Int J Comput Vis (2010) 87: 75–92
DOI 10.1007/s11263-008-0173-1

Optimization and Filtering for Human Motion Capture

A Multi-Layer Framework

Juergen Gall · Bodo Rosenhahn · Thomas Brox · Hans-Peter Seidel

Received: 11 January 2008 / Accepted: 21 August 2008 / Published online: 15 November 2008
© The Author(s) 2008. This article is published with open access at Springerlink.com

Abstract Local optimization and filtering have been widely applied to model-based 3D human motion capture. Global stochastic optimization has recently been proposed as promising alternative solution for tracking and initialization. In order to benefit from optimization and filtering, we introduce a multi-layer framework that combines stochastic optimization, filtering, and local optimization. While the first layer relies on interacting simulated annealing and some weak prior information on physical constraints, the second layer refines the estimates by filtering and local optimization such that the accuracy is increased and ambiguities are resolved over time without imposing restrictions on the dynamics. In our experimental evaluation, we demonstrate the significant improvements of the multi-layer framework and provide quantitative 3D pose tracking results for the complete `HumanEva-II` dataset. The paper further comprises a comparison of global stochastic optimization with particle filtering, annealed particle filtering, and local optimization.

Keywords Human motion capture · Stochastic optimization · Filtering · Tracking

J. Gall (✉) · B. Rosenhahn · H.-P. Seidel
Max-Planck-Institute for Computer Science,
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
e-mail: jgall@mpi-inf.mpg.de

B. Rosenhahn
e-mail: rosenhahn@mpi-inf.mpg.de

H.-P. Seidel
e-mail: hpseidel@mpi-inf.mpg.de

T. Brox
Department of Computer Science, University of Dresden,
01162 Dresden, Germany
e-mail: brox@inf.tu-dresden.de

1 Introduction

The 3D reconstruction of human motion from multi-view video sequences has applications in many areas including computer graphics, biomechanics, medicine, and sport science, see e.g. (Rosenhahn et al. 2008). Besides robustness, accuracy, and low computational cost, many applications require a general solution without imposing strong assumptions on the dynamics and the appearance of the human, i.e. neither motion patterns nor clothing are known a-priori. Nonetheless, the use of prior poses or motion patterns learned from a motion database has become very popular in order to achieve robust tracking also in difficult and ambiguous scenarios (Rosenhahn et al. 2007a; Sidenbladh et al. 2000; Urtasun and Fua 2004). In Agarwal and Triggs (2006) the pose is directly recovered from silhouettes by learning the mapping between silhouettes and markers. Gaussian process dynamical models (Moon and Pavlovic 2006; Urtasun et al. 2006) have been used for embedding motion in a low-dimensional latent space. Although these learning strategies allow for tracking even in monocular video sequences, they impose strong assumptions on the tracked motion. The restriction to a small subset of human motion patterns limits their application in practice. When, for example, the movement of a person with an artificial hip joint is measured using training data from persons with natural hip joints, the estimates are likely to be biased towards the movement of a person with natural hip joints, i.e., one eliminates exactly the information that is important for the medical application. Hence, in the present paper, we will focus on a tracking system that allows for robust and accurate tracking without relying on strong motion priors.

Another kind of prior knowledge frequently used in human tracking is a surface model with an underlying skeleton, see e.g. (Hogg 1983) or the survey (Moeslund et al. 2006).

These so-called model-based approaches estimate the position, rotation, and joint configuration (pose) of the human model for each frame, where the large number of degrees of freedom (DoF) results in a high-dimensional state space. Although the use of a model-based approach also limits the general applicability of the tracking framework, we assume here the existence of such a body model.

The strategies for model-based pose estimation can be classified into global optimization, filtering, and local optimization. All these strategies have some drawbacks. The main contribution of the present paper is therefore a multi-layer framework that employs the basic ideas of all three concepts.

1.1 Global Optimization

A stochastic global optimization approach, called interacting simulated annealing (ISA) (Gall et al. 2007b), has recently been proposed for human motion capture (Gall et al. 2007a). Since it searches for the globally best solution, it is also suitable for initialization of model-based approaches (Gall et al. 2007c). Its ability to recover from errors and its precise estimates satisfy the requirements for the first layer where robustness and accuracy are essential. However when the estimates are observed over time, some jitter is noticeable which is typical for stochastic approaches like ISA that sample from a distribution of interest. Variations between estimates of two frames might also occur, when the tracker recovers from an ambiguity in the previous frame. Moreover, while stochastic global optimization provides estimates close to the global optimum in reasonable time, the ratio between accuracy and computation cost is unsatisfactory when more precise estimates are required, as we will show.

1.2 Filtering/Smoothing

Filtering approaches estimate the unknown true state x_t from some noisy observations y_t, e.g. images. In general, the estimation is called prediction, filtering, or smoothing if observations before frame t, including t, or also after t are taken into account. The filtering problem is typically solved by Kalman filtering (Kalman 1960) or particle filtering (Doucet et al. 2001) where it is assumed that the underlying stochastic processes

$$x_{t+1} = f_t(x_t) + v_t, \tag{1}$$

$$y_t = h_t(x_t) + w_t \tag{2}$$

with noise v_t and w_t are known. Isard and Blake (1996) applied a particle filter to 2D tracking and extended it to a two-pass smoothing algorithm (Isard and Blake 1998). For 3D human motion capture, particle filters were combined with Markov chains, called Hybrid Monte Carlo filter (Choo and Fleet 2001), and graphical models, called nonparametric belief propagation (Lee and Nevatia 2006; Sigal et al. 2004). In Bregler (1997) a Kalman filter was used to model the human dynamics by multiple abstraction levels. Even though filtering approaches exploit temporal coherence, handle noise and are able to recover from errors, they are usually too imprecise for motion analysis in high dimensional spaces. Since accurate models for f_t and h_t are rarely available, the model's weakness is compensated by overestimating the noise vectors v_t and w_t at the expense of poor performance.

For this reason, some heuristics based on particle filters were developed to combine local optimization with filtering. Sminchisescu and Triggs (2003) propose covariance scaled sampling to guide the particles to the local maxima of a posterior distribution. To find the local maxima, the particles are broadly spread in the search space by inflating the covariance of the dynamic prior and refined by a local optimization with respect to the likelihood. The posterior is then modeled by a mixture of Gaussians where the means and covariance matrices are given by the detected local maxima and their Hessians. Smart particle filtering (Bray et al. 2007) combines a particle filter with stochastic meta descent (Schraudolph 1999) for local optimization. Since the optimization of the particles changes the approximated distribution, a correction factor is used to compensate for the additional set of particles. The correction factor, however, depends on the unknown distribution after prediction. Hence, a regularization (Doucet et al. 2001, Chap. 12), which introduces an error, is performed to estimate the continuous distribution from the finite set of particles before the optimization step. Particularly, the low number of particles prevents an accurate estimation of the correction factor. Deutscher et al. propose an annealed particle filter (Deutscher et al. 2000; Deutscher and Reid 2005) that follows the idea of annealing to guide the particles to the global maximum of the likelihood. To this end, the shape of the likelihood is gradually changed and the sampling is repeated. The approach does not perform annealing in the classical sense where the temperature is monotonically decreased, but relies on the fluctuating survival rate of the particles. Hence, the annealed particle filter is not suitable for global optimization and requires an additional technique for initialization like other approaches that combine local optimization with particle filtering. Although it has been shown that these heuristics work well for tracking hands or humans, there is no evidence that they converge to the optimal solution of the filtering problem stated in (1) and (2) in contrast to Kalman or particle filtering.

1.3 Local Optimization

Local optimization has been widely used for 3D human motion capture (Bregler and Malik 1998; Cheung et al. 2005; Gall et al. 2008; Gavrila and Davis 1996; Kakadiaris and Metaxas 1996; Kehl et al. 2005; Mundermann et al. 2007). It provides very accurate results provided that the state vector is initialized near the global optimum. Since it searches only for the locally best solution, it usually cannot recover from errors and requires an initialization. Without additional prior information, the tracking often fails in case of fast motions and ambiguities. The optimization for pose estimation has recently been coupled with level-set segmentation (Brox et al. 2005; Rosenhahn et al. 2006) and graph-cut segmentation (Bray et al. 2006) where the estimated pose serves as shape prior for segmentation. Even though the shape prior yields better segmentation results and can be applied more generally than background subtraction, it introduces a local term for energy minimization that depends on the previous estimate. Hence, these approaches are not able to recover from errors since a wrong estimate results in a wrong shape prior and a wrong segmentation for the next frame.

The idea of several layers has been used for tracking-by-detection approaches (Fossati et al. 2007; Ramanan et al. 2007) which rely on a learned template model. Since the detection is usually limited to canonical poses like lateral walking, the human poses are only detected on a subset of frames. A second step is therefore required to interpolate or track between the detected frames. While the tracking is usually done offline since the detected poses are used to learn a subject specific appearance model, our framework processes the image data online or with a very short delay.

1.4 Overview and Contribution

In this work, we propose a model-based approach for 3D human motion capture that meets important needs of motion analysis since it does not rely on prior knowledge of the dynamics. In order to increase the accuracy and resolve ambiguities over time without imposing restrictions on the dynamics, we introduce a multi-layer framework that combines global optimization, filtering, and local optimization. While the first layer relies on global stochastic optimization, the second layer refines the estimates by filtering and local optimization as outlined in Fig. 1.

For the first layer, the images are processed and silhouettes are extracted (Sect. 2). A recently developed stochastic global optimization technique, namely interacting simulated annealing, initializes the tracker and estimates the pose for each frame by minimizing an image-based energy function, which relies on silhouettes and color, as well as some weak prior on physical constraints (Sect. 3). Although the first layer provides a robust and relatively accurate estimate

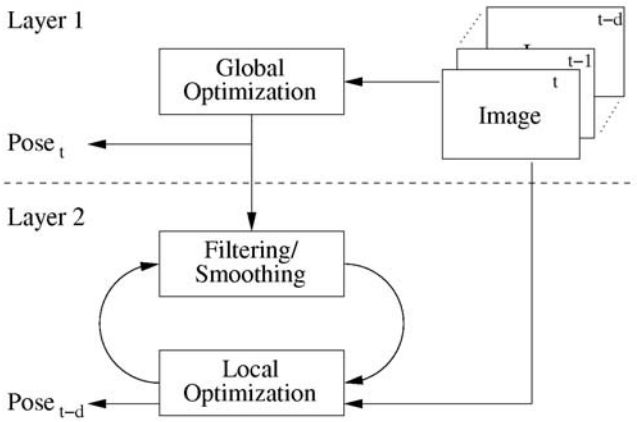


Fig. 1 A multi-layer framework for tracking. While the first layer based on global stochastic optimization provides robust and relatively accurate estimates, the second layer increases the accuracy and reduces jitter and potential bias from the first layer with a short delay d

of the human pose in the current frame, the estimate is still corrupted by noise due to sampling and the unsteady quality of the image features. Besides the missing temporal consistency, some bias might have been introduced by the weak prior.

