2D Action Recognition Serves 3D Human Pose Estimation

Juergen Gall[1], Angela Yao[1], and Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zurich, Switzerland
[2] KU Leuven, Belgium
{gall,yaoa,vangool}@vision.ee.ethz.ch

Abstract. 3D human pose estimation in multi-view settings benefits from embeddings of human actions in low-dimensional manifolds, but the complexity of the embeddings increases with the number of actions. Creating separate, action-specific manifolds seems to be a more practical solution. Using multiple manifolds for pose estimation, however, requires a joint optimization over the set of manifolds and the human pose embedded in the manifolds. In order to solve this problem, we propose a particle-based optimization algorithm that can efficiently estimate human pose even in challenging in-house scenarios. In addition, the algorithm can directly integrate the results of a 2D action recognition system as prior distribution for optimization. In our experiments, we demonstrate that the optimization handles an 84D search space and provides already competitive results on HumanEva with as few as 25 particles.

1 Introduction

3D human pose estimation in multi-view scenarios is an active field of research [14]. While recent approaches [3,6,11,12] report impressive results on benchmarks like HumanEva [23], real-world applications such as in-house monitoring still pose many challenges. For example, background clutter, occlusions, and interactions with objects are all difficulties not encountered in studio recordings.

To maintain robustness in more unconstrained scenarios, the use of priors on human actions and dynamics have become very popular. For instance, the poses of a certain group of actions can be embedded into a low-dimensional manifold [12,15,29]. While 'full-body' motions like walking, jogging, and golf swings can be nicely embedded, learning embeddings for more ambiguous actions like 'carrying an object', particularly from sparse and noisy data, is a much more difficult task. Furthermore, the complexity increases with the number of actions and many dimensionality reduction techniques struggle to establish useful embeddings for a high number of actions. Instead of embedding all actions into a single manifold, creating separate, action-specific manifolds is an easier task to solve. Moreover, this allows for the incremental addition of new actions, which is an important property to have in practice. Using multiple manifolds, however, leads to an unsolved problem: how can we estimate the pose from a set of manifolds? An approach would be to learn the transitions between each manifold,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 425–438, 2010.
© Springer-Verlag Berlin Heidelberg 2010

using techniques like motion graphs [10] or switching models [4], but this does not scale with the number of actions.

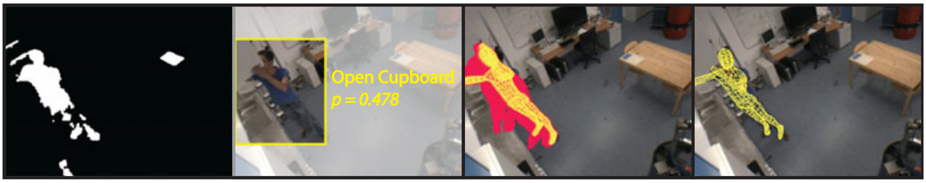
Here, we propose a new algorithm for optimizing over a set of manifolds that can efficiently estimate human pose even in challenging scenarios like the TUM kitchen dataset [27]. We have adapted a particle-based annealing optimization scheme [7] to jointly optimize over the action-specific manifolds and the human poses embedded in the manifolds. The approach scales in the worst case linearly with the number of manifolds, under the assumption that each manifold can be optimized with the same amount of time, i.e. they have the same dimensionality, which is more efficient than modeling transitions between the manifolds. Since a linear scaling is not optimal for a high number of action classes, we also propose a prior on the distribution of the actions obtained by a 2D action recognition system. In our experiments, we demonstrate that the action prior improves the tracking performance and that the optimization provides already competitive results with as few as 25 particles. The action recognition and tracking performance are evaluated on two state-of-the-art benchmarks, the HumanEva dataset [23] and the TUM Kitchen dataset [27].

2 Related Work

Using priors learned from motion capture databases is now very popular for robust tracking in difficult scenarios [22,30]. By learning a mapping between the image space and the pose space, the pose can be recovered directly from silhouettes and image features [1,3,8,11,24]. In [15,28,29], Gaussian process dynamical models were used for embedding motion in a low-dimensional latent space, while in [12] locally linear coordination is proposed for dimensionality reduction. Retrieved motions from databases have also been used [2] to refine tracked poses.

Action recognition is a rich sub-field of computer vision research in itself; we refer the reader to the recent review [18] and limit our discussion to multi-camera methods. Most work in multi-view action recognition has been focused on achieving view-invariant recognition. One line of approach has been to model the changes with respect to view, either of the location of feature points, using linear basis functions [21] or of the action's appearance, using low-dimensional manifolds [25]. A second line of approach has been to construct templates in either 3D [13,31,32] (2D space and time) or 4D [17] (3D space and time) and then projecting them back to a lower dimension from an arbitrary view, for matching either silhouettes or visual hulls.

Little work, however, has been done in coupling action recognition with pose estimation, as much of the previous work in pose estimation has been focused on sequences of single action classes rather than multi-actioned longer sequences. An exception is the switching Gaussian process dynamic model [4], in which the action is modelled as a hidden switching state. We follow a different approach since we do not model pose estimation as a filtering problem over time but as an optimization problem over the manifolds for each frame. Hence, we do not need to observe transitions between actions for training.



