Coupled Action Recognition and Pose Estimation from Multiple Views

Originally published in:
International Journal of Computer Vision 100(1), https://doi.org/10.1007/s11263-012-0532-9

Int J Comput Vis (2012) 100:16–37
DOI 10.1007/s11263-012-0532-9

Coupled Action Recognition and Pose Estimation from Multiple Views

Angela Yao · Juergen Gall · Luc Van Gool

Received: 8 September 2011 / Accepted: 27 April 2012 / Published online: 30 May 2012

Abstract Action recognition and pose estimation are two closely related topics in understanding human body movements; information from one task can be leveraged to assist the other, yet the two are often treated separately. We present here a framework for coupled action recognition and pose estimation by formulating pose estimation as an optimization over a set of action-specific manifolds. The framework allows for integration of a 2D appearance-based action recognition system as a prior for 3D pose estimation and for refinement of the action labels using relational pose features based on the extracted 3D poses. Our experiments show that our pose estimation system is able to estimate body poses with high degrees of freedom using very few particles and can achieve state-of-the-art results on the HumanEva-II benchmark. We also thoroughly investigate the impact of pose estimation and action recognition accuracy on each other on the challenging TUM kitchen dataset. We demonstrate not only the feasibility of using extracted 3D poses for action recognition, but also improved performance in comparison to action recognition using low-level appearance features.

A. Yao (✉) · J. Gall · L. Van Gool
Computer Vision Laboratory, ETH Zurich, Sternwartstrasse 7, 8092 Zurich, Switzerland
e-mail: yaoa@vision.ee.ethz.ch

J. Gall
Max Planck Institute for Intelligent Systems, Spemannstrasse 41, 72076 Tubingen, Germany
e-mail: jgall@tue.mpg.de

L. Van Gool
Department of Electrical Engineering/IBBT, K.U. Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium
e-mail: luc.vangool@esat.kuleuven.be

Keywords Human pose estimation · Human action recognition · Tracking · Stochastic optimization · Hough transform

1 Introduction

Vision-based human motion analysis attempts to understand the movements of the human body using computer vision and machine learning techniques. The movements of the body can be interpreted on a physical level through *pose estimation*, i.e. reconstruction of the 3D articulated motions, or on a higher, semantic level through *action recognition*, i.e. understanding the body's movements over time. While the objectives of the two tasks differ, they share a significant information overlap. For instance, poses from a given action tend to be a constrained subset of all possible configurations within the space of physiologically possible poses. Therefore, many state-of-the-art pose estimation systems use action-specific priors to simplify the pose estimation problem, e.g. (Geiger et al. 2009; Li et al. 2010; Taylor et al. 2010; Lee and Elgammal 2010; Chen et al. 2009). At the same time, pose information can be a very strong indicator of actions and action labels can be determined from as little as a single frame (Schindler and Van Gool 2008; Thurau and Hlavac 2008; Yang et al. 2010; Maji et al. 2011). However, as neither pose estimation nor action recognition are trivial tasks, few systems have tried to couple the two tasks together into a single system. On the one hand, priors from many state-of-the-art pose estimation systems are of a single activity, thereby assuming that the activity is already known, and cannot handle sequences of multiple activities (Taylor et al. 2010). On the other hand, action recognition approaches either model poses implicitly through pose-related descriptors (Thurau

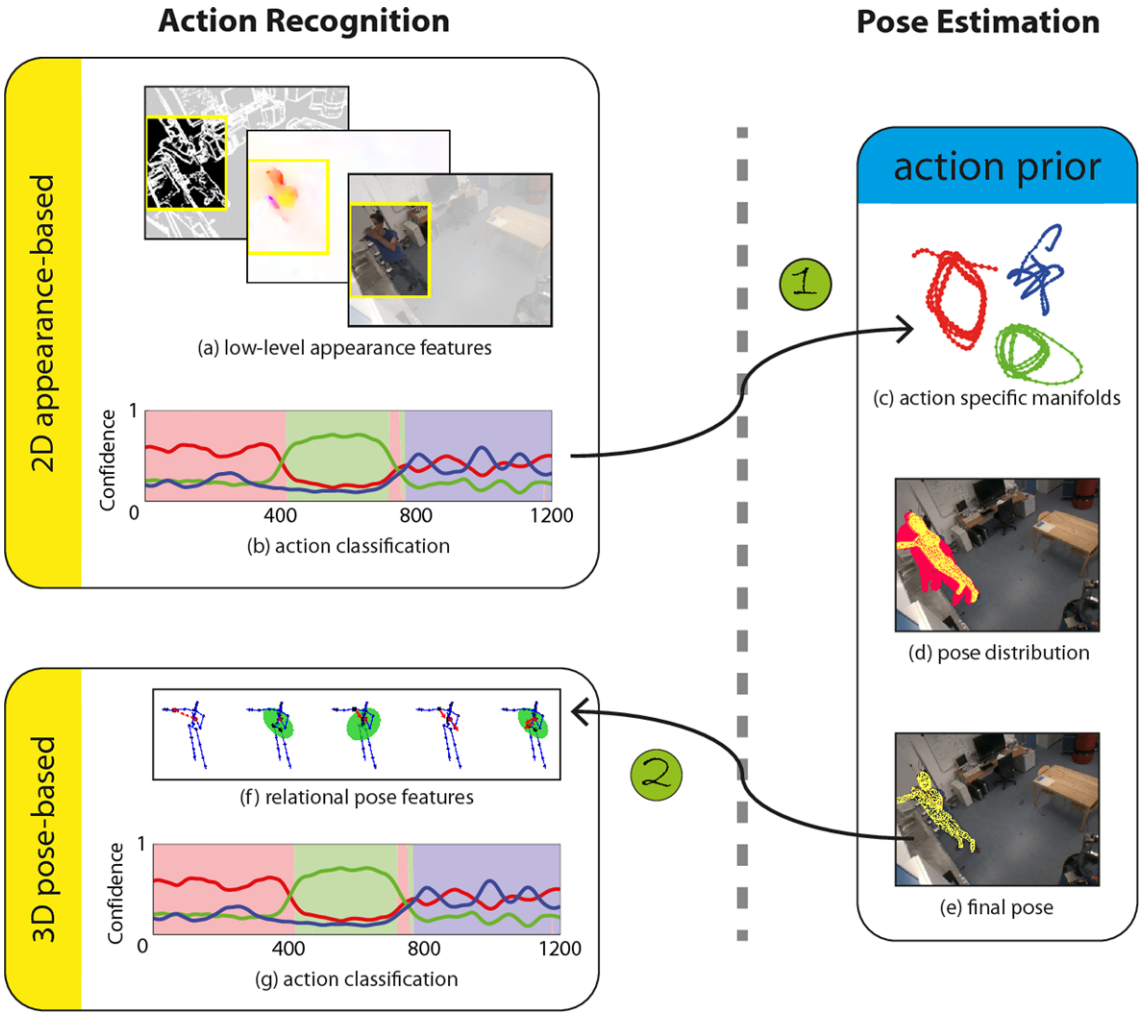


Fig. 1 Overview of the coupled action recognition and pose estimation framework. The framework begins with 2D appearance-based action recognition based on low-level appearance features (**a**) such as colour, optical flow and spatio-temporal gradients. Outputs of the 2D action recognition (**b**) are used as a prior distribution (**c**) for the particle-based optimization for 3D pose estimation (**d**) (*Arrow 1*). Finally, 3D pose-based action recognition (**g**) is then performed based on pose-based features (**f**) extracted from the estimated poses (**e**) (*Arrow 2*)

and Hlavac 2008; Kläser et al. 2010; Natarajan et al. 2010; Yang et al. 2010) or completely bypass the difficulties of pose estimation and directly classify actions with abstract and low-level appearance features (Dollar et al. 2005; Efros et al. 2003; Jhuang et al. 2007; Laptev and Lindeberg 2003; Schindler and Van Gool 2008; Willems et al. 2009).

Given that human pose estimation and action recognition are such closely intertwined tasks, information from one task can be leveraged to assist the other and vice versa. Therefore, we advocate in this paper the use of information from action recognition to help with pose estimation and vice versa for the following reasons. First, using the results of an action classifier is a simple way to bring together many single-activity priors for pose estimation in multi-activity sequences. Secondly, pose-based action recognition has several advantages. For example, pose representations suffer little of the intra-class variances common in appearance-based systems; in particular, 3D skeleton poses are viewpoint and appearance invariant, such that actions vary less from actor to actor. Furthermore, using pose-based representations greatly simplifies learning for the action recognition itself, since the relevant high-level information has already been extracted.

We introduce a framework which builds upon the results of action recognition to help with human pose estimation, the results of which are then used to refine the action label, as illustrated in Fig. 1. Our framework begins with 2D appearance-based action recognition using low-level appearance features such as colour, optical flow and spatio-temporal gradients. The outputs of the 2D action recognition are used as a prior distribution for the particle-based optimization for 3D pose estimation (Arrow 1 in Fig. 1). Finally, we perform 3D pose-based action recognition using pose-based features extracted from the estimated poses (Ar-

row 2 in Fig. 1). While we acknowledge the difficulties of both action and pose estimation as individual tasks, we show that perfect results from either are not necessary to have an impact. In summary, the contributions of the framework are twofold:

1. **Action recognition helps pose estimation.**
   We propose a new algorithm that integrates the results of a 2D action recognition system as a prior distribution for optimization. Low-dimensional manifolds are often used to simplify 3D pose estimation, but the complexity of the embeddings increases with the number of actions. Separate, action-specific manifolds seem to be more practical; here, we adapt a particle-based annealing optimization scheme (Gall et al. 2008b) to jointly optimize over the action-specific manifolds and the human poses embedded in each of the manifolds. The approach scales in the worst case linearly with the number of manifolds but can be made much more efficient with an action prior.
2. **Robust pose-based action recognition.**
   We demonstrate the robustness of using relational pose features for pose-based action recognition. Because semantically similar motions which can be grouped into a single action are not necessarily numerically similar (Kovar and Gleicher 2004; Müller et al. 2005), previous works (Thurau and Hlavac 2008; Kläser et al. 2010; Natarajan et al. 2010; Yang et al. 2010) have encoded pose implicitly. As such, we also do not directly compare 3D skeleton joints in space and time. Instead, we use relational pose features, which describe geometric relations between specific joints in a single pose or a short sequence of poses. Relational pose features, introduced in Müller et al. (2005), have been used previously for indexing and retrieval of motion capture data. Here, we modify a subset of them for action recognition and show that with these features, it is not necessary to have perfect poses to perform action recognition.

Preliminary versions of this paper appeared in Gall et al. (2010b), which described how 2D action recognition could be used as a prior for improving 3D pose estimation and in Yao et al. (2011), which classified actions based on 3D poses. The current paper couples the two into a single framework and contains an extensive experimental section investigating the impact of pose estimation accuracy on action recognition and as well as the impact of differently sized training data.

2 Related Works

As action recognition and 3D pose estimation are both very active fields of research, there exists a large body of literature on both topics. We refer the reader to the excellent reviews (Poppe 2010; Aggarwal and Ryoo 2010) on action recognition and (Moeslund et al. 2006; Forsyth et al. 2006; Sigal et al. 2010) on human pose estimation and tracking for a more complete overview. We focus our discussion here on a comparison of appearance- versus pose-based action recognition and pose estimation with priors.

2.1 Action Recognition

Early works in recognising human actions relied on recovering articulated poses from frame to frame and then linking together either the poses or pose-derived features into sequences. Pose information was obtained from motion capture systems (Campbell and Bobick 1995) or segmentation (Yacoob and Black 1999; Rao et al. 2002). The sequences themselves were then classified either through exemplar matching (Gavrila and Davis 1995; Yacoob and Black 1999; Rao et al. 2002) or with state-space models such as HMMs (Campbell and Bobick 1995).

