International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

A Self-Training Approach for Visual Tracking and Recognition of Complex Human Activity Patterns

Jan Bandouch · Odest Chadwicke Jenkins · Michael Beetz

Received: date / Accepted: date

Abstract Automatically observing and understanding human activities is one of the big challenges in computer vision research. Among the potential fields of application are areas such as robotics, human computer interaction or medical research.

In this article we present our work on unintrusive observation and interpretation of human activities for the precise recognition of human fullbody motions. The presented system requires no more than three cameras and is capable of tracking a large spectrum of motions in a wide variety of scenarios. This includes scenarios where the subject is partially occluded, where it manipulates objects as part of its activities, or where it interacts with the environment or other humans. Our system is self-training, *i.e.* it is capable of learning models of human motion over time. These are used both to improve the prediction of human dynamics and to provide the basis for the recognition and interpretation of observed activities.

The accuracy and robustness obtained by our system is the combined result of several contributions. By taking an anthropometric human model and optimizing it towards use in a probabilistic tracking framework we obtain a detailed biomechanical representation of human shape, posture and motion. Furthermore, we introduce a sophisticated hierarchical sampling strategy for tracking that is embedded in a probabilistic framework and outperforms state-of-the-art Bayesian methods. We then show how to track complex manipulation activities in everyday environments using a combination of learned human appearance models and implicit environment models. Finally, we discuss a locally consistent representation of human motion that we use as a basis for learning environment- and task-specific motion models.

All methods presented in this article have been subject to extensive experimental evaluation on today's benchmarks and several challenging sequences ranging from athletic exercises to ergonomic case studies to everyday manipulation tasks in a kitchen environment.

Keywords Markerless Human Motion Capture · Probabilistic State Estimation · Self-Trained Models of Human Motion · Activity Recognition

J. Bandouch
Intelligent Autonomous Systems Group, Technische Universität München, Boltzmannstr. 3, 85748 Garching bei München, Germany E-mail: bandouch@cs.tum.edu

O. C. Jenkins
Department of Computer Science, Brown University, 115 Waterman St., Providence, RI, 02912-1910, USA E-mail: cjenkins@cs.brown.edu

M. Beetz
Intelligent Autonomous Systems Group, Technische Universität München, Boltzmannstr. 3, 85748 Garching bei München, Germany E-mail: beetz@cs.tum.edu

1 Introduction

Observing and interpreting human activities is a constant topic of interest in *artificial intelligence* (AI) and *computer vision* (CV) research. The ability to understand human behavior and to act with respect to human actions or intentions is an ambitious goal. Once achieved, it will prove beneficial in application areas ranging from *robotics* to *human computer interaction* (HCI) or *medical research* such as *gait analysis*. However, despite the broad progress made in the field in the last decades, the road ahead is still paved.

In this article we will present our contributions to this highly active field of research. We present a markerless system for human fullbody motion tracking from

three or more cameras that utilizes a realistic human model to estimate the observed motions at high accuracy. Our system is capable to extract this information for arbitrary types of motions in realistic environments. This includes manipulation activities where objects are being handled and where interactions with the environment or between humans take place. Furthermore, our system is self-adaptive in that it can improve its efficiency over time by learning environment specific motion patterns. Lastly, these motion patterns can be used to infer semantic labels for the observed activities, and thus to reason about human activities and intentions.

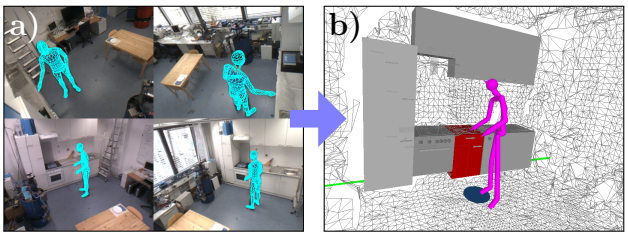

Fig. 1 Example application where human motion capture data is integrated into a knowledge base: a) human motion capture data as retrieved using our multi-view motion tracker; b) the data is integrated into a knowledge base where it is aligned with input from other perception modules (*e.g.* semantic environment maps or readings from an embedded sensor network) to enable higher-level reasoning about human activities (Figure b courtesy of Moritz Tenorth [63]).

Figure 1 shows one example application that is made possible by our system. The unintrusive retrieval of human motion capture data is used to complement sensor readings from an embedded sensor network to create a rich representation of human activities. The resulting knowledge base can be accessed using semantic queries to reason about ongoing activities and human intentions, or to filter sensor readings for relevant events, such as *places where humans stand while grasping a cup* [63]. Further example applications that have been implemented on top of our system include the integration of realistic human motions into a robotic simulator, and the motion transfer to humanoid robots.

The accuracy and maturity of our system is the combined result of several contributions that we will discuss in this paper. The basis of our model-based approach is an anthropometric human model that is adapted for use in tracking applications. Among our optimizations are a reduced set of shape parameters learned from exemplar shapes, a physiologically realistic coupling of pose parameters in the spine, and realistic temporal motion limits. We present *branched iterative hierarchical sampling* as a sampling strategy for recursive pose estimation. This strategy proved to be extremely effective in searching the high-dimensional space of human postures despite its nonlinearity with many local maxima. It is reliable and enables us to use a general motion model with a parameterization that allows for a large area of convergence during tracking. Furthermore, we present simple yet effective layered environment models that can be used in combination with color-based appearance models to implicitly deal with two cases of environmental occlusions by (1) subtracting dynamic non-human objects from the region of interest and (2) modeling objects (*e.g.* tables) that both occlude and can be occluded by human subjects. Figure 2 shows the relevance of these implicit models in dynamic environments. Our last contribution is an automated method for training models of human motion *on the fly*. These are used to improve the motion prediction, and for recognizing patterns of activities. We present a graph-based approach to model transitions between poses and a position invariant similarity measure based on the comparison of short *spatio-temporal motion snippets* to detect previously observed motion patterns.

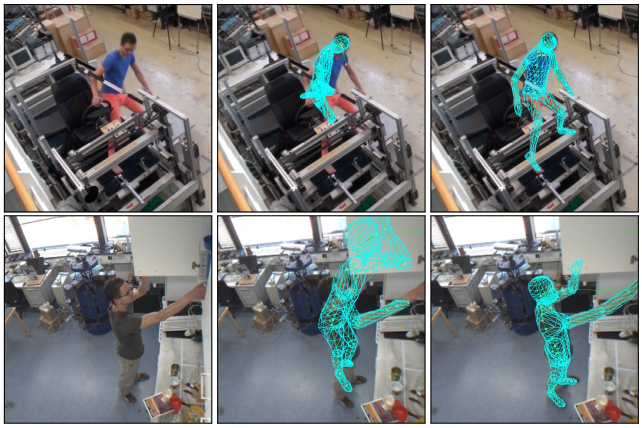

Fig. 2 Challenging setups for human motion capture in cluttered and dynamic environments. First row: A scenario where the tracking subject is occluded by a car-mockup; Second row: A scenario featuring a subject manipulating its environment. Even sophisticated motion tracking methods will fail without a model of the dynamic environment (center column). The last column shows tracking results using our system.

Extensive evaluations on the HumanEva 2 benchmarks show the potential of our method when compared to state-of-the-art Bayesian techniques. Besides the HumanEva 2 benchmarks, we present results on more challenging sequences, including kitchen tasks, sports sequences, ergonomic case studies and multi-target tracking. Furthermore, our tracker has played an integral role in the creation of the TUM Kitchen

dataset of everyday manipulation activities [62]. Here it was used to complement sensor readings in a sensor-equipped kitchen environment with fullbody motion capture data, without the need for an intrusive marker-based setup.

The remainder of this article is organized as follows. We start by introducing the anthropometric human model RAMSIS and our modifications to it in Section 2. In Section 3 we will propose a novel hierarchical sampling strategy for human pose tracking that is highly efficient in parsing the high-dimensional state space of human postures. We present a simple yet effective way to implicitely model the environment in Section 4, which enables us to track manipulation activities in complex environments. We show how to use our unconstrained motion tracking system to automatically learn environment- and task-specific models of human motion in Section 5. These are used to improve tracking efficiency as well as for recognizing observed activities. Experimental evaluation is presented in Section 6. Finally, we discuss related work in Section 7 and conclude in Section 8.

2 Anthropometric Human Model

In our work we take the new approach to integrate the digital human model RAMSIS for tracking of human motions (Figure 4). RAMSIS is an advanced and industry-proven model from the ergonomics community, that is widely-used especially in the automotive community [15]. It was initially developed to ease CAD-based design of car interior and human workspaces, as well as for use in ergonomic studies. The following advantages come with the use of this model:

- The model is capable of capturing different body types according to anthropometric considerations, *i.e.* the different appearance of a wide range of humans. Its design has been guided by ergonomic considerations from leading experts in the field.
- The locations of the inner joints correspond precisely to the real human joint locations, making this model an ideal choice for motion analysis tasks *e.g.* in sport analytics or ergonomic studies.
- The model is able to capture most of the movements humans can perform while retaining a correct outer appearance. Absolute motion limits are integrated and help to reduce the search space when tracking. Motion limits can be queried for different percentiles of the population using anthropometric knowledge.

The model consists of an inner kinematic model that closely resembles a real human skeleton (Figure 3), and an outer surface model representing human flesh and skin (Figure 4). The main model definition comprises aspects such as the hierarchy of body parts, the relative positions of body joints and surface vertices in a corresponding local part coordinate system, and absolute motion limits for body parts. All corresponding parameters are fixed and have been specified during the initial development of the RAMSIS model. In addition, variable parameters describe the pose $\boldsymbol{\psi}$ (dependant on the underlying kinematic structure) and the shape $\boldsymbol{\phi}$ (based on metric length values that provide an absolute scale for the relative positions in the model definition) of the model.

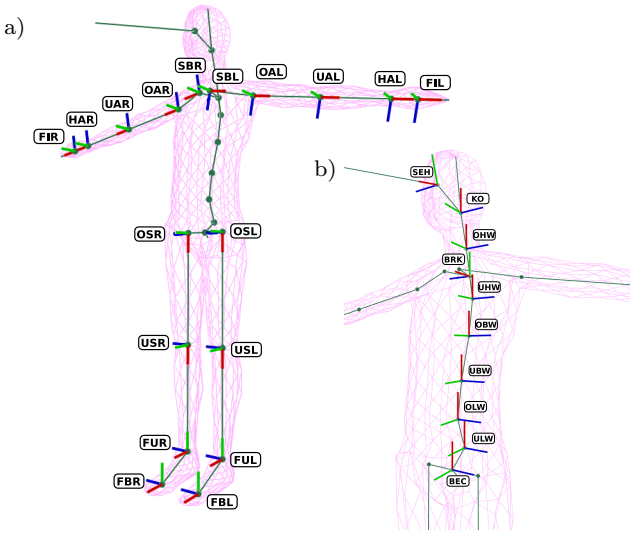

Fig. 3 The inner kinematic structure of the digital human model RAMSIS. Shown are the local part coordinate systems for a) the limbs b) the spine.

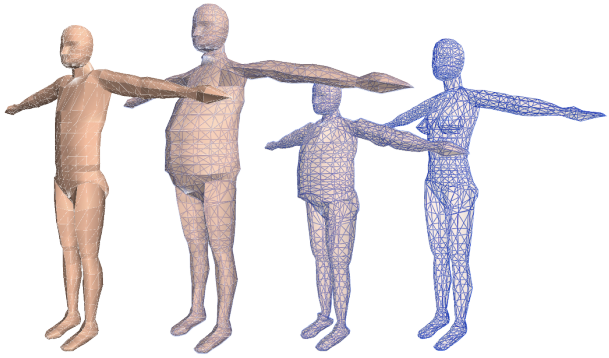

Fig. 4 The outer model of RAMSIS for different body shapes and gender.

We will now describe our model optimizations with respect to the original RAMSIS specification. These are tailored towards the specific needs of motion tracking applications, such as the ones presented later in this article.

Learning Shape Parameters: Both inner and outer model are adaptable to different anthropometries (height, figure, body mass, etc.) by setting the shape parameters $\boldsymbol{\phi}$. This is usually done by hand in an initialization step. In the original model specification of RAMSIS, a total of 643 shape parameters (43 inner parameters $\boldsymbol{\phi}_I$ and 600 outer parameters $\boldsymbol{\phi}_O$) is provided. Most of these parameters model local changes to the shape, and especially the outer parameters $\boldsymbol{\phi}_O$ influence only the placement of a few outer vertices. Such an over-parameterization makes it difficult to adapt the model shape to different humans in an intuitive manner.

To reduce the number of shape parameters, we have statistically analyzed the parameters by means of *principal component analysis* (PCA). We have taken $N = 139$ different (male) model adaptations for RAMSIS that cover a broad spectrum of human shapes. The principal components of the set of N known shape parameters $\boldsymbol{\phi}^{(i)};\ i \in 1 \dots N$ of dimensionality d are given by the *eigenvectors* $\mathbf{e}_1 \dots \mathbf{e}_d$ of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$. The eigenvectors can be extracted from $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ *e.g.* by means of *singular value decomposition* (SVD), which in the case of positive semi-definite symmetric matrices such as $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ results in $\boldsymbol{\Sigma}_{\boldsymbol{\phi}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$. The column vectors of $\mathbf{U} = (\mathbf{e}_1 \dots \mathbf{e}_d)$ correspond to the requested eigenvectors. Furthermore, $\boldsymbol{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues λ_i as diagonal entries.

The orthogonal space spanned by the principal components provides an alternative parameterization of the body shape. By ignoring principle components with small corresponding eigenvectors, we can reduce the number of parameters by ignoring irrelevant ones. A common criterion is to set the number of parameters based on the percentage of variance in the training set that is explained by the first x principal components. The variance in the direction of each principal component is given by the corresponding eigenvalue λ_i. Figure 5 plots the cumulative variance as a function of the first x eigenvectors for our application.

Coupling of Spinal Motion: We have reduced the d.o.f. of the original RAMSIS model as much as possible, to soften the complexity of tracking high-dimensional articulated models. While the limbs are already modeled with the minimal number of parameters necessary to express all possible poses, potential for reduction is given in the spine parameters. In RAMSIS, the spine is modeled by 7 independent joints (including the head) with 3 d.o.f. each. In the human anatomy however, the spine is a coupled structure that allows only for a restricted set of motions. It is *e.g.* impossible to bend the spine in several opposing directions. To provide a physiologically sound coupling of the spine, we split it into a lower spine, an upper spine, and the head spine. Joints belonging to each group are interpolated based on the relation of their absolute motion limits. Such a coupling is reasonable, as most of the common motions are performed by deforming only the upper spine (*i.e.* the thoracic and the cervical spine), while the lower spine (*i.e.* the lumbar spine) is only used to support extreme bending motions once the deformation of the upper spine reaches its limits.