(a) Silhouette *(b)* Action Recognition *(c)* Pose Distribution *(d)* Final Pose

Fig. 1. System Overview. *(a)* Silhouettes are extracted by background subtraction. *(b)* Tracks are built over the entire sequence and classified by a 2D action recognition system. *(c)* Confidences of each action are used to distribute the particles over the action-specific manifolds. *(d)* Final pose is obtained by optimizing over the manifolds.

3 Framework

The multi-view system can be decomposed into action recognition on the 2D images and 3D pose estimation, with the action-specific manifolds acting as a link between the two. First, silhouettes are used to establish a track of the person over the sequence; the action recognition system then assigns labels for the track over time (Section 4). The confidence measure of the action labels are then used to distribute the particles in the particle-based optimization scheme over the action-specific manifolds (Section 5.1). Finally, the pose is estimated by an optimization over the entire set of manifolds (Section 5.3).

4 2D Action Recognition

For 2D action recognition, a separate classifier is trained for each of the cameras in the multi-view setup; results from the individual classifiers are then combined with standard classifier ensemble methods. Motivation for fusing the single views is based on the assumption that actions which are ambiguous in one view, e.g. due to self-occlusion, is more distinguishable from another view.

2D action recognition is performed according to the Hough-transform voting method presented in [33]. It breaks down the action recognition problem into an initial localization stage, which generates tracks of the individual performing the action, and a subsequent classification stage, which assigns action labels to the tracks. In scenarios where the cameras are fixed, it is not necessary to build the tracks with a tracking-by-detection technique as presented in [33]. Instead, background subtraction is used to generate silhouettes of the person performing the action (Fig. 1). Bounding boxes are then extrapolated around the silhouette and the trajectory of the bounding boxes is smoothed to build the track.

The output of the classification stage is a confidence score of each action class over time, normalized such that the confidences over all classes at any time point sum up to 1. A classifier combination strategy such as the max-rule is then used to combine the outputs from the multiple cameras [9].

5 Optimizing over a Set of Manifolds

Having a skeleton and a surface model of the human, the human pose is represented by a vector in a bounded, high-dimensional state space $\mathbb{E} \subset \mathbb{R}^{D+6}$. While $\Theta = \theta_1, \cdots, \theta_D \in \mathbb{E}_\Theta$ denotes the joint angles, the global orientation and position are encoded by the 6D vector (r, t). An element of the search space is given by $x = (r, t, \Theta)$. We formulate pose estimation as an optimization problem over $\mathbb{E}$ for a given positive energy function V, i.e. $\min_{x \in \mathbb{E}} V(x)$. The energy function measures the consistency between the images of all camera views and the projections of the model's surface for a given pose x. As consistency measure, we use edges and silhouettes [20]. Although these features are not optimal for human pose estimation, since edges are sensitive to background clutter and silhouettes are sensitive to occlusions and background changes, the associated energy function is fast to compute and fixed for all our experiments. As a baseline, we implemented the particle-based annealing optimization scheme ISA over $\mathbb{E}$, which has been used in the multi-layer framework [6]. The optimization scheme, based on the theory of Feynman-Kac models [16], iterates over a selection and mutation step, and is also the underlying principle of the annealed particle filter [5].

We modify the baseline algorithm to optimize over a set of manifolds instead of a single state space. To this end, we consider a set of action classes $\mathcal{A} = \{a_1, \cdots, a_{|\mathcal{A}|}\}$, where we learn for each class an action-specific low-dimensional manifold $\mathbb{M}_a \subset \mathbb{R}^{d_a}$ with $d_a \ll D$. We assume that the following mappings are available:

$$f_a : \mathbb{E}_\Theta \mapsto \mathbb{M}_a, \quad g_a : \mathbb{M}_a \mapsto \mathbb{E}_\Theta, \quad h_a : \mathbb{M}_a \mapsto \mathbb{M}_a, \tag{1}$$

where f_a denotes the mapping from the state space to the low-dimensional manifolds, g_a the projection back to the state space, and h_a the prediction within an action-specific manifold. Since the manifolds encode only the space of joint angles, a low-dimensional representation of the full pose is denoted by $y_a = (r, t, \Theta_a)$ with $\Theta_a = f_a(\Theta)$. A particle $s^i = (y^i_a, a^i)$ stores the corresponding manifold label a^i in addition to the vector $y^i_a = (r^i, t^i, \Theta^i_a)$ and the set of particles is denoted by $\mathcal{S}$. Our algorithm operates both in the state space as well as in the manifolds. An overview of the algorithm is given in Fig. 2.

5.1 Action-Specific Manifolds

Each of the action-specific low-dimensional manifolds, $\mathbb{M}_a$, are learned from the joint angles Θ in motion capture data using Isomap [26], a non-linear dimensionality reduction technique. As Isomap does not provide mappings between the high- and low-dimensional pose spaces, we learn two separate Gaussian Process (GP) regressions [19], f_a and g_a (2), to map from the high-dimensional space to the low-dimensional space and back, respectively, where $m(\cdot)$ and $k(\cdot)$ denote the mean and covariance functions.

$$y = f_a(x) \sim \mathcal{GP}\left(m(x), k(x, x\prime)\right); \quad x = g_a(y) \sim \mathcal{GP}\left(m(y), k(y, y\prime)\right). \tag{2}$$

In addition, a third GP regression, h_a, is learned to model temporal transitions between successive poses within each action-specific manifold:

$$y_t = h_a\left(y_{t-1}\right) \sim \mathcal{GP}\left(m\left(y_{t-1}\right)\right), k\left(y_{t-1}, y\prime_{t-1}\right)\right). \tag{3}$$