The second layer refines the estimate with a short delay of $d \geq 0$ frames, where the estimate is filtered or smoothed (Sect. 4). Although the smoothing reduces the jitter from the stochastic global optimization by introducing temporal consistency, it improves only slightly the accuracy of the estimate. The latter is achieved by local optimization and segmentation where the smoothed estimate for frame $t - d$ serves as initial pose for optimization and as shape prior for the level-set segmentation (Sect. 5). The additional local segmentation improves the quality of the silhouettes of the first layer, which are obtained by global segmentation like background subtraction and often contain severe artifacts like shadows and holes. Since both segmentation and local optimization are initialized by good estimates from the first layer for each frame, an error accumulation due to the shape prior is prevented. We show that the second layer consisting of smoothing, local optimization, and local segmentation not only increases the accuracy, but also reduces jitter and potential bias from the first layer.

Indeed, our experimental evaluation in Sect. 6 demonstrates the improvements of the multi-layer framework in comparison to an increased number of iterations and samples for global optimization. It further comprises a quantitative error analysis using the `HumanEva-II` dataset (Sigal and Black 2006), where we also compare interacting simulated annealing with particle filtering, annealed particle filtering, and local optimization.

2 Image Processing

In our multi-layer framework, global and local optimization are applied to the same images, see Fig. 1. Hence, the images need to be processed once such that they are suitable for the appearance model used for global optimization (Sect. 3.3.2) and the level-set segmentation in the second layer (Sect. 5.1). Both for segmentation and the appearance model, good results are obtained with the CIELab color space that mimics the human perception of color differences. In order to reduce noise without smoothing over the edges that separate body parts and background, we apply the edge-enhancing diffusivity function (Brox et al. 2003)

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|^p + \varepsilon} \tag{3}$$

with $\varepsilon = 0.001$ and $p = 1.5$, where the smoothing is efficiently implemented by the AOS scheme (Weickert et al. 1998).

3 Global Optimization

The first layer of our tracking framework relies on interacting simulated annealing (ISA) (Gall et al. 2007b), which is a global stochastic optimization technique. Since we assume that a 3D skeletal model as shown in Fig. 2a is available, the pose can be represented by a vector x containing the position, orientation, and joint angles, where rotations are converted to the axis-angle representation. For each frame, the pose $\hat{x}$ is obtained by searching for the global minimum of an energy function $V \geq 0$, which is described in Sect. 3.3.

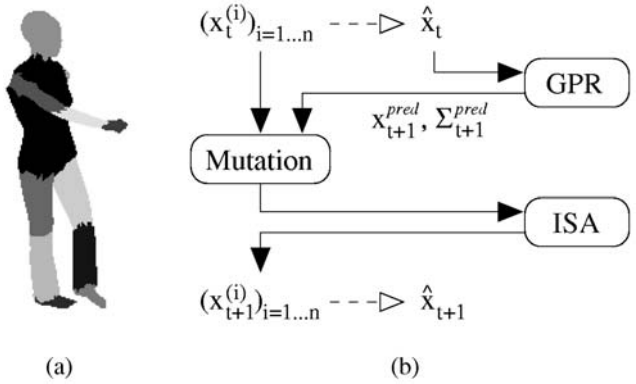


Fig. 2 *Left*: (**a**) The triangles of the human model encode the body parts. *Right*: (**b**) Outline of the first layer. While the particle set $(x_t^{(i)})_i$ represents the distribution of the solution, the mean $\hat{x}_t$ provides a single estimate for the pose. The pose for the next frame x_{t+1}^{pred} is predicted by Gaussian process regression (GPR), and an additional mutation operator spreads the particles in the search space. The pose is then estimated by stochastic optimization (ISA). The system is closed in the sense that any uncertainty that arises from the prediction and estimation is preserved in terms of Σ_{t+1}^{pred} and $(x_{t+1}^{(i)})_i$

Instead of searching for a single estimate $\hat{x}$, ISA approximates a distribution η_k whose mass concentrates in the region of global minima of the energy function V as k tends to infinity, see Fig. 3. This behavior is described by the following convergence theorem (Moral 2004) saying that for any $\varepsilon > 0$

$$\lim_{k\to\infty} \eta_k(V \geq \sup\{v \geq 0;\ V \geq v \text{ a.e.}\} + \varepsilon) = 0. \tag{4}$$

Similar to particle filters, where the posterior distribution is approximated by so-called particles, η_k needs to be approximated by n samples $x_k^{(i)}$ with weights $\pi^{(i)}$ since an analytical solution is usually not available. The approximate distribution

$$\eta_k^n := \sum_{i=1}^{n} \pi^{(i)} \delta_{x_k^{(i)}}, \tag{5}$$

where δ denotes the Dirac measure, converges to η_k as the number of particles increases (Moral 2004). A single estimate for the human pose from the set of particles is obtained by the mean $\hat{x} = \int \eta_k^n(x)dx$ where the mean of rotations is computed according to (Pennec and Ayache 1998). The details of ISA are discussed in Sect. 3.2.

During tracking the solution is represented by the set of particles $(x_t^{(i)})_i$ as outlined in Fig. 2b. Since the particles approximate a distribution, uncertainties from the pose estimation are propagated to the next frame making the estimation robust to ambiguities. An additional mutation operator between two frames spreads the particles in the search space where the predicted pose x_{t+1}^{pred} and its confidence Σ_{t+1}^{pred} are taken into account, see Sect. 3.1. The initial pose is also determined by ISA as described in Sect. 3.4.

3.1 Mutation

After estimating the pose $\hat{x}_t$, the particles $x_t^{(i)}$ congregate around the global optimum for frame t. Since this set is not well distributed for estimating the pose in the next frame, a mutation step spreads the particles in the search space. For this purpose, the pose is predicted from the previous estimates by a 3rd order autoregression, i.e. $x_{t+1}^{pred} = f(\hat{x}_{t:3})$ where $\hat{x}_{t:3} = (\hat{x}_t, \hat{x}_{t-1}, \hat{x}_{t-2})$ denotes the last three estimates. The function f can be learned during tracking from the history of estimates given by the equations

$$\hat{x}_{t-r+1} = f(\hat{x}_{t-r:3}) \quad \text{for } r = 1 \ldots R. \tag{6}$$

The regression is implemented by Gaussian processes (GP) (Williams and Rasmussen 1996) where the prediction is given by a Gaussian distribution with mean x_{t+1}^{pred} and covariance matrix Σ_{t+1}^{pred}. Since GP regression provides a predictive distribution and works well for a small set of training data, it meets the needs for the first layer.

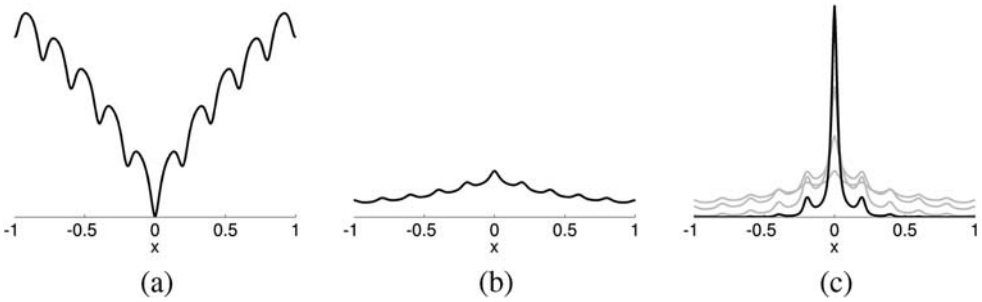


Fig. 3 *From left to right*: (**a**) Energy function V with global minimum at zero. (**b**) η_1. (**c**) The mass of η_k concentrates around the global minimum as k increases. For a limited number of iterations, η_k is multimodal

To simplify matters, we briefly summarize only the one-dimensional prediction by Gaussian processes where the set of training data is given by $\hat{x}_R = (\hat{x}_{t-1:3}, \ldots, \hat{x}_{t-R:3})^T$ and $f(\hat{x}_R) = (f(\hat{x}_{t-1:3}), \ldots, f(\hat{x}_{t-R:3}))^T$. The predictive distribution for the last three estimates $\hat{x}_{t:3}$ is obtained by the conditional Gaussian distribution $p(\hat{x}_{t+1}|\hat{x}_{t:3}, \hat{x}_R, f(\hat{x}_R))$ with mean and variance

$$x_{t+1}^{pred} = k(\hat{x}_{t:3}, \hat{x}_R)^T \mathbf{K}^{-1} f(\hat{x}_R), \tag{7}$$

$$(\sigma_{t+1}^{pred})^2 = k(\hat{x}_{t:3}, \hat{x}_{t:3}) - k(\hat{x}_{t:3}, \hat{x}_R)^T \mathbf{K}^{-1} k(\hat{x}_{t:3}, \hat{x}_R). \tag{8}$$

The covariance matrix for the training data $\mathbf{K}$ is modeled by the general covariance function

$$k(\hat{x}_{r:3}, \hat{x}_{s:3}) = a_0 \exp\left(-\frac{1}{2}\sum_{j=0}^{2} a_{j+1}(\hat{x}_{r-j} - \hat{x}_{s-j})^2\right) + \sum_{j=0}^{2} a_{j+4}\hat{x}_{r-j}\hat{x}_{s-j} + \sigma_{noise}^2 \delta_{rs}, \tag{9}$$

where the hyperparameters a_j and σ_{noise}^2 are learned offline[1] by minimizing the log likelihood as proposed in Williams and Rasmussen (1996). Due to computational efficiency, all parameters of the search space are assumed to be independent yielding a one-dimensional prediction for each degree of freedom.

Since the dynamics are learned online, the prediction adapts to the current motion but it also might be corrupted by tracking errors in the past. Hence, we shift only 40% of the particles according to x_{t+1}^{pred}, another 30% is kept as it is and 30% are mutated. The mutation is motivated by evolutionary algorithms where a larger variety among a population helps to recover from errors. We propose two human specific mutation operators as illustrated in Fig. 4. The first swaps two kinematic branches like the left and the right leg and helps to recover from ambiguous silhouettes which often occur when the legs are next to each other. The second is useful when

[1]The hyperparameters are learned from the sequences shown in rows 2–4 of Fig. 8 in Gall et al. (2008). The sequences differ from the test sequences in motion, frame rate, and subject.