An alternative line of work models the entire body as a single entity, using silhouettes or visual hulls (Bobick and Davis 2001; Lv and Nevatia 2007; Weinland et al. 2007; Weinland and Boyer 2008; Blank et al. 2005). These works are sometimes called pose-based approaches, in reference to the extracted silhouettes of the human body; however, we consider silhouettes to be a specialised appearance feature, since it offers little interpretation of the individual body parts, and categorise these works as appearance-based approaches.

To avoid articulated tracking or segmentation, recent works have shifted towards the use of local, low-level appearance features such as Gabor filter responses (Jhuang et al. 2007; Schindler and Van Gool 2008) and optical flow (Efros et al. 2003). Lately, spatio-temporal interest points have become especially popular, e.g. cuboids (Dollar et al. 2005), 3D Harris corners (Laptev and Lindeberg 2003; Schuldt et al. 2004) and 3D Hessians (Willems et al. 2009). Most of these are extensions of their 2D counterparts used in object detection and their usage follows a traditional object detection approach. After interest point detection at multiple scales, feature descriptors are computed, clustered, and assigned to a code-book to be used in some bag-of-words representation (Laptev et al. 2008; Dollar et al. 2005; Liu et al. 2009). These approaches have shown great success in natural and unconstrained videos, such as feature films (Laptev et al. 2008), broadcast sports (Rodriguez et al. 2008) and YouTube (Liu et al. 2009). The use of low-level appearance features requires little to no high-level preprocessing and is clearly more advantageous in scenarios in which pose estimation is extremely difficult (e.g. monocular views) or even impossible (e.g. very low resolutions (Efros et al. 2003)).

Despite their success, low-level appearance-based features offer little intuition with regards to the actor performing the action, much less the various poses that constitute the action itself. In many action recognition applications in which the scenario is slightly more constrained, it is not only helpful but natural to infer activity from the actor and his pose. In an attempt to bring back the "human" to human action recognition, works such as (Thurau and Hlavac 2008; Kläser et al. 2010; Natarajan et al. 2010; Yang et al. 2010) have tried to couple person detectors with the action recognition task and focus on features which are related to the human pose. However, the pose is never solved for explicitly and is instead handled implicitly by the various models and/or classifiers.

2.2 Pose Estimation

One of the most popular ways to reduce the complexity of the human pose estimation problem is to use a prior model learned from motion capture databases. The most basic approaches rely on database matching, where the previously estimated poses in the sequence are used as a query to search for the most similar motion exemplar in a database. Approaches can be either on-line, to predict the pose for the next frame (Sidenbladh et al. 2002; Rosenhahn et al. 2007), or offline, to refine the tracked poses (Baak et al. 2009).

Since exemplar-based models do not generalise well, several methods have been proposed to model priors in low-dimensional spaces. Among the simplest are those based on PCA (Baumberg and Hogg 1994; Sidenbladh et al. 2000; Urtasun et al. 2005). More complex priors include those generated from dimensionality reduction techniques such as Isomap (Tenenbaum et al. 2000) (see Gall et al. 2010b), LLE (Roweis and Saul 2000) (see Elgammal and Lee 2004; Jaeggli et al. 2009; Lee and Elgammal 2010) and Laplacian Eigenmaps (Belkin and Niyogi 2002) (see Sminchisescu and Jepson 2004) or probabilistic latent variable models such as the commonly used GPLVM (Lawrence 2005) and GPDM (Wang et al. 2008) (see Urtasun et al. 2006; Moon and Pavlovic 2006; Hou et al. 2007; Geiger et al. 2009; Ukita et al. 2009). More recently, Taylor et al. (2010) introduced the use of Conditional Restricted Boltzmann Machines, composed of large collections of discrete latent variables.

Instead of building priors based on poses or motion models, other approaches learn a mapping between the image space and the pose space. These approaches recover the pose directly from silhouettes and image features (Rosales and Sclaroff 2001; Agarwal and Triggs 2006; Sminchisescu et al. 2007; Bo and Sminchisescu 2010). In Taycher et al. (2006), for instance, pose estimation is formulated as inference in a conditional random field model where the observation potential function is learned from a large set of training data.

2.3 Integrated Action Recognition and Pose Estimation

Using pose information for labeling actions is not new. As previously discussed in Sect. 2.1, some of the earliest works in action recognition focused on tracking body parts and classifying the joint movements. More recent approaches which follow this line of work include (Yilmaz and Shah 2005; Ali et al. 2007; Husz et al. 2011), though they all assume that poses are readily available, either from hand labeling (Yilmaz and Shah 2005; Ali et al. 2007) or from an independent tracker (Husz et al. 2011). In the context of gesture and sign language recognition, as well as facial expression recognition, a common model is to first track the hands and or face and then perform classification based on the estimated pose parameters. Gesture recognition is beyond the scope of the present work and we refer the readers to the review article (Mitra and Acharya 2007).

Little work, however, has been done to leverage action labels for pose estimation, as much of the previous work in pose estimation has been focused on sequences of single action classes rather than longer multi-activity sequences. In Raskin et al. (2011), an annealed particle filter (Deutscher and Reid 2005) was used for tracking in a single low dimensional space trained on a few basic actions; action classification was then performed on the tracked poses. A similar approach was proposed in Darby et al. (2010) where PCA was used for dimensionality reduction and a hidden Markov model for modeling dynamics, but in contrast to (Raskin et al. 2011), transitions between different actions are modeled explicitly. Finally, in Jenkins et al. (2007), multiple particle filters were used in parallel in activity-specific latent spaces; pose likelihoods from each of the particle filters were then combined and normalized into a pseudo-distribution from which the individual pose and action label are selected, based on the highest probability.

Since complexity increases with the number of actions and many dimensionality reduction techniques struggle to establish useful embeddings for a high number of actions, mixture models (Lin et al. 2006; Li et al. 2007, 2010) or switching models (Pavlovic et al. 2000; Jaeggli et al. 2009; Chen et al. 2009) that rely on action-specific manifolds have been shown to be more flexible. We also follow the concept of action-specific manifolds. However, we do not need to observe transitions between actions for training since we do not model pose estimation as a filtering problem over time but as an optimization problem over the manifolds for each frame.

3 Overview

As illustrated in Fig. 1, our framework begins with 2D appearance-based action recognition based on low-level appearance features (Sect. 4.2). 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 and the pose is estimated by an optimization over the entire set of manifolds (Sect. 5). Finally, we perform 3D pose-based action recognition based on pose-based features extracted from the estimated poses (Sect. 4.3).

4 Action Recognition

4.1 Hough Forest Classifier

For classifying the actions, we use the Hough-transform voting method of Yao et al. (2010), which can be easily adapted to use both appearance features as well as pose-based features. A random forest, i.e. a *Hough forest* (Gall et al. 2011), is trained to learn a mapping between features extracted from the data (either from appearance or pose) and a corresponding vote in an action Hough space. Each tree T in the Hough forest is constructed from a set of annotated features $\mathcal{P} = \{(\mathcal{F}_i, a_i, d_i)\}$. $\mathcal{F}_i$, feature i, can be either appearance-based or pose based; a_i is the action label ($a_i \in \mathcal{A}$) and $\boldsymbol{d}_i$ is the temporal displacement of the feature center with respect to the action center in the sequence.

Trees are built recursively, starting with the entire collection of features at the root. At each non-leaf node, a large pool of binary tests t associated with the feature values are randomly generated to split the annotated features $\mathcal{P}$ into two subsets, $\mathcal{P}_L(t)$ and $\mathcal{P}_R(t)$. The optimal binary test $t^\star$ maximizes the gain $\Delta H(t)$, where

$$\Delta H(t) = H(\mathcal{P}) - \sum_{S\in\{L,R\}} \frac{|\mathcal{P}_S(t)|}{|\mathcal{P}|} \cdot H\big(\mathcal{P}_S(t)\big). \tag{1}$$

Depending on the measure H used, nodes can be either classification or regression nodes. For classification, entropy

$$H(\mathcal{P}) = -\sum_{a\in\mathcal{A}} p(a|\mathcal{P}) \log p(a|\mathcal{P}) \tag{2}$$

is used, where $p(a|\mathcal{P})$ is given by the percentage of samples with class label a in the set $\mathcal{A}$. For regression, the sum-of-squared-differences is used as an objective function:

$$H(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{a\in\mathcal{A}} \sum_{i:a_i=a} \|d_i - \overline{d}_a\|_2^2, \tag{3}$$

where $\overline{d}_a$ is the mean of the temporal displacement vectors for class a. The $t^\star$ found is stored at the node and the sets $\mathcal{P}_L(t^\star)$ and $\mathcal{P}_R(t^\star)$ are passed to the left and right child node. The tree grows until some stopping criterion is met, i.e. the child node is of a maximum depth, or there are less than a minimum number of patches remaining. When training is complete, the leaves store the proportion of features per action label which reached the leaf L (p_a^L) and the features' respective displacement vectors (D_a^L).

At classification time, features are densely extracted from the test track and passed through all trees in the forest. The features are split according to the binary tests stored in the non-leaf nodes and, depending on the reached leaf L, cast votes proportional to p_a^L for the action label a and the associated temporal center.

4.2 Appearance Features

When using appearance features with Hough forests, $\mathcal{F}_i$ is a spatio-temporal cuboid ($15 \times 15 \times 5$ pixels) extracted from feature channels such as spatial gradients or optical flow, i.e. $\mathcal{F}_i = (I_i^1, \ldots, I_i^f, \ldots, I_i^F)$, where each I_i^f is channel f at patch i and F is the total number of channels. We use the same low-level appearance features as (Yao et al. 2010): colour, dense optical flow (Brox et al. 2004) and spatio-temporal gradients. The binary tests at each node are comparisons of two pixels at locations $\boldsymbol{p} \in \mathbb{R}^3$ and $\boldsymbol{q} \in \mathbb{R}^3$ in feature channel f with some offset τ:

$$t(f; \boldsymbol{p}, \boldsymbol{q}; \tau) = \begin{cases} 0 & \text{if } I^f(\boldsymbol{p}) - I^f(\boldsymbol{q}) < \tau, \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where $\boldsymbol{p}, \boldsymbol{q}$, f and τ are learned during training by optimizing (1).

4.3 Pose Features

For encoding pose information, we have adopted the relational features introduced by Müller et al. (2005). These features describe geometric relations between specific joints in a single pose or a short sequence of poses (e.g. the distance between the shoulder and the wrist, (see Fig. 2(a)) or the distance of the wrist with respect to a plane formed by the shoulder and hip joints (see Fig. 2(b)).

Given multiple instances of an action, relational features are more robust to spatial variations than the poses themselves (Müller et al. 2005). Previous works have also shown that semantically similar motions belonging to the same action are not necessarily numerically similar (Kovar and Gleicher 2004; Müller et al. 2005); by encoding the pose in a relative manner, it is easier to capture semantic similarity (Müller et al. 2005). While Müller et al. (2005) hand-tuned the features for indexing and retrieval of motion capture data, the Hough Forest framework selects the optimal features during training.

Let $p_{j_i,t} \in \mathbb{R}^3$ and $v_{j_i,t} \in \mathbb{R}^3$ be the 3D location and velocity of joint j_i at time t. The joint distance feature F^{jd} (see Fig. 2(a)) is defined as the Euclidean distance between joints j_1 and j_2 at times t_1 and t_2 respectively:

$$F^{jd}(j_1, j_2; t_1, t_2) = \|p_{j_1,t_1} - p_{j_2,t_2}\|. \tag{5}$$

(a) joint distance (b) plane (c) normal plane (d) velocity (e) normal velocity

Fig. 2 Pose-based features used in the 3D pose-based action recognition. (**a**) Euclidean distance between two joints (*red*). (**b**) Plane feature: distance between a joint (*red*) and a plane (defined by three joints—*black*). (**c**) Normal plane feature: same as plane feature, but the plane is defined by its normal direction of two joints (*black squares*) centered at a third joint (*black circle*). (**d**) Velocity feature: velocity component of a joint (*red*) in the direction of two joints (*black*). (**e**) Normal velocity feature: velocity component of a joint in normal direction of the plane defined by three other joints (*black*)

If $t_1 = t_2$, then F^{jd} is the distance between two joints in a single pose; if $t_1 \neq t_2$, then F^{jd} would encode distances between joints separated by time.