5.2 Theoretical Discussion

As mentioned in Section 5, one seeks the solution of the minimization problem $\min_{x\in\mathbb{E}} V(x)$. When optimizing over a set of manifolds the problem becomes

$$\min_{a\in\mathcal{A}}\left(\min_{y\in\mathbb{M}_a} V(g_a(y))\right). \tag{4}$$

Minimizing the problem this way, i.e. searching the global minimum in all manifolds $\mathbb{M}_a$ and then taking the best solution mapped back to the state space, does not scale well with the number of manifolds. Hence, we propose to optimize over all manifolds jointly. Before outlining the optimization procedure in Section 5.3, we briefly discuss the existence and the uniqueness of the solution. Since g_a and f_a are not direct inverses of each other, i.e. $(g_a \circ f_a)$ does not equal the identity function, the optimization over the manifolds (4) does not provide the same solution as the original optimization problem over the state space. Indeed, this is the case only if the following is satisfied:

$$\exists a \in \mathcal{A}, \exists y \in \mathbb{M}_a : \min_{x\in\mathbb{E}} V(x) = V(g_a(y)). \tag{5}$$

The uniqueness of the solution for the manifold and thus of the action a is interesting from the point of action recognition. It is given if and only if

$$\forall a_1, a_2 \in \mathcal{A} \quad \text{with} \quad a_1 \neq a_2 : \min_{y\in\mathbb{M}_{a_1}} V(g_{a_1}(y)) \neq \min_{y'\in\mathbb{M}_{a_2}} V(g_{a_2}(y')). \tag{6}$$

In most cases, optimization of the pose propagates the particles into the "right" manifold, i.e. the correct action, as plotted in Fig. 3. However, there is usually an overlap of poses between the manifolds such that Eq. (6) is not satisfied. Note that in comparison to the action recognition, which takes a sequence of frames into account (Section 4), the pose is optimized only for the current frame.

To cope with the problem defined in (5), we introduce two optimization steps

$$(\hat{y}, \hat{a}) = \operatorname*{argmin}_{a\in\mathcal{A}, y\in\mathbb{M}_a} V(g_a(y)) \qquad \text{and} \tag{7}$$

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{E}} V(x), \quad \text{with} \quad x_0 = g_{\hat{a}}(\hat{y}) \tag{8}$$

as the initialization. In other words, we first search for the nearest approximation by optimizing over the manifolds and then use this result to initialize the optimization over the state space. With this procedure, we can design an optimization that converges to the global minimum in the state space, see Fig. 2.

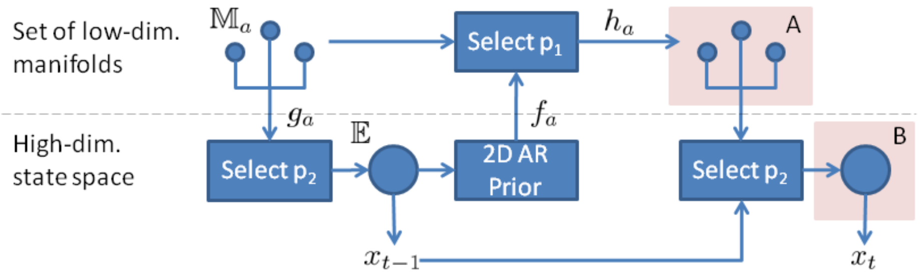


Fig. 2. For each action class a, we learn an embedding in a low-dimensional manifold $\mathbb{M}_a$. The manifolds are indicated by the small circles and the high-dimensional state space $\mathbb{E}$ is indicated by the large circle. Having estimated the pose x_{t-1}, a set of particles is selected from the previous particle sets (*Select* p_1). To this end, the particles in $\mathbb{E}$ are mapped by f_a to $\mathbb{M}_a$ where each particle is associated to one of the manifolds. This process is steered by a prior distribution on the actions obtained by a 2D action recognition system. Since the manifolds are action-specific, the pose for the next frame can be predicted by the function h_a. The first optimization step, *Optimization A*, optimizes jointly over the manifolds and the human poses embedded in the manifolds. Since our manifolds do not cover transitions between actions, we run a second optimization step, *Optimization B*, over the particles mapped back to the state space $\mathbb{E}$ by g_a. Before the optimization, the particle set is augmented by making use of the embedding error of the previous pose x_{t-1} (*Select* p_2).

5.3 Algorithm

Optimization A: Since ISA [7] is not directly applicable for optimizing over a set of manifolds, we have to modify the algorithm. For the weighting, the particles are mapped back to the full space in order to evaluate the energy function V:

$$w^i = \exp\left(-\beta_k \cdot V\left(r^i, t^i, g_{a^i}(\Theta_a^i)\right)\right), \tag{9}$$

where k is the iteration parameter of the optimization. The weights of all particles are normalized such that $\sum_{s^i} w^i = 1$. Note that the normalization does not take the label of the manifold a^i into account. As result, particles in a certain manifold might have higher weights than particles in another manifold since their poses fit the image data better. Since particles with higher weights are more likely to be selected, the distribution of the particles among the manifolds $\mathbb{M}_a$ changes after the selection step. This is desirable since the particles should migrate to the most likely manifold to get a better estimate within this manifold. While the selection is performed as in [7][1], the mutation step needs to be adapted since the particles are spread in different spaces. To this end, we use $|\mathcal{A}|$ mutation kernels K_a, one for each manifold, and an additional kernel K_0 for the global position and orientation. In our implementation, we use Gaussian kernels with covariance matrices Σ_a proportional to the sample covariance within a manifold,

[1] Using the selection kernel $\epsilon_k(\eta) = \frac{1}{\inf\{y\,:\,\eta(\{x\in E\,:\,\exp(-\beta_k\,V(x))>y\})=0\}}$.