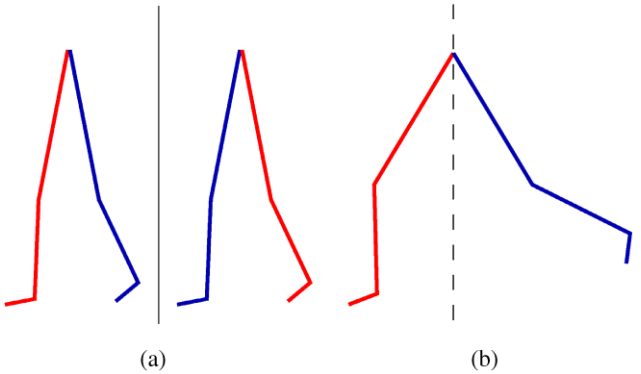


Fig. 4 Two mutation operators. *From left to right*: (**a**) The left branch (*red*) and the right branch (*blue*) are swapped. (**b**) The left branch (*red*) is reconstructed from the right branch (*blue*) by mirroring the first joint

only one of two legs or arms is well estimated due to occlusions. In order to reconstruct its counterpart, we imitate the behavior of humans to use their arms or legs to balance. For this purpose, the first joint of the kinematic branch is mirrored while the other joint angles remain unchanged. Even though the mutated particles will be mostly rejected after the first iterations of the optimization, they support the tracker in recovering from errors. Finally, all particles are propagated by a zero-mean Gaussian distribution with covariance matrix proportional to Σ_{t+1}^{pred}.

The prediction by Gaussian process regression has two advantages. When the movement is fast or the frame rate is low, x_{t+1}^{pred} guides some particles towards the next potential pose such that less iterations are required for optimization as illustrated in Fig. 5. More important, however, is Σ_{t+1}^{pred} which spreads the particles in the search space before optimization. Without the prediction, it would be necessary to set Σ_{t+1}^{pred} manually but the optimal values depend on the motion and the frame rate. GPR provides this information where the variance becomes larger for fast motions or a reduced frame rate. Note that we do not require a first order Markov process for the transitions as it is usually assumed for filtering approaches. In our experiments, we have observed that a 3rd order autoregression performs well for human motion whereas models with higher order improve only marginally the prediction.

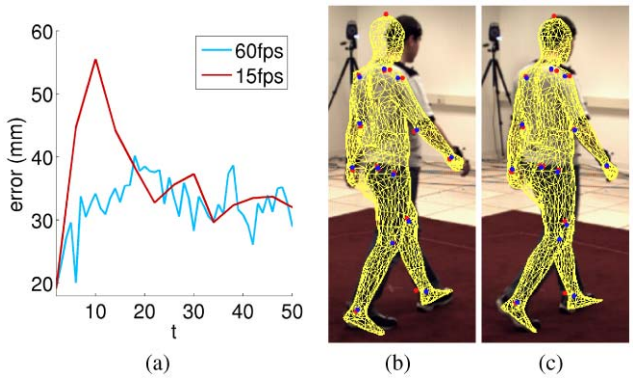


Fig. 5 Impact of learning the motion model online. *From left to right*: (**a**) To simulate the effect of a fast movement, only every 4th frame is used, i.e., the frame rate of the camera is reduced from 60 fps to 15 fps. Since the dynamics are learned online, it takes some frames until good estimates for x_{t+1}^{pred} and Σ_{t+1}^{pred} are obtained. When the number of iterations for ISA remains unchanged, the error increases for the first frames. After the motion model is learned, the error is comparable to the 60 Hz sequence. (**b**) Estimated pose for frame 3 of the 15 Hz sequence *(frame 10)*. (**c**) After 5 frames at 15 Hz *(frame 18)*, the motion model is learned and the pose is well estimated

Fig. 6 The set of particles converges to the global minimum. The weighted particles are shown for iterations $k = 5$, 10, 20, and 35, where particles with higher weights are brighter

3.2 ISA

The optimization consists of a weighting, a selection, and a mutation step that are iterated several times. For each iteration k, the distribution η_k is approximated by the set of particles, see Fig. 6. The particles are initialized by the mutation operator from Sect. 3.1 as illustrated in Fig. 2.

Weighting Assuming that a set of particles $(x_k^{(i)})_{i=1\ldots n}$ exists, each particle is weighted by the Boltzmann-Gibbs measure

$$\pi^{(i)} = \exp(-\beta_k V(x_k^{(i)})), \tag{10}$$

where $\beta_k = (k+1)^b$ with $b = 0.7$ is an annealing scheme that increases monotonically. After normalizing the weights such that $\sum_i \pi^{(i)} = 1$, the weight indicates the probability that a particle is selected for the next step.

Selection In a first stage, particles are accepted with probability $\pi^{(i)} / \max_l \pi^{(l)}$, i.e. the particle with the highest weight is always accepted. Since after this first stage only m particles are selected, additional $n - m$ particles are drawn in a second stage, replacing those from the old set. This is efficiently done by stratified resampling (Douc et al. 2005) using the normalized weights $\pi^{(i)}$. Due to the selection operation, similar particles with high weights are contained several times in the new set whereas particles with low weights might disappear completely.

Mutation In order to explore the search space, the particles are spread out according to a Gaussian K_k whose covariance matrix is the sampling covariance matrix

$$\Sigma_k = \frac{\alpha_\Sigma}{n-1}\left(\rho I + \sum_{i=1}^{n}(x_k^{(i)} - \mu_k)(x_k^{(i)} - \mu_k)^T\right) \tag{11}$$

scaled by $\alpha_\Sigma = 0.4$, where μ_k is the average, I the identity matrix, and ρ a small positive constant that ensures that the covariance does not become singular. The computational cost is reduced by using a sparse matrix that takes only correlations of joints into account that belong to the same skeleton branch. In general, the Gaussian distribution can be replaced by any distribution that satisfies the mixing condition to ensure the convergence on a bounded search space; see (Gall et al. 2007b) or (Moral 2004).

For a comparison of different annealing schemes and parameter settings for ISA, we refer to Gall et al. (2007b). The optimal number of iterations and particles is a trade-off between accuracy and computation cost, which is discussed in Sect. 6.

3.3 Energy

As energy function for global optimization, we use

$$V(x) = \nu V_{silh}(x) + \tau V_{app}(x) + \upsilon V_{phys}(x), \tag{12}$$

where the parameters ν, τ, and υ control the influence of the three terms, namely silhouettes, appearance, and physical constraints that are explained in Sects. 3.3.1, 3.3.2, and 3.3.3, respectively. The impact of the appearance term has been evaluated in Gall et al. (2007a). Throughout this paper, we use the recommended parameters $\nu = 2$, $\tau = 40$, and $\upsilon = 2$.

3.3.1 Silhouettes

In order to model an error function between a particle x and a silhouette image I_v extracted by background subtraction, a template image $T_v(x)$ is generated by projecting the

Fig. 7 *From left to right*: (**a**) Template image $T_v(x)$. (**b**) Silhouette image I_v. (**c**) Smoothed a-channel. (**d**) Smoothed b-channel

surface of the human model that is translated, rotated, and deformed according to the particle as shown in Fig. 7a. The inconsistent areas between the silhouette and the template are then measured for each view v by

$$V_v(x) = \frac{1}{2|T_v^0(x)|} \sum_{p \in T_v^0(x)} |T_v(x,p) - I_v(p)| + \frac{1}{2|I_v^0|} \sum_{p \in I_v^0} |I_v(p) - T_v(x,p)|, \tag{13}$$

where $I_v(p)$ and $T_v(x,p)$ are the pixel values for a pixel p and the sets of pixels inside the silhouettes are denoted by I_v^0 and $T_v^0(x)$. Since pixels that are far away from the silhouette should be penalized more severely, a Chamfer distance transform (Borgefors 1986) is previously applied to I_v as shown in Fig. 7b. In the optimal case, the Chamfer distance transform is also applied to the template $T_v(x)$, but this would be very expansive since the transform needs to be computed for each particle. Hence, we use only a constant value where pixels inside the silhouette are set to 0, as it is the case for the distance transform, and pixels outside the silhouette have a constant 'distance' to compensate for the differences between the error of the first and the second term of (13). In our experiments, we have found that a value of 8 is a proper compensation factor. The energy term V_{silh} is finally defined as the average error of all views.

3.3.2 Appearance

To obtain an appearance model that is robust to 3D rotations, we combine the pixel information from all views to model the statistics of different body parts rather than their separate projections to the images. Since the L-channel of the CIELab color space is very sensitive to illumination changes, we use only the a- and b-channel, see Fig. 7. Furthermore, we assume the image channels u_c to be uncorrelated for efficiency reasons. Hence, the joint probability density function for a body part s can be written as

$$p_s(u) = \prod_c p_{s,c}(u_c). \tag{14}$$

Instead of assuming a certain family of distribution functions, we approximate the probabilities $p_{s,c}$ in a more general manner by normalized histograms $H^{(s,c)}$ where we fixed the number of bins to $K = 64$.

In order to measure deviations of the appearance of a particle x from the appearance model given by $H^{(s,c)}$, the particle's appearance $\tilde{H}^{(s,c)}(x)$ is estimated by sampling from all views. For this purpose, the triangles of the human model are used to encode the body parts of the projected surface as shown in Fig. 2a. Hence, a pixel p that belongs to a body part s contributes for each channel u_c to the histogram $\tilde{H}^{(s,c)}(x)$. For histogram comparison, we choose the Bhattacharya distance since it is also stable for empty bins in contrast to χ^2-statistics or Kullback–Leibler divergence (Puzicha et al. 1999). The total deviation is then measured according to (14) by

$$V_{app}(x) = \sum_s \frac{w_s}{C} \sum_{c=1}^{C} \left(1 - \sum_{k=1}^{K} \sqrt{h_k^{(s,c)} \tilde{h}_k^{(s,c)}(x)}\right), \tag{15}$$

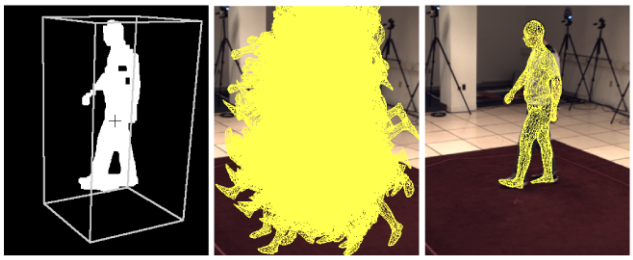
where the weights w_s reflect the size of the body parts and are determined during initialization, see Sect. 3.4. In general, the appearance model needs to be updated during tracking. However when the lighting conditions are controlled as it is the case for the `HumanEva-II` dataset, an update is not necessary.

3.3.3 Physical Constraints

Since human motion is subject to physical restrictions like anatomical constraints and self-intersections, the search can be focused on poses with higher probabilities by adding a soft constraint to the energy function. For this purpose, the probability of a skeleton configuration p_{pose} is estimated from a set of training samples y_l taken from the CMU motion database (CMU 2007). Since self-intersections between the head, the upper body, and the lower body rarely occur, the sample size L can be reduced by regarding the probabilities for the three body parts, denoted by p_{pose}^{head}, p_{pose}^{upper}, and p_{pose}^{lower}, as uncorrelated. The probability for a body part is approximated by a Parzen-Rosenblatt estimator with a Gaussian kernel K:

$$p_{pose}(x) = \frac{1}{Lh^d} \sum_l K\left(\frac{x - y_l}{h}\right), \tag{16}$$

where the d-dimensional vectors x and y_l contain only the joint angles for the body part. The bandwidth h is given



(a) (b) (c)

Fig. 8 Initialization. *From left to right*: (**a**) The search space is bounded by a cube. (**b**) The initial set of particles is randomly distributed around the center of the cube. (**c**) The pose is correctly initialized after 35 iterations. Intermediate steps are shown in Fig. 6

by the maximum second nearest neighbor distance between all training samples. Finally, we used less than 200 samples from different motions for modeling the physical constraints by

$$V_{phys}(x) = -\frac{1}{3}\ln(p_{pose}^{head}(x)\,p_{pose}^{upper}(x)\,p_{pose}^{lower}(x)). \tag{17}$$

Although the term V_{phys} is only a weak prior, it might still introduce some bias that is reduced by the second layer.