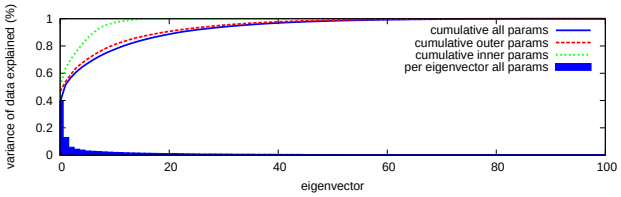

Fig. 5 Plotting the variance accounted for by the principal components of the shape parameters $\boldsymbol{\phi}$. The cumulative plots show the total percentage of the variance in the training set that is accounted for by the first x eigenvectors when estimating all 643 shape parameters, only the 600 outer shape parameters $\boldsymbol{\phi}_O$, or only the 43 inner shape parameters $\boldsymbol{\phi}_I$. Additionally, the variance for each eigenvector is plotted for the combination of $\boldsymbol{\phi}_I$ and $\boldsymbol{\phi}_O$.

By using this parameterization we are able to reduce the dimensionality of the spine from 21 to 9 d.o.f. while keeping the physiological expressivity. The reduced model used for tracking thus comprises 51 d.o.f. (or 41 when omitting hands and feet). Figure 6 shows some renderings of extreme spinal deformations that are possible with our modified RAMSIS model.

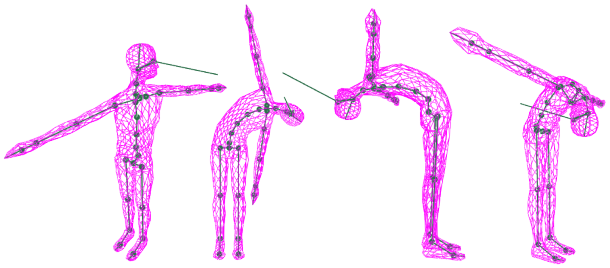

Fig. 6 Motion limits of the combined spine joints when keeping the pelvis fixed. Bending can be further increased by rotating the pelvis.

Biomechanical Inter-Frame Motion Limits: We performed a biomechanical study to estimate the maximum angular velocities for each body part in the model [21]. In this study, a human subject performed predefined isolated motions (*i.e.* each d.o.f. separately) at maximum speed while being recorded with two high-speed video cameras at 250 f.p.s. Our estimates have been derived based on the measured deviation between two manually adapted keyframes around the center of

the range of motion. In addition, we used marker-based motion capture data from sports science to refine the estimates.

For tracking applications, *e.g.* when doing stochastic search, the maximum angular velocities can be transformed into angular inter-frame standard deviations by considering the f.p.s. at which the videos were recorded.

Caching of Pose Calculations: Another optimization that helps to improve computational efficiency is the caching of body-part dependant pose calculations. It works by keeping the previous pose calculations in memory, which includes local coordinate systems, 3D coordinates of surface vertices, and their 2D image plane projections. Whenever a new pose is provided for model synthesis, all body parts whose parameters are changed when compared to the last calculated pose are invalidated. In addition, all dependencies of these body parts are also invalidated, *i.e.* all parts with invalidated kinematic predecessors. Such a modification can be powerful when used in combination with hierarchical particle filters as presented in Section 3.3. These algorithms repeatedly modify only parts of the parameter space for large numbers of parallel evaluations, and caching helps in reducing the number of required computational operations (about a factor of 4 in practice).

In addition to our modifications, several extensions to Ramsis such as posture prediction using internal and external forces as well as discomfort [56] have been presented. Integrating such cues provides promising opportunities for future research.

3 Human Pose Tracking

We will now discuss the key components of our sampling strategy for human pose tracking. Note that we believe that it is reasonable to consider human motion tracking as high-dimensional search, given that good approximations of the posterior density as needed in Bayesian estimation are hard to come by. This can be attributed to the complex high-dimensional distributions that are difficult to describe analytically and virtually impossible to approximate using discrete samples or histograms. Furthermore, the presence of local modes and the difficulty to generate accurate priors for arbitrary types of human motions provides hard challenges for gradient-based optimization techniques. A good strategy is to combine the parallel exploration capabilities of particle filters with techniques adopted from optimization methods [19,13]. We will now approach such a strategy from a Bayesian perspective.

3.1 Particle Filtering

Figure 7 gives a schematic introduction of one time step of the traditional *sampling importance resampling* (SIR) particle filter for recursive Bayesian estimation [4]. In particle filtering, the *posterior* probability density function (pdf) $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t})$ for a pose $\mathbf{x}_t$ at time t given a sequence of image observations $\mathbf{y}_{0:t}$ up to time t is represented by a set of N weighted particles $\mathcal{S}_t = \{\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \rangle\}_{i=1}^{N}$ where $w_t^{(i)}$ are the normalized weights. Tracking is performed by predicting a new state for each particle using the motion model $p(\mathbf{x}_t \,|\, \mathbf{x}_{t\text{-}1})$ before updating the weights according to the current observation likelihood $p(\mathbf{y}_t \,|\, \mathbf{x}_t)$. The resampling in each timestep is necessary to prevent the particle set from degeneracy [4].

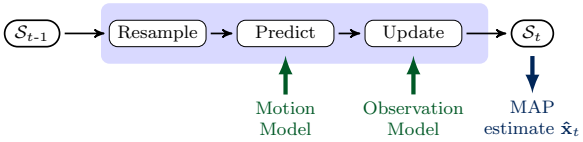

Fig. 7 One timestep of Sampling Importance Resampling. The weighted particle set $\mathcal{S}_{t\text{-}1}$ from the previous timestep is resampled to create an unweighted set where particles have been drawn with probability proportional to their weights. The resampled particles are then used to predict new states according to the motion model. In the last step, particle weights are updated based on the current observation. This yields the new weighted particle set $\mathcal{S}_t$, from which a final state estimate $\hat{\mathbf{x}}_t$ can be computed.

As we are interested in tracking arbitrary motions, we use the following motion model in our application:

$$\mathbf{x}_t = \mathbf{x}_{t\text{-}1} + \boldsymbol{\delta} \quad ; \qquad \boldsymbol{\delta} \sim \mathcal{N}(0,\, \boldsymbol{\Sigma}) \tag{1}$$

Here, $\boldsymbol{\Sigma}$ is a diagonal matrix where the diagonal entries correspond to the variance σ_j^2 of the j-th component of $\mathbf{x}$. The variance influences the amount of diffusion for each pose parameter or joint angle in the state vector $\mathbf{x}$. We have initialized these parameters once according to the biomechanical inter-frame motion limits of our model as derived in Section 2.

The weight update $w_t^{(i)} = \omega(\mathbf{x}_t^{(i)}) \sim p(\mathbf{y}_t \,|\, \mathbf{x}_t)$ is performed by comparing rendered binary projection masks I_P of the particle states with binary foreground masks I_F extracted using background subtraction (Figure 8). The number of inconsistent pixels corresponds to the error e_s for the predicted state:

$$e_s = \mathrm{COUNT}(\mathrm{XOR}(I_P, I_F)) \tag{2}$$

Here, XOR corresponds to a pixelwise symmetric difference ($\triangle$) operator that works on the image masks of all cameras in parallel, and the COUNT operator sums up

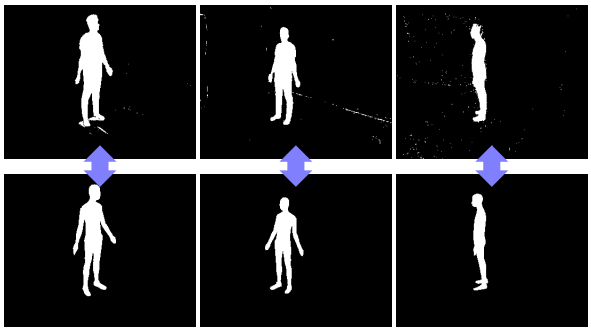

Fig. 8 Estimating the likelihood of a predicted state estimate $\mathbf{x}_t^{(i)}$ by comparing binary foreground masks I_F from background subtraction (top row) to rendered projection masks I_P (bottom row).

the non-zero pixels of all cameras. The final weights $w_t^{(i)}$ are scaled between 0 (highest encountered error) and 1 (lowest encountered error), with subsequent normalization. Such a likelihood model provides good results when using at least three camera views and a fitting human shape model.

3.2 Multi-Layered Search

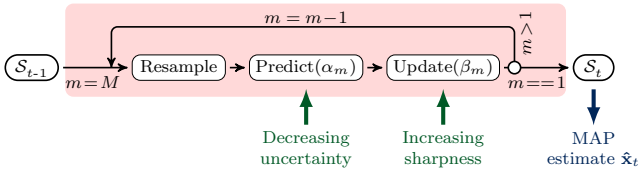

Fig. 9 One timestep of Annealed Particle Filtering. The final particle set is estimated in M iterations . Each iteration is comparable to a SIR step, where the amount of diffusion (*i.e.* uncertainty) added during the prediction step is reduced with each iteration by a factor α_m, and where the weight function in the update step is annealed according to the scheme $\langle \beta_M, \beta_{M-1}, \ldots, \beta_1 \rangle$. This annealing has the effect of gradually sharpening the weight function to slowly push particles towards the global maximum.

Pure SIR particle filtering is inapplicable to the problem of human motion tracking as the number of necessary particles grows exponentially with the dimensionality of the state space. Deutscher and Reid [19] motivate a multi-layered search strategy related to simulated annealing [37] to overcome this problem by focusing particles around the (global) modes of the pdf, which is similar in spirit to optimization methods. This *annealed particle filter* (APF) is illustrated in Figure 9. Over the course of M iterations, the particle set evolves towards the final set $\mathcal{S}_t$. With each iteration, the diffusion $\boldsymbol{\delta}$ introduced during the prediction step (Equation 1) is scaled by a factor α_m (*e.g.* 0.5) to decrease uncertainty. At the same time the weight function $w_m(\mathbf{x}_t^{(i)}) = \omega(\mathbf{x}_t^{(i)})^{\beta_m}$ is *sharpened* by exponentiating with values from an annealing scheme $\beta_M < \beta_{M-1} < \ldots < \beta_1$. As β_m increases with each iteration, particle survival changes from a broad survival to a survival of the fittest, which helps to overcome local maxima in the early iterations while focusing on the modes towards the latter ones.

Good performance on short tracking sequences has been reported [19], however it was shown that APF still needs an exponential amount of particles as all dimensions are estimated simultaneously [7]. A more efficient use of particles is achieved by *covariance scaled diffusion*, where the diffusion vector $\boldsymbol{\delta}$ is sampled from the state covariance matrix $\boldsymbol{\Sigma}_{t,m+1}$ estimated from the particle set $\mathcal{S}_{t,m+1}$ of the last iteration (except when $m = M$). Therefore, diffusion is adaptively guided based on how well a parameter has already been estimated, and the search becomes focused in regions where the optimal parameters could not yet be determined [19]. A similar idea was presented by Sminchisescu and Triggs [60] in the context of monocular tracking to favor particle diffusion along the directions of the unobservable d.o.f.

In practice, the diffusion vector $\boldsymbol{\delta}$ is sampled as follows:

$$\boldsymbol{\Sigma}_{t,m+1} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}} \tag{3}$$

$$\boldsymbol{\delta} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\mathbf{z} \quad ; \qquad \mathbf{z} \sim \mathcal{N}(0, I) \tag{4}$$

In Equation 3 we perform an eigendecomposition of $\boldsymbol{\Sigma}_{t,m+1}$ (*e.g.* by means of *singular value decomposition*), such that the column vectors of $\mathbf{U} = (\mathbf{e}_1 \ldots \mathbf{e}_d)$ correspond to its eigenvectors, and $\boldsymbol{\Lambda}$ is the diagonal matrix with the corresponding eigenvalues λ_i as diagonal entries. Then, a random vector $\mathbf{z}$ made up of standard normal distributed random variables is sampled, scaled by the squareroots of the eigenvalues, and transformed from eigenvector space into state space (Equation 4).

3.3 Hierarchical Partitioning

Although covariance scaled diffusion provides a soft partitioning, the problem with APF is that all state parameters are still effectively estimated in parallel. This is especially critical as typical human body models consist of hierarchically dependant parameters (*e.g.* the elbow joint parameters depend on the shoulder joint parameters), so that a parallel estimation is inefficient and can even lead to bad estimates [7]. An alternative approach to high dimensional state estimation is

to subdivide the state space $\mathcal{X}$ into several partitions $\mathcal{X} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_K$ that can be estimated sequentially, following the hierarchical structure of the human body (*e.g.* torso, left upper arm, left lower arm, etc.). This is a valid approach when the dynamics of one partition do not influence the dynamics of a hierarchically preceding partition and a local weight function can be evaluated for each partition. Figure 10 illustrates this algorithm known as *partitioned sampling* (PS) [44]. Note that although Maccormick and Isard [43] originally introduced a *weighted resampling* operator for mathematical elegance, partitioned sampling can in fact be seen as a sequentially coupled series of SIR particle filters (Figure 11) when using *importance resampling* in combination with a local update step instead, which is equivalent.

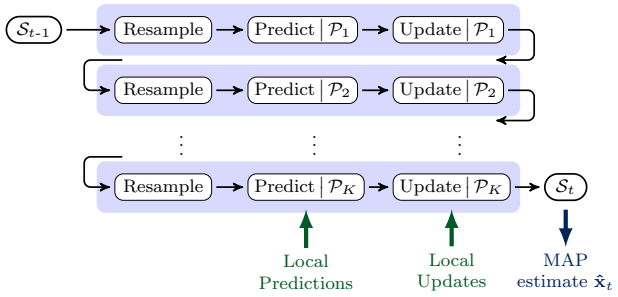

Fig. 10 One timestep of Partitioned Sampling. Each partition is estimated using the resampling, prediction and update pattern known from the SIR particle filter (Figure 7). However, predictions and updates for each partition are local, *i.e.* only parts of the state space corresponding to the current partition $\mathcal{P}_k$ are estimated. The particle set $\mathcal{S}_{t\text{-}1}$ evolves into the new particle set $\mathcal{S}_t$ by traversing all partitions sequentially. The order of partitions is subject to kinematic precedence relations.

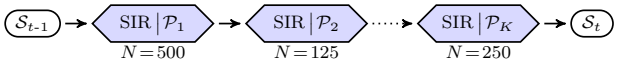

Fig. 11 Simplified representation of PS as sequentially coupled SIR filters. The particle count N for each partition is variable.