The plane feature F^{pl} (see Fig. 2(b)) is defined as

$$F^{pl}(j_1, j_2, j_3, j_4; t_1, t_2) = \operatorname{dist}\big(p_{j_1,t_1}, \langle p_{j_2,t_2}, p_{j_3,t_2}, p_{j_4,t_2}\rangle\big), \tag{6}$$

where $\langle p_{j_2}, p_{j_3}, p_{j_4}\rangle$ indicates the plane spanned by p_{j_2}, p_{j_3}, p_{j_4}, and $\operatorname{dist}(p_j, \langle\cdot\rangle)$ is the Euclidean distance from point p_j to the plane $\langle\cdot\rangle$. Similarly, the normal plane feature F^{np} (see Fig. 2(c)) is defined as

$$F^{np}(j_1, j_2, j_3, j_4; t_1, t_2) = \operatorname{dist}\big(p_{j_1,t_1}, \langle p_{j_2,t_2}, p_{j_3,t_2}, p_{j_4,t_2}\rangle_n\big), \tag{7}$$

where $\langle p_{j_2}, p_{j_3}, p_{j_4}\rangle_n$ indicates the plane with normal vector $p_{j_2} - p_{j_3}$ passing through p_{j_4}.

The velocity feature F^{ve} (see Fig. 2(d)) is defined as the component of v_{j_1,t_1} along the direction of $p_{j_2} - p_{j_3}$ at time t_2:

$$F^{ve}(j_1, j_2, j_3; t_1, t_2) = \frac{v_{j_1,t_1} \cdot (p_{j_2,t_2} - p_{j_3,t_2})}{\|(p_{j_2,t_2} - p_{j_3,t_2})\|}. \tag{8}$$

Similarly, the normal velocity feature F^{nv} (see Fig. 2(e)) is defined as the component of v_{j_1,t_1} in the direction of the normal vector of the plane spanned by p_{j_2}, p_{j_3} and p_{j_4} at time t_2:

$$F^{nv}(j_1, j_2, j_3, j_4; t_1, t_2) = v_{j_1,t_1} \cdot \hat{n}_{\langle p_{j_2,t_2}, p_{j_3,t_2}, p_{j_4,t_2}\rangle}, \tag{9}$$

where $\hat{n}_{\langle\cdot\rangle}$ is the unit normal vector of the plane $\langle\cdot\rangle$.

Binary tests for the pose features can be defined as follows:

$$t(f; j_1, \ldots, j_n; t_1, t_2; \tau) = \begin{cases} 0 & \text{if } F^f(j_1, \ldots, j_n; t_1, t_2) < \tau, \\ 1 & \text{otherwise,} \end{cases} \tag{10}$$

where f, $j_1, \ldots, j_n$, t_1, t_2, τ are pose-based feature types, joints, times and thresholds respectively.[1] The parameters of the binary tests are selected during training as for the appearance features.

4.4 Multiview 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, may be more distinguishable from another view.

In Yao et al. (2010), the action recognition problem is broken down 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 Yao et al. (2010). Instead, background subtraction is used to generate silhouettes of the person performing the action (see 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. As classifier combination strategy, the max-rule is used to fuse the outputs from the multiple cameras (Kittler et al. 1998) (see Fig. 3).

[1] We have kept all planes to be defined by joints at t_2, though planes can in theory be defined in space-time by joints at different time points.

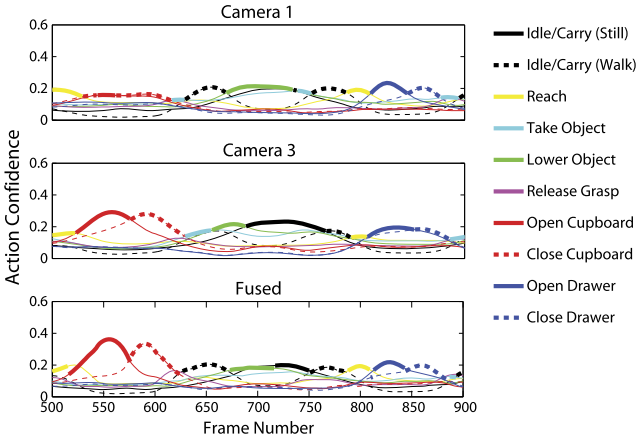


Fig. 3 Normalized action confidences for 2D appearance-based action recognition. Two camera views as well as the fused confidences of all four camera views are shown for frames 500-900 of episode *0-11*. Action confidences are generally higher after fusion, i.e. higher peaks

Algorithm 1 Interacting Simulated Annealing over $\mathbb{E}$

For $k = 1, \ldots, It$

- *Selection*
  - $\forall s^i \in \mathcal{S}_{k-1}$: $w^i = \exp(-\beta_k \cdot V(r^i, t^i, \Theta^i))$
  - $\forall s^i \in \mathcal{S}_{k-1}$: $w^i = w^i / \sum_{s^j \in \mathcal{S}_{k-1}} w^j$
  - $\mathcal{S}_k = \emptyset$; $\forall s^i \in \mathcal{S}_{k-1}$ draw u from $\mathcal{U}[0, 1]$:
    If $u \leq w^i / \max_{s^j \in \mathcal{S}_{k-1}} w^j$ then
    - $\mathcal{S}_k = \mathcal{S}_k \cup \{s^i\}$

    Otherwise
    - $\mathcal{S}_k = \mathcal{S}_k \cup \{s^j\}$, where s^j is selected with probability w^j
- *Mutation*
  - $\mu = \frac{1}{|\mathcal{S}_k|} \sum_{s^j \in \mathcal{S}_k} (r^j, t^j, \Theta^j)$
    $\Sigma = \frac{\alpha_\Sigma}{|\mathcal{S}_k|-1} (\rho\, I + \sum_{s^j \in \mathcal{S}_k} ((r^j, t^j, \Theta^j) - \mu)((r^j, t^j, \Theta^j) - \mu)^T)$
  - $\forall s^i \in \mathcal{S}_k$ sample (r^i, t^i, Θ^i) from $\mathcal{N}((r^i, t^i, \Theta^i), \Sigma)$

5 3D Human Pose Estimation

5.1 Optimizing Over a Set of Manifolds

Having a skeleton and a surface model of the human, the human pose is represented by the joint angles $\Theta = \theta_1, \ldots, \theta_D \in \mathbb{E}_\Theta$ where each joint has between 1 and 3 degrees-of-freedom (DOF). For instance, the skeleton provided for the TUM kitchen dataset (Tenorth et al. 2009) comprised 26 joints, where the number of angles associated with each joint is 3 ($D = 78$). The global rotation r and translation t define the position of the root, which is the middle of the pelvis. This yields a $D + 6$ dimensional state space denoted by $\mathbb{E}$. In this paper, we formulate pose estimation as an optimization problem over $\mathbb{E}$ for a given positive energy function V, i.e. $\min_{x \in \mathbb{E}} V(x)$.

We use the negative log-likelihood based on edge and silhouette features in Shaheen et al. (2009), as the energy function:

$$V(x) = \lambda_{edge} \cdot V_{edge}(x) + \lambda_{silh} \cdot V_{silh}(x). \tag{11}$$

λ_{edge} and λ_{silh} controls the influence of the edge and silhouette terms respectively. V_{edge} and V_{silh} are determined by comparing the edges and silhouettes in the observed image versus that which is generated by projecting the human model according to the pose encoded in x. More precisely,

$$V_{edge} = \frac{|E_P(x) \notin E_I|}{|E_P(x)|}, \tag{12}$$

i.e., the fraction of pixels observed in $E_P(x)$, the projected edge map from the model, which do not overlap with E_I, the edge map observed in image I by applying a Sobel operator. Similarly,

$$V_{silh} = \frac{|S_P(x) \notin S_I|}{2 \cdot |S_P(x)|} + \frac{|S_I \notin S_P(x)|}{2 \cdot |S_I|}, \tag{13}$$

i.e., the fraction of pixels in $S_P(x)$, the projected silhouette from the model, which do not overlap with S_I, the silhouette observed in image I by applying background subtraction and vice versa. Note that edges and silhouettes are not optimal features for human pose estimation, since edges are sensitive to background clutter, clothing textures and wrinkles, while silhouettes are sensitive to occlusions and background changes. However, the associated energy function is fast to compute and therefore fixed for all our experiments.

As a baseline, we implemented Interacting Simulated Annealing (ISA), a particle-based annealing optimization scheme over $\mathbb{E}$ (Algorithm 1). ISA has been used previously in the multi-layer pose estimation framework in Gall et al. (2010a). The optimization scheme, based on the theory of Feynman-Kac models (Del Moral 2004), iterates over a selection and mutation step, and is also the underlying principle of the annealed particle filter (Deutscher and Reid 2005).

In the following, we briefly describe the notations used in Algorithm 1 and throughout the paper. The set of particles is denoted by $\mathcal{S}$. When optimizing over $\mathbb{E}$, a particle is given by $s^i = (r^i, t^i, \Theta^i)$ and an estimate of the pose is given by the weighted mean of the particles after the last iteration, i.e. $\hat{x} = \sum_{s^i \in \mathcal{S}} (w^i \cdot s^i)$. Although the weighted mean may not be the optimal choice for a multi-modal distribution, the annealing emphasizes the dominant mode and justifies the use of the weighted mean (Gall et al. 2008b).

In our experiments, we use a polynomial annealing scheme:

$$\beta_k = (k + 1)^b, \tag{14}$$

where β_k is the annealing temperature, k is the iteration and $b = 0.7$. The mutation step is implemented with the scaling factor $\alpha_\Sigma = 0.4$ and the positive constant $\rho = 0.0001$. The

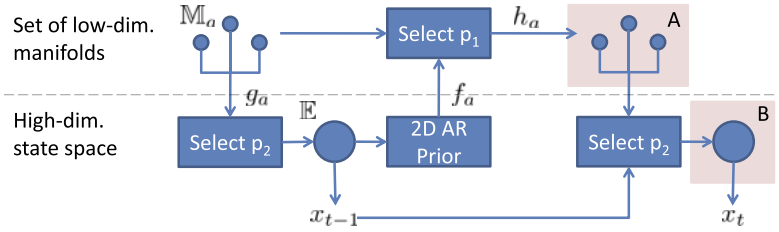


Fig. 4 Overview of the particle-based optimization scheme for pose estimation. For each action class a, we learn an embedding in a low-dimensional manifold $\mathbb{M}_a$. The manifolds are indicated by the *small circles* and the high-dimensional state space $\mathbb{E}$ is indicated by the *large circle*. 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)

uniform distribution and the normal distribution are denoted by $\mathcal{U}[0,1]$ and $\mathcal{N}(\mu,\Sigma)$, respectively.

We modify the baseline algorithm to optimize over a set of manifolds instead of a single state space. To this end, we learn for each class a 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:

$$f_a : \mathbb{E}_\Theta \mapsto \mathbb{M}_a, \qquad g_a : \mathbb{M}_a \mapsto \mathbb{E}_\Theta, \qquad h_a : \mathbb{M}_a \mapsto \mathbb{M}_a, \tag{15}$$

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 $\Theta_a = f_a(\Theta)$. When optimizing over the set of manifolds $\mathbb{M}_a$, a particle $s^i = (y_a^i, a^i)$ stores the corresponding manifold label a^i in addition to the vector $y_a^i = (r^i, t^i, \Theta_a^i)$. Our algorithm operates both in the state space as well as in the manifolds. An overview of the algorithm is given in Fig. 4.