3.4 Initialization

For finding the initial pose, ISA searches for the global minimum of the energy function defined in (12) where only the terms V_{silh} and V_{phys} are used since the appearance of the model is unknown a priori. To this end, the search space is bounded by a cube that is determined by the silhouettes, where the intersections of the projection rays of the silhouettes' bounding boxes are the corners of the cube (Gall et al. 2007c). The particles are then randomly distributed around the center of the cube and optimized by ISA, see Fig. 8. Finally, the pose is refined by local optimization as discussed in Sect. 5.2. After the pose $\hat{x}_0$ is estimated for the first frame, the histograms $H^{(s,c)}$ are generated by sampling from the images as described in Sect. 3.3.2. During sampling, the range of each feature channel is also determined and divided into uniform bins. Furthermore, the weights w_s in (15) are given by the sample size for each body part s after normalizing such that $\sum_s w_s = 1$.

4 Smoothing

Using the noisy mean estimates $\hat{x}_t$ from global optimization as observations instead of images, the filtering problem specified by (1) and (2) is simplified such that h_t becomes the identity map. In addition, for considering the solutions of many frames for smoothing and not only a single one, we formulate the filtering as a regression problem.

As outlined in Fig. 1, the second layer refines the estimates $\hat{x}_t$ from global optimization with a short delay of $d \geq 0$ frames by means of local optimization, as described later in Sect. 5. This yields more precise estimates x_t. We propose to couple regression and local optimization. Having R estimates

$$x_{t-R}, \ldots, x_{t-d-1}, \hat{x}_{t-d}, \ldots, \hat{x}_t, \tag{18}$$

we seek the function f that provides a smoothed version for frame $t-d$, i.e. $x_{t-d}^{smooth} = f(t-d)$. Since the refined values x_t should have more impact in the regression than the values $\hat{x}_t$, we add a binary indicator variable i_t as additional dimension to the input space. $i_t = 1$ indicates that the estimate has been already refined. The regressor $f(t, i_t)$ is then learned from the data

$$x_{t-r} = f(t-r, 1) \quad \text{for } r = R \ldots d-1 \tag{19}$$

$$\hat{x}_{t-r} = f(t-r, 0) \quad \text{for } r = d \ldots 0. \tag{20}$$

Similar to the prediction in Sect. 3.1, we apply Gaussian process regression. Let $\mathbf{t} := (t, i_t)$ and $\mathbf{t}_R := (\mathbf{t}-\mathbf{R}, \ldots, \mathbf{t})^T$. The smoothed estimate is then given by the mean

$$x_{t-d}^{smooth} = k((t-d, 1), \mathbf{t}_R)^T \mathbf{K}^{-1} f(\mathbf{t}_R), \tag{21}$$

where the covariance matrix $\mathbf{K}$ is modeled by

$$\begin{aligned} &k(\mathbf{t}-\mathbf{r}, \mathbf{t}-\mathbf{s}) \\ &\quad = a_0 \exp\left(-\frac{1}{2}(a_1(r-s)^2 + a_2(i_{t-r} - i_{t-s})^2)\right) \\ &\qquad + \sigma_{noise}^2 \delta_{rs}. \end{aligned} \tag{22}$$

The hyperparameters are learned offline as explained in Sect. 3.1. Since the correlation depends only on the temporal distance but not on the current value of t, $\mathbf{K}^{-1}$ needs to be calculated only once for a fixed number of training data R. Basically the regression comes down to linear filtering with an asymmetric filter mask and the weights being learned from training data. Figure 9 shows the impact of d where we use $R = 10 + d$.

In general, a Kalman or particle filter could also be used for smoothing. However, the parameters need to be learned as well and we have not observed a significant improvement when the smoothing is performed with only a short delay.

5 Local Optimization

After smoothing, the accuracy of the estimated pose is increased by local optimization. Since the silhouettes from background subtraction often contain severe artifacts like shadows and holes, we improve the quality of the silhouettes by local segmentation before optimizing the pose, see

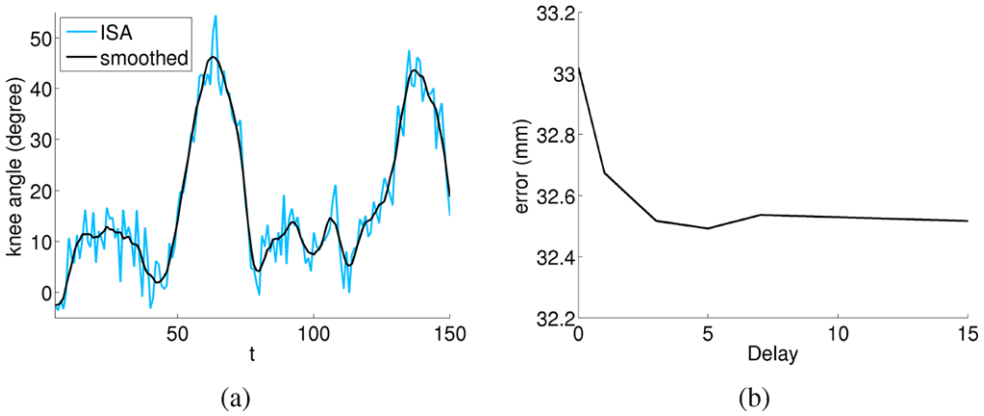


Fig. 9 Impact of smoothing. *From left to right*: (**a**) The smoothing reduces the jitter from global stochastic optimization. (**b**) The absolute tracking error of the second layer with respect to the introduced delay d (Frames 2–821 of sequence S4). The best result is achieved with a delay of only 5 frames. This corresponds to a delay of 83 ms for a sequence with 60 fps. For $d = 0$, the estimates are filtered without delay

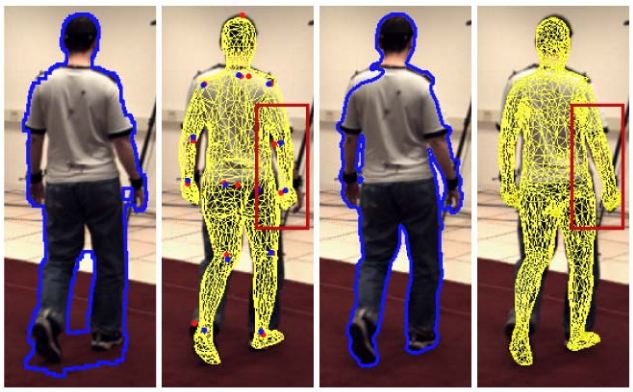


Fig. 10 *From left to right*: (**a**) Silhouette from background subtraction. (**b**) Estimate from global optimization. (**c**) Silhouette from level-set segmentation. (**d**) Improved estimate by local optimization. The right and left arms are better estimated

Fig. 10. The smoothed pose x_{t-d}^{smooth} serves both as shape prior for the segmentation and as initial estimate for local optimization.

5.1 Local Segmentation

The silhouette of the human is extracted by a level-set segmentation that divides the image into fore- and background where the contour is given by the zero-line of a level-set function Φ. As proposed in Rosenhahn et al. (2007b), the level-set function Φ is the minimum of the energy functional

$$E(\Phi) = -\int_\Omega H(\Phi)\ln p_1 + (1-H(\Phi))\ln p_2 dx + \vartheta\int_\Omega |\nabla H(\Phi)| dx + \lambda\int_\Omega (\Phi-\Phi_0))^2 dx, \quad (23)$$

where H is a regularized version of the Heaviside step function. The probability densities of the fore- and background, p_1 and p_2, are modeled by local Gaussian densities using the color channels L, a, and b that are assumed to be independent as in (14). While the first term maximizes the likelihood, the second term, weighted by the fixed parameter $\vartheta = 2$, regulates the smoothness of the contour. The last term penalizes deviations from the projected surface of the smoothed pose x_{t-d}^{smooth} given as level-set function Φ_0, where the influence of the shape prior is controlled by the parameter $\lambda = 0.08$. For minimizing (23), local optimization is performed with gradient

$$\partial_k \Phi = H'(\Phi)\left(\log\frac{p_1}{p_2} + \vartheta\,\mathrm{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right) + 2\lambda(\Phi_0 - \Phi) \quad (24)$$

and Φ_0 as initial estimate.

5.2 Pose Estimation

The pose x_{t-d}^{smooth} is finally refined by an iterated closest point (ICP) approach. To this end, 2D-2D correspondences between the zero-level of Φ and $\Phi_0(x_{t-d}^{smooth})$ are established by a closest point algorithm (Zhang 1994). Since the points on the contour of the projected surface of x_{t-d}^{smooth} relate to 3D vertices of the mesh, 3D-2D correspondences between the model and the image can be derived. According to ICP, the pose estimation is performed iteratively where the set of correspondences is updated after each optimization until the pose converges to a local minimum.

For estimating the pose, we seek for the relative transformation that minimizes the error of given 3D-2D correspondences denoted by pairs (X_i, x_i) of homogeneous coordinates. A suitable representation for local optimization are twists $\theta\hat{\xi}$ (Bregler et al. 2004) that express 3D rigid motions as $M = \exp(\theta\hat{\xi})$. A joint j is modeled as zero-pitch screw around a given axis, i.e., the joint motion depends only on the rotation angle θ_j. Hence, a transformation of a point X_i on the limb k_i influenced by n_{k_i} joints is given by

$$X_i' = M(\theta\hat{\xi})M(\theta_{\iota_{k_i}(1)})\ldots M(\theta_{\iota_{k_i}(n_{k_i})})X_i, \quad (25)$$

where the mapping ι_{k_i} represents the order of the joints in the kinematic chain. Since each 2D point x_i defines a projection ray that can be represented as Plücker line $L_i = (n_i, m_i)$

(Stolfi 1991), the error of a pair (X_i', x_i) is given by the norm of the perpendicular vector between the line L_i and the point X_i'

$$\|\Pi(X_i') \times n_i - m_i\|_2, \tag{26}$$

where Π denotes the projection from homogeneous coordinates to non-homogeneous coordinates. Using the Taylor approximation $\exp(\theta\hat{\xi}) \approx I + \theta\hat{\xi}$, where I denotes the identity matrix, (25) can be linearized. Hence, the sought transformation is obtained by solving the linear least squares problem

$$\frac{1}{2}\sum_i \left\| \Pi\left(\left(I + \theta\hat{\xi} + \sum_j^{n_i} \theta_{t_{k_i}(j)}\hat{\xi}_j\right) X_i\right) \times n_i - m_i \right\|_2^2, \tag{27}$$

i.e. by solving a system of linear equations.