3.4 Branched Iterative Hierarchical Sampling

PS can be an efficient tool in overcoming the exponential growth in particles needed for high-dimensional tracking. By splitting the search space into manageable chunks, the particle growth can be kept within linear bounds. The downsides are that (•) individual partitions must be reasonably small to be manageable by SIR filters, which can be problematic when *e.g.* the torso needs to be split into several partitions, (•) and that errors in early iterations are propagated without chance of recovery, which is particularly problematic when using inaccurate human models such as cylindrical approximations.

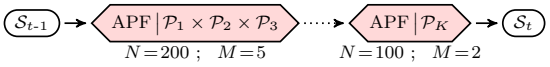

Fig. 12 Iterative Hierarchical Sampling (IHS). Enabling bigger partitions in PS (Figure 11) by exchanging SIR filters (Figure 7) with APF (Figure 9).

A good solution to these problems is to exchange the local SIR filters in PS with APF partitions (Figure 12). This increases the potential size of each partition and helps to avoid sampling errors in early iterations. We have already proven that this *iterative hierarchical sampling* (IHS) strategy works extremely well in simulation [7], however noisy observation data (such as partially missed limbs in the foreground masks) still leads to erroneous outcomes (Figure 13). As a solution to this problem, we propose to evaluate several partitioning schemes in parallel. By concatenating the resulting particle sets and performing a final update step, this corresponds to an implicit *voting* of the best partitioning scheme (Figure 14). This idea was originally published by Maccormick and Isard [43] as *branched partitioned sampling* in the context of occlusion handling of multi-target tracking, to make sure that the unoccluded target will be evaluated first. We thus term our combined algorithm *branched iterative hierarchical sampling* (BIHS).

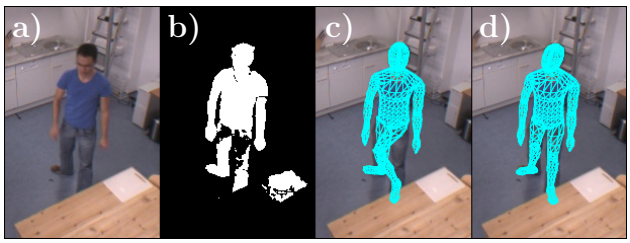

Fig. 13 How the order of the limb partitions can influence the outcome when observations are noisy: a) original image b) noisy foreground mask c) left leg first partitioning scheme d) right leg first partitioning scheme. For this example, strategy d) works best.

Parallel evaluation of several compatible partitioning schemes is a powerful means to circumvent problems arising from noisy observation data. The resulting particle diversity in the exploration of the state space $\mathcal{X}$ is

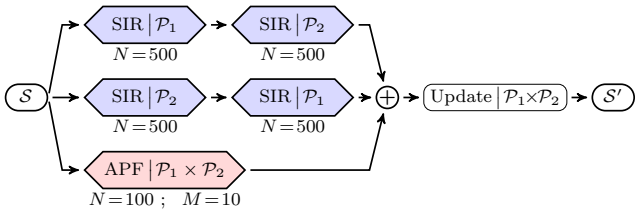

Fig. 14 Branched Iterative Hierarchical Sampling (BIHS). When the order of evaluation is arbitrary (*e.g.* left leg then right leg or vice versa), particle evaluations can be split into parallel pipelines and concatenated again as long as identical parts of the statespace are estimated. A final update step acts as a voting for the best partitioning strategy.

the key to avoiding getting trapped in local maxima of the weight function $\omega(\mathbf{x}_t^{(i)})$.

We will now sketch a meaningful partitioning scheme for human pose tracking, but we will restrict ourselves to the lower body. The extension to the full body is straightforward. In our experiments we found it best to start with an APF partition for the full state space $\mathcal{X}$, which we will denote as $APF(\mathcal{X})$. This results in a good torso estimate after only a few iterations, whereas the limbs will still be badly estimated. Then, the lower body is estimated by branching the following five schemes that are made up from the four minimal partitions *left upper leg* (U_L), *right upper leg* (U_R), *left lower leg* (L_L), *right lower leg* (L_R):

1. $APF(U_L) \rightarrow APF(L_L) \rightarrow APF(U_R) \rightarrow APF(L_R)$
2. $APF(U_R) \rightarrow APF(L_R) \rightarrow APF(U_L) \rightarrow APF(L_L)$
3. $APF(U_L \times L_L) \rightarrow APF(U_R \times L_R)$
4. $APF(U_R \times L_R) \rightarrow APF(U_L \times L_L)$
5. $APF(U_L \times L_L \times U_R \times L_R)$

The larger the dimensionality of the partitions and the larger the corresponding inter-frame motion limits, the more iterations and particles should be used.

In theory, the partitioning schemes could be automatically selected as all ordered combinations of elements from the power set $\boldsymbol{\mathcal{P}}(S)$ (with $S = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\}$ corresponding to the set of smallest meaningful subpartitions of $\mathcal{X}$) where each $\mathcal{P}_k$ appears exactly once per branch and hierarchical precedence relations are not violated. However, with growing number K of minimal partitions there will be too many schemes, and only a subset of these should be selected in practice. In all our experiments we have used 5 parallel partitions for the lower body, and 5 for the upper body.

4 Implicit Environment Models

In Section 3.1 we briefly introduced our shape-based observation model, where binary segmented foreground masks I_F are compared to model projection masks I_P of the current pose state (Figure 8). Such observation models are prone to errors once scenes become dynamic and subjects start interacting with the environment. In such cases, I_F can contain dynamic parts of the environment in addition to the human subjects. Furthermore, humans might be contained only partially in I_F due to occlusions from the environment.

We propose the use of layered environment models to approach the problem of humans interacting with dynamic environments. These are simple 2D models that are capable to implicitly model occlusions and dynamic regions directly in the image plane. To do so, we learn simple color-based appearance models for our human model, such that each surface triangle is associated with a color distribution (Figure 15). The learning can be done in an initialization step, or adaptively as tracking progresses. Up to now, we have used color information solely to distinguish between foreground and background. In the environment models that we present next, we use the learned color information to distuinguish between different types of foreground (*i.e.* humans or dynamic objects). The layered structure of these environment models makes them fast and easy to use in practice, without requiring explicit 3D models of the environment.

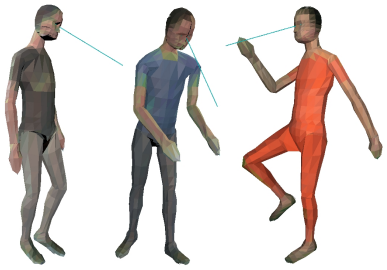

Fig. 15 Human model examples with learned appearance (color) information.

To deal with cases of dynamic non-human foreground objects and environmental occlusions, we introduce a new binary layer mask I_B into our observation model, that will be used to block regions from processing. This mask is set (1) for all pixels that should be processed, and unset (0) for regions to be blocked. Blocking is then achieved by masking out the respective parts in both the foreground mask I_F and the projection mask I_P before evaluating the shape error e_s from

Equation 2:

$$I_F = \text{AND}(I_F, I_B) \quad (5)$$

$$I_P = \text{AND}(I_P, I_B) \quad (6)$$

The filtering of blocked regions is achieved by a pixelwise logical AND operation between the blocking mask I_B and I_F respectively I_P. This principle is depicted in Figure 16 for the example of blocking an occluding table. As a result, the blocked regions in the image plane are removed from I_F and I_P. Silhouette differences between I_F and I_P in these blocked regions do not add to the shape error e_s, nor can these regions be used to favor incorrect model projections when containing incorrectly detected foreground shapes (*e.g.* dynamic objects).

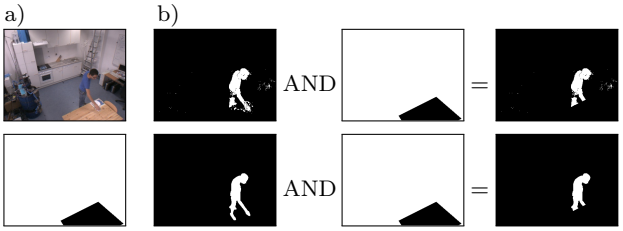

Fig. 16 Principle of blocking layers: a) original camera image (top) and corresponding blocking layer I_B with marked table (bottom) b) the blocking layer is used to remove uncertain regions from both foreground mask I_F (top row) and projection mask I_P (bottom row) by means of a bitwise AND operator. Therefore, the regions marked as blocked in I_B (black) are effectively ignored when calculating the shape error e_s (Equation 2).

We will now show how these blocking layers can be used to filter dynamic non-human foreground objects and how they can help in modeling occlusions.

Dynamic non-human foreground objects: In this case dynamic objects (possibly manipulated by the human subject) or dynamic parts of the environment (doors, cupboards, drawers) appear inside the foreground mask and mix with the human silhouettes. To filter these parts, we introduce a human appearance mask I_H that is set whenever a pixel's color resembles a color in our appearance model. Such a mask can be calculated efficiently using a binary lookup table when color values are discretized to 16 bit. The lookup table is updated whenever the appearance model is updated. We then remove non-human parts from the foreground mask I_F and add them to the blocking mask I_B using the following operations:

$$I'_F = \text{AND}(I_F, I_H) \quad (7)$$

$$I'_B = \text{AND}(I_B, \text{NOT}(\text{DIFF}(I_F, I'_F))) \quad (8)$$

Here, DIFF corresponds to the pixelwise difference ($\ominus$) operator and NOT specifies inversion ($\neg$). Adding non-human parts to the blocking region is important, as we have no idea whether the dynamic object might be occluding the human. It also weakens the influence of erroneously removed parts of the humans, as they will be ignored without penalizing the shape model. Fig. 17 gives an example on the effectivity of these simple operations.

Environmental occlusions: In this case static objects in the environment partially occlude the observed subjects (*e.g.* tables, chairs). However, these objects can also be occluded by the subjects, as there is no persistent spatial ordering. We mark regions that are candidates for occlusions (*e.g.* tables) by unsetting these regions in the blocking mask I_B. This needs to be done once during camera setup and can be done by a user within seconds by choosing a polygonal region to be considered. Such regions will by default be ignored during evaluation. To prevent valid observations of human body parts to be blocked, *e.g.* when arms are visible above a table, we exclude all human-like foreground regions from blocking:

$$I''_B = \text{OR}(I'_B, I'_F) \quad (9)$$

Here, OR corresponds to the pixelwise union ($\vee$) operator. Fig. 18 shows exemplar results using this kind of occlusion modeling. The amount of occlusion that can be compensated depends on the number of cameras used and their placement, *i.e.* each body part should always be observable from at least three cameras. Scenarios with more occlusion thus require more cameras.

5 Self-Trained Motion Models

As will be shown in our experimental results (Section 6), the system presented so far is capable of tracking a wide variety of motions in complex environments. Due to the general motion model from Equation 1, we are able to track any kind of motion that is compliant with our biomechanical motion limits (both intra- and inter-frame). However, computational costs are high, as uninformed sampling requires much more particle evaluations than a more informed sampling would.

Prior learning of human motions from training data is one way of creating such informed predictions. The downside is that the space of possible human poses is very large. Good training data that encompasses the often complex and diverse human motion patterns is difficult to come by, and almost always requires the use of expensive marker-based motion capture systems. In the end, trained motions are often unrelated to motions that are specific to an environment, and thus of no practical use.

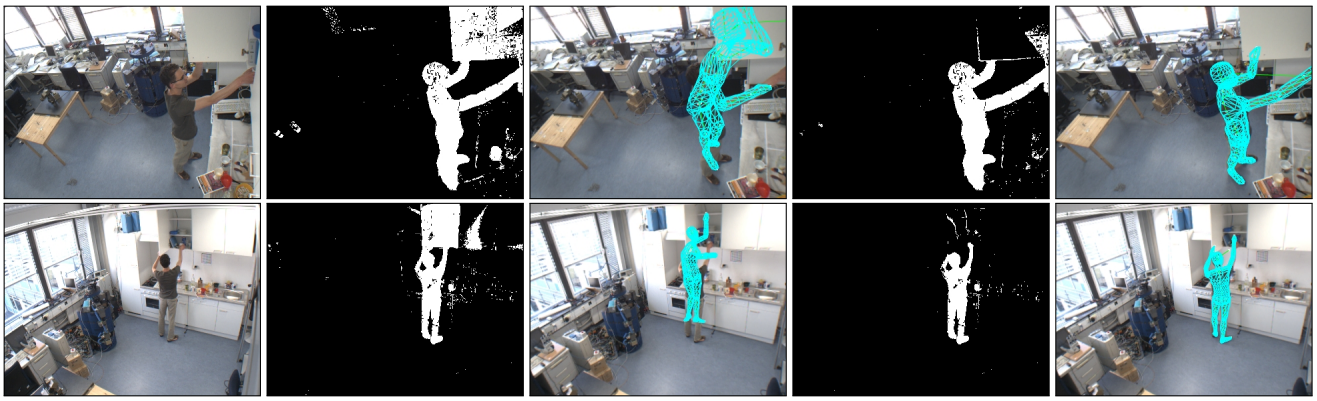

Fig. 17 Example frame with opened cupboard (from left to right): a) original image b) unmodified foreground mask I_F c) tracking results without using appearance d) foreground mask I'_F after non-human foreground removal e) tracking results using our method. Each row shows one camera view. White color in the mask represents set bits.

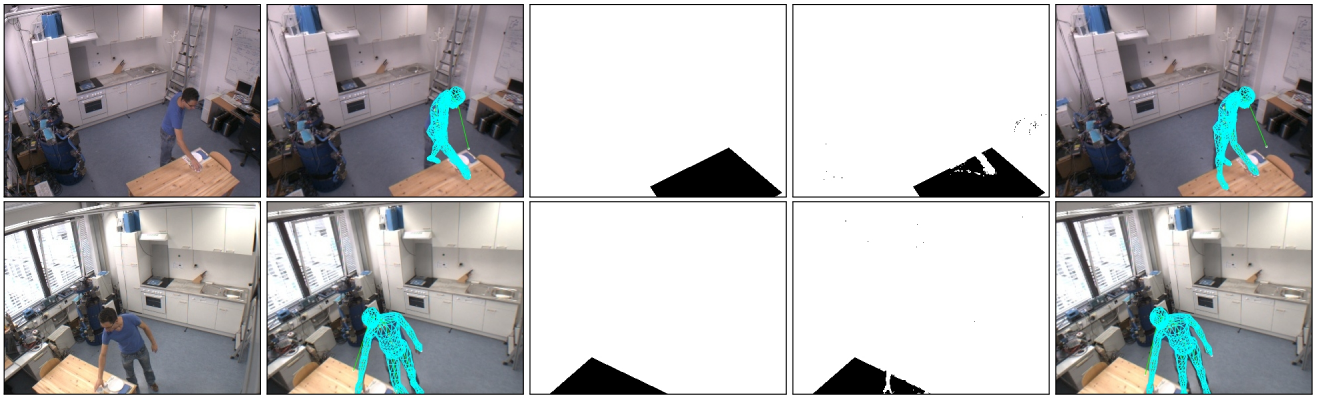

Fig. 18 Example frame with occlusion from table (from left to right): a) original image b) tracking results without using blocking layer and appearance c) original user-specified blocking mask I_B d) blocking mask I''_B after exclusion of human-like parts e) tracking results using our method. Each row shows one camera view. White color in the mask represents set bits (blocked regions are black).