5.2 Action-Specific Manifolds

Each of the action-specific low-dimensional manifolds, $\mathbb{M}_a$, is learned from the joint angles Θ in motion capture data using Isomap (Tenenbaum et al. 2000), 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 (Rasmussen and Williams 2006), f_a (16) and g_a (17), 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.

$$y = f_a(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big), \tag{16}$$

$$x = g_a(y) \sim \mathcal{GP}\big(m(y), k(y, y')\big). \tag{17}$$

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

$$y_t = h_a(y_{t-1}) \sim \mathcal{GP}\big(m(y_{t-1}), k(y_{t-1}, y'_{t-1})\big). \tag{18}$$

While we have chosen Isomap for dimensionality reduction and GP regression to learn the mappings, other dimensionality reduction and regression techniques are also suitable.

5.3 Theoretical Discussion

As mentioned in Sect. 5.1, 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

$$\min_{a\in\mathcal{A}}\Big(\min_{y\in\mathbb{M}_a} V\big(g_a(y)\big)\Big). \tag{19}$$

Minimizing the problem this way, i.e. searching for the global minimum in each of the 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:

$$(\hat{y}, \hat{a}) = \operatorname*{argmin}_{a\in\mathcal{A}, y\in\mathbb{M}_a} V\big(g_a(y)\big). \tag{20}$$

The optimization over the manifolds (20), however, does not provide the same solution as the original optimization problem over the state space since a low-dimensional manifold cannot represent the high dimensional poses exactly. Furthermore, the data used for learning the manifolds $\mathbb{M}_a$ might not contain the correct pose at all. Therefore, we perform a second optimization over the full state space starting with the solution of (20), $x_0 = g_{\hat{a}}(\hat{y})$, as initial point:

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{E}} V(x). \tag{21}$$

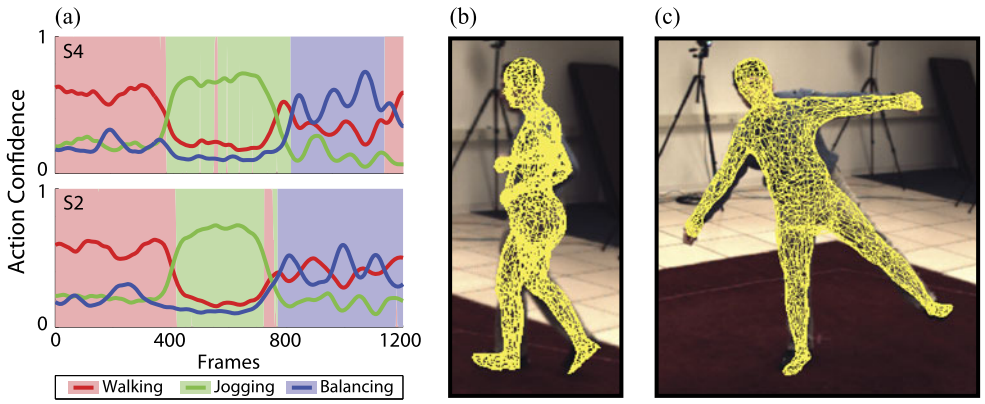


Fig. 5 HumanEva-II. Action recognition prior from camera C1 (**a**). The *curves* show the action confidence per frame. Note the smooth transitions between the actions around frame 800 for subject S4. After jogging, the subject walks a few steps before balancing. At the end of the sequence, the person 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**)

Since ISA converges to the global optimum in the probabilistic sense (Gall et al. 2008b), the original problem is solved. While ISA can be directly used to optimize (21) without using the solution of (20) (*baseline*), we will show that the two-step optimization will drastically reduce the pose estimation error if the number of iterations and particles for ISA are limited. A more detailed description of the two optimization steps is given Sect. 5.4.

Note that the solution of (20), $\hat{a}$, is not unique since there is usually an overlap of poses between the manifolds. If the manifolds do not overlap much, the optimization of the pose propagates the particles into the "right" manifold, i.e. the correct action, as plotted in Fig. 5.

5.4 Algorithm

The proposed algorithm at a glance is outlined as Algorithms 2 and 3 and illustrated in Fig. 4. The different steps *Optimization A*, *Select* p_2, *Optimization B* and *Select* p_1 are described in the following:

Optimization A: Since ISA (Algorithm 1) is not directly applicable for optimizing over a set of manifolds, we have to modify the algorithm. For the weighting, the particles $s^i = (r^i, t^i, \Theta_a^i, a^i)$ with $\Theta_a^i \in \mathbb{M}_{a^i}$ are mapped back to the full space in order to evaluate the energy function V:

$$w^i = \exp\bigl(-\beta_k \cdot V\bigl(r^i, t^i, g_{a^i}\bigl(\Theta_a^i\bigr)\bigr)\bigr), \tag{22}$$

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

$$\Sigma_a = \frac{\alpha_\Sigma}{|\mathcal{S}_a| - 1}\Biggl(\rho I + \sum_{s^i \in \mathcal{S}_a} \bigl(\Theta_a^i - \mu_a\bigr)\bigl(\Theta_a^i - \mu_a\bigr)^T\Biggr), \tag{23}$$

$$\mu_a = \frac{1}{|\mathcal{S}_a|} \sum_{s^i \in \mathcal{S}_a} \Theta_a^i. \tag{24}$$

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 $\mathcal{S}$:

$$\Sigma_0 = \frac{\alpha_\Sigma}{|\mathcal{S}| - 1}\Biggl(\rho I + \sum_{s^i \in \mathcal{S}} \bigl(\bigl(r^i, t^i\bigr) - \mu\bigr)\bigl(\bigl(r^i, t^i\bigr) - \mu\bigr)^T\Biggr), \tag{25}$$

$$\mu = \frac{1}{|\mathcal{S}|} \sum_{s^i \in \mathcal{S}} \bigl(r^i, t^i\bigr). \tag{26}$$

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

Algorithm 2 Optimizing over $\mathbb{M}_a$

Optimization A:

For $k = 1, \ldots, It_A$

- *Selection*
  - $\forall s^i \in \mathcal{S}_{k-1}^{\mathbb{M}}$: $w^i = \exp(-\beta_k \cdot V(r^i, t^i, g_{a^i}(\Theta_a^i)))$
  - $\forall s^i \in \mathcal{S}_{k-1}^{\mathbb{M}}$: $w^i = w^i / \sum_{s^j \in \mathcal{S}_{k-1}^{\mathbb{M}}} w^j$
  - $\mathcal{S}_k^{\mathbb{M}} = \emptyset$; $\forall s^i \in \mathcal{S}_{k-1}^{\mathbb{M}}$ draw u from $\mathcal{U}[0,1]$:
    If $u \leq w^i / \max_{s^j \in \mathcal{S}_{k-1}^{\mathbb{M}}} w^j$ then
    - $\mathcal{S}_k^{\mathbb{M}} = \mathcal{S}_k^{\mathbb{M}} \cup \{s^i\}$

    Otherwise
    - $\mathcal{S}_k^{\mathbb{M}} = \mathcal{S}_k^{\mathbb{M}} \cup \{s^j\}$, where s^j is selected with probability w^j
- *Mutation*
  - $\forall a \in \mathcal{A}$: $\mu_a = \frac{1}{|\mathcal{S}_a|} \sum_{s^j \in \mathcal{S}_a} \Theta_a^j$ with $\mathcal{S}_a = \{s^i \in \mathcal{S}_k^{\mathbb{M}} : a^i = a\}$
    $\forall a \in \mathcal{A}$: $\Sigma_a = \frac{\alpha_\Sigma}{|\mathcal{S}_a|-1}(\rho I + \sum_{s^j \in \mathcal{S}_a} (\Theta_a^j - \mu_a)(\Theta_a^j - \mu_a)^T)$
    $\mu_0 = \frac{1}{|\mathcal{S}_k^{\mathbb{M}}|} \sum_{s^j \in \mathcal{S}_k^{\mathbb{M}}} (r^j, t^j)$
    $\Sigma_0 = \frac{\alpha_\Sigma}{|\mathcal{S}_k^{\mathbb{M}}|-1}(\rho I + \sum_{s^j \in \mathcal{S}_k^{\mathbb{M}}} ((r^j, t^j) - \mu_0)((r^j, t^j) - \mu_0)^T)$
  - $\forall s^i \in \mathcal{S}_k^{\mathbb{M}}$ sample Θ_a^i from $\mathcal{N}(\Theta_a^i, \Sigma_{a^i})$ and (r^i, t^i) from $\mathcal{N}((r^i, t^i), \Sigma_0)$

Select p_2:

- $\hat{a} = \operatorname{argmin}_{a \in \mathcal{A}} \|\hat{\Theta}_{t-1} - g_a(f_a(\hat{\Theta}_{t-1}))\|$
  $(\Sigma_{\hat{a}})_{ii} = \frac{|\hat{\Theta}_{t-1} - g_{\hat{a}}(f_{\hat{a}}(\hat{\Theta}_{t-1}))|_i}{3}$
- $\mathcal{S}_{It_A}^{\mathbb{E}} = \emptyset$; $\forall s^i \in \mathcal{S}_{It_A}^{\mathbb{M}}$ draw u from $\mathcal{U}[0,1]$:
  If $u < p_2$ then
  - $\mathcal{S}_{It_A}^{\mathbb{E}} = \mathcal{S}_{It_A}^{\mathbb{E}} \cup \{(r^i, t^i, g_{a^i}(\Theta_a^i))\}$

  Otherwise
  - $\mathcal{S}_{It_A}^{\mathbb{E}} = \mathcal{S}_{It_A}^{\mathbb{E}} \cup \{(r^i, t^i, \hat{\Theta})\}$, where $\hat{\Theta}$ is sampled from $\mathcal{N}(\hat{\Theta}_{t-1}, \Sigma_{\hat{a}})$

Optimization B:

For $k = It_A + 1, \ldots, It_B$

- *Selection*
  - $\forall s^i \in \mathcal{S}_{k-1}^{\mathbb{E}}$: $w^i = \exp(-\beta_k \cdot V(r^i, t^i, \Theta^i))$
  - $\forall s^i \in \mathcal{S}_{k-1}^{\mathbb{E}}$: $w^i = w^i / \sum_{s^j \in \mathcal{S}_{k-1}^{\mathbb{E}}} w^j$
  - $\mathcal{S}_k^{\mathbb{E}} = \emptyset$; $\forall s^i \in \mathcal{S}_{k-1}^{\mathbb{E}}$ draw u from $\mathcal{U}[0,1]$:
    If $u \leq w^i / \max_{s^j \in \mathcal{S}_{k-1}^{\mathbb{E}}} w^j$ then
    - $\mathcal{S}_k^{\mathbb{E}} = \mathcal{S}_k^{\mathbb{E}} \cup \{s^i\}$

    Otherwise
    - $\mathcal{S}_k^{\mathbb{E}} = \mathcal{S}_k^{\mathbb{E}} \cup \{s^j\}$, where s^j is selected with probability w^j
- *Mutation*
  - $\mu = \frac{1}{|\mathcal{S}_k^{\mathbb{E}}|} \sum_{s^j \in \mathcal{S}_k^{\mathbb{E}}} (r^j, t^j, \Theta^j)$
    $\Sigma = \frac{\alpha_\Sigma}{|\mathcal{S}_k^{\mathbb{E}}|-1}(\rho I + \sum_{s^j \in \mathcal{S}_k^{\mathbb{E}}} ((r^j, t^j, \Theta^j) - \mu)((r^j, t^j, \Theta^j) - \mu)^T)$
  - $\forall s^i \in \mathcal{S}_k^{\mathbb{E}}$ sample (r^i, t^i, Θ^i) from $\mathcal{N}((r^i, t^i, \Theta^i), \Sigma)$