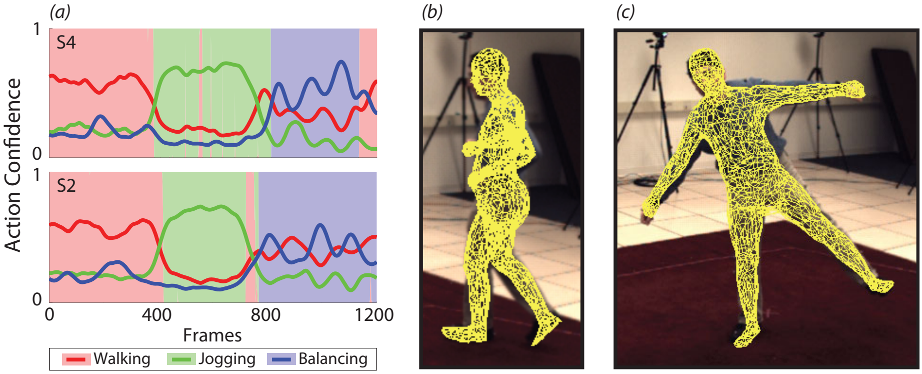


Fig. 3. HumanEva. Action recognition prior from camera C1 *(a)*. The curves show the action confidence per frame. Note the smooth transitions between the actions around frame 800 for subject S4. After jogging, the subject walks a few steps before balancing. At the end of the sequence, the person walks away, as recognized by the action recognition system. The distribution of the particles among the action-specific manifolds after *Optimization A* is shown by the area plot. The particles move to the correct manifold for nearly all frames. Pose estimate for jogging *(b)* and balancing *(c)*.

i.e. $\mathcal{S}_a = \{s^i \in \mathcal{S} : a^i = a\}$:

$$\Sigma_a = \frac{\alpha_\Sigma}{|\mathcal{S}_a| - 1}\left(\rho\, I + \sum_{s^i \in \mathcal{S}_a} (\Theta_a^i - \mu_a)\,(\Theta_a^i - \mu_a)^T\right), \quad \mu_a = \frac{1}{|\mathcal{S}_a|}\sum_{s^i \in \mathcal{S}_a} \Theta_a^i. \tag{10}$$

The scaling factor $\alpha_\Sigma = 0.4$ and the positive constant $\rho = 0.0001$, which ensures that the covariance does not become singular, are fixed for all kernels. The kernel K_0 for rotation and translation is computed over the full set of particles $\mathcal{S}$:

$$\Sigma_0 = \frac{\alpha_\Sigma}{|\mathcal{S}| - 1}\left(\rho\, I + \sum_{s^i \in \mathcal{S}} \left((r^i, t^i) - \mu\right)\,\left((r^i, t^i) - \mu\right)^T\right), \quad \mu = \frac{1}{|\mathcal{S}|}\sum_{s^i \in \mathcal{S}} (r^i, t^i). \tag{11}$$

Since we compute the extra kernel K_0 instead of taking (r, t) as additional dimensions for the kernels K_a, the correlation between (r, t) and Θ_a is not taken into account. However, the number of particles per manifold can be very small, such that K_0 computed over all particles provides a better estimate of the correlation between the global pose parameters (r, t).

Select p_2: Before continuing with the optimization in the full state, the set of particles $\mathcal{S}$ needs to be mapped from the manifolds $\mathbb{M}_a$ to $\mathbb{E}$, where the particles build the initial distribution for the next optimization step. However, it can happen that the true pose is not well represented by any of the manifolds. This is typical of transitions from one action to another, which are not modelled in our setting. As we will show in our experiments, it is useful to use the previous estimate $\hat{x}_{t-1}$ to augment the initial particle set. To measure the discrepancy between the last estimated pose and the poses modeled by the manifolds, we compute $\Sigma_{\hat{a}}$ based on the reconstruction error for $\hat{x}_{t-1}$:

$$\hat{a} = \operatorname*{argmin}_{a \in \mathcal{A}} \left\| \hat{\Theta}_{t-1} - g_a(f_a(\hat{\Theta}_{t-1})) \right\|, \quad \sigma_{\hat{a},i} = \frac{|\hat{\Theta}_{t-1} - g_{\hat{a}}(f_{\hat{a}}(\hat{\Theta}_{t-1}))|_i}{3}. \tag{12}$$