We propose to use our general motion tracker to learn environment- and task-specific motion models over time. These can then be used to improve the prediction for repeatedly observed activities. Figure 19 depicts this strategy. In each timestep, we use the current motion history to create so-called *motion snippets* that represent a short motion pattern. These motion snippets are matched against the learned motion model to check for corresponding motion patterns. Whenever no match is found, we use the uninformed (and computationally expensive) motion model to create the new pose hypotheses. The final pose estimate is then used to self-train the observed motion and therefore to account for future detections of similar motion patterns. Whenever known motion patterns are detected however, we use knowledge about likely successors that is based on our previous observations to create potentially much more accurate pose hypotheses. Due to the improved prediction, we can then switch to a more efficient tracking step that requires less particle evaluations. Furthermore, by labeling the learned motion data, we can use correspondences between observed and trained motion patterns for activity recognition.

5.1 Spatio-Temporal Neighborhood Graphs

Our aim is to create generative models of human motion that are able to predict likely follow-up poses to the current pose estimate during recursive estimation. Many approaches try to improve prediction by learning motion models in terms of low-dimensional embeddings of training poses [65, 67]. Such models are global in that they process the full training set in one step to create a consistent embedding for all data points. However, these global constraints can lead to loss of relevant details in the data when compared to models that aim at local consistency of the data only [18]. For near-term prediction, local consistency is preferable.

We propose a graph-based representation of human motion that is inspired by ST-Isomap [34], a spatio-

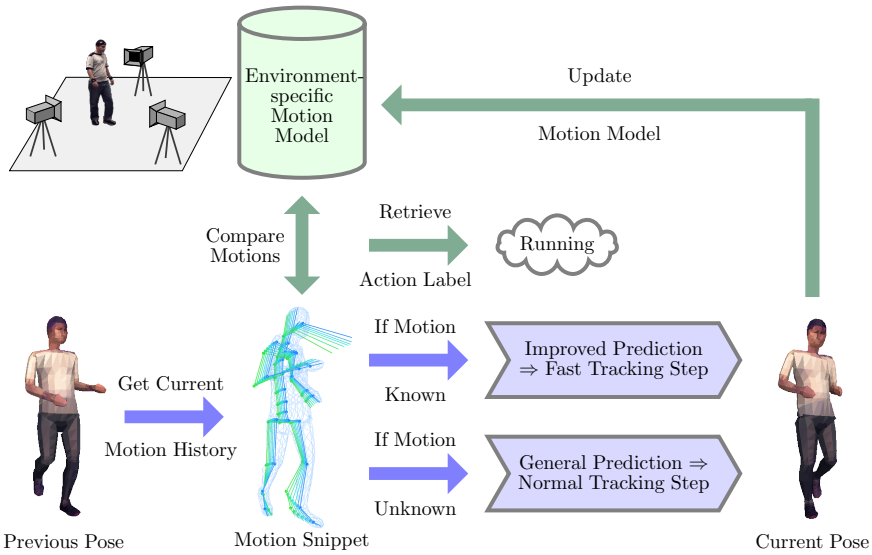

Fig. 19 Using environment-specific motion models to improve the prediction for human motion tracking and to recognize human activities. The motion model is trained with previously observed motion patterns that are typical for a specific environment. During tracking it is used to compare trained patterns with motion snippets that encode the short-term motion history of the last estimated pose. Whenever correspondences are detected, they are used to predict the current pose with greater accuracy using the information stored in the motion model. This allows us to use a computationally more efficient scheme for tracking. If no correspondences were found, a normal tracking step with a general motion model is performed. The newly estimated (unknown) pose is then used to update the motion model. In addition to its use for motion prediction, the motion model can be used to recognize actions or activities whenever semantic labels have been provided with the training data.

temporal extension of Isomap [61]. Isomap (and also ST-Isomap) is an approach for nonlinear dimensionality reduction that tries to estimate the intrinsic geometry of a set of data points in three steps. First, it creates an undirected weighted graph that connects all data points within a local neighborhood. Second, the *all-pair-shortest-path* matrix for the data points is computed from the graph, which contains the geodesic distances between all data points. Finally, a lower-dimensional embedding that tries to preserve the relative distances between all data points is computed by means of *multidimensional scaling* (MDS).

In our work, we omit the calculation of the *all-pair-shortest-path* matrix and the embedding using MDS. These steps serve to find a global embedding of the geodesic distance graph computed in the first step, and do not add additional information (in contrast, information might be lost when the dimensionality of the embedding is chosen too low). Furthermore, the computational savings of omitting the second and third step reduce the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ (or even lower when using approximate nearest-neighbor methods). This allows us to use more data points, which in turn results in a better approximation of the underlying manifold and thus in a better prediction. Another advantage is that the geodesic distance graph by itself can be incrementally updated.

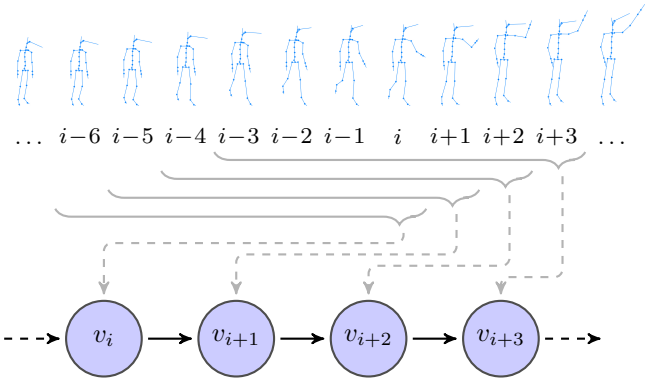

Fig. 20 Sequential creation of a spatio-temporal neighborhood graph. The graph is created from an ordered temporal sequence of s poses ($s = 7$ in this example). One vertex is created for each new pose and associated with a motion snippet based on the pose and its direct temporal history. Directed edges are added to encode the succession of poses. Note that the temporal windows encoded within each vertex are overlapping.

As in ST-Isomap, we use a directed graph structure to model the temporal progression of human poses. Training data is expected to arrive in sequential order, and we remove all poses that are not sufficiently different from their predecessor to ensure that the sequence of poses corresponds to actual motion. This is done by calculating the mean Euclidean distance be-

tween corresponding body joints of both poses (we set the threshold at 2 cm). This also helps to normalize the speed of motions, as more frames will be removed during slow motions when compared to fast motions. We then create a sequential graph of vertices from the remaining poses that is connected via directed edges from each vertex to its direct temporal successor. This process is illustrated in Figure 20. To capture the temporal structure of human motions, we associate each vertex not only with its corresponding pose, but with a motion snippet that consists of the pose and its short-term temporal history. Therefore, when computing the similarity of vertices, the corresponding temporal windows are compared. Note that the temporal windows are overlapping (Figure 20).

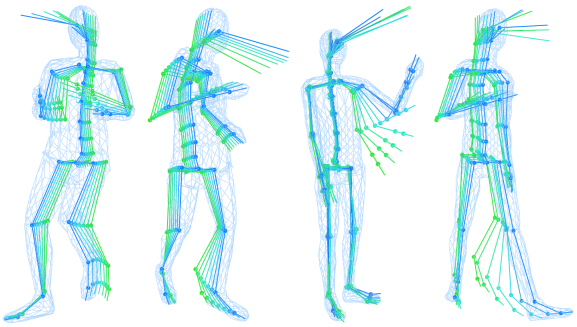

Fig. 21 Examples of spatio-temporal motion snippets. Motion snippets encode short patterns of human motion as a vector of 3D body joint locations of the current pose and its short-term temporal predecessors (marked by the dots in the images). All coordinates are relative to the origin of the current pose (*i.e.* its pelvis position), so that motion snippets can be recognized independently of their absolute position.

As stated, each vertex v_i is associated with a spatio-temporal motion snippet $\boldsymbol{\varsigma}^{(i)}$ that encodes a short-term motion pattern corresponding to the i-th pose. We use these motion snippets to compute a similarity measure between observed motion patterns. Figure 21 shows some examples. As can be seen, it is relatively easy for humans to assess the current motion and to predict the immediately following pose from these short pose sequences. Mathematically, a motion snippet is a vector $\boldsymbol{\varsigma}$ of body joint locations for s consecutive poses:

$$\boldsymbol{\varsigma}^{(i)} = \left(\boldsymbol{\rho}^{(i,-(s-1))^{\mathrm{T}}}, \ldots, \boldsymbol{\rho}^{(i,-1)^{\mathrm{T}}}, \boldsymbol{\rho}^{(i,0)^{\mathrm{T}}}\right)^{\mathrm{T}} \quad (10)$$

$$\boldsymbol{\rho}^{(i,j)} = \left(\boldsymbol{\tau}_{\mathtt{BEC}}^{(i,j)^{\mathrm{T}}}, \ldots, \boldsymbol{\tau}_{\mathtt{FIR}}^{(i,j)^{\mathrm{T}}}\right)^{\mathrm{T}} ; \quad -s < j \leq 0 \quad (11)$$

Here, $\boldsymbol{\varsigma}^{(i)}$ is computed from poses in its short-term temporal pose history, *i.e.* from all poses in the time interval $[\,i-(s-1) : i\,]$. It is a concatenation of the 28 body joint locations $\boldsymbol{\tau}_{\mathtt{bp}}^{(i,j)} = (x_{\mathtt{bp}}^{(i,j)}, y_{\mathtt{bp}}^{(i,j)}, z_{\mathtt{bp}}^{(i,j)})^{\mathrm{T}}$ of each of these s poses. The dimensionality d of a vector $\boldsymbol{\varsigma}$ is thus $d = 28 \cdot 3 \cdot s$. Note that the superscript $.^{(i,j)}$ of a parameter vector denotes the index i of the motion snippet and the relative index $j \leq 0$ of the pose inside its temporal window (with $j = 0$ being the current and most recent pose). The subscript $._{\mathtt{bp}}$ is the body part identifier for our model (see Figure 3).

An important aspect of spatio-temporal motion snippets $\boldsymbol{\varsigma}^{(i)}$ is that they are encoded in coordinates that are relative to the origin of the associated pose. In other words, all body joint locations are given in the pelvis coordinate system $\mathbf{H}_{\mathtt{BEC}}^{(i)}$ of the i-th pose. Therefore, motion snippets become independent of the position in the world at which they have been originally observed. This helps to improve the efficiency of the training set by removing the necessity to store similar patterns occurring at different spatial locations in the world. Note that all poses inside the temporal window of a motion snippet share the same coordinate system, so that the relative spatial extent of a motion is preserved (thus the term *spatio-temporal*, see Figure 21).

The distance measure $dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(k)})$ for comparing the vector representation of two motion snippets $\boldsymbol{\varsigma}^{(i)}$ and $\boldsymbol{\varsigma}^{(k)}$ is given as the mean Euclidean distance between all body joint locations encoded in the vectors:

$$dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(k)}) = \frac{1}{28 \cdot s} \sum_{j=0}^{s-1} \sum_{\mathtt{bp}=\mathtt{BEC}}^{\mathtt{FIR}} \|\boldsymbol{\tau}_{\mathtt{bp}}^{(i,j)} - \boldsymbol{\tau}_{\mathtt{bp}}^{(k,j)}\| \quad (12)$$

This formula based on the Euclidean distances of the joint locations in $\mathbb{R}^3$ gives us a very intuitive measure for the similarity of two motion snippets. In [7] we have evaluated the mean Euclidean joint errors and thus the accuracy of our tracking algorithm as approximately 2 cm. Thus we can assume that two motion snippets with a comparable distance are still very similar.

So far the graph corresponds to a linear list, as we only insert edges to denote the temporal succession in the training data. We also need to detect reoccurences of similar patterns and model these structural similarities in the graph. This enables us to provide multiple hypotheses about likely successors when a frequently observed motion pattern leads to different follow-up motions (*e.g.* when approaching a table, both the left or the right hand can be used to reach for an object). To model these cases, we adopt the strategy used in ST-Isomap [34], where two important types of neighborhood relations are distinguished (Figure 22). *Adjacent temporal neighbors* (ATN) are trivial relations where two poses are direct temporal neighbors. This relation is already captured in the initial graph creation. *Common temporal neighbors* (CTN) are non-trivial neighbors that correspond to the spatially closest points on each temporally distant trajectory that passes through

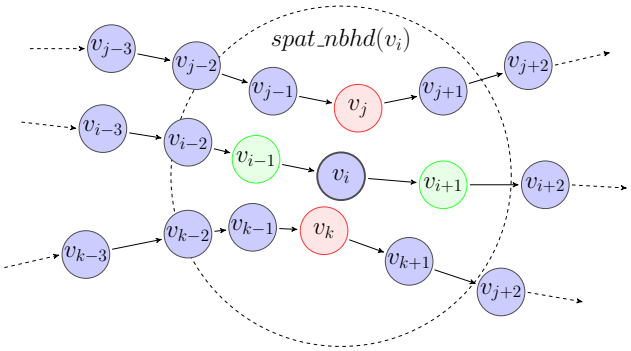

Fig. 22 Neighborhood relations in spatio-temporal neighborhood graphs. The neighborhood relations for a vertex v_i are defined by its spatial neighborhood $spat\_nbhd(v_i)$ and the temporal occurrence of vertices. *Adjacent temporal neighbors* (ATN) are trivial neighbors as they correspond to the direct temporal neighbors on the current trajectory (green vertices). *Common temporal neighbors* (CTN) are non-trivial neighbors that correspond to the spatially closest point on each temporally distant trajectory that passes through the spatial neighborhood $spat\_nbhd(v_i)$ (red vertices).

the spatial neighborhood of a vertex v_i. Whenever we detect CTNs (using Equation 12), we connect the corresponding vertices with edges in both directions to reflect their relation in the graph. *Motion graphs* as introduced by Kovar *et al.* [40] are also similar in spirit and have been used in a generative way to create realistic and controllable motion.