Algorithm 3 *Select p_1*

- $\mathcal{S}^{\mathbb{M}} = \emptyset$; $\forall s^i \in \mathcal{S}_{It_A}^{\mathbb{M}}$ draw u from $\mathcal{U}[0,1]$:
  If $u < p_1$ then
  - $\mathcal{S}^{\mathbb{M}} = \mathcal{S}^{\mathbb{M}} \cup \{s^i\}$

  Otherwise
  - $\mathcal{S}^{\mathbb{M}} = \mathcal{S}^{\mathbb{M}} \cup \{(r^j, t^j, f_{a^j}(\Theta^j), a^j)\}$, where $(r^j, t^j, \Theta^j) \in \mathcal{S}_{It_B}^{\mathbb{E}}$ and a^j is selected with probability $p(A \mid T = t, \mathcal{I})$

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 modeled in our setting. As shown in Fig. 8(b), it is useful to use the previous estimate $\hat{x}_{t-1} = (\hat{r}_{t-1}, \hat{t}_{t-1}, \hat{\Theta}_{t-1})$ 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 $\hat{x}_{t-1}$:

$$\hat{a} = \operatorname*{argmin}_{a \in \mathcal{A}} \left\| \hat{\Theta}_{t-1} - g_a\big(f_a(\hat{\Theta}_{t-1})\big) \right\|, \tag{27}$$

$$\sigma_{\hat{a},i} = \frac{|\hat{\Theta}_{t-1} - g_{\hat{a}}(f_{\hat{a}}(\hat{\Theta}_{t-1}))|_i}{3}. \tag{28}$$

We create a new set of particles by sampling from the normal distribution $\mathcal{N}(\hat{\Theta}_{t-1}, \Sigma_{\hat{a}})$, 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 (Algorithm 1) 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 *Optimization A*, we continue with β_{It_A+1} instead of β_1.

Select p_1: After *Optimization 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 *Optimization A* and the particles in the state space $\mathcal{S}^{\mathbb{E}}$ after *Optimization B*. The selection is controlled by the parameter $p_1 \in [0,1]$ (Algorithm 3). 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 $p(A|T = t, \mathcal{I})$:

$$p(A = a \mid T = t, \mathcal{I}) = \frac{1}{|\mathcal{A}|}, \tag{29}$$

$$p(A = a \mid T = t, \mathcal{I}) = p(A = a \mid I_{t-l} \cdots I_{t+l}). \tag{30}$$

The first prior (29) is a *uniform prior* that is independent of the current frame and results in a joint optimization over the manifolds $\mathbb{M}_{a \in \mathcal{A}}$ and poses $y \in \mathbb{M}_a$. The second prior (30) is termed *action prior*. It distributes the particles to manifolds that are more likely a-priori based on the 2D action recognition results. Since a manifold $\mathbb{M}_a$ cannot be explored when $p(A = a|T = t, \mathcal{I}) = 0$ and $\{s^i \in \mathcal{S}^{\mathbb{M}} : a^i = a\} = \emptyset$, we use the particle set $\mathcal{S}^{\mathbb{M}}$ to increase the robustness to temporary errors in the *action prior* as demonstrated in Fig. 8(a). Note that a zero-probability error for the true manifold over many frames cannot be compensated. In our framework (Fig. 1), $p(A|I_{t-l} \cdots I_{t+l})$ is obtained by the Hough-transform voting method described in Sect. 4 using appearance features.

Finally, we want to emphasize that the proposed method for optimizing over the manifolds relates to other sampling strategies like stratified sampling. While the manifolds can be regarded as subsets of the full state space, the *action prior* distributes the samples over the subspaces. In contrast to stratified sampling, however, the subsets are not assumed to be disjoint and to cover the full space.

6 Experiments

For the action prior-based pose estimation system, we test on both HumanEva-II (Sigal et al. 2010) and the TUM kitchen dataset (Tenorth et al. 2009). We also illustrate the complete framework on the TUM kitchen dataset and run two sets of experiments, one with multi-subject training data (full training set) and one with a single subject (S1 training set) for training to test the generalization capabilities of the framework.

6.1 Datasets

HumanEva-II The HumanEva-II (Sigal et al. 2010) dataset is a standard benchmark on 3D human pose estimation. It contains two sequences, one for each subject, S2 and S4; each sequence has three actions (see Fig. 5). 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 (13 joints, $D = 22$) (Gall et al. 2010a). We perform two trials: training on S2 and testing on S4 and vice versa. For learning the action-specific manifolds, we use the tracking results of the multi-layer tracker (Gall et al. 2010a).[2] We split the data into the three action classes and discard the transitions between the actions.

TUM Kitchen The TUM kitchen dataset (Tenorth et al. 2009) is a more challenging dataset than HumanEva-II. 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 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 standard action recognition benchmarks such as KTH (Schuldt et al. 2004) and Weizmann (Blank et al. 2005) and parts of the body are often occluded by objects such as drawers, cupboard doors and tables (see Fig. 1). Sample images of the actions can be seen in Fig. 13. The pose is represented by the provided model with 84 parameters (26 joints,[3] $D = 78$).

Testing was done on episodes *0-2*, *0-4*, *0-6*, *0-8*, *0-10*, *0-11*, and *1-6*. We use two sets of training data, a full set (i.e. all episodes in the dataset except those used for testing) as well as a limited set on episodes *1-0* to *1-5*, recorded only from subject 1, to test generalization capabilities of the framework. For the action recognition, we use the 9 labels that are annotated for the 'left hand' (Tenorth et al. 2009). Since these 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.

6.2 2D Appearance-Based Action Recognition

HumanEva-II We do not quantitatively evaluate the 2D action recognition on the HumanEva-II sequences as the actions are very simple and the system correctly identifies each of the actions. Examples of the action confidences from camera C1 are shown in Fig. 5(a).

[2] We have used tracking results to create the training data since the motion capture data for HumanEva II is withheld for evaluation purposes. Note that training data from markerless tracking approaches is in general noisier and less accurate than data from marker-based systems.

[3] The original model has 28 joints but we do not consider the gaze since it has 0 DOF. The root joint is represented by the global orientation and position (6 DOF).

Table 1 Individual camera and fused action recognition performance for subjects 1–4; fused performance is higher than any individual camera view for each subject. While the average performance from each camera view using the S1 training set is less than using the full training set, fusing each of the camera views makes up for the difference, such that the average fused performance is equal

	S1 training set					full training set				
	C1	C2	C3	C4	Fused	C1	C2	C3	C4	Fused
S1	0.60	0.54	0.62	0.61	0.63	0.59	0.63	0.62	0.61	0.63
S2	0.58	0.55	0.45	0.61	0.65	0.63	0.62	0.60	0.53	0.66
S3	0.72	0.75	0.68	0.69	0.77	0.76	0.80	0.78	0.79	0.79
S4	0.68	0.57	0.66	0.67	0.78	0.73	0.73	0.71	0.69	0.75
average	0.65	0.60	0.60	0.64	0.71	0.68	0.70	0.68	0.65	0.71

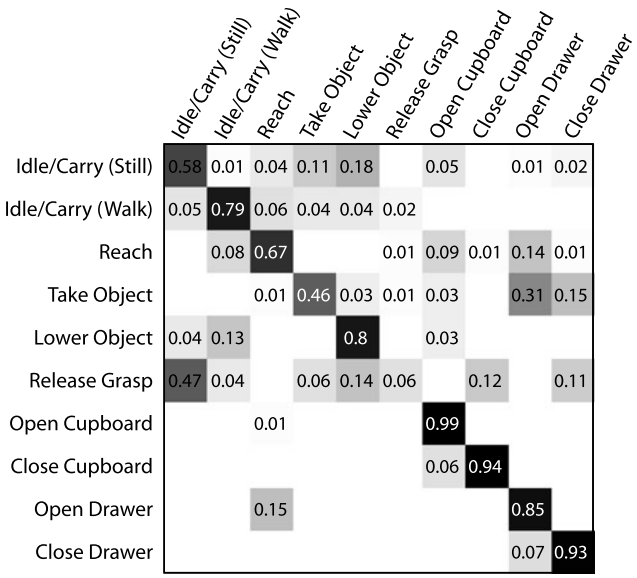
TUM Kitchen For each camera view, we trained a forest of 15 trees of depth 17 each with 50000 random tests generated at all the nodes. Results of the appearance-based action recognition for each individual camera view and for the two different training sets are shown in Table 1. We report here the classification rate from a frame-by-frame basis, averaged over the different action classes. For each sequence, we disregard a time window of 4 frames on either side of a transition from one action to another. Action recognition performance does not vary much from camera to camera, though there is a significant variation between subjects, i.e. for both training sets, S3 and S4 are easier to classify than S1 and S2.

For classifier fusion, we used the max-rule, which gave the best performance in comparison to other standard ensemble methods (Kittler et al. 1998). The classifier fusion has a greater effect on the S1 training set (increased performances of up to 21 %) than on the full training set (increased performances of up to 13 %), so that even with lower average performance on the individual cameras, the fused performance is still equal (0.71). A sample of the normalized classifier output for cameras 1 and 3 as well as the the fused results are shown in Fig. 3.

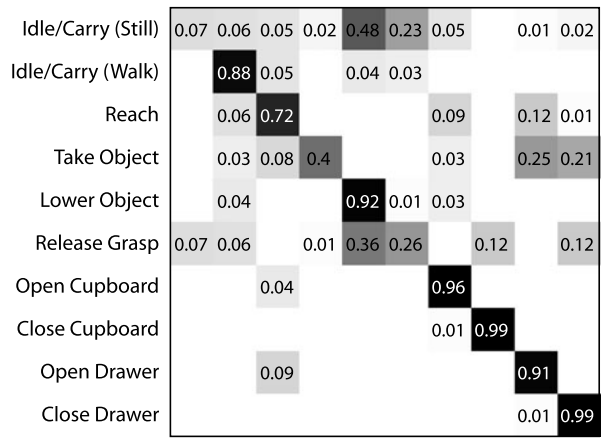
We show a confusion matrix of the fused results for the S1 training set in Fig. 6(a); results for the full training set are similar in trend. The most difficult actions to identify are *"idle/carry (still)"*, *"take object"* and *"release grasp"*. In particular, *"take object"* and *"release grasp"* are transition actions;[4] such high-level movements may be difficult to define based on low-level features alone.

6.3 3D Pose Estimation

For evaluating the pose estimation, we measure the absolute 3D error of the estimated joint positions and report the mean



(a) S1 training set



(b) Full training set

Fig. 6 Confusion matrices for fused action outputs of 2D appearance-based action recognition using (**a**) the S1 training set and (**b**) the full training set. Average performance over all classes is 0.71 for both training sets

[4] *"Take object"* always occurs between *"reach"* and *"idle/carry"* while *"release grasp"* always occurs before *"idle/carry"*, after interacting with an object, the drawer or the cupboard.

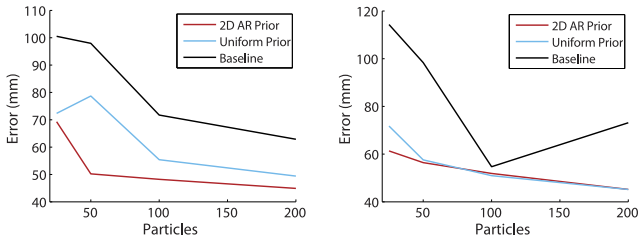


Fig. 7 3D pose estimation error with respect to number of particles. The proposed approach performs significantly better than the direct optimization in the state space $\mathbb{E}$ (*baseline*), particularly for a small number of particles. The discrepancy between the *uniform prior* and the prior obtained from 2D action recognition gets larger for fewer 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 4

error and standard deviation over frames. We perform only one run for each sequence and method, but we use the same random generator with the same seed for all experiments. The standard deviation over several runs will be reported for the TUM kitchen dataset. The methods are initialized with the ground-truth pose, but the initial pose can be also recovered from the silhouettes as in Gall et al. (2010a).