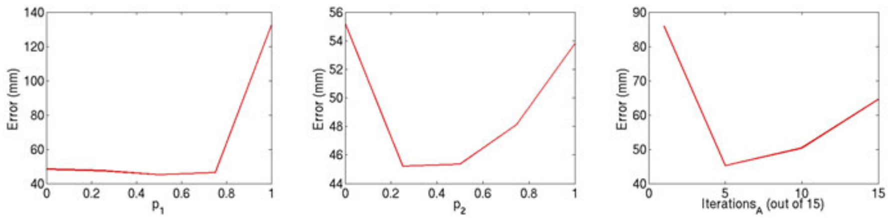
We create a new set of particles by sampling from $\mathcal{N}(\hat{\Theta}_{t-1}, \Sigma_{\hat{a}})$, where $\Sigma_{\hat{a}}$ is the diagonal matrix with $\sigma_{\hat{a},i}$ as entries. According to the 3σ rule, this means that nearly all samples are within the distance of the reconstruction error. The selection process between the two particle sets is controlled by the parameter $p_2 \in [0,1]$. For all $s^i \in \mathcal{S}$, we draw u from the uniform distribution $\mathcal{U}[0,1]$. If $u < p_2$, $s^i = (r^i, t^i, \Theta^i)$ is added to the new set; otherwise the particle $(r^i, t^i, \hat{\Theta})$ is added to the set, where $\hat{\Theta}$ is sampled from $\mathcal{N}(\hat{\Theta}_{t-1}, \Sigma_{\hat{a}})$.

Optimization B: The second optimization step eventually runs ISA [7] on the full state space. However, we do not start from the beginning but continue with the optimization, i.e. when It_A is the number of iterations used for *Opt. A*, we continue with β_{It_A+1} instead of β_1.

Select p_1*:* After *Opt. A*, all the particles may aggregate into one single manifold, so we distribute the particles again amongst the manifolds $\mathbb{M}_a$ when moving to the next frame I_t; otherwise, we get stuck in a single action class. Similar to the previous selection, we make use of two particle sets; the particles $\mathcal{S}^{\mathbb{M}}$ in the manifolds $\mathbb{M}_a$ after *Opt. A* and the particles in the state space $\mathcal{S}^{\mathbb{E}}$ after *Opt. B*. The selection is controlled by the parameter $p_1 \in [0,1]$. For all $s^i \in \mathcal{S}^{\mathbb{M}}$, we draw u from the uniform distribution $\mathcal{U}[0,1]$. If $u < p_1$, s^i is added to the new set; otherwise the particle $(r^i, t^i, \Theta^i) \in \mathcal{S}^{\mathbb{E}}$ is mapped to one of the manifolds and added to the set. The manifold $\mathbb{M}_{a^i}$ is selected according to the probability $p(A = a|T = t, \mathcal{I})$, yielding the mapped particle $(r^i, t^i, f_{a^i}(\Theta^i), a^i)$. In our experiments, we use two choices for $p(A|T = t, \mathcal{I})$:

$$\begin{aligned} p(A = a \,|\, T = t, \mathcal{I}) &= p(A = a) = \tfrac{1}{|\mathcal{A}|} && \textit{(Uniform Prior)} \\ p(A = a \,|\, T = t, \mathcal{I}) &= p(A = a \,|\, I_{t-l} \cdots I_{t+l}) && \textit{(Action Prior)} \end{aligned}$$

The *uniform prior* is independent of the current frame and results in a joint optimization over the manifolds $\mathbb{M}_{a \in \mathcal{A}}$ and poses $y \in \mathbb{M}_a$. However, the prior does not scale well with the number of manifolds since the total number of particles is fixed and there must be a sufficient number of particles available for each manifold. The *action prior* distributes the particles to manifolds that are more likely a-priori, meaning that a manifold $\mathbb{M}_a$ cannot be explored when $p(A = a|T = t, \mathcal{I}) = 0$ and $\{s^i \in \mathcal{S}^{\mathbb{M}} : a^i = a\} = \emptyset$. This also motivates the use of the particle set $\mathcal{S}^{\mathbb{M}}$ to increase the robustness to temporary errors in the *action prior* as demonstrated in Fig. 4*(a)*. Note that a zero-probability error for the true manifold over many frames cannot be compensated. In our experiments, $p(A|I_{t-l} \cdots I_{t+l})$ is obtained by an action recognition system which takes a set of frames in the neighborhood of t into account (Section 4).



(a) p_1 (b) p_2 (c) It_A

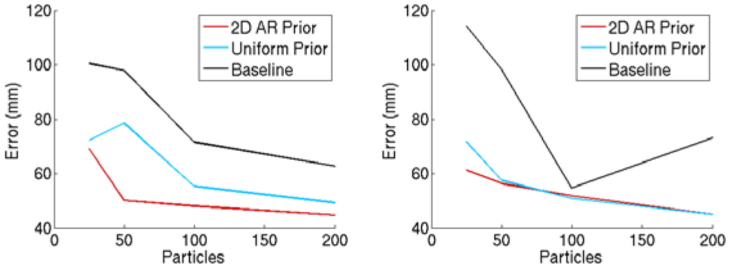
Fig. 4. Evaluation of parameters. *(a) Select* p_1: The best result is obtained by $p_1 = 0.5$, which shows the benefit of taking both particle sets $\mathcal{S}^{\mathbb{M}}$ and $\mathcal{S}^{\mathbb{E}}$ into account. For $p_1 = 1$, the particles $\mathcal{S}^{\mathbb{E}}$ from *Opt. B* are discarded. *(b) Select* p_2: The best results are achieved with $p_2 \in [0.25, 0.5]$. It shows the benefit of taking the reconstruction error for $\hat{x}_{t-1}$ into account. *(c)* Number of iterations for *Opt. A* (It_A) and *Opt. B* (15-It_A). The summed number of iterations was fixed to 15. Without a second optimization step (It_A=15), the error is significantly higher than for the optimal setting (It_A=5).