In order to penalize strong deviations from x_{t-d}^{smooth} and to avoid an underdetermined system, we extend the linear system by an additional equation

$$\alpha\theta_j = \alpha(\theta_j^{smooth} - \tilde{\theta}_j) \tag{28}$$

for each joint j, where $\tilde{\theta}_j$ is the previously estimated absolute joint angle. The parameter α is set relative to the number of correspondences to achieve a constant weighting for each frame. In practice, we use $\alpha = 0.2 \cdot |\{(X_i, x_i)_i\}|$. Since the local optimization provides only a relative transformation, the refined pose x_{t-d} is obtained by applying the relative transformation to the previously estimated pose. We remark that the particular choice of the parameters for local segmentation and optimization influences only marginally the results of the second layer. The values therefore remain fixed in our experiments.

6 Experiments

For an experimental evaluation of the proposed multi-layer framework, we use the `HumanEva-II` dataset (Sigal and Black 2006) that contains two sequences that were captured by 4 calibrated cameras with resolution of 656×490 pixels and 60 fps. The ground truth has been obtained by a marker-based motion capture system that was synchronized with the cameras. The sequences show two different subjects S2 and S4 performing the motions walking, jogging, and balancing. We use the 3D surface mesh model that is available for subject S4 and does not contain the clothing. Both sequences S2 and S4 are tracked with this model although the mesh model does not fit subject S2 as shown in Fig. 19. Furthermore, we reduced the number of triangles to 5000 and added a skeleton with 28 degrees of freedom to the mesh. Since not all 20 points of the 3D pose from the marker-based system relate to joints of our mesh, we have used the first frame of each sequence to register the 3D markers of the ground-truth to our mesh. In Fig. 12, the registered markers are shown by red dots and the joint locations by blue dots. For computing the 2D and 3D error, we take the joint locations of the model, if they are available. Otherwise we use the registered markers. Since the joint locations of subject S4 do not fit subject S2, we have used only the registered markers for S2. In order to register the 3D markers as accurately as possible to the model, we have manually segmented the first frame and estimated the initial pose as described in Sect. 3.4. We remark that not only the tracking and initialization contribute to the overall error, but also the registration and the marker-based system introduce some errors. Hence, the reported errors should be regarded only as upper bounds that allow comparison of different approaches. The experiments are split into two sections. While Sect. 6.1 compares filtering approaches to optimization approaches, Sect. 6.2 demonstrates the performance of the proposed multi-layer framework.

6.1 Optimization vs. Filtering

We have compared interacting simulated annealing (ISA) to local optimization (ICP), a standard particle filter (PF) (Doucet et al. 2001), a variant of the smart particle filter (PFICP) (Bray et al. 2007), and the annealed particle filter (APF) (Deutscher and Reid 2005). The comparison is performed on the first 820 frames of sequence S4 using the absolute 3D error as measurement (Sigal and Black 2006). Since the ground truth is corrupted for the frames 298–335, these frames are neglected in the error analysis. For local optimization, we apply the iterative closest point approach described in Sect. 5.2 to the silhouettes obtained by background subtraction, where the prior on physical constraints (16) is integrated according to Brox et al. (2006). ISA, PF, and APF use the same energy model defined in Sect. 3.3. For the particle filter, we employ the weighting function (10) with $\beta_t = 1$. This is similar to the assumption that the likelihood is proportional to a product of normal densities. The particles are predicted as described in Sect. 3.1 without using the mutation operator since it is not supported by a filtering framework, i.e., 50% of the particles are shifted according to the predicted mean and the remaining 50% are directly selected. While ISA and APF are executed with 250 particles and 15 iterations, which are called layers for APF, we set the number of particles to 3750 for the particle filter to obtain the same computational cost. Though the smart particle filter as proposed in Bray et al. (2007) uses stochastic meta descent (SMD) (Schraudolph 1999) for local optimization, any local optimization like ICP can be used in principle. Since our ICP implementation is slower than SMD, we use 16 particles for PFICP to achieve the same computation time as PF according to Bray et al. (2007).

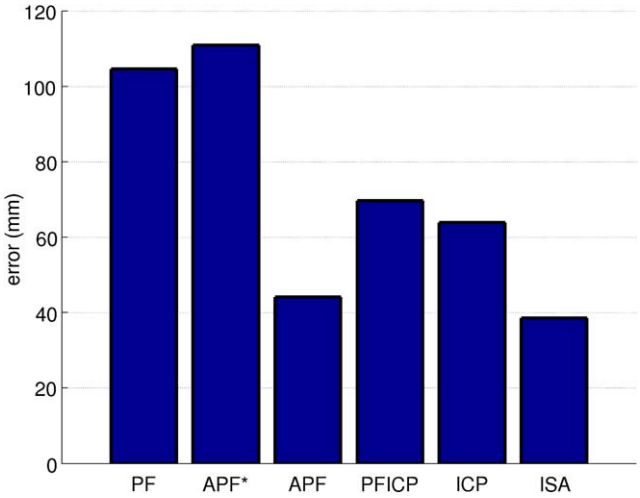
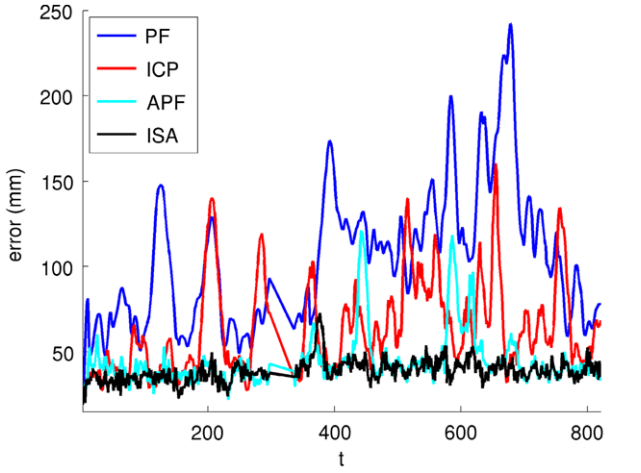


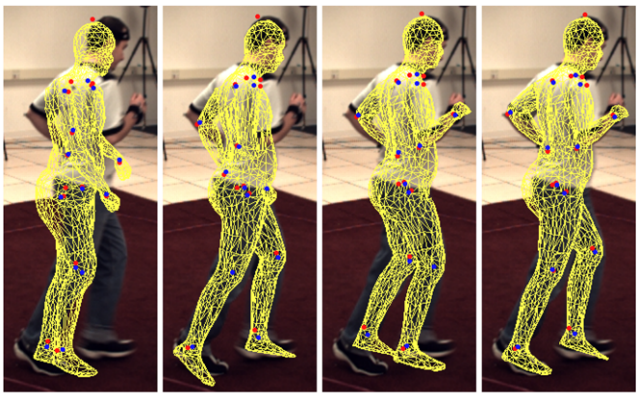
Fig. 11 Comparison between filtering and optimization approaches. Global stochastic optimization (ISA) provides the best estimates whereas the standard particle filter (PF) and local optimization (ICP) perform poorly. The annealed particle filter (APF) performs better than a combination of particle filtering with local optimization (PFICP) provided that the parameter for adaptive diffusion is well chosen. Otherwise, the error for APF becomes very large. The detailed errors with standard deviations are listed in Table 1

Since neither PF, APF, PFICP, nor ICP are suitable for initialization, the initial pose is provided by ISA.

The errors are plotted in Figs. 11 and 12. It shows that the global stochastic optimization approach clearly outperforms the local optimization and the particle filter. While ICP gets stuck in local minima, the estimates of PF are imprecise. The annealed particle filter performs better than the standard particle filter but it still produces two severe errors. This is reflected in the standard deviation for APF given in Table 1, which is large in comparison to ISA that performs very well for the entire sequence. The result that APF performs better than PF seems to contradict the comparison in Balan et al. (2005) where only slightly better results were obtained by APF. The outcome of APF, however, depends strongly on the parameter for adaptive diffusion (Deutscher and Reid 2005) which was not implemented in the previous comparison. The errors for two different settings, namely 0.4 (APF*) and 0.2 (APF), are plotted in Fig. 11. PFICP does not necessary improve ICP where the best result has been achieved with a very large window size for estimating the correction factor. Approaches like PFICP are in general relatively inefficient since the additional optimization step limits the number of particles such that a good approximation of a distribution is infeasible. Furthermore, a lot of computation time is wasted when the particles migrate to the same local minimum.

The performance of APF and ISA on a very fast sequence has been evaluated by reducing the frame rate from 60 fps to 15 fps. For the comparison shown in Fig. 13, the parameters for both algorithms are unchanged. While ISA performs





(a) PF (b) ICP (c) APF (d) ISA

Fig. 12 *Top*: Absolute 3D errors for frames 2–821 of sequence S4. While the estimates of the particle filter (PF) are imprecise, local optimization (ICP) gets stuck in local minima. The annealed particle filter (APF) contains two severe errors (>100 mm) around frames 440 and 590 yielding a large standard deviation, see Table 1. Global stochastic optimization (ISA) performs very well for the entire sequence. *Bottom*: Estimates for frame 580 by PF, ICP, APF, and ISA (*from left to right*). ICP fails to track the right arm and the legs are disarranged by the APF

Table 1 Averages and standard deviations of the absolute tracking error for frames 2–821 of sequence S4. ISA shows clearly the best results where the standard deviation is significantly lower than for APF

Error (mm)	PF	ICP	PFICP	APF	ISA
avg	104.61	63.86	69.70	44.15	38.58
std dev	40.77	27.07	24.75	15.39	6.54

very well for 60 Hz and 15 Hz, the error for APF increases by more than 30% when the speed is quadrupled. It might be that the result of APF can be improved by optimizing the parameter for adaptive diffusion on 15 Hz but it is clear that the faster the motion is the more important global optimization becomes.

Although the optimal numbers of particles and iterations for ISA are trade-offs between accuracy and computation

cost, Fig. 14 shows that large numbers of iterations and particles improve the estimates only marginally. Indeed, the error drops until 200 particles and 15 iterations, however after 30 iterations the absolute error is still 36.75 mm. For comparison, an error of 38.58 mm is obtained by 15 iterations. This indicates that ISA provides estimates near the global optimum in reasonable time, but when more precise estimates are required the ratio between accuracy and computation cost is unsatisfactory.

6.2 Multi-layer

For evaluating the performance of the proposed multi-layer framework, the absolute 3D tracking errors are measured for the entire sequence S4 that consists of 1257 frames. Figure 15 shows that the second layer increases the accuracy of the estimates from the first layer, where 250 particles and 15 iterations are used for ISA and the second layer refines the estimates with a delay of 5 frames. In particular, the largest error around frame 380 is significantly reduced by the second layer. This is reflected by the results given in Table 2, where the average error is reduced by 15.9% and the standard deviation by 22.4%. The second layer clearly provides more precise estimates, which cannot be achieved by an increased number of particles and iterations in reasonable time; see Fig. 14. Our current implementation requires 76 seconds per frame for the first layer and 48 seconds per frame for the second layer on a standard computer whereas ISA with 30 iterations would require 152 seconds per frame.