Let us quickly discuss alternative representations for motion snippets. Euler representations of joint angles could also be used to decouple motion patterns from their spatial occurence. However, they are redundant and do not provide intuitive distance measures. Quaternions are a slightly better alternative, but the remaining problem with all representations based on relative orientations is that the mean distance values can be deceiving in the case of articulated structures. Small distances can indicate both a similar pose as expected but also a completely differing pose due to error accumulations. Thus, we believe it is best to use absolute values that share a local reference frame, as is the case in our representation.

5.2 Incremental Training for Improved Prediction

During the recursive estimation of human poses, we need to predict the pose for the current timestep based on the last pose estimate. Our general motion model from Equation 1 predicts the new pose by adding random diffusion with body-part dependant variance to the last pose estimate. The variance of the diffusion must be large enough to capture fast movements in any direction. A more informed solution is to add the random diffusion to a prediction of where the current pose is expected to be. When we can assure that this prediction is closer to the requested pose than the last pose, we can reduce the variance of the diffusion without sacrificing estimation accuracy.

Assuming that we have already observed some motion snippets and incorporated them into our graph structure, we can distinguish between 3 relevant cases during each estimation step:

1. No motion is detected, *i.e.* the subject does not move.
2. An already known motion is observed.
3. An unknown motion is observed.

We treat these cases differently. Whenever no motion is observed at all, we can skip the estimation and just assume the last pose is still valid. Alternatively, we can perform one estimation step with very low variance to refine the last estimate. When we detect a familiar motion pattern that we have already learned, we can use this knowledge to create a potentially accurate prediction and then perform an estimation step with low variance to refine our prediction. Whenever we observe a motion pattern that we cannot assign to any of the learned motion snippets, we have to stick to our unconstrained tracking algorithm with full variance. We can then however incorporate the new pose estimate into the graph to extend our trained motion model, in case a similar pattern will be observed in the future. Algorithm 1 sketches the computational steps that are needed for improving human pose tracking with self-trained motion models.

We test whether motion is observed at all by using a simple change detection between two image frames. This is done by calculating the difference image and counting the number of pixels that are above a threshold.

In cases where motion has been observed, we first compute the motion snippet ς_c corresponding to the current motion history (line 4 of Algorithm 1). We then search for correspondences in our neighborhood graph (lines 5 to 10). Whenever we find a vertex v_i whose corresponding motion snippet differs from our current motion snippet by at most ε_{sn}, we add the pose associated with this vertex to a set of predicted poses. Additionally, we add all poses associated with the successors of v_i up to a traversal depth of $maxdepth$ to this set (we have been using $maxdepth = 3$ in our experiments). The reason for this is that we want to provide several likely options to increase our chances of hitting a good prediction. Note that all predicted poses have to be transformed such that the relative motion between the matching vertex v_i and the predicted successor of

Algorithm 1 Human pose tracking with self-trained motion models.

```
1: if NoMotionDetected() then
2:    RunTrackerMinimalVariance()
3: else
4:    ς_c = GetCurrentMotionHistory()
5:    predicted_poses = ∅
6:    for all v_i do
7:       if dist( ς_c, GetMotionSnippet(v_i) ) ≤ ε_sn then
8:          predicted_poses = predicted_poses ∪
            GetSuccessors(v_i, maxdepth)
9:       end if
10:   end for
11:   if predicted_poses ≠ ∅ then
12:      ClearParticleSet()
13:      AddToParticleSet( GetCurrentPose() )
14:      AddToParticleSet( predicted_poses )
15:      UpdateParticleWeights()
16:      RunTrackerLowVariance()
17:   else
18:      RunTrackerFullVariance()
19:      UpdateGraph( GetCurrentMotionHistory() )
20:   end if
21: end if
```

v_i is added to the base pose of the current estimate. We then perform one update step to weight the new predictions based on the current observation. As the first step during BIHS tracking will be a weighted resampling of the particle set, good predictions will multiply in the particle set while bad predictions will be eliminated. Using this particle set, we then run a BIHS step of our tracker with reduced variance and particle count (we use a 4th of the particles and the variance).

It should be noted that recognizing a common motion history does not guarantee that there will be a common future. However, we observed that motions do not deviate instantly, but rather over a short period of time. The reduced variance tracker is able to account for these incremental deviations until the motion history eventually differs from all known motion patterns.

When the current motion does not match our trained data we switch to unconstrained tracking with full variance. After the new pose has been estimated (assuming it is sufficiently different from the last pose), we use it to update the graph (line 19). A new vertex is inserted either at the end of the last insertion sequence, or after the particle that won the last prediction step. To prevent the new motion to be immediately recognized as a correspondence in the following timesteps, we buffer newly created vertices and add them to the graph with a short delay of about 2 sec.

6 Experimental Results

We will now present and discuss the experimental evaluation of the presented system. Note that corresponding videos to our experiments can be viewed online at http://memoman.cs.tum.edu.

6.1 HumanEva Data Set

The HumanEva data sets [58] are two publicly available data sets for the evaluation of human motion tracking algorithms, with corresponding ground-truth measurements as provided by a marker-based motion capture system. We have evaluated our system on the newer HumanEvaII data set. The ground-truth data is kept back by the authors, to prevent misuse such as manual modifications of tracking results, or the use of strong motion-specific priors in order to reduce the complexity of the problem. The mean Euclidean joint errors for evaluation are computed remotely via a XML-based web-interface.

The HumanEvaII sequences consist of a person walking in a circle, then accelerating into a jogging motion, and finally changing into a slightly unusual stretching motion (Figure 23). The transitions between the three types of motion are especially critical for tracking algorithms that rely on strong motion priors, *e.g.* when a motion model has been learned from exemplars of walking motions only. We have adapted our human model manually to fit the observed human shapes (with loose clothing) as good as possible.

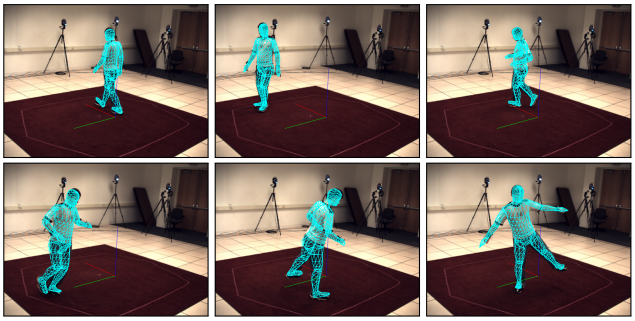

Fig. 23 Tracking results on the HumanEvaII S4 sequence for the BIHS strategy (10 layers, 800 particles). We show random frames for one of the four camera views.

Our first experiments aimed at comparing our BIHS sampling strategy with several variants of state-of-the-art sampling algorithms like APF or PS (see Section 3). The resulting mean 3D joint error plots for sequence S4 of the data set are depicted in Figure 24. Note that we have omitted the estimation of the lower spine parameters as this results in more stable estimates for most

kinds of upright motions (we have done so in particular to favor the APF and PS strategies; BIHS is shown in Figure 25 to compensate well for the additional d.o.f.).

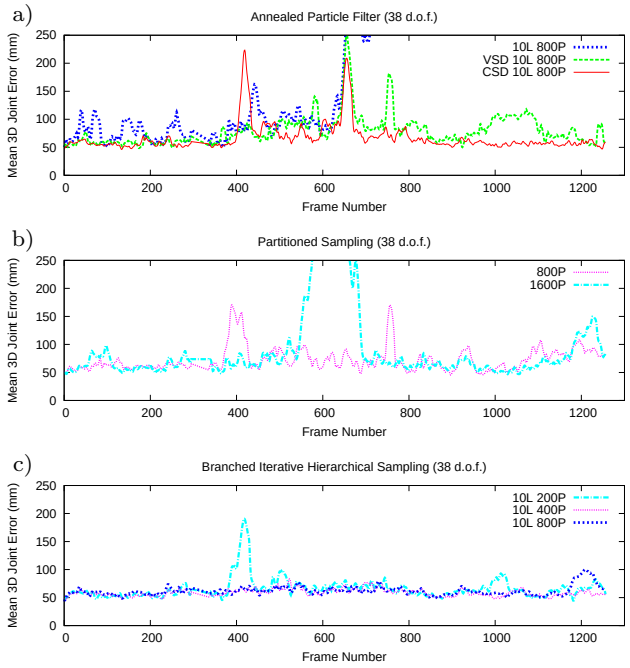

Fig. 24 Tracking results on the HumanEvaII benchmark (sequence S4) for: a) the APF with 10 layers and 800 particles, once without scaled diffusion, once with variance scaled diffusion (VSD), and once with covariance scaled diffusion (CSD); b) PS with 11 partitions and 1600 or 800 particles; c) BIHS with 10 layers and 800 or 400 or 200 particles. Even when using only about a quarter of the particle evaluations, BIHS outperforms APF and PS and shows robust tracking behavior despite noisy image observations. Note that there seems to be a systematic error (the error never drops below 5 cm) that can be attributed to the differences in relative joint positions between the ground-truth model and ours. Visually, our BIHS method delivers near perfect results.

We first evaluated the APF (Figure 24a) with three different variants for setting the amount of diffusion applied to each body part with each iteration. In the initial version, the diffusion is reduced using a constant factor ($\alpha_M = \alpha_{M-1} = \ldots = \alpha_1 = 0.5$). In the second version, the amount of diffusion is controlled using *variance scaled diffusion* (VSD), *i.e.* according to the state variance in the particle set from the last iteration. In the third version, particles are diffused using the state covariance matrix from the particle set in the last iteration. This corresponds to *covariance scaled diffusion* (CSD) as presented in Section 3.2. The last strategy provides the best results, although the overall performance of the APF is unsatisfying. A visual inspection of the results shows that while the torso is correctly estimated most of the time, the limb positions are often wrong.

We then tested PS (Figure 24b) at comparable processing times (PS with 1600 particles $\sim$ APF with 10 layers and 800 particles). The order of partitioning is pose and lower torso, upper torso, left upper leg, left lower leg, right upper leg, right lower leg, left upper arm, left lower arm, right upper arm, right lower arm, and head. The tracking results are again unsatisfying, which we attribute to the large ambiguity in the local weight function when evaluating the lower torso. We have observed that contrary to the good results on simulated sequences with perfect observations [7], the PS strategy is strongly affected by noisy or partially erronous observation data.

Finally, we have evaluated the novel BIHS strategy as presented in Section 3.4 (Figure 24c). Recall that we derived this strategy from the observation that APF manages to find good estimates of the torso parameters but fails in estimating the limbs, while PS is good at estimating the limbs in case that it got the torso parameters right. Despite the often noisy and imprecise foreground masks I_F that are segmented from the HumanEvaII sequences, BIHS is capable of producing estimates with constantly low mean errors around 50 mm. Even variants that use only a quarter of the particle evaluations when compared to APF or PS still give better results, although the robustness is slightly reduced. We believe that the true mean errors for the BIHS strategy are even lower, but there seems to be a systematic error that can be attributed to the differences in relative joint positions between the ground-truth model and ours. This explains why the mean errors never drop below 50 mm during 1250 frames, an observation that is unlikely given chance. We have shown in a previous experiment with simulated ground-truth observations that the true achievable accuracy is around 25 mm [7].

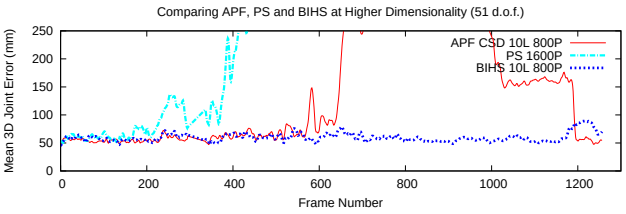

Fig. 25 Comparing accuracy and reliability of APF, PS and BIHS when tracking the full 51 d.o.f. of our human model, including hands and feet. BIHS is nearly unaffected by the increased dimensionality when compared to Figure 24, whereas both APF and PS fail.

We repeated the experiments on sequence S4 using the best variants of each algorithm, this time tracking the full 51 d.o.f. of our model, including the lower

spine, hands and feet. Both APF and PS completely loose track early in the sequence, while our BIHS strategy provides almost the same quality results as in the 38 d.o.f. case (Figure 25). The reasons for this are the hard partitioning of the search space, where the additional parameters are estimated last without obscuring previous results. To the best of our knowledge, the presented BIHS strategy is the first Bayesian approach to have shown successful tracking of such high-dimensional articulated models without the use of training data. All sequences have been processed without reinitializing the model pose and using the same tracking parameterizations.

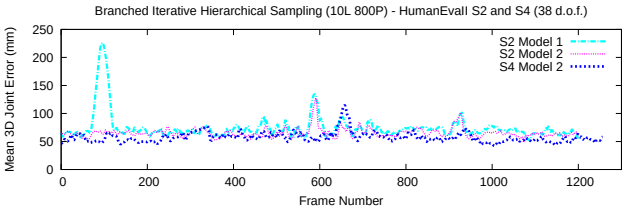

Fig. 26 Tracking results for BIHS on both sequences (S2 and S4) of the HumanEvaII benchmark in combination with different human models. The tracking quality can be affected when the inner shape parameter ϕ_I (*i.e.* the estimates of the limb lengths) are imprecise, as has been the case for model 1 for subject S2. Model 2 for subject S4 is also less accurate than the model used for the results presented in Figures 24 and 25.

In Figure 26 we show two tracking results on sequence S2 with two different model adaptations, and another tracking result on sequence S4 with a different model than the one used in Figures 24 and 25. There is a slight decrease in tracking accuracy for some models, that comes from inaccurate model adaptations. The results confirm our believe that accurate models of the tracked humans are integral to the tracking accuracy and robustness.

Self-Training: In our next experiment we have used the HumanEvaII S4 sequence to evaluate the self-training capacity of our system. The sequence consists of three different classes of repetitive actions that make it well-suited for evaluating the incremental learning strategy. During the first 380 frames, the subject is *walking* slowly in a circle, before accelerating into a *running* motion that lasts until frame 820. The final segment consists of a random *exercise* that shows no clear repetitive patterns. No training data has been provided for this sequence, *i.e.* the tracking started with an empty neighborhood graph that was incrementally growing as new motions were estimated.