HumanEva For comparison, we report the results for optimizing over the state space $\mathbb{E}$ (*baseline*), i.e. Algorithm 1, and the proposed algorithm with a *uniform prior* and an *action prior*, where the *action prior* is computed as described in Sect. 4. For evaluation, we use 5 iterations for *Optimization A*, and 10 iterations for *Optimization B* unless otherwise specified. For the *baseline*, we run Algorithm 1 with 15 iterations. Sample pose estimates using the *action prior* are shown in Fig. 5(b) and (c).

According to (Gall et al. 2010a; Sigal et al. 2010), 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 as plotted in Fig. 7. Unsurprisingly, 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 3 action classes, the *uniform prior* performs very well and differences between the two priors become prominent only when using 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 scenario where the action recognition is not better than a random guess.[5] Timings and mean errors are given in Table 4. It is interesting to note that the errors for subject S2 is either comparable or even lower than for S4, suggesting that having a perfect body model is not essential to achieve reasonable pose estimates.

In Fig. 8, we plot the impact of the parameters on the pose estimation error. The results clearly support our design decisions for the algorithm (Sect. 5.4).

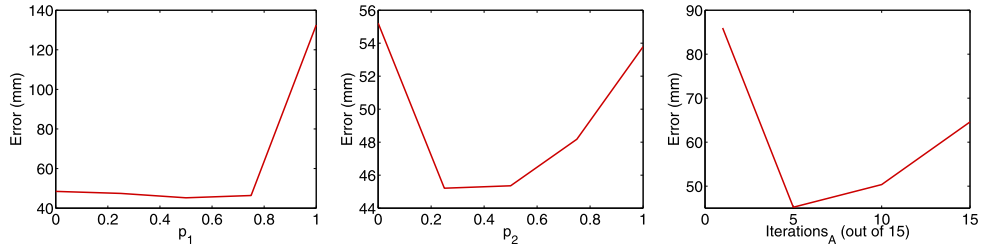
In Fig. 9 and Table 2, we show the pose estimation error with respect to number of camera views using 200 particles. Again, the proposed approach significantly outperforms the *baseline*. At first glance, the *uniform prior* and the *action prior* seem to perform similarly, due to the scaling of the 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 when less particles are used, as shown in Fig. 7, since this results in even fewer particles being distributed to each action class.

We also evaluated the impact of smoothing the estimated joint estimates over time. Since the pose estimates are obtained by computing the mean of a high-dimensional distribution approximated by as few as 200 particles, the estimates are very noisy. Therefore, we filtered the 3D joint positions with a low pass filter. In our experiments, we processed the data by 3 passes of a moving average with a span of 5 frames. As the results in Table 3 show, the smoothing reduces the average error by about 4–6 %.

In Table 5, we compare our approach to state-of-the-art methods reporting results for HumanEva-II. Although the methods are often not directly comparable, since they rely on different assumptions, the results show that the proposed method achieves state-of-the-art performance with respect to accuracy and run time. Though the multi-layer framework (Gall et al. 2010a) does achieve a higher accuracy on the full dataset, it is much slower (124 seconds per frame) than the proposed method (4 seconds per frame) since it uses more expensive image features and a second layer for segmentation-based pose refinement.

TUM Kitchen Based on the fused results of the action recognition, we also evaluate the pose estimation. For the TUM kitchen dataset, we use the provided models with 84 parameters. The large errors for the *baseline* in Fig. 10 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 an accuracy comparable to HumanEva-II, 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 in average by 9–11 % depending on the different training setups.

[5]Note that the worst-case scenario would be if the action recognition is biased and always misclassified certain actions as others.



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

Fig. 8 Evaluation of optimization parameters for pose estimation. (**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 *Optimization 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 *Optimization A* (It_A) and *Optimization B* (15-It_A). The total 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 ($It_A = 5$)

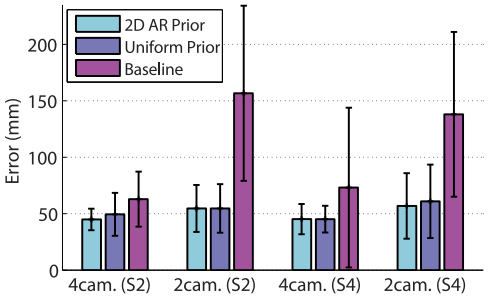


Fig. 9 3D pose estimation error with respect to number of views for HumanEva-II. For the setting with two views, cameras C1 and C2 are taken. The reduced number of views results in more ambiguities. The proposed approach handles these ambiguities better than the direct optimization in the state space $\mathbb{E}$ (*baseline*). The mean errors are also given in Table 2

Table 2 3D pose estimation error (mean ± standard deviation over frames) of the optimization with respect to number of views (*camera*). *ap: action prior; up: uniform prior*

seq.	cameras	ap (mm)	up (mm)
S2	C1-C2	54.6 ± 20.8	54.7 ± 21.5
S2	C1-C4	44.9 ± 9.5	49.4 ± 19.0
S4	C1-C2	56.9 ± 29.0	60.9 ± 32.5
S4	C1-C4	45.2 ± 13.4	45.2 ± 11.8

Table 3 Impact of smoothing

seq.	smoothing	ap (mm)	up (mm)
S2	no	44.9 ± 9.5	49.4 ± 19.0
S2	yes	42.4 ± 8.9	47.0 ± 18.5
S4	no	45.2 ± 13.4	45.2 ± 11.8
S4	yes	42.4 ± 13.0	42.4 ± 11.0

The detailed results in Tables 6 and 7 show that the smoothing reduces the error by 7–8 % for the *uniform prior* as well as the *action prior*. Since the training data may influence not only the action prior but also the learned manifolds $\mathbb{M}_a$, we evaluated the method for both training sets. Even when the system is trained only on one subject (S1 training set), the human poses are well estimated; showing that the method generalises well across subjects.

To give an idea of the standard deviation over several runs, we performed 5 runs with different seeds for episode *0-2* using the *action prior*; see Table 7. Depending on smoothing, the standard deviation over runs is 1 mm and 1.2 mm.

Using the full training set, we also evaluated the pose estimation error with 300 particles and provide the results in Table 8. Similar to HumanEva-II, the differences between the *action prior* and the *uniform prior* become marginal with an increasing number of particles (see Fig. 7). Increasing the particles from 200 to 300 reduces the error by 11 % for the *uniform prior*, whereas the error is only reduced by 2–3 % for the *action prior*. The error reduction is independent of the smoothing, which reduces the error on average by 8 % in both cases, as with 200 particles. This shows that the *action prior* is only beneficial when the number of particles per manifold is very small. Otherwise, the pose estimation efficiently allocates most of the particles to the relevant manifolds and achieves accurate pose estimates even with a *uniform prior*.

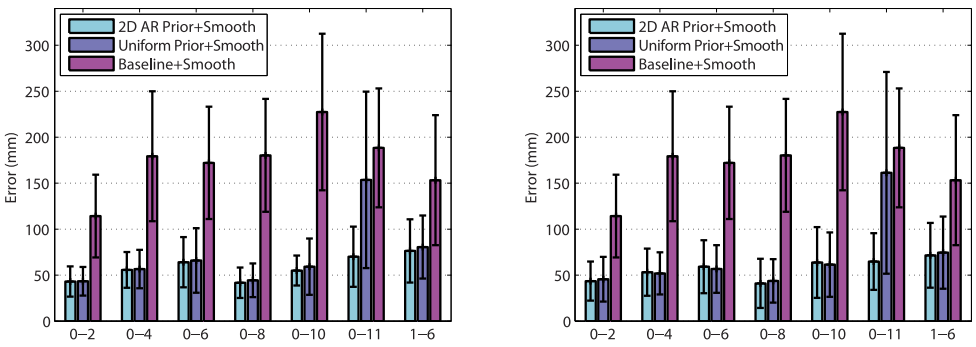
We have chosen to use Isomap, a non-linear embedding technique for creating the action-specific manifolds, though any dimensionality reduction method can be used. As a comparison, we perform pose estimation with the action prior using manifolds created by PCA instead of Isomap. PCA has an advantage over Isomap in that it does not require additional mappings for transitioning to and from the high and low-dimensional spaces. However, it is linear and less expressive than Isomap; we had to use a 20 dimensional PCA space in order to retain 90 % of the data variance. Transitions between successive poses were modeled with a GP regression in the same manner as the Isomap embeddings as described in Sect. 5.2. Pose estimation errors are shown in

Table 4 Computation time per frame and 3D estimation error (mean ± standard deviation over frames) 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*

n	Time (sec.)		S2 Error (mm)			S4 Error (mm)		
	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

Table 5 Comparison of pose estimation errors to state-of-the-art for HumanEva-II. Note that the methods are often not directly comparable since they rely on different assumptions. For instance, several methods have been applied only to a subset of frames (*frames*), e.g. only the walking activity (up to frame 350) or only every 20th frame. The number of cameras (*cam*) also varies. The error (*error*) is the average 3D error in mm and the approximate computational time (*time*) is measured in seconds per frame

method	frames	time	cam	error(S2)	error(S4)
(Baak et al. 2009)		28	4	–	48
(Andriluka et al. 2010)	−350	28	1	101	–
(Sigal et al. 2010)		250	4	83	78
(Peursum et al. 2010)	−380	36	4	107	92
(Gall et al. 2010a)		124	4	38	32
(Bergtholdt et al. 2010)	20th	–	4	207	292
(Brubaker et al. 2010)	−350	–	2	53	54
(Corazza et al. 2010)	−150	–	4	78	80
(Schmaltz et al. 2011)		15	4	–	49
(Gall et al. 2008a)	−400	30	4	–	36
(Gall et al. 2009)		9	4	–	50
(Darby et al. 2010)		15–20	2–3	97	93
ap + sm		**4**	**4**	**42**	**42**
up + sm		**4**	**4**	**47**	**42**



(a) S1 training set (b) full training set

Fig. 10 3D pose estimation error for the TUM kitchen dataset using a limited S1 training set and a full training set. The proposed approach using a 2D action recognition prior performs significantly better than the direct optimization in the state space $\mathbb{E}$ (*baseline*). Impact of the different training sets, however, is small. Mean and standard deviation are provided in Tables 6 and 7

Table 9. Using PCA, the error is only 3–4 % higher, emphasizing the fact that the specific manifold being used is not essential.

In Sect. 5.3, we have pointed out that the solution of the action $\hat{a}$ is not unique when the manifolds share poses. Since this is not the case for HumanEva as shown in Fig. 5, we evaluated the estimated manifold for action recognition on the TUM kitchen dataset. The estimate $\hat{a}$ is given by the manifold that contains most particles after Optimization A (see Sect. 5.4).[6] Depending on the prior used and the number of particles, $\hat{a}$ corresponds only in 30–33 % the cases to the manifold of the action label. This shows that the estimated manifold cannot be used for action recognition when the manifolds overlap since the particles might end up in any of the manifolds that contain the right pose.

[6]This is equivalent to summing the weights of the particles before re-sampling.

