynchronized multi-view video. Unlike previous approaches that rely on synchronized input video, our method makes use of the additional temporal resolution to successfully track fast moving actors with low frame-rate cameras. It also enables setting up simpler and cheaper capture setups, as there is no need anymore for hardware based synchronization and high frame rate cameras.

7. Acknowledgements

This work has been developed within the Max-Planck-Center for Visual Computing and Communication collaboration, and has been co-financed by the Intel Visual Computing Institute.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, 2010. 2

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

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

[4] 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

[5] 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

[6] R. Carceroni, F. Padua, M. Santos, and K. Kutulakos. Linear sequencetosequence alignment. In *CVPR*, 2004. 2, 5

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

[8] 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. 2

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

[10] 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. 2, 7

[11] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR*, 2009. 1, 2, 5

[12] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(2):328 –333, feb. 2006. 2

[13] C. Ionescu, F. Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *ICCV*, 2011. 2

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

[15] 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. 2

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[17] B. Meyer, T. Stich, M. Magnor, and M. Pollefeys. Subframe temporal alignment of non-stationary cameras. In *CVPR*, 2009. 2, 5

[18] 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, 2

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

[20] 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

[21] H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV*, volume 1, pages 784–800, 2002. 2

[22] 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, 2

[23] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *ICCV*, 2011. 1, 2, 3, 4, 5, 6, 7

[24] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. In *Adv. in Comput. Math.*, pages 389–396, 1995. 4, 5