Figure 27 shows some examples from the processed sequence where the self-trained motion model was used

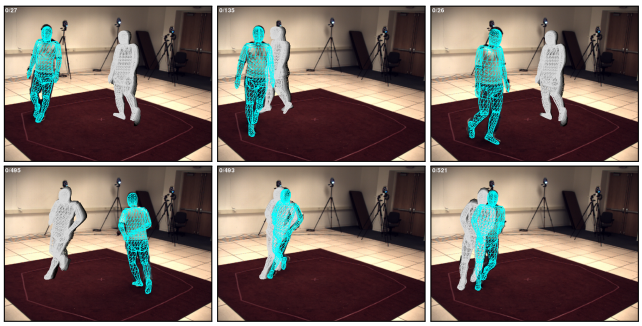

Fig. 27 Screenshots from the HumanEvaII S4 sequence when using incrementally learned motion models for improved prediction. The detected motion snippet correspondences (greyscale) have been trained earlier in the sequence and increase the computational efficiency when dealing with previously observed motions. The tracker first learns a sequence of motion snippets corresponding to *walking*, and later a sequence corresponding to *running* (see also Figure 28). As can be seen, motion snippets are detected independently of their absolute position, which increases detection rates and improves motion prediction.

to predict the motion for the next timestep. The final pose estimate is shown in blue, whereas the learned motion snippet that was used for prediction is shown in greyscale, with the temporally preceding poses fading to black. Due to the incremental learning of our motion model, the detected motion snippets correspond to motion patterns that occurred earlier in the same sequence. As can be seen, the global occurrence of the pattern is not encoded in our motion snippet representation. This allows us to detect matching patterns independent of the location they have been observed at, which increases the detection rates for correspondences and reduces the redundancy of motion snippets stored in the graph.

We have analyzed the processed sequence in detail to see how our new motion model influences the quality of pose estimates and the computational efficiency of our tracker. Figure 28a shows the errorplot with the mean Euclidean joint errors for our pose estimates. The quality is comparable to our previous experiments. Note that we did not expect the quality to improve when using the learned motion models, as the parameters of our tracker are set such that they are able to retrieve accurate estimates even in the uninformed case. Instead, we expect that the computational efficiency of our tracker improves once it can access the learned motion patterns. As shown in Figure 28b, this is exactly what happens once motion snippets for the walking motion and for the running motion have been self-trained by the model. When no correspondences are found, the estimation time per frame is around 15 sec (blue pulses). This changes once motion correspondences are detected and the tracker is able to switch to low-variance mode. In

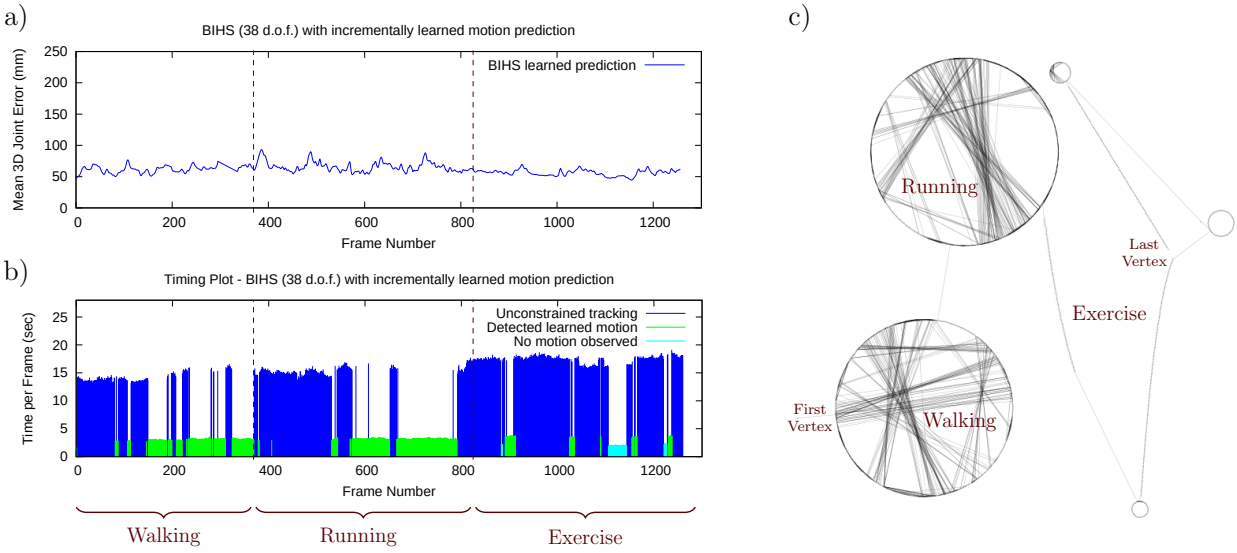

Fig. 28 Errorplot, timingplot and learned motion graph for the HumanEvaII S4 sequence when using incrementally learned motion models for improved prediction: a) the errorplot shows constant accuracy over the whole sequence b) the timing plot shows the improved efficiency of our tracker once models of the walking and running motion have been learned c) a 2D visualization of the learned motion graph (715 vertices, 1617 edges) shows the clear separation between the walking, running and exercise motions, indicating its potential use for motion segmentation and action recognition. The graph has been self-trained without any prior information on the motion.

these cases, estimation time drops to around 2.5 sec per frame (green pulses). When transitioning from walking to running, and later from running to the exercise motion, the new motion patterns force the tracker to switch to the general motion model until the new motions have been trained and added to the graph. Due to the random nature of the exercise at the end of the sequence, only a few correspondences are found and most of this segment has to be processed using the general motion model. Overall, the processing of 40.1% of the frames could be accelerated due to the self-training of the repetitive motion patterns.

The immediate loss of motion snippet correspondences when transitioning from walking to running and from running to exercise indicates that these motions show strong stylistic differences. In Figure 28c we have plotted the spatio-temporal neighborhood graph in two dimensions to see if these differences are reflected in the structure of the graph. As can be seen, both the walking and the running segments are directly recognizable when looking at the visualization. All edges that connect CTNs are restricted to the respective subgraphs that contain all vertices for either the walking or the running motion. These two subgraphs are connected by just a bridge. The vertices corresponding to the exercise motion are less distinctive due to the non-repetitive pattern of the exercise. The visualization shows that spatio-temporal neighborhood graphs have the potential to be used for motion segmentation or activity recognition, as structural similarities are well-reflected in the graph.

6.2 TUM Kitchen Data Set

The second setup in which we evaluated our methods is the Assistive Kitchen [8] at the TUM. This kitchen environment is observed by four ceiling mounted cameras and a smart sensor network that is embedded into the environment, consisting of additional sensors such as laser scanners, RFID readers or magnetic (reed) sensors.

We recorded several sequences of persons that were told to set the table. This involved pick and place actions of objects that were located at different positions in the environment. Most of the sequences range around 1500 frames, but we also recorded longer sequences of random manipulation tasks of up to 5000 frames. We processed a total of 21 sequences from 4 different subjects with our markerless motion capture system, see Figure 29 for screenshots from one of the sequences. The tracking results on these sequences are accurate and reliable despite frequent interactions with the environment, object manipulations, and partial occlusions. Some of the sequences required small amounts of post-processing due to partial tracking failures. The most common failure appeared when subjects were reaching into the rightmost cupboard, and the lifting of the right arm was missed by the tracker. The cause for this is that

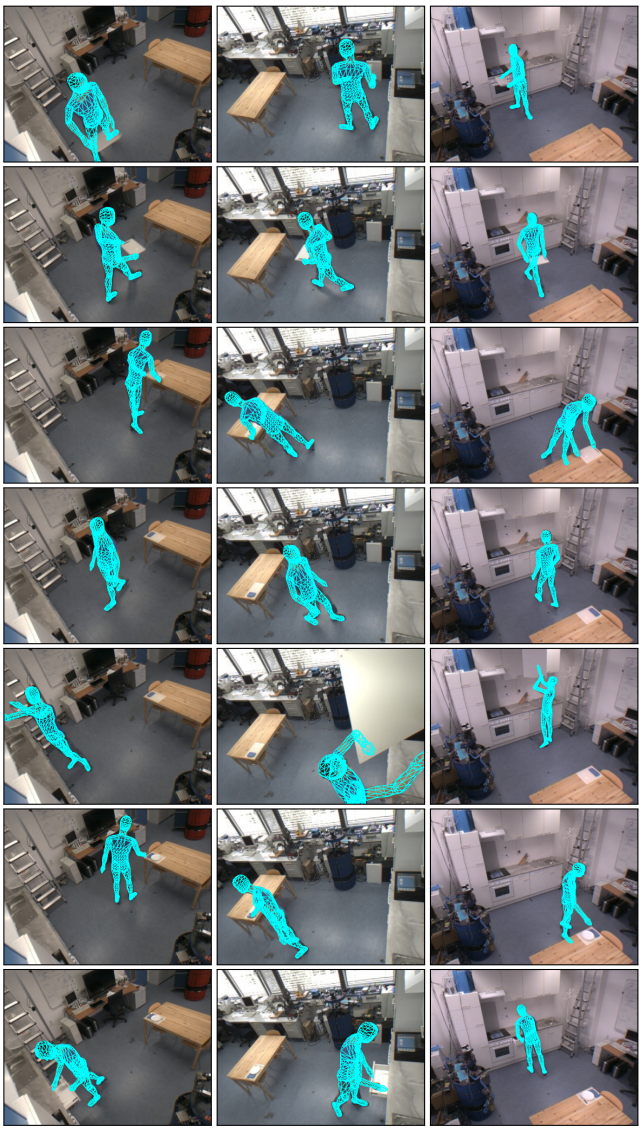

Fig. 29 Screenshots from a kitchen sequence (1500 frames or 60 sec). Each column corresponds to one camera view (3 out of 4 shown). Notice the interactions with objects, drawers and cupboards.

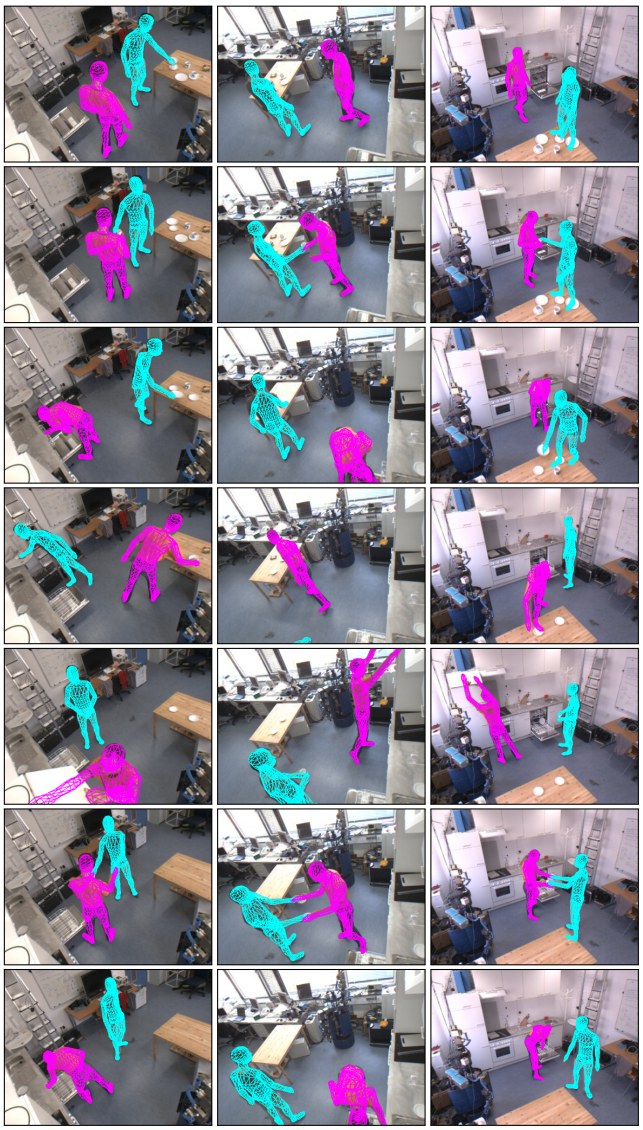

Fig. 30 Screenshots from a kitchen sequence featuring the joint action of two subjects (1300 frames or 50 sec). Each column corresponds to one camera view (3 out of 4 shown). The two subjects perform the joint action of clearing the table and loading the dishwasher.

the lifting motion is not visible in any of the cameras due to a blind spot caused by bad placement of the cameras. Other infrequently observed failures are that one of the arms temporarily sticks to the body or that legs get crossed for a couple of frames only.

We decided to provide the original video sequences including the retrieved motion capture data to the research community. The resulting TUM Kitchen data set [62] is publicly available for download from http://kitchendata.cs.tum.edu. In addition to the original calibrated video sequences and our motion capture data, we have added synchronized sensor readings from the sensor network in the Assistive Kitchen. These consist of RFID readings from three fixed readers embedded in the environment (table, counter-top and cupboard) to detect the location of tagged objects such as plates or cups, and of magnetic (reed) sensor readings to detect whether doors or drawers in the environment are opened or closed. In combination with the additionally provided fine-grained semantic action labels (for the torso and for both hands separately), this data set is equally suited for the evaluation of algorithms for human motion capture as well as for motion segmentation or activity recognition. One of the key advantages of our data set is the high level of realism in both the modeling of the scene and the type of activities that has been recorded.

Joint Activities: In a separate experiment we have recorded a sequence where two humans jointly clear the table and load a dishwasher. During this joint activity, several objects are being handed over between the subjects. One of the subject opens the dishwasher to fill it with the objects it receives, and afterwards closes it again. Furthermore, some objects are placed back into the cupboards. We have used the concept of layered environment models as a means to distinguish between multiple humans acting in the same scene. Usually, the combined segmentation of several human silhouettes results in a merged shape that will confuse shape-based trackers. We can use the appearance models to soften this effect by considering the colors corresponding to each human. Of course, this requires that the subjects wear visually distinctive clothes. The respective other subject is then treated as a dynamic object and filtered from the segmentation. However, the filtering is not perfect due to unavoidable overlap in colors between the humans, *e.g.* for the skin colors.

Each subject was initialized and tracked individually, and the resulting poses were overlaid with different colors for the final video sequence shown in Figure 30. As the screenshots indicate, the sequence was tracked with good accuracy and very little imperfections. Despite the motion of the subjects that occlude each other several times, the tracking quality is comparable to that of single subject tracking.

Activity Recognition: Finally, we will present experimental results on activity recognition using the action labels of the TUM Kitchen data set. We have used nearest-neighbor classification as a proof-of-concept that the spatio-temporal motion snippets from Section 5.1 are suitable for the recognition of challenging and subtle activities. More sophisticated methods will likely involve the use of graphical models such as *hidden markov models* (HMM) or *conditional random fields* (CRF) to incorporate transition probabilities between activity states. These extensions should be straightforward given the information contained in our learned graphs.