The errors and quantiles for individual joints are provided in Fig. 17. The quantiles show that most joints, particularly the knees, are very well estimated. It also reveals that the limb extremities, namely wrists and ankles, are more difficult to track since hands and feet are relatively small body parts. The lower quantiles indicate the registration errors of the joint positions, particularly of the ankles. Since the distances between the upper and lower quantiles for the wrists and ankles are similar, the larger error of the ankles might be explained by the registration error.

We have also evaluated the impact of coupling local optimization and smoothing for the second layer, which performs better than each of these steps alone. This is shown in Fig. 16. Tables 2 and 3 reveal that the accuracy is primarily increased by local optimization whereas smoothing reduces the jitter, as indicated by the decreased standard deviation. The best results for the second layer were achieved with a short delay of 5 frames as plotted in Fig. 9. Even

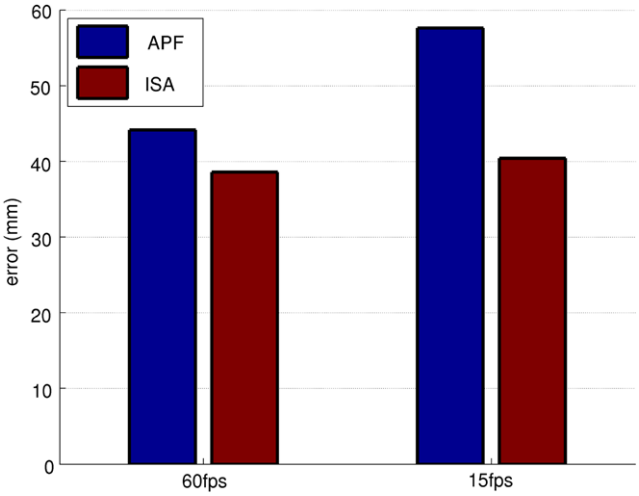


Fig. 13 The effect of a very fast movement is simulated by using only every 4th frame of sequence S4 (frames 2–821). This corresponds to a walking and running sequence recorded with 15 fps. While the error increases slightly by 4.68% to 40.39 mm for ISA, the error for APF rises to 57.63 mm by 30.5%

Table 2 Averages and standard deviations of the absolute tracking error for the complete sequence S4 (frames 2–1258). The error of the first layer using only global optimization is significantly reduced by the second layer. Clearly, a coupling of smoothing and local optimization provides more precise results than each of them alone

Error (mm)	Layer1	L1 + Smooth	L1 + LocOpt	L1 + Layer2
avg	38.07	35.58	33.23	32.01
std dev	5.84	5.09	5.08	4.53

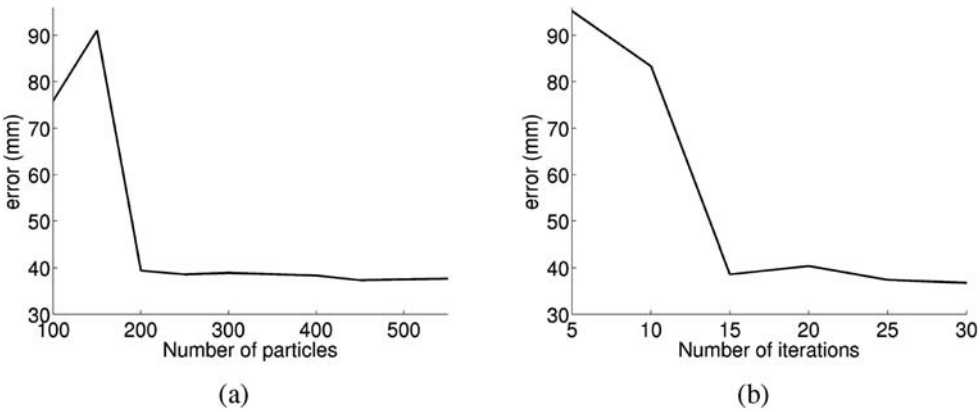


Fig. 14 Absolute tracking error of global optimization for frames 2–821 of sequence S4. Large numbers of iterations and particles improve the estimates only marginally. *From left to right*: (**a**) Error with respect to the number of particles using 15 iterations. (**b**) Error with respect to the number of iterations using 250 particles

Table 3 Averages and standard deviations of the absolute tracking error for the complete sequence S2 (frames 1–1202)

Error (mm)	Layer1	L1 + Smooth	L1 + LocOpt	L1 + Layer2
avg	43.82	41.44	39.20	37.53
std dev	10.65	9.67	10.05	9.00

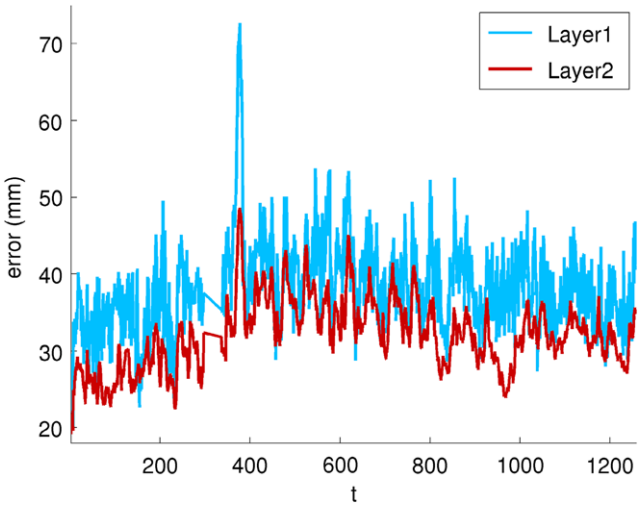


Fig. 15 Absolute tracking error for the sequence S4 (frames 2–1258). The second layer reduces jitter and increases the accuracy of the estimates from the first layer. In particular, the largest error around frame 380 is significantly reduced by the second layer

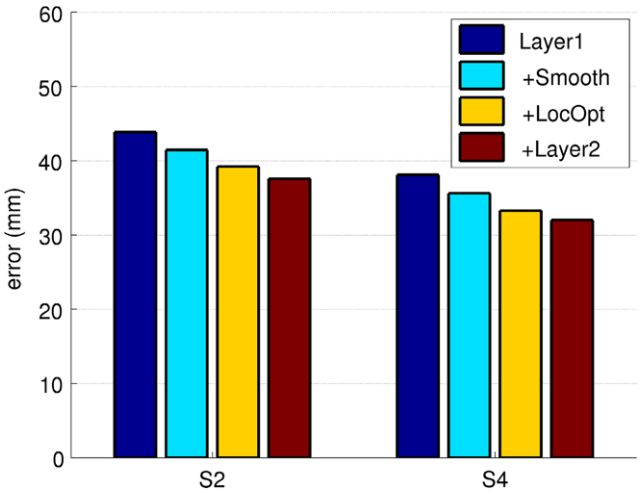


Fig. 16 A comparison of the average errors for the complete sequences S2 and S4 shows the improvements of our multi-layer framework. The detailed errors with standard deviations are given in Tables 2 and 3

without delay, the error is slightly reduced compared to applying only local optimization. The computation times are listed in Table 4. For convenience, we also provide the error of the second layer in Table 5 when a particle filter approach is used as first layer.

Table 4 Overall computation time on a standard PC for a frame with 4 images

	Layer1	L1 + Smooth	L1 + LocOpt	L1 + Layer2
sec/frame	76	76	124	124

Table 5 Averages and standard deviations of the absolute tracking error for frames 2–821 of sequence S4. The second layer (L2) improves the results for all sampling approaches. The results without the second layer are given in Table 1

Error (mm)	PF + L2	PFICP + L2	APF + L2	ISA + L2
avg	82.70	58.38	37.26	32.49
std dev	43.77	25.32	14.67	5.21

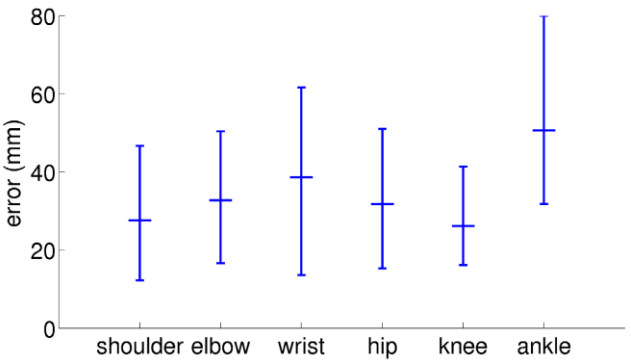


Fig. 17 Average errors and 0.025-quantiles for individual joints obtained by the multi-layer framework on the entire sequence S4. While the knees are very well estimated, the error bars for the limb extremities such as wrists and ankles are larger than for other joints. The quantiles of the ankles indicate that the ankle joints are not well registered

We further applied the multi-layer framework to sequence S2 that consists of 1202 frames. Since we use the 3D surface mesh model of subject S4, the model does not fit subject S2, see Fig. 19. Nevertheless, competitive results are obtained even though the error is larger by 6 mm than for sequence S4, see Tables 2 and 3. The increase of the error seems to be mainly caused by the wrong model since the camera setting and movement are very similar to S4. Particularly, the elbow joints of the model are at the wrong position which causes problems when the elbows are angled. This indicates that our approach would also work with a generic surface model like the SCAPE model (Anguelov et al. 2005; Balan et al. 2007). However, it also reveals that the quality of the surface mesh has a significant impact on the accuracy of the estimates.