Table 6 3D pose estimation error for the TUM kitchen dataset in mm (using S1 training set). *ap: action prior; up: uniform prior; base: baseline*

(mm)	*0-2*	*0-4*	*0-6*	*0-8*	*0-10*	*0-11*	*1-6*
ap	47.8 ± 18.1	60.6 ± 20.7	69.1 ± 29.3	46.9 ± 18.9	60.2 ± 18.4	74.0 ± 33.5	80.2 ± 35.7
up	48.2 ± 17.6	61.5 ± 22.6	71.1 ± 37.1	49.6 ± 20.0	64.7 ± 32.4	159.2 ± 98.0	84.7 ± 35.9
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
ap+smooth	43.1 ± 16.5	55.7 ± 19.4	64.1 ± 27.3	41.7 ± 16.5	55.0 ± 16.3	70.0 ± 32.7	76.4 ± 34.4
up+smooth	43.4 ± 15.6	56.6 ± 20.9	66.0 ± 35.1	44.5 ± 18.3	59.3 ± 30.6	153.6 ± 95.9	80.6 ± 34.3
base+smooth	114.3 ± 45.0	179.3 ± 70.7	172.1 ± 61.1	180.3 ± 61.4	227.4 ± 85.1	188.4 ± 64.7	153.3 ± 70.7

Table 7 3D pose estimation error for the TUM kitchen dataset in mm (using full training set). *ap: action prior; up: uniform prior; base: baseline*

(mm)	*0-2*	*0-4*	*0-6*	*0-8*	*0-10*	*0-11*	*1-6*
ap	48.1 ± 22.4	58.4 ± 27.1	64.6 ± 30.4	45.8 ± 27.8	69.3 ± 39.9	68.7 ± 31.6	75.9 ± 36.5
up	50.3 ± 25.4	57.2 ± 25.5	61.9 ± 27.4	49.0 ± 25.7	67.2 ± 36.9	167.1 ± 114.0	78.6 ± 40.4
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
ap+smooth	43.5 ± 21.3	53.2 ± 25.7	59.2 ± 28.8	41.0 ± 26.7	63.8 ± 38.5	64.8 ± 30.8	71.6 ± 35.2
up+smooth	45.6 ± 24.3	51.9 ± 22.9	56.7 ± 25.9	43.7 ± 23.6	61.5 ± 35.0	161.3 ± 109.7	74.5 ± 39.3
base+smooth	114.3 ± 45.0	179.3 ± 70.7	172.1 ± 61.1	180.3 ± 61.4	227.4 ± 85.1	188.4 ± 64.7	153.3 ± 70.7

Table 8 3D pose estimation error for the TUM kitchen dataset in mm (using full training set and 300 particles). *ap: action prior; up: uniform prior*

(mm)	*0-2*	*0-4*	*0-6*	*0-8*	*0-10*	*0-11*	*1-6*
ap	46.9 ± 23.0	57.4 ± 28.6	59.5 ± 27.3	49.2 ± 33.8	63.2 ± 36.0	66.6 ± 32.2	74.3 ± 37.0
up	47.5 ± 24.0	56.2 ± 23.4	61.9 ± 31.6	47.8 ± 35.0	65.7 ± 41.7	67.4 ± 33.0	72.1 ± 34.6
ap+smooth	42.4 ± 22.1	52.4 ± 26.1	54.4 ± 26.1	44.6 ± 33.1	58.1 ± 35.2	63.1 ± 31.0	70.4 ± 35.9
up+smooth	43.0 ± 22.7	51.1 ± 21.3	56.5 ± 27.4	43.1 ± 34.3	60.4 ± 40.0	63.5 ± 31.9	68.1 ± 33.0

Table 9 3D pose estimation error for the TUM kitchen dataset in mm (using full training set)

(mm)	*0-2*	*0-4*	*0-6*	*0-8*	*0-10*	*0-11*	*1-6*
ap(Isomap)	48.1 ± 22.4	58.4 ± 27.1	64.6 ± 30.4	45.8 ± 27.8	69.3 ± 39.9	68.7 ± 31.6	75.9 ± 36.5
ap(PCA)	46.3 ± 15.9	60.6 ± 35.9	65.6 ± 32.2	50.1 ± 22.3	79.2 ± 61.5	66.2 ± 29.8	74.4 ± 32.3
ap(Isomap)+smooth	43.5 ± 21.3	53.2 ± 25.7	59.2 ± 28.8	41.0 ± 26.7	63.8 ± 38.5	64.8 ± 30.8	71.6 ± 35.2
ap(PCA)+smooth	41.9 ± 14.7	55.7 ± 34.0	60.7 ± 30.6	45.5 ± 20.4	74.4 ± 60.7	62.6 ± 28.9	70.7 ± 30.8

6.4 3D action recognition

TUM Kitchen From the estimated 3D poses, we perform 3D action recognition using the pose features described in Sect. 4.3. For each type of feature, we trained a forest of 15 trees of depth 15[7] each with 20000 random tests generated at all the nodes. We train on the 3D joint positions provided in the TUM dataset; note that these poses were determined by a markerless motion capture system where large errors were manually corrected. We test using the "ground truth" poses, i.e. the poses provided in the dataset *TUM* as well as the estimated poses using our *action prior*, *uniform prior* and *baseline*. As per the 3D pose estimation, we compare two different training sets (S1 training set vs. full training set) and also look at the effects of smoothing the poses over time.

[7]We use a lower depth than the trees trained for 2D appearance-based features since the possible number of unique $\mathcal{F}_i$ for pose-based features is much smaller than that of appearance-based features.

Table 10 Action recognition performance for different relational pose features extracted from the ground truth, action prior, uniform prior and baseline pose estimates. For the action prior, uniform prior and baseline, we report two values to indicate the effects of smoothing on the action recognition performance (unsmoothed/smoothed). *TUM: ground truth; ap: action prior; up: uniform prior; base: baseline*

	particles	S1 training set				full training set			
		joint dist.	plane	velocity	combined	joint dist.	plane	velocity	combined
TUM	–	0.59	0.68	0.68	0.70	0.76	0.79	0.81	0.81
ap	200	0.55/0.54	0.56/0.57	0.31/0.54	0.54/0.57	0.67/0.67	0.70/0.68	0.46/0.73	0.73/0.74
up	200	0.52/0.52	0.51/0.50	0.30/0.49	0.54/0.55	0.67/0.65	0.64/0.62	0.41/0.68	0.66/0.68
base	200	0.19/0.19	0.20/0.20	0.13/0.23	0.25/0.24	0.28/0.28	0.27/0.27	0.21/0.36	0.29/0.28
ap	300					0.68/0.68	0.72/0.72	0.52/**0.77**	0.75/0.75
up	300					0.69/0.69	0.71/0.69	0.52/0.76	0.73/0.73

We show the action recognition performance in Table 10. Unlike the fused 2D action recognition, there is about 10 % performance drop from the full training set to using only the S1 training set. A similar drop in performance does occur for 2D action recognition in the single view case (Table 1), though the classifier fusion scheme compensates for the loss in performance in the 2D case. In contrast to the pose estimation, the action recognition clearly benefits from more training data.

When testing with the TUM poses, there is little difference between the joint distance, plane features and velocity features; combining the three different types of features does not improve the action recognition and average classification remains at 0.81. When using the poses estimated from the *action prior* with 200 particles, there is a 7–10 % performance drop from the TUM poses; the best performance is achieved using the combined features (0.74) though the velocity features on the smoothed poses are similar (0.73). When using the poses estimated from the *uniform prior*, there is a further 5 % drop; again, the best performance is achieved using either the combined features or the velocity features from the smoothed poses (both 0.68). The poses estimated from the *baseline* algorithm are too poor to be used for action recognition, indicating that a pose estimation error over 100 mm is to much for reliable pose-based action recognition.

While temporal smoothing has no effect on the joint distance features and the plane features, it is essential for the velocity features, which are by definition more sensitive to noise. This effect was also observed in the synthetic experiments of Yao et al. (2011) when the TUM poses were corrupted by additive Gaussian noise. If we use the poses estimated with 300 particles, which are on average 2–3 % (*action prior*) or 11 % (*uniform prior*) lower in 3D error in comparison to poses estimated with 200 particles, then the action recognition performance is about 4 % higher. Since the pose estimates with (*action prior*) and (*uniform prior*) become similar with more particles, the action recognition performance becomes similar as well.

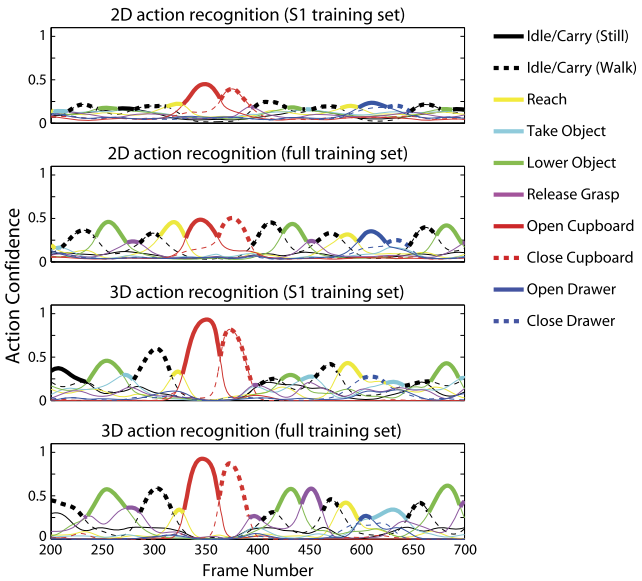


Fig. 11 Normalized action confidences for the 2D appearance-based action recognition and 3D pose-based action recognition from the two training sets for frames 200-700 of episode 0-8. In general, the action confidences are higher for the 3D pose features than the 2D appearance features, i.e. higher peaks

In comparison to the 2D action recognition performance (Table 1), using the poses from the *action prior* (0.77) or *uniform prior* (0.76) with the full training set and 300 particles is better than the fused results (0.71). Using the S1 training set and 200 particles, however, performance is about 10 % worse. Comparing the action confidence outputs shown in Fig. 11, confidences for the 3D pose-based action recognition is higher than the 2D appearance-based recognition. Using the full training versus the S1 training set also results in slightly higher confidences, though this effect is more pronounced in the 2D appearance-based outputs than the 3D pose-based outputs. When looking at the confu-

Table 11 3D pose estimation error for the TUM kitchen dataset in mm (using full training set and 300 particles). *ap(2D): action prior from 2D action recognition (Table 1); ap(3D): action prior from 3D action recognition (Table 10)*

(mm)	*0-2*	*0-4*	*0-6*	*0-8*	*0-10*	*0-11*	*1-6*
ap(2D)	46.9 ± 23.0	57.4 ± 28.6	59.5 ± 27.3	49.2 ± 33.8	63.2 ± 36.0	66.6 ± 32.2	74.3 ± 37.0
ap(3D)	44.1 ± 15.7	58.7 ± 26.7	58.3 ± 23.9	46.6 ± 31.5	61.0 ± 30.0	68.1 ± 34.8	68.5 ± 31.1
ap(2D)+smooth	42.4 ± 22.1	52.4 ± 26.1	54.4 ± 26.1	44.6 ± 33.1	58.1 ± 35.2	63.1 ± 31.0	70.4 ± 35.9
ap(3D)+smooth	39.3 ± 13.9	53.7 ± 24.9	53.0 ± 21.9	42.0 ± 30.6	55.8 ± 28.6	64.4 ± 34.0	64.6 ± 29.9

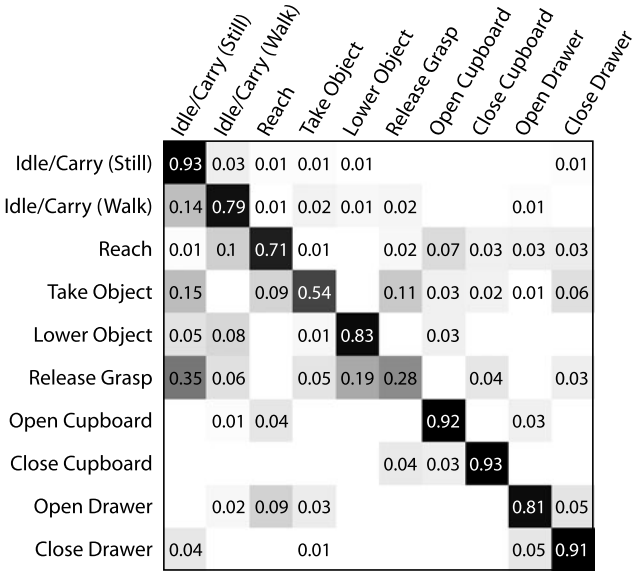


Fig. 12 Confusion matrix for the 3D pose-based action recognition using the full training set with velocity features extracted from smoothed pose estimates, estimated with 300 particles

sion matrix (Fig. 12), one sees that the most difficult classes are again the ambiguous actions such as *"release grasp"* and *"take object"*, though performance is better than the 2D appearance-based system (Fig. 6).