6 Experiments

HumanEva. The HumanEva-II [23] dataset is the standard benchmark on 3D human pose estimation. It comprises two sequences S2 and S4 with three actions, see Fig. 3. The dataset provides a model for subject S4, which we also use for subject S2 despite differences in body shape. The human pose is represented by 28 parameters. We perform two trials: testing on S2 and training on S4 and vice versa. For learning the action-specific manifolds, we use the tracking results of the multi-layer tracker [6] where we split the data into the three action classes and discard the transitions between the actions. Note that training data from marker-less tracking approaches is in general noisier and less accurate than data from marker-based systems.

In Fig. 4, we plot the impact of the parameters on the tracking accuracy. For evaluation, we use 200 particles, 5 iterations for *Opt. A*, and 10 iterations for *Opt. B* unless otherwise specified. The optimization is run with a polynomial annealing scheme with $b = 0.7$ [7]. The results clearly support our design decisions for the algorithm (Section 5.3).

In Fig. 5, we plot the 3D estimation error of the joints with respect to the number of particles. For comparison, we show the mean and standard deviation for optimizing over the state space $\mathbb{E}$ (*baseline*) and the proposed algorithm with a *uniform prior* and an *action prior*, with the *action prior* computed as described in Section 4. For the baseline, we run *Opt. B* with 15 iterations and without taking the manifolds $\mathbb{M}_a$ into account. Note that according to [6,23], pose estimation requires usually at least 200-250 particles to achieve good results on this dataset. We perform the optimization of the 28 parameters with 200 down to 25 particles. Unsurprisingly, that the error for the *baseline* increases significantly when the number of particles drops below 100. When optimizing over the manifolds and the poses embedded in the manifolds, the error increases gently with a decreasing number of particles. Since the dataset contains only



(a) S2 (b) S4

Fig. 5. 3D Estimation error with respect to number of particles. The proposed approach performs significantly better than the direct optimization in the state space $\mathbb{E}$ (baseline), particularly for a small number of particles. The discrepancy between uniform prior and the prior obtained from 2D action recognition is getting larger for very few particles. In this case, the number of particles per manifold becomes very small for a uniform distribution. Note that competitive results are still achieved with only 25 particles. Timings are given in Table 1.

Table 1. Computation time per frame and 3D estimation error of the optimization with respect to number of particles. The 2D action recognition takes additional 0.4 seconds for each frame consisting of 4 images, which is roughly the computation time for 20 particles. *ap: action prior; up: uniform prior; base: baseline.*

	Time (sec.)		S2 Error (mm)			S4 Error (mm)		
n	ap,up	base	ap	up	base	ap	up	base
200	3.89	3.80	44.9 ± 9.5	49.4 ± 19.0	62.9 ± 24.4	45.2 ± 13.4	45.2 ± 11.8	73.1 ± 70.7
100	1.96	1.92	48.2 ± 12.7	55.4 ± 37.8	71.7 ± 25.7	51.9 ± 20.9	51.0 ± 21.3	54.7 ± 25.0
50	0.98	0.96	50.2 ± 13.4	78.7 ± 72.4	98.0 ± 61.1	56.4 ± 19.2	57.6 ± 19.2	98.3 ± 67.4
25	0.5	0.49	69.3 ± 51.1	72.3 ± 51.2	100.5 ± 40.4	61.3 ± 21.2	71.8 ± 29.3	114.3 ± 85.4

3 action classes, the *uniform prior* performs very well. Differences between the two priors become more prominent for very few particles per action class. This indicates that the *action prior* scales better with a large number of classes since this basically limits the number of particles per action class. In general, the *uniform prior* describes the worst case scenario where the action recognition is not better than a random guess. Timings and mean errors are given in Table 1.

Finally, we show the tracking performance with respect to number of camera views in Fig. 6*(a)*; using 200 particles. Again, the proposed approach significantly outperforms the *baseline*. At first glance, the *uniform prior* and the *action prior* seem to perform similarly, due to the scaling of the plot from the large error of the *baseline*, though the *action prior* actually reduces the error on average by 4%. The benefit of the *action prior* is more evident for very few particles per action class as shown in Fig. 5.

TUM Kitchen dataset. A more challenging dataset than HumanEva is the newly released TUM Kitchen dataset [27]. The dataset contains 20 episodes of recordings from 4 views of 4 subjects setting a table. In each episode, a subject moves back and forth between the kitchen and a dining table, each time fetching

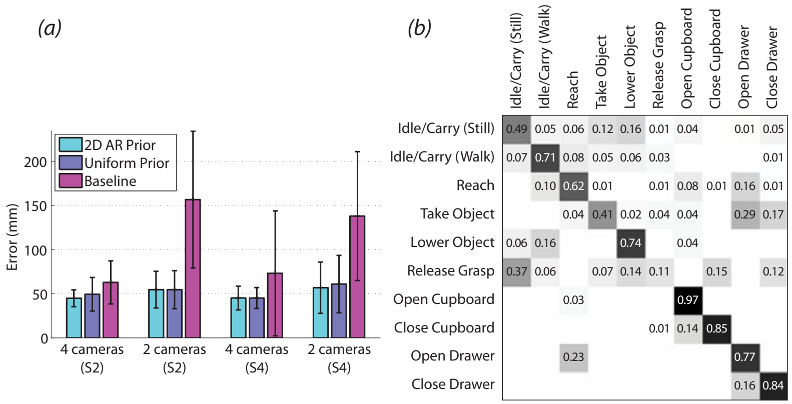


Fig. 6. *(a)* 3D Estimation error with respect to number of views for HumanEva. For the setting with two views, cameras C1 and C2 are taken. The reduced number of views results in more ambiguities. The proposed approach handles these ambiguities better than the direct optimization in the state space $\mathbb{E}$ (baseline). *(b)* Confusion matrix for fused results according to the max-rule for TUM kitchen.