The TUM Kitchen data set provides ground-truth labels for the actions of the left hand, the right hand and the trunk separately to take into account the high degree of parallelism in the observed activities [62]. Often, one hand is opening a cupboard door, while the other one is reaching towards a cup, and while the body is still moving towards the cupboard. Each of the sequences 1-0 to 1-2 used in the following experiments corresponds to one observed instance of a table setting task of the same subject. The table setting activity can be further segmented into several subactions, such as taking something, carrying something, or putting down something. These subactions match the granularity of the semantic labels.

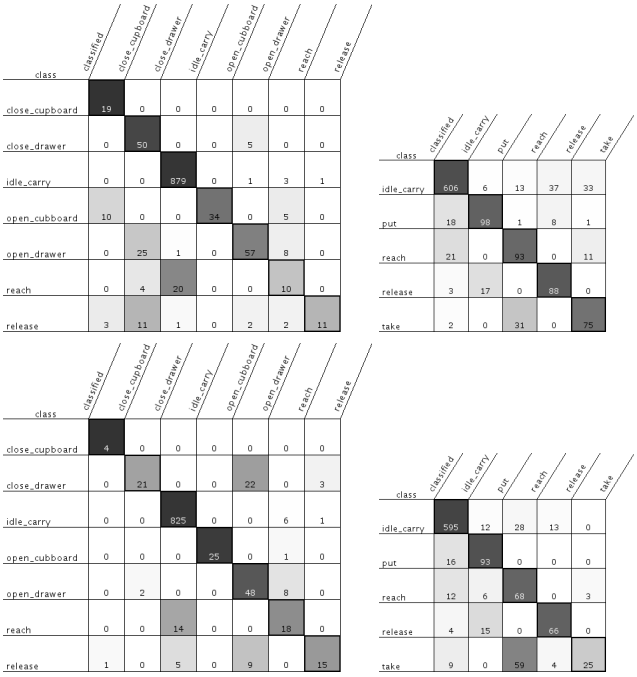

Fig. 31 Confusion matrices for activity recognition on the TUM Kitchen data set. Shown are the matrices for the left hand (left column) and right hand (right column) labels of sequence 1-1 (top row) and 1-2 (bottom row). Sequence 1-1 has been trained with sequence 1-0, sequence 1-2 with sequences 1-0 and 1-1.

For each test sequence, semantic labels for each frame have been assigned according to the best-matching known motion snippet. Figure 31 depicts the resulting confusion matrices for the sequences 1-1 and 1-2 of our data set for the left hand and right hand labels. As can be seen, the majority of retrieved labels are correctly assigned. The overall precision and recall rates range around 90.0% for the left hand and 82.0% for the right hand actions. Typical confusions include *reach* misclassified as *idle carry* (which corresponds to a catch-all class), and the frequently confused *open drawer* and *close drawer* actions. Furthermore, the *release* action is often falsely classified. For the right hand, the most common errors include *take* misclassified as *reach*, *release* misclassified as *put*, and *put* and *reach* misclassified as *idle carry*.

We have found some typical causes for these misclassifications. First, the semantic labels take into account the handling of objects, *e.g.* the *reach*, *take*, *put* and *release* labels implicitly assume that an object is being manipulated. However, as we only compare the motion patterns of a human without taking into account any object detections, these actions cannot al-

ways be clearly distinguished. In addition, *take* actions always follow immediately after *reach* actions, and *release* actions follow immediately after *put actions*. Because the transitions are seamless, this is a common cause of confusion. Another problem is that the temporal extent of motion snippets is far shorter than the extent of the labeled actions. Thus, it can happen that similar motion snippets are observed for different actions, which is the case for the *open drawer* and *close drawer* actions. When opening, the pattern corresponds to reaching for the drawer and pulling it out. When closing, it corresponds to pushing the drawer and retracting the arm. Both actions share the almost identical motion pattern of moving the arm forward then backward again. Distinction between these actions could be achieved by modeling the temporal succession of snippet detections (*e.g.* using HMMs) or through complementary sensor input from RFID readers (for object detections) or magnetic sensors (for detecting the states of cupboards or drawers). Note that the observed confusions are irrelevant when using the motion snippets for prediction, due to the similar motion patterns that result in similar predictions for the tracker.

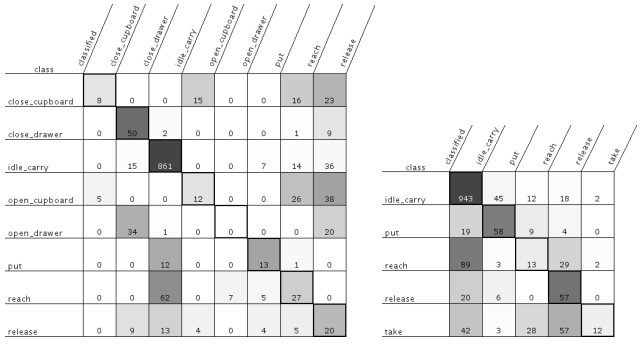

Fig. 32 Confusion matrices for cross subject activity recognition (sequence 1-7 of subject S03). The motion model has been trained with sequences 1-0 and 1-5 of subject S01. Confusion matrices are shown for the left hand (left) and right hand (right) labels.

The experiments presented so far have been trained and tested on the same subject. Figure 32 shows the confusion matrices for a cross subject activity recognition. The training and test subjects differ in body height by about 25 cm and their style of motion is very different. To normalize the motion patterns, we have performed all calculations of the body joint locations of each motion snippet based on a mean human model. As expected, confusions are now much more evident. Still, the overall precision and recall rates amount to 74.0% for the left hand and 78.8% for the right hand actions. None of the detections were good enough to be used for improving the prediction of our motion tracker, though. We take these results as an indication that it is best to learn person-specific motion models and to switch between these models based on an external recognition component.

6.3 Other Sequences

We have evaluated our algorithms on several other challenging sequences. A particularly noteworthy sequence corresponds to an ergonomic case study of a human getting into and out of a car mock-up. The mock-up serves as an approximation of a passenger cabin and consists of a driver's seat, armatures, steering wheels, foot pedals, gear shift, and a surrounding skeleton made from steel bars. Due to the missing body panels, mock-ups are well suited for recording the human subjects from different viewpoints. Still, the scene features large amounts of occlusions (up to 50% per camera view) such that shape-based observation models without additional environment modeling are bound to fail (Figure 2). Using our layered environment model, we have been able to track the subject through the whole sequence at good accuracy (Figure 33).

Another interesting sequence was recorded outdoors under direct sunlight and features an athlete exercising in shot-put (Figure 34). The observed action is performed in one sudden and fast motion, and had to be recorded at 60 Hz to avoid motion blur. The scene was processed at *Full HD* resolution (1920×1080) in one go. The estimated motion is smooth and fits very well to the observed human, with the exception of the left arm that lost track early in the sequence. After closer investigation, we noticed that the left arm was kept close to the body during the first half of the sequence. Due to suboptimal placement of the cameras (two of the three cameras were nearly opposite of each other, which resulted in mirrored silhouette shapes), the left arm was not visible in any of the binary foreground masks. Thus, the tracker made uninformed estimates of the arm's motion and was not able to recover the arm's position once it suddenly reappeared during a fast swing motion away from the body. This shows that at least three cameras with substantially differing and non-opposite viewing directions are needed to resolve the ambiguities resulting from self-occlusions of the human body. On a side note, the strong shadows on the ground caused by the direct sunlight did not distract the tracker, as they were well separated from the human silhouette.

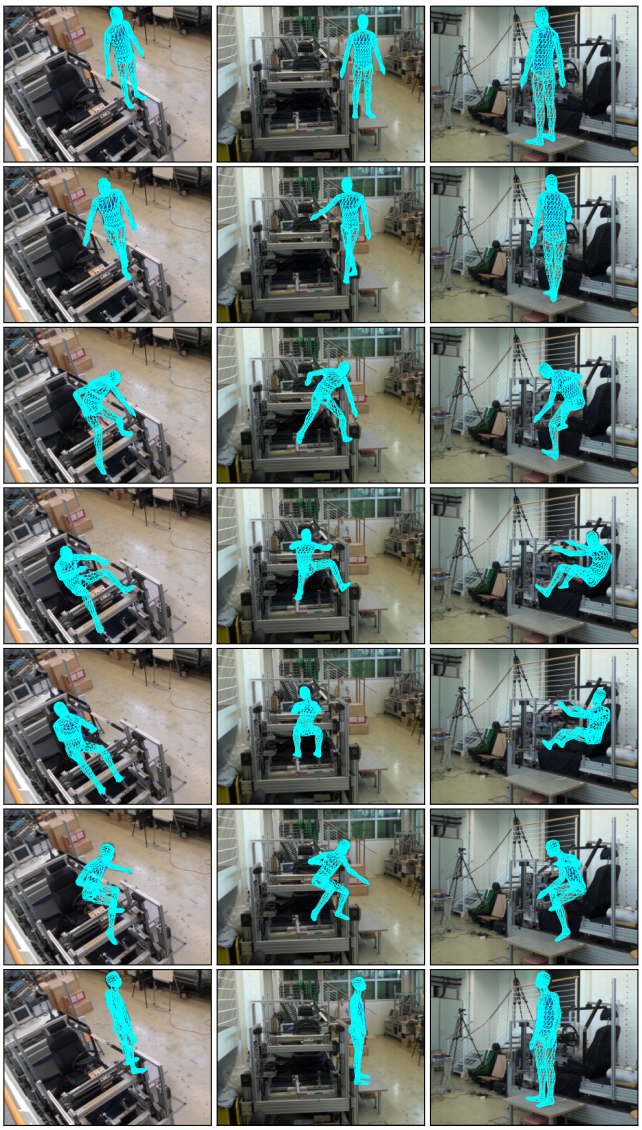

Fig. 33 Screenshots from a car mock-up sequence (2500 frames or 100 sec). Each column corresponds to one camera view (3 out of 4 shown). Notice the occlusions from the environment, that are strongest in the second and third column.

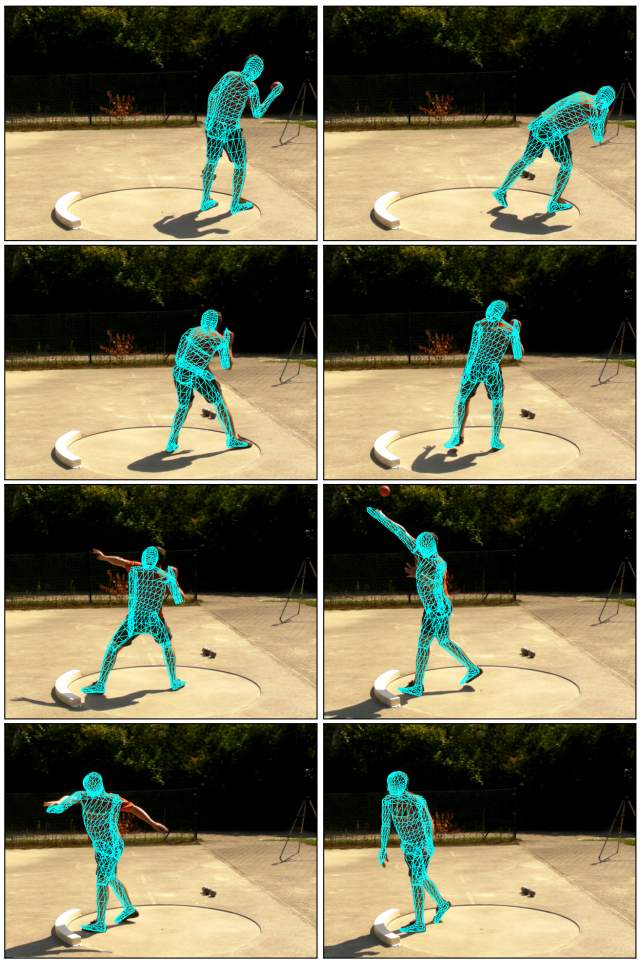

Fig. 34 Screenshots from an outdoor shot-put sequence (580 frames or 10 sec). The motion is extremely fast and had to be recorded at 60 Hz. The estimated motion is realistic and smooth, with the exception of the left arm that lost track early in the sequence. We attribute this to a bad placement of the cameras on our behalf, as two of the three cameras used were placed almost opposite of each other. The resulting pose ambiguities of the left arm when moving close to the body could not be resolved. Images have been cropped.

7 Related Work

A large body of related work is available both in the context of human motion capture and human activity recognition.

Markerless Human Motion Capture: Early approaches on markerless vision-based human motion capture mostly target monocular tracking and mostly rely on two-dimensional human models [16,35]. These are often made up of image patches that correspond to individual body parts and that are connected at the body joints. Felzenszwalb and Huttenlocher introduced *pictorial structures* [22,54] that are related graphical models where the spatial arrangement of parts as well as their appearance is parameterized by probability distributions.

Agarwal and Triggs [1] extract more detailed and precise 3D poses from monocular camera views by learning a mapping from observed shape features to the corresponding pose parameters of human models. The mapping is learned from humans silhouettes that have been synthetically generated using available human motion capture data (a similar approach using multiple cameras is taken by Grauman *et al.* [27]). Statistical inference is then used to recover unknown 3D pose parameters from the mapping and extracted foreground silhouettes. A related approach by Howe *et al.* [32] tries to compensate for missing 3D information by learning

patterns of human motion from training data. Sigal and Black [59] first estimate 2D postures in a bottom-up approach before attempting to classify the corresponding 3D motion sequence.

As only small subsets of all possible human motions (*e.g.* walking) can be learned at reasonable expense, several methods use more than one camera view to obtain precise 3D measurements of unconstrained human motions. Stereo cameras can provide a compromise to acquire the missing 3D information for accurate pose estimation [51,28,5], still they remain viewpoint-dependant to some point. Approaches that use 3D models with multiple distributed cameras are usually *top-down* approaches, *i.e.* the pose parameters of the model are first predicted and then evaluated based on the available image information.

Probabilistic approaches try to infer the solution using recursive Bayesian estimation. The solutions are usually approximated using sequential *Monte Carlo* methods (*particle filters*). Common approaches such as *sampling importance resampling* fail due to the high dimensionality of the problem. Deutscher and Reid [19] proposed the *annealed particle filter* that is related to the method of *simulated annealing* [37] to evolve the particle set towards the global maximum of the weight function. MacCormick and Isard [44] introduced *partitioned sampling* to estimate the parameters of articulated models by splitting the state space into hierarchical partitions that can be evaluated sequentially, thus reducing an initially high-dimensional estimation problem into several lower-dimensional problems. Mitchelson and Hilton [46] presented a variant of hierarchical sampling where partitions are evaluated in parallel to increase efficiency.