The influence of a strong prior is demonstrated in Fig. 18. To this end, we learned the physical constraints of the head movement only by joint samples around zero. While the estimates from the first layer are biased towards the training data and do not fit the image data, the second layer reduces the bias since it does not rely on the prior. We emphasize that

Table 6 3D and 2D errors for subject S2. Accurate results are obtained by our multi-layer framework although the sequence has been tracked with a wrong surface mesh model, see Fig. 19

♯Layers	Dataset	3D (mm)		2D/C1 (pix)		2D/C2 (pix)	
		Absolute	Relative	Absolute	Relative	Absolute	Relative
1	Set1 (1–350)	**41.50 ± 7.98**	45.78 ± 9.00	5.45 ± 1.49	5.85 ± 1.74	5.54 ± 1.78	5.66 ± 1.84
2	Set1 (1–350)	**32.23 ± 5.71**	33.49 ± 6.03	4.10 ± 1.11	4.24 ± 1.25	4.38 ± 1.36	4.28 ± 1.33
1	Set2 (1–700)	**45.04 ± 12.85**	48.36 ± 13.68	5.79 ± 1.89	6.04 ± 2.04	6.07 ± 2.35	6.22 ± 2.41
2	Set2 (1–700)	**35.86 ± 10.73**	37.62 ± 11.42	4.49 ± 1.44	4.65 ± 1.55	4.85 ± 1.86	4.92 ± 2.01
1	Set3 (1–1202)	**43.82 ± 10.65**	46.57 ± 11.44	5.61 ± 1.57	5.89 ± 1.71	5.95 ± 1.91	6.14 ± 1.96
2	Set3 (1–1202)	**37.53 ± 9.00**	39.36 ± 9.70	4.77 ± 1.25	4.99 ± 1.34	5.13 ± 1.55	5.25 ± 1.69

Table 7 3D and 2D errors for subject S4. The frames 298–335 are neglected since the ground truth is corrupted for these frames

♯Layers	Dataset (Frames)	3D (mm)		2D/C1 (pix)		2D/C2 (pix)	
		Absolute	Relative	Absolute	Relative	Absolute	Relative
1	Set1 (2–350)	**34.59 ± 4.63**	43.93 ± 8.24	4.48 ± 1.00	5.66 ± 1.69	4.17 ± 0.72	4.93 ± 1.17
2	Set1 (2–350)	**27.65 ± 2.96**	33.91 ± 4.97	3.58 ± 0.74	4.40 ± 1.03	3.35 ± 0.51	3.91 ± 0.86
1	Set2 (2–700)	**38.53 ± 6.90**	47.00 ± 10.60	5.14 ± 1.30	6.22 ± 1.90	5.01 ± 1.38	5.70 ± 1.76
2	Set2 (2–700)	**32.14 ± 5.42**	37.31 ± 6.55	4.34 ± 1.05	5.04 ± 1.21	4.24 ± 1.14	4.72 ± 1.35
1	Set3 (2–1258)	**38.07 ± 5.84**	45.25 ± 9.13	5.25 ± 1.17	6.12 ± 1.62	5.00 ± 1.12	5.71 ± 1.53
2	Set3 (2–1258)	**32.01 ± 4.53**	36.01 ± 5.79	4.42 ± 0.92	4.99 ± 1.04	4.30 ± 0.93	4.71 ± 1.10

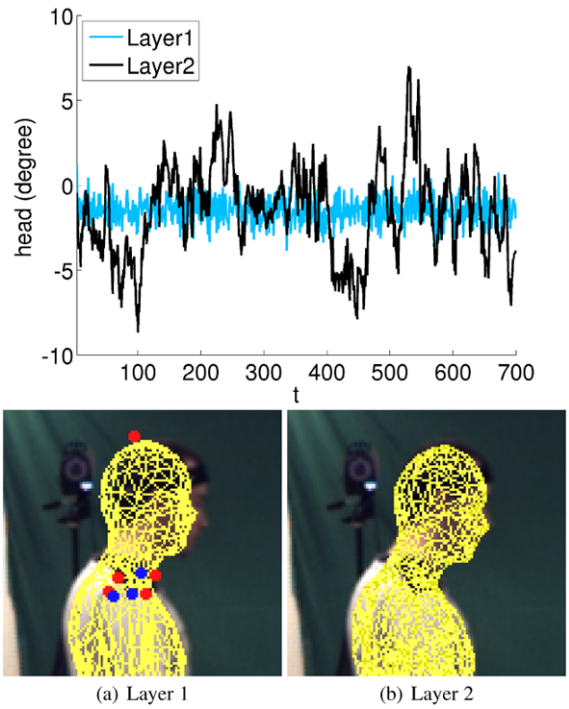


Fig. 18 Biased estimates. *Top*: When the physical constraints are modeled by a strong prior, the estimates are biased towards the training data. For this example, only joint samples around zero have been used. Since the second layer does not make use of the prior, the bias is reduced. *Bottom*: Biased estimate of the head by the first layer (*left*). The estimate of the second layer better fits the image data (*right*)

the bias is not completely removed, since the second layer is initialized by the estimates of the first layer, but the example shows that the estimates of our multi-layer framework better fit the image data.

In order to allow comparison to other approaches that have not been mentioned in this section, we provide various error metrics for the sequences S2 and S4 in Tables 6 and 7. Each sequence is split into three sets, where the first set contains only the walking motion, the second the walking and jogging motion and the third set the entire sequence consisting of walking, jogging, and balancing. The average errors and standard deviations are given for global stochastic optimization (one layer) and the multi-layer framework (two layers). The 2D errors are computed for cameras C1 and C2. The relative error is computed with respect to the pelvis joint. For a detailed description on the error metrics, we refer to Sigal and Black (2006). We remark that the relative error is higher than the absolute error. This indicates that the marker for the pelvis joint has not been accurately registered to the surface mesh model. In addition, some estimated human body poses of the multi-layer framework are shown in Figs. 19 and 20.

7 Discussion

In this work, we have compared optimization and filtering approaches for model-based human motion capture that do

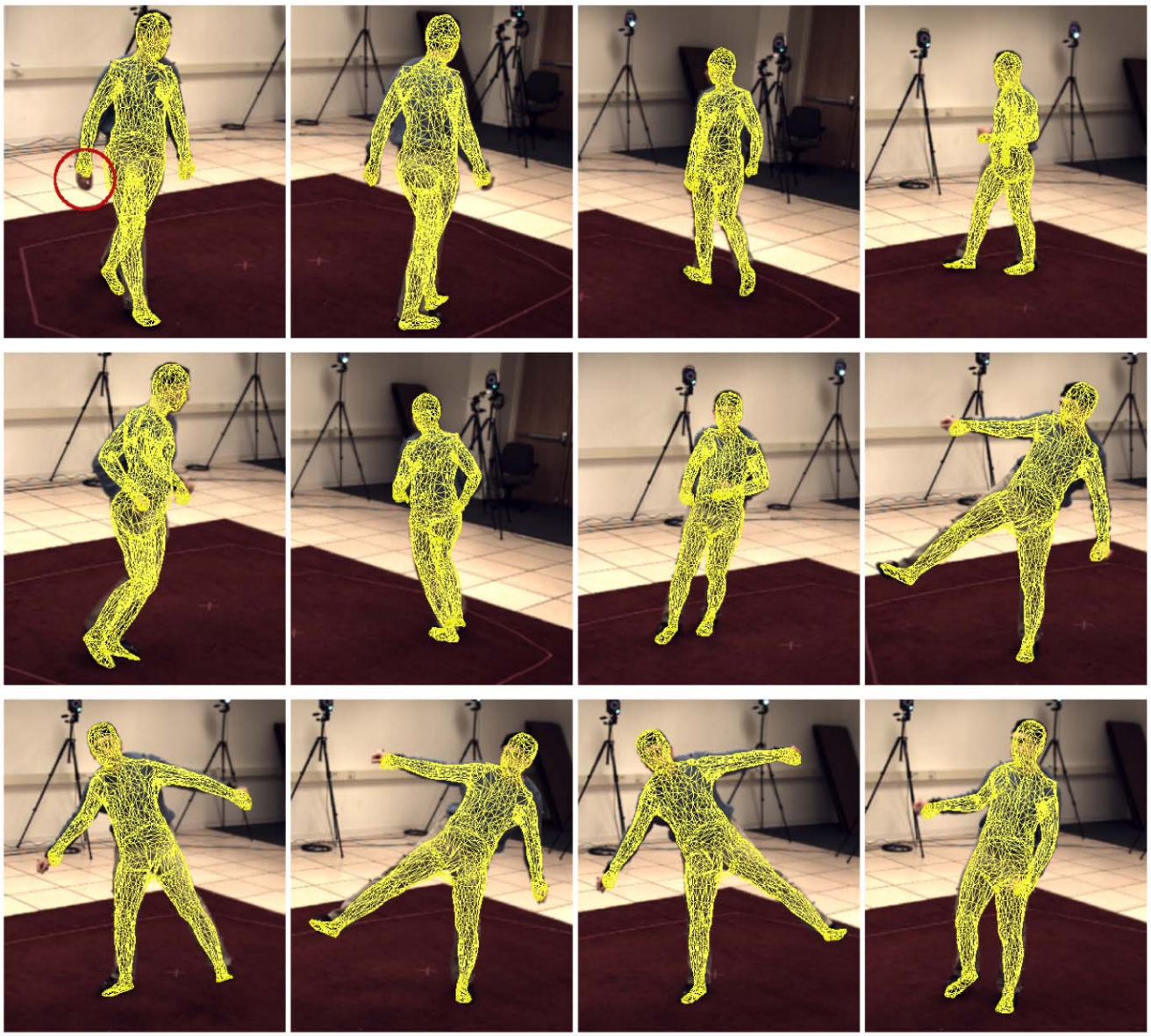


Fig. 19 Estimates for subject S2. Note that the arms of the surface mesh model are too short, since the model of subject S4 has been used for tracking (*top left*). *From top left to bottom right*: The meshes of the estimates are projected on the images of camera C1 for frames 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, and 1200

not rely on prior knowledge on the dynamics. A quantitative error analysis has revealed that a recently proposed stochastic optimization technique (ISA) provides significantly better estimates than an iterative closest point approach, a standard particle filter, a variant of the smart particle filter, or the annealed particle filter. While ISA provides robust and relatively accurate estimates of the human pose, an even higher precision is only achieved at the expense of high computational cost. To address this problem, we have introduced a multi-layer framework that combines the advantages of global stochastic optimization, local optimization, and filtering. While the first layer relies on ISA, the second layer refines the estimates where filtering and local optimization are coupled. The second layer not only increases the accuracy, but also reduces jitter and potential bias from the first layer. The latter is an important issue particularly in medical applications. In practice, the two layers can be run in parallel such that the processing time is not increased. So far real-time performance cannot be achieved, but we intend to reduce the computation time further by exploiting the parallel structure of ISA and graphics hardware.