Table 2. Individual camera and fused action recognition performance for subjects 1-4; fused performance is higher than any individual camera view for each subject

	Camera 1	Camera 2	Camera 3	Camera 4	Fused
Subject 1	0.542	0.493	0.569	0.564	0.574
Subject 2	0.532	0.501	0.456	0.560	0.585
Subject 3	0.690	0.718	0.652	0.666	0.740
Subject 4	0.619	0.529	0.610	0.610	0.706
Average	0.596	0.560	0.572	0.600	0.651

objects such as cutlery, plates and cups and then transporting them to the table. The dataset is particularly challenging for both action recognition as well as pose estimation, as the actions are more subtle than those of standard action recognition datasets and parts of the body are often occluded by objects such as drawers, cupboard doors and tables (see Fig. 1). Training was done on episodes *1-0* to *1-5*, all of which are recorded from subject 1 and testing was done on episodes *0-2*, *0-4*, *0-6*, *0-8*, *0-10*, *0-11*, and *1-6*, which are recorded from all 4 subjects. For the action recognition, we use the 9 labels that are annotated for the 'left hand' [27]. Since the labels are determined by the activity of the arms and we would like the manifolds to be representative of the entire body, we further split the idle/carry class according to whether the subject is walking or standing; see Fig. 6*(b)*.

Results of the action recognition for cameras 0 and 2, as well as the fused results are shown in Table 2. For classifier fusion, we use the max-rule that gave the best performance compared to other standard ensemble methods [9], though results were similar for all the methods. Fused results and the confusion matrix are shown in Fig. 7 and Fig. 6*(b)*.

Based on the fused results of the action recognition, we also evaluate the tracking performance. For the dataset, we use the provided models with 84

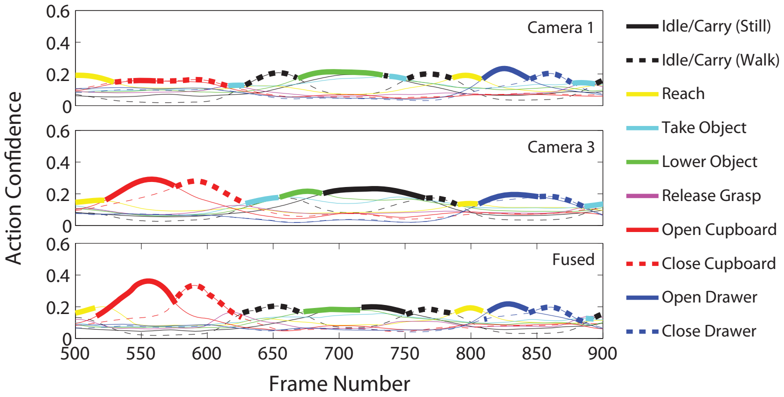


Fig. 7. Normalized action confidences for two camera views as well as fused confidences for frames 500-900 of episode *0-11*

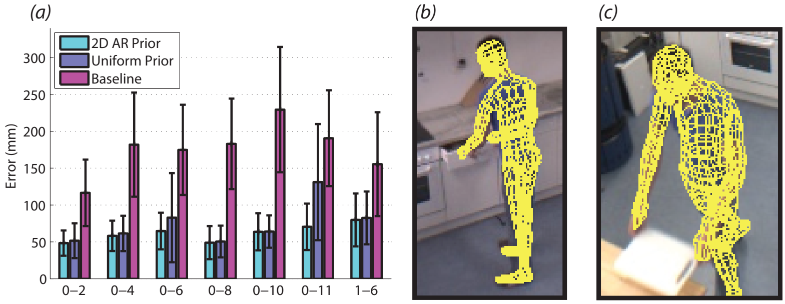


Fig. 8. 3D Error for TUM kitchen dataset *(a)*. The proposed approach performs significantly better than the direct optimization in the state space $\mathbb{E}$ (baseline). The error for the sequences *0-2* and *0-8* are the lowest since the action-specific manifolds were trained on the same subject. Mean and standard deviation are provided in Table 3. Pose estimates for opening drawer *(b)* and lowering object *(c)*.

Table 3. 3D Error for TUM kitchen dataset in mm. *ap: action prior; up: uniform prior; base: baseline.*

(mm)	*0-2*	*0-4*	*0-6*	*0-8*	*0-10*	*0-11*	*1-6*
ap	48.4 ± 17.1	58.2 ± 20.5	64.7 ± 24.9	49.0 ± 22.5	63.7 ± 25.2	70.5 ± 31.5	79.8 ± 35.9
up	51.6 ± 23.6	61.4 ± 23.9	82.9 ± 60.5	50.5 ± 21.5	64.0 ± 22.0	131.1 ± 78.8	82.5 ± 35.8
base	116.5 ± 45.1	181.9 ± 70.6	174.8 ± 61.2	183.0 ± 61.4	229.4 ± 85.0	190.6 ± 65.0	155.4 ± 70.4

parameters. The large errors for the *baseline* in Fig. 8 show that 200 particles are not enough to optimize over a 84 dimensional search space. Note that we do not make use of any joint limits or geometric information about the kitchen and use only the images as input. The proposed approach estimates the sequences with a comparable accuracy as HumanEva, although the dimensions of the state space increased from 28 to 84, the number of action classes from 3 to 8 (the 'open' and 'close' actions are embedded in one manifold), and the silhouette quality is much worse due to truncations and occlusions. Compared to the *uniform prior*, the *action prior* reduces the error on average by 12%.