Other work uses deterministic as opposed to stochastic methods. Here, the parameters are estimated based on nonlinear optimization [14,36,24,39]. A good initial estimate is necessary to avoid getting stuck in local minima of the objective function. Rosenhahn *et al.* [55] use several randomly chosen starting points for the optimization to reduce the risk of getting stuck in local minima. Bray *et al.* [13] circumvent this problem by combining nonlinear optimization with *Monte Carlo* methods. Ivekovič *et al.* [33] presented an approach that is closely related to ours, where they use *particle swarm optimization* to track upperbody movements. Other approaches use variants of the *iterative closest point* (ICP) algorithm to register the human model to a 3D point cloud [29,50,38].

Several approaches build 3D reconstructions of the human surface from multiple cameras (so-called *visual hulls*). They can be used to measure the deviation between the surface of the human model and the nearest reconstructed 3D points [45,36,31]. Other related approaches do not use explicit models at all and estimate the body joint positions directly from clusters of 3D points that share the same motion patterns [17,2]. Methods that utilize *visual hulls* require a large density of cameras in the environment to be able to reconstruct the observed subject at good accuracy (usually a minimum of 8 cameras).

We can also distuinguish approaches by the 3D human models used, that often differ in the way the surface is modeled. Volumetric outer models approximate body parts with geometric primitives such as ellipsoids [70], cylindrical shapes [66,38] or truncated cones [19], superquadrics or superellipsoids [36]. In contrast to such implicit surfaces, some models use explicit surfaces such as polygonal meshes [55,23]. These can be derived from body scans of the subject and often seem more realistic than their implicit counterparts. Plänkers and Fua [51] and Horaud *et al.* [31] use smooth implicit surfaces that provide a level of realism that is comparable to explicit surfaces through blending of several implicit geometric primitives.Anguelov *et al.* [3] introduced the SCAPE model [48,6,30] whose surface parameters have been learned from 3D laser scans of several human subjects in varying poses. The resulting surface mesh is deformable with respect to body size and posture, and enables accurate display of human shapes up to the level of muscle contractions. SCAPE is not associated with an inner kinematic structure, although extensions have been presented to estimate the underlying skeleton [2]. The impressive realism of the model comes at increased computational expense, which makes it difficult to apply SCAPE-like models for tracking tasks.

To cope with the high dimensionality of the state space for human poses and the resulting difficulties in estimating these parameters, many approaches make use of machine learning techniques to solve the task. In the simplest case one can learn strong motion priors for specific motions to limit the search space when tracking. Some approaches project training motions from the high-dimensional state space to a low-dimensional manifold, using techniques such as *principal component analysis* [65] or *Gaussian process dynamical models* [67]. The low-dimensional manifold can then be used in a generative manner to create predictions for the next states.

Another class of exemplar-based methods [27,1,64, 10,12] learns a direct mapping between observed image features and the corresponding human poses, which makes them easy to implement and to apply. Such methods are constricted to the type of features they have been trained with and training has to be redone

when transferring to environments with different camera perspectives.

A more comprehensive summary of related work on markerless human motion capture is provided by the surveys from Gavrila [25], Moeslund *et al.* [47] and Poppe [52].

Activity Recognition: Activity recognition aims at assigning higher-level semantic labels to human motion patterns to facilitate the recognition of ongoing activities and intentions. We will distinguish between two research directions.

Model-free methods provide a direct mapping between image cues and action classes without intermediate parameter estimation steps. Holistic approaches consider image information as a whole, *e.g.* by using silhouettes [11], edge images [68] or optical flow fields [20]. Bobick *et al.* [11] have introduced *motion-history-images* as a 2.5D representation of actions, where silhouettes are overlaid on a single image with brightness corresponding to the distance in time. Blank *et al.* [9,26] create *space-time-shapes* from extracted silhouettes by using time as the third dimension. A similar technique was proposed by Yilmaz and Shah [71] as *spatiotemporal volumes*. Weinland *et al.* [69] create view-independent representations of human motions by fusing silhouettes from multiple cameras to obtain visual hulls, that are then augmented with temporal information to form 3.5D *motion-history-volumes* (extending *motion-history-images* [11] by another dimension). In contrast to holistic approaches, patch-based approaches extract only selected salient features and group them for classification, which helps to overcome problems in the case of partial occlusions [49].

In model-based approaches, parameters of a representative model are determined in a first step before performing action classification in phase space. The most commonly used parameters for action classification are joint angles or joint positions of articulated human models [57,42]. Such parameters are invariant to translations, scale and rotations, and form a rich and comprehensive representation of human movements. They are well suited for recognition and provide high recognition rates even with more subtle motions. However, the model parameters are difficult to extract by unintrusive means, and most methods depend on commercial marker-based motion capture systems for parameter retrieval.

More detailed surveys on human action recognition are provided by Krüger *et al.* [41] and by Poppe [53].

8 Conclusion

We have presented a system for markerless human pose tracking and activity recognition that is capable of estimating complex human motion with high accuracy. It is based around an anthropometric human model of up to 51 d.o.f. that provides a detailed representation of human shape and posture, and that we have optimized towards efficient use for tracking. We have shown how reliable and accurate tracking is possible despite an unconstrained motion model and a high-dimensional state space. For this, we proposed to combine concepts from hierarchical particle filtering with stochastic optimization, and we have discussed the importance of a highly diversive exploration strategy. Finally,our system is self-adaptive by learning environment- and task-specific motions over time, which helps to improve performance and which can be used as a basis for model-based activity recognition, and thus for higher-level reasoning about human intentions.

Among the open problems that need to be addressed is the automatic initialization and failure recovery of tracking systems. A solution to this problem could come in two steps. First, a robust classifier could be trained to detect distinct keyposes that are well-suited for initialization. Second, knowledge about the detected keyposes could be used to provide a rough initialization that is then used as a starting point for optimization. Note that the statistical set of shape parameters that we have learned for our human model is well-suited for automated parameter estimation. The final step towards interactive usage of our system is then the careful optimization towards real-time performance. We plan to accomplish this goal mainly through parallelization of particle evaluations (*e.g.* using GPUs).

Future work will also include the adaption to single viewpoints by utilizing sensor technologies such as *time-of-flight* cameras or *structured light stereo* that provide dense depth maps of a scene. This will also help to get rid of our main source of error, which are problems with the foreground-background segmentation in our observation model due to difficult lighting conditions or clothing that is indistinguishable from the background.

Acknowledgements This work was supported by the DFG (Deutsche Forschungsgemeinschaft) cluster of excellence CoTeSys (Cognition for Technical Systems) at the Technische Universität München.

References

1. Ankur Agarwal and Bill Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1), jan 2006.

2. Dragomir Anguelov, Daphne Koller, Hoi-Cheung Pang, Praveen Srinivasan, and Sebastian Thrun. Recovering articulated object models from 3d range data. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 18–26, Arlington, Virginia, United States, 2004. AUAI Press.
3. Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.
4. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.
5. Pedram Azad, Ales Ude, Tamim Asfour, and Rüdiger Dillmann. Stereo-based markerless human motion capture for humanoid robot systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3951–3956. IEEE, 2007.
6. A. O. Balan and M. J. Black. The naked truth: Estimating body shape under clothing. In *European Conference on Computer Vision, ECCV*, volume 5303, pages 15–29, 2008.
7. Jan Bandouch, Florian Engstler, and Michael Beetz. Evaluation of hierarchical sampling strategies in 3d human pose estimation. In *Proceedings of the 19th British Machine Vision Conference (BMVC)*, 2008.
8. Michael Beetz, Freek Stulp, Bernd Radig, Jan Bandouch, Nico Blodow, Mihai Dolha, Andreas Fedrizzi, Dominik Jain, Uli Klank, Ingo Kresse, Alexis Maldonado, Zoltan Marton, Lorenz Mösenlechner, Federico Ruiz, Radu Bogdan Rusu, and Moritz Tenorth. The assistive kitchen — a demonstration scenario for cognitive technical systems. In *IEEE 17th International Symposium on Robot and Human Interactive Communication (RO-MAN), Muenchen, Germany*, 2008. Invited paper.
9. Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005.
10. Liefeng Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.
11. Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.
12. Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, 2009.
13. M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding (CVIU)*, 106(1):116–129, 2007.
14. Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, February 2004.
15. H. Bubb, F. Engstler, F. Fritzsche, Ch. Mergl, O. Sabbah, P. Schaefer, and I. Zacher. The development of RAMSIS in past and future as an example for the cooperation between industry and university. *International Journal of Human Factors Modelling and Simulation*, 1(1):140–157, 2006.
16. Tat-Jen Cham and James M. Rehg. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1999.
17. Kong Man Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.
18. Ankur Datta, Yaser Ajmal Sheikh, and Takeo Kanade. Modeling the product manifold of posture and motion. In *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV2009*, 2009.
19. Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision (IJCV)*, 61(2):185–205, 2005.
20. Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, 2003.
21. Florian Engstler, Jan Bandouch, and Heiner Bubb. Memoman - model based markerless capturing of human motion. In *The 17th World Congress on Ergonomics (International Ergonomics Association, IEA)*, Beijing, China, 2009.
22. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.
23. J. Gall, B. Rosenhahn, and H.P. Seidel. Drift-free tracking of rigid and articulated objects. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
24. Jürgen Gall, Carsten Stoll, Edilson de Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 1–8, Miami, USA, 2009. IEEE Computer Society.
25. D. M. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding (CVIU)*, 73(1):82–98, 1999.
26. Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.
27. Kristen Grauman, Gregory Shakhnarovich, and Trevor Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 641, Washington, DC, USA, 2003. IEEE Computer Society.
28. Daniel Grest and Volker Krüger. Gradient-enhanced particle filter for vision-based motion capture. In Ahmed M. Elgammal, Bodo Rosenhahn, and Reinhard Klette, editors, *Workshop on Human Motion*, volume 4814 of *Lecture Notes in Computer Science*, pages 28–41. Springer, 2007.
29. Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear body pose estimation from depth images. In *DAGM-Symposium*, 2005.
30. P. Guan, A. Weiss, A. O. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *IEEE International Conference on Computer Vision, ICCV*, 2009.
31. Radu P. Horaud, Matti Niskanen, Guillaume Dewaele, and Edmond Boyer. Human motion tracking by registering an articulated surface to 3-d points and normals.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008.
32. Nicholas R. Howe, Michael E. Leventon, and William T. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. In *Advances in Neural Information Processing Systems (NIPS) 12*, pages 820–826, Denver, CO, November 2000. MIT Press.
33. Špela Ivekovič, Emanuele Trucco, and Yvan R. Petillot. Human body pose estimation with particle swarm optimisation. *Evolutionary Computation*, 16(4), 2008.
34. Odest Chadwicke Jenkins and Maja J Matarić. A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction. In *The International Conference on Machine Learning (ICML 2004)*, pages 441–448, Banff, Alberta, Canada, Jul 2004.
35. S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, 1996.
36. Roland Kehl and Luc Van Gool. Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding (CVIU)*, 104(2):190–209, 2006.
37. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
38. Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann. Fusion of 2d and 3d sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321–329, 2009.
39. David Knossow, Remi Ronfard, and Radu P. Horaud. Human motion tracking with a kinematic parameterization of extremal contours. *International Journal of Computer Vision*, 2008. To appear.
40. Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion Graphs. In *29th annual conference on computer graphics and interactive techniques (SIGGRAPH 2002)*, 2002.
41. Volker Krüger, Danica Kragic, Ales Ude, and Christopher Geib. The meaning of action: a review on action recognition and mapping. *Advanced Robotics*, 21(13):1473–1501, 2007.
42. Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *International Journal of Robotics Research*, 27(7):761–784, 2008.
43. John MacCormick and Andrew Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.
44. John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 3–19, London, UK, 2000. Springer-Verlag.
45. Ivana Mikic, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Articulated body posture estimation from multi-camera voxel data. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
46. J.R. Mitchelson and A. Hilton. Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling. In *British Machine Vision Conference (BMVC)*, 2003.
47. Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU)*, 104(2):90–126, 2006.
48. L. Muendermann, S. Corazza, and T.P. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.
49. A. Oikonomopoulos, P. Ioannis, and M. Pantic. Spatio-Temporal Salient Points for Visual Recognition of Human Actions. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(3):710–719, 2006.
50. S. Pellegrini, K. Schindler, and D. Nardi. A generalization of the ICP algorithm for articulated bodies. In *British Machine Vision Conference (BMVC)*, 2008.
51. Rolf Plänkers and Pascal Fua. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding (CVIU)*, 81(3):285–302, March 2001.
52. Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):4–18, 2007.
53. Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 2010.
54. Deva Ramanan, David A. Forsyth, and Andrew Zisserman. Tracking people by learning their appearance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):65–81, 2007.
55. Bodo Rosenhahn, Thomas Brox, Uwe Kersting, Andrew Smith, Jason Gurney, and Reinhard Klette. A system for marker-less motion capture. *Künstliche Intelligenz*, 20(1):45–51, January 2006.
56. T. Seitz, D. Recluta, and D. Zimmermann. An approach for a human posture prediction model using internal/external forces and discomfort. In *Proceedings of the SAE 2005 World Congress*, 2005.
57. Yaser Sheikh, Mumtaz Sheikh, and Mubarak Shah. Exploring the space of a human action. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 144–149. IEEE Computer Society, 2005.
58. Leonid Sigal and Michael J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, 2006.
59. Leonid Sigal and Michael J. Black. Predicting 3D people from 2D pictures. In *Proceedings of the International Conference on Articulated Motion and Deformable Objects (AMDO'06)*, number 4069 in Lecture Notes in Computer Science, pages 185–195, Port d'Andratx, Spain, July 2006. Springer-Verlag.
60. Cristian Sminchisescu and Bill Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotic Research*, 22(6):371–392, June 2003.
61. J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.
62. Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV2009*, 2009.
63. Moritz Tenorth and Michael Beetz. KnowRob — Knowledge Processing for Autonomous Personal Robots. In *IEEE/RSJ International Conference on Intelligent RObots and Systems.*, 2009.
64. R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In

Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, June 2008.
65. Raquel Urtasun and Pascal Fua. 3d human body tracking using deterministic temporal motion models. In *European Conference on Computer Vision (ECCV)*, pages 92–106, 2004.
66. M. Vondrak, L. Sigal, and O.C. Jenkins. Physical simulation for probabilistic motion tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
67. J. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. *Advances in Neural Information Processing Systems*, 18:1441–1448, 2006.
68. Liang Wang and David Suter. Informative shape representations for human action recognition. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 1266–1269. IEEE Computer Society, 2006.
69. Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, November/December 2006.
70. Christopher R. Wren, Ali J. Azarbayejani, Trevor Darrell, and Alex P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):780–785, July 1997.
71. A. Yilmaz and Mubarak Shah. Actions sketch: A novel action representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 984–989, 2005.