We finally remark that the pose estimation with the *uniform prior* already provides reliable estimates for pose-based action recognition (0.76) and performs better than appearance-based action recognition (0.71), although there is room for further improvement to match the performance with the "ground truth" (*TUM*) poses (0.81). This is particularly relevant for scenarios when view- and environment-dependent training data is difficult to acquire and only MoCap training data is available.

6.5 Closing the Loop

In our system, we transition from an action label to pose estimates and then back to actions again. One can continue and re-estimate the pose based on the 3D pose-based action labels; based on the re-estimated poses, one can again solve for the action labels. In a subsequent iteration, the pose estimation error is reduced by about 3 % (see Table 11) and the action recognition by 1 % error (using velocity features from smoothed poses, we improve from 0.77 to 0.78). These results highlight that for both action recognition and pose estimation, the more accurate the information being leveraged, the better the results.

7 Conclusion

We have presented a system for coupling the closely intertwined tasks of action recognition and pose estimation. The success of the proposed method for pose estimation which achieves state-of-the art performance builds on the ability to jointly optimize over several low-dimensional spaces that represent poses of various activities. Beyond that, unobserved pose variations or unobserved transitions between actions are resolved by continuing the optimization in the high-dimensional space of all human poses. Our experiments have shown that this combination is superior compared to optimization in either space individually. On the one hand, the full human pose has too many degrees of freedom to be optimized efficiently. On the other hand, learned low-dimensional embeddings can be poor at generalization, such that poses which are not present in the training data can not be well estimated. The proposed method benefits from the efficiency of low-dimensional embeddings but also overcomes the problem of generalization. Our experiments have also shown that an action prior improves the pose estimation when the number of particles for optimization are limited. The benefit of the action prior compared to a uniform prior, however, becomes smaller with an increasing number of particles.

Within our proposed action recognition system, 3D pose-based features have been shown to be more successful at classifying the actions than 2D appearance-based features. The same has been shown to be true even when the pose-based features were extracted from the estimated poses of our pose estimation system, indicating that the quality of estimated poses with an average error between 42–70 mm is sufficient enough for reliable action recognition. Since 3D

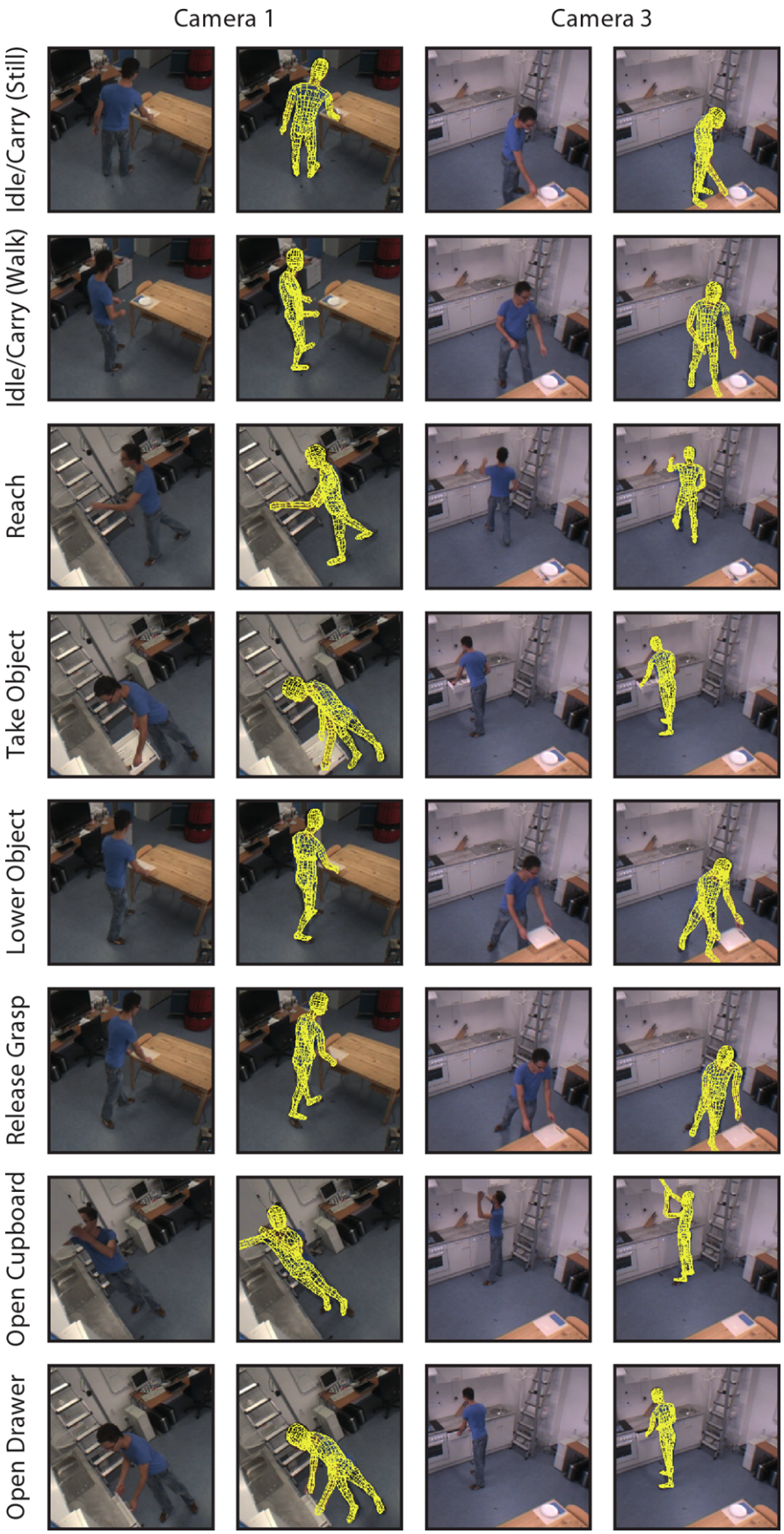


Fig. 13 Cropped images and pose estimates for the actions of TUM kitchen dataset, shown for cameras 1 and 3; *Close cupboard* and *close drawer* are not shown

pose-based features are, in contrast to 2D appearance features, view-independent, it is easier to acquire training data from other datasets. In this way, the pose estimation system with a uniform prior and the pose-based action recognition method can be easily set-up at any location without requiring additional view-specific training data.

In our system, we have shown the advantages of using action recognition for pose estimation and the advantages of using pose estimation for action recognition. The selection of priors for pose estimation, be it the action prior or the uniform prior, is related to the amount of computational resources available at hand, i.e. the number of particles to be used and hence the amount of time required for the pose estimation algorithm. The less the resources, the more benefit there is to be gained from using action information, e.g. at 200 particles, the action prior outperforms the uniform prior. With more resources, e.g. at 300 particles, differences between the action prior and the uniform prior are no longer distinguishable. Given unlimited resources, however, even the baseline algorithm which does not make use of any action information is expected to perform reasonably well. Performance of the pose-based action recognition, while tolerant of errors, is directly related to the pose accuracy. As such, we envision two possible settings to use the current system. If one has more computational resources for pose estimation, then it is preferable to use the uniform prior and bypass the initial 2D action recognition stage, since the benefits of the action prior for pose estimation is no longer distinguishable from the uniform prior. On the other hand, with more limited resources and a focus on pose estimation, it is more preferable to keep the 2D action recognition to improve the accuracy of the pose estimates.

To advance vision-based human motion analysis beyond isolated actions and poses, one should integrate contextual information, either from the environment or objects. Environmental context, e.g. the type of scene or even specific locations within a scene can provide strong indicators to the types of actions and therefore poses which can be expected. Furthermore, interactions with objects can often be the defining characteristic of an action and having a better understanding of human-object interactions would lead to improved recognition on high-level actions such as *"take object"* or *"release grasp"*. Future work will be focused on methods of encoding the contextual information so that it can be efficiently integrated into coupled action recognition and pose estimation systems.

Acknowledgements This work has been supported by funding from the Swiss National Foundation NCCR project IM2 as well as the EC projects IURO, TANGO and RADHAR. Angela Yao was also supported by funding from NSERC Canada.

References

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

Aggarwal, J., & Ryoo, M. (2010). Human activity analysis: a review. *ACM Computing Surveys*.

Ali, S., Basharat, A., & Shah, M. (2007). Chaotic invariants for human action recognition. In *Proceedings international conference on computer vision*.

Andriluka, M., Roth, S., & Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Baak, A., Rosenhahn, B., Mueller, M., & Seidel, H. P. (2009). Stabilizing motion tracking using retrieved motion priors. In *Proceedings international conference on computer vision*.

Baumberg, A., & Hogg, D. (1994). An efficient method for contour tracking using active shape models. In *Proceeding of the workshop on motion of nonrigid and articulated objects*. Los Alamitos: IEEE Computer Society.

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Neural information processing systems*.

Bergtholdt, M., Kappes, J., Schmidt, S., & Schnörr, C. (2010). A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, *87*, 93–117.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2005). Actions as space-time shapes. In *Proceedings international conference on computer vision*.

Bo, L., & Sminchisescu, C. (2010). Twin Gaussian processes for structured prediction. *International Journal of Computer Vision*, *87*, 28–52.

Bobick, A., & Davis, J. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(3), 257–267.

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Proceedings European conference on computer vision*.

Brubaker, M., Fleet, D., & Hertzmann, A. (2010). Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, *87*, 140–155.

Campbell, L., & Bobick, A. (1995). Recognition of human body motion using phase space constraints. In *Proceedings international conference on computer vision*.

Chen, J., Kim, M., Wang, Y., & Ji, Q. (2009). Switching Gaussian process dynamic models for simultaneous composite motion tracking and recognition. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Corazza, S., Mündermann, L., Gambaretto, E., Ferrigno, G., & Andriacchi, T. (2010). Markerless motion capture through visual hull, articulated icp and subject specific model generation. *International Journal of Computer Vision*, *87*, 156–169.

Darby, J., Li, B., & Costen, N. (2010). Tracking human pose with multiple activity models. *Pattern Recognition*, *43*, 3042–3058.

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

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

Dollar, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance (VS-PETS)*.

Efros, A., Berg, A., Mori, G., & Malik, J. (2003). Recognizing action at a distance. In *Proceedings international conference on computer vision*.

Elgammal, A., & Lee, C. S. (2004). Inferring 3d body pose from silhouettes using activity manifold learning. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., & Ramanan, D. (2006). Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, *1*.

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

Gall, J., Rosenhahn, B., & Seidel, H. P. (2008b). An introduction to interacting simulated annealing. In *Human motion: understanding, modelling, capture and animation* (pp. 319–343). Berlin: Springer.

Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., & Seidel, H. P. (2009). Motion capture using joint skeleton tracking and surface estimation. In *Proceedings IEEE conference on computer vision and pattern recognition* (pp. 1746–1753).

Gall, J., Rosenhahn, B., Brox, T., & Seidel, H. P. (2010a). Optimization and filtering for human motion capture—a multi-layer framework. *International Journal of Computer Vision*, *87*, 75–92.

Gall, J., Yao, A., & Van Gool, L. (2010b). 2d action recognition serves 3d human pose estimation. In *Proceedings European conference on computer vision*.

Gall, J., Yao, A., Razavi, N., Van Gool, L., & Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gavrila, D., & Davis, L. (1995). Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International workshop on face and gesture recognition*.

Geiger, A., Urtasun, R., & Darrell, T