7 Conclusion

We have presented an algorithm[2] that efficiently solves the problem of optimizing over a set of manifolds. In the context of 3D pose estimation, we demonstrated that the algorithm handles high-dimensional spaces with very few particles. Since transitions between actions are not explicitly modeled, as in previous work, it is an important step towards pose estimation with many action classes. Furthermore, we have shown that a prior distribution based on action recognition improves the performance. This is interesting since it is expected that the algorithm scales very well with the number of classes when the action recognition system does as well. In this way, 3D human pose estimation can be linked to the progress in the field of action recognition. As there are very few datasets for pose estimation and action recognition available and none contains many action classes, new datasets are required to investigate scalability more in detail.

Acknowledgments. This work has been supported by funding from the Swiss National Foundation NCCR project IM2 as well as the EC project IURO. Angela Yao was also supported by funding from NSERC Canada. We thank Jan Bandouch and Moritz Tenorth for providing the TUM Kitchen dataset and Fabian Nater for helpful discussions.

References

1. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. TPAMI 28(1), 44–58 (2006)
2. Baak, A., Rosenhahn, B., Müller, M., Seidel, H.P.: Stabilizing motion tracking using retrieved motion priors. In: ICCV (2009)
3. Bo, L., Sminchisescu, C.: Twin gaussian processes for structured prediction. IJCV 87, 28–52 (2010)
4. Chen, J., Kim, M., Wang, Y., Ji, Q.: Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. In: CVPR (2009)
5. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. IJCV 61(2), 185–205 (2005)
6. Gall, J., Rosenhahn, B., Brox, T., Seidel, H.P.: Optimization and filtering for human motion capture – a multi-layer framework. IJCV 87, 75–92 (2010)
7. Gall, J., Rosenhahn, B., Seidel, H.P.: An Introduction to Interacting Simulated Annealing. In: Human Motion: Understanding, Modelling, Capture and Animation, pp. 319–343. Springer, Heidelberg (2008)
8. Grauman, K., Shakhnarovich, G., Darrell, T.: Inferring 3d structure with a statistical image-based shape model. In: ICCV, pp. 641–648 (2003)
9. Kittler, J., Society, I.C., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. TPAMI 20, 226–239 (1998)
10. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Trans. Graph. 21(3), 473–482 (2002)

[2] Source code is available at `http://www.vision.ee.ethz.ch/~gallju`

11. Lee, C.S., Elgammal, A.: Coupled visual and kinematic manifold models for tracking. IJCV 87, 118–139 (2010)
12. Li, R., Tian, T.P., Sclaroff, S., Yang, M.H.: 3d human motion tracking with a coordinated mixture of factor analyzers. IJCV 87, 170–190 (2010)
13. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: CVPR (2007)
14. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104(2), 90–126 (2006)
15. Moon, K., Pavlovic, V.: Impact of dynamics on subspace embedding and tracking of sequences. In: CVPR, pp. 198–205 (2006)
16. Moral, P.D.: Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications. Springer, New York (2004)
17. Pingkun Yan, S.M.K., Shah, M.: Learning 4d action feature models for arbitrary view action recognition. In: CVPR (2008)
18. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing (2010)
19. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). MIT Press, Cambridge (2005)
20. Shaheen, M., Gall, J., Strzodka, R., van Gool, L., Seidel, H.P.: A comparison of 3d model-based tracking approaches for human motion capture in uncontrolled environments. In: IEEE Workshop on Applications of Computer Vision (2009)
21. Sheikh, Y., Sheikh, M., Shah, M.: Exploring the space of a human action. In: ICCV (2005)
22. Sidenbladh, H., Black, M., Sigal, L.: Implicit probabilistic models of human motion for synthesis and tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 784–800. Springer, Heidelberg (2002)
23. Sigal, L., Balan, A., Black, M.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. IJCV 87, 4–27 (2010)
24. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Bme: Discriminative density propagation for visual tracking. TPAMI 29(11), 2030–2044 (2007)
25. Souvenir, R., Babbs, J.: Learning the viewpoint manifold for action recognition. In: CVPR (2008)
26. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(5500), 2319–2323 (2000)
27. Tenorth, M., Bandouch, J., Beetz, M.: The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In: IEEE Workshop on THEMIS (2009)
28. Ukita, N., Hirai, M., Kidode, M.: Complex volume and pose tracking with probabilistic dynamical model and visual hull constraint. In: ICCV (2009)
29. Urtasun, R., Fleet, D., Fua, P.: 3d people tracking with gaussian process dynamical models. In: CVPR, pp. 238–245 (2006)
30. Urtasun, R., Fua, P.: 3d human body tracking using deterministic temporal motion models. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 92–106. Springer, Heidelberg (2004)
31. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV, pp. 1–7 (2007)
32. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. CVIU 104(2-3), 249–257 (2006)
33. Yao, A., Gall, J., van Gool, L.: A hough transform-based voting framework for action recognition. In: CVPR (2010)