Since the described approach is based on a fixed surface model, its general applicability is still limited. Although good results are obtained even with a wrong surface model,

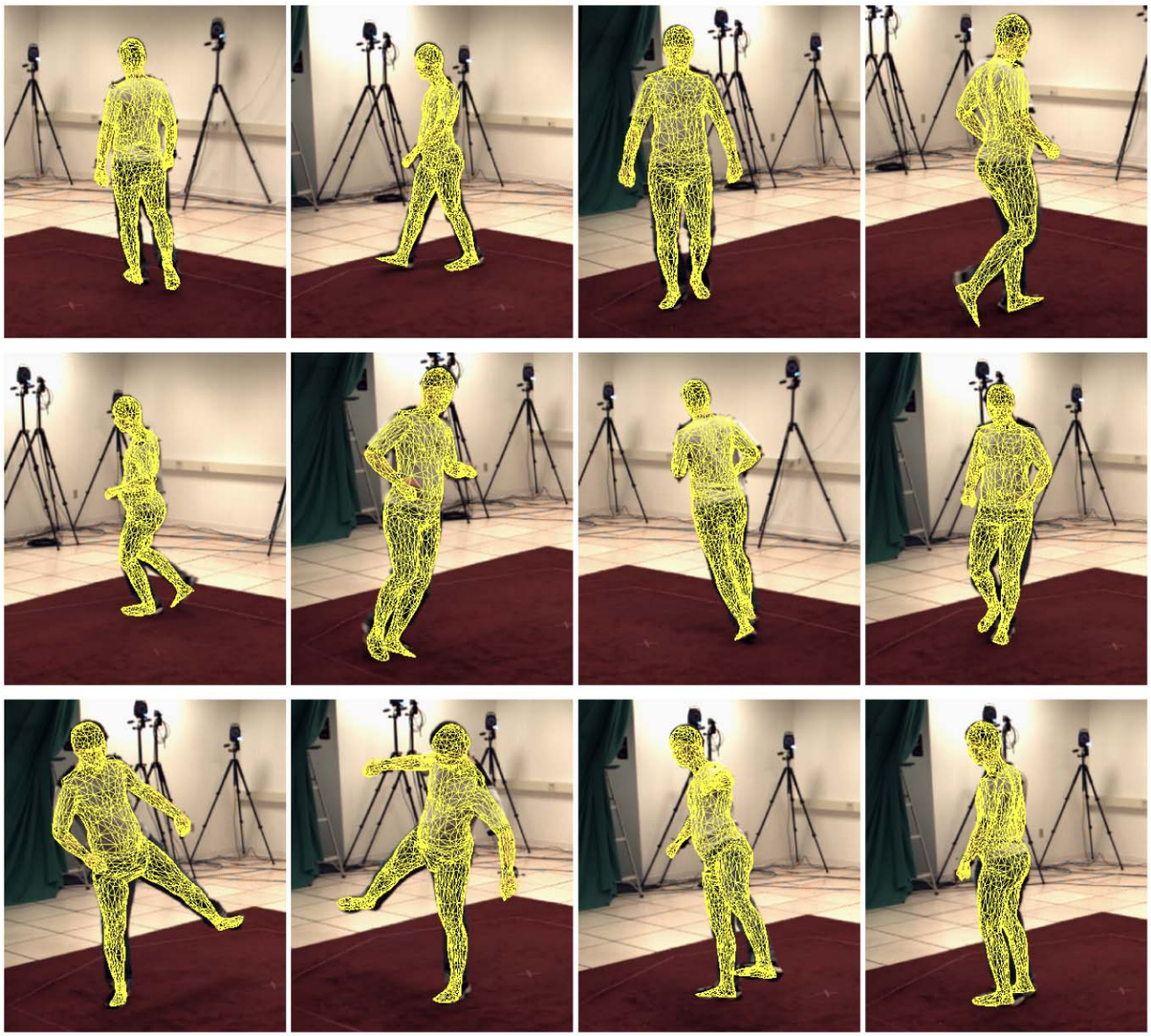


Fig. 20 Estimates for subject S4. *From top left to bottom right*: The meshes of the estimates are projected on the images of camera C2 for frames 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, and 1200

we have demonstrated that the quality of the surface mesh has an impact on the accuracy of the estimates. A solution to be investigated in future works might be to adapt a generic human model to the image data. The framework could also be combined with motion priors which might be useful in monocular scenarios. The multi-layer framework is appealing in this case, since the motion priors would reduce the search space for ISA and the second layer would be necessary to reduce the bias introduced by the priors.

Acknowledgements This research is partially funded by the Max-Planck Center for Visual Computing and Communication and the Cluster of Excellence on Multimodal Computing and Interaction. We would like to thank Leonid Sigal for maintaining the online evaluation for the `HumanEva` dataset and Stefano Corazza for providing the 3D model for subject S4.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

Agarwal, A., & Triggs, B. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(1), 44–58.

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

Balan, A., Sigal, L., & Black, M. (2005). A quantitative evaluation of video-based 3D person tracking. In *IEEE workshop on VS-PETS* (pp. 349–356).

Balan, A., Sigal, L., Black, M., Davis, J., & Haussecker, H. (2007). Detailed human shape and pose from images. In *IEEE conference on computer vision and pattern recognition*.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, *34*(3).

Bray, M., Kohli, P., & Torr, P. (2006). Posecut: simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts. In *European conference on computer vision* (pp. 642–655).

Bray, M., Koller-Meier, E., & Gool, L. V. (2007). Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, *106*(1), 116–129.

Bregler, C. (1997). Learning and recognizing human dynamics in video sequences. In *IEEE conference on computer vision and pattern recognition*.

Bregler, C., & Malik, J. (1998). Tracking people with twists and exponential maps. In *IEEE conference on computer vision and pattern recognition* (pp. 8–15).

Bregler, C., Malik, J., & Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, *56*(3), 179–194.

Brox, T., Rousson, M., Deriche, R., & Weickert, J. (2003). Unsupervised segmentation incorporating colour, texture, and motion. In *Lecture notes in computer science: Vol. 2756. Computer analysis of images and patterns* (pp. 353–360). Berlin: Springer.

Brox, T., Rosenhahn, B., & Weickert, J. (2005). Three-dimensional shape knowledge for joint image segmentation and pose estimation. In *Lecture notes in computer science: Vol. 3663. Pattern recognition (DAGM)* (pp. 109–116). Berlin: Springer.

Brox, T., Rosenhahn, B., Kersting, U., & Cremers, D. (2006). Nonparametric density estimation for human pose tracking. In *Lecture notes in computer science: Vol. 4174. Pattern recognition (DAGM)* (pp. 546–555). Berlin: Springer.

Cheung, K., Baker, S., & Kanade, T. (2005). Shape-from-silhouette across time, part Ii: applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, *63*(3), 225–245.

Choo, K., & Fleet, D. (2001). People tracking using hybrid Monte Carlo filtering. In *International conference on Computer vision* (pp. 321–328).

CMU (2007). *Graphics lab motion capture database*. http://mocap.cs.cmu.edu.

Deutscher, J., & Reid, I. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, *61*(2), 185–205.

Deutscher, J., Blake, A., & Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 1144–1149).

Douc, R., Cappe, O., & Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *International symposium on image and signal processing and analysis* (pp. 64–69).

Doucet, A., de Freitas, N., & Gordon, N. (Eds.) (2001). *Sequential Monte Carlo methods in practice*. New York: Springer.

Fossati, A., Dimitrijevic, M., Lepetit, V., & Fua, P. (2007). Bridging the gap between detection and tracking for 3D monocular video-based motion capture. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Gall, J., Brox, T., Rosenhahn, B., & Seidel, H. P. (2007a). *Global stochastic optimization for robust and accurate human motion capture*. (Tech. Rep. MPI-I-2007-4-008). Max-Planck-Institut für Informatik, Germany.

Gall, J., Potthoff, J., Schnoerr, C., Rosenhahn, B., & Seidel, H. P. (2007b). Interacting and annealing particle filters: mathematics and a recipe for applications. *Journal of Mathematical Imaging and Vision*, *28*(1), 1–18.

Gall, J., Rosenhahn, B., & Seidel, H. P. (2007c). Clustered stochastic optimization for object recognition and pose estimation. In *Lecture notes in computer science: Vol. 4713. Pattern recognition* (pp. 32–41). Berlin: Springer.

Gall, J., Rosenhahn, B., & Seidel, H. P. (2008). Drift-free tracking of rigid and articulated objects. In *IEEE conference on computer vision and pattern recognition*.

Gavrila, D., & Davis, L. (1996). 3D model-based tracking of humans in action: a multi-view approach. In *IEEE conference on computer vision and pattern recognition* (pp. 73–80).

Hogg, D. (1983). Model-based vision: a program to see a walking person. *Image and Vision Computing*, *1*(1), 5–20.

Isard, M., & Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. In *European conference on computer vision* (pp. 343–356).

Isard, M., & Blake, A. (1998). A smoothing filter for condensation. In *European conference on computer vision* (pp. 767–781).

Kakadiaris, I., & Metaxas, D. (1996). Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *IEEE conference on computer vision and pattern recognition* (pp. 81–87).

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, *82*(Series D), 35–45.

Kehl, R., Bray, M., & Gool, L. V. (2005). Full body tracking from multiple views using stochastic sampling. In *IEEE conference on computer vision and pattern recognition* (pp. 129–136).

Lee, M., & Nevatia, R. (2006). Human pose tracking using multi-level structured models. In *European conference on computer vision* (pp. 368–381).

Moeslund, T., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*(2), 90–126.

Moon, K., & Pavlovic, V. (2006). Impact of dynamics on subspace embedding and tracking of sequences. In *IEEE conference on computer vision and pattern recognition* (pp. 198–205).

Moral, P. D. (2004). *Feynman-Kac formulae. Genealogical and interacting particle systems with applications*. New York: Springer.

Mundermann, L., Corazza, S., & Andriacchi, T. (2007). Accurately measuring human movement using articulated ICP with soft-joint constraints and a repository of articulated models. In *Computer vision and pattern recognition* (pp. 1–6).

Pennec, X., & Ayache, N. (1998). Uniform distribution, distance and expectation problems for geometric features processing. *Journal of Mathematical Imaging and Vision*, *9*(1), 49–67.

Puzicha, J., Buhmann, J. M., Rubner, Y., & Tomasi, C. (1999). Empirical evaluation of dissimilarity measures for color and texture. In *International conference on computer vision* (pp. 1165–1172).

Ramanan, D., Forsyth, D., & Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 65–81.

Rosenhahn, B., Brox, T., Smith, D., Gurney, J., & Klette, R. (2006). A system for marker-less human motion estimation. *Künstliche Intelligenz*, *1*, 45–51.

Rosenhahn, B., Brox, T., & Seidel, H. P. (2007a). Scaled motion dynamics for markerless motion capture. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Rosenhahn, B., Brox, T., & Weickert, J. (2007b). Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, *73*(3), 243–262.

Rosenhahn, B., Klette, R., & Metaxas, D. (Eds.) (2008). *Computational imaging and vision: Vol. 36. Human motion—understanding, modelling, capture and animation*. Netherlands: Springer.

Schraudolph, N. (1999). Local gain adaptation in stochastic gradient descent. In *International conference on artificial neural networks* (pp. 569–574).

Sidenbladh, H., Black, M., & Fleet, D. (2000). Stochastic tracking of 3D human figures using 2D image motion. In *European conference on computer vision* (pp. 702–718).

Sigal, L., & Black, M. (2006). *Humaneva: synchronized video and motion capture dataset for evaluation of articulated human motion* (Tech. Rep. CS-06-08). Brown University.

Sigal, L., Bhatia, S., Roth, S., Black, M., & Isard, M. (2004). Tracking loose-limbed people. In *IEEE conference on computer vision and pattern recognition* (pp. 421–428).

Sminchisescu, C., & Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, *22*(6), 371–391.

Stolfi, J. (1991). *Oriented projective geometry: a framework for geometric computation*. Boston: Academic Press.

Urtasun, R., & Fua, P. (2004). 3D human body tracking using deterministic temporal motion models. In *European conference on computer vision* (pp. 92–106).

Urtasun, R., Fleet, D. J., & Fua, P. (2006). 3D people tracking with Gaussian process dynamical models. In *IEEE conference on computer vision and pattern recognition* (pp. 238–245).

Weickert, J., ter Haar Romeny, B., & Viergever, M. (1998). Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, *7*, 398–410.

Williams, C., & Rasmussen, C. (1996). Gaussian processes for regression. In *Advances in neural information processing systems*.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, *13*(2), 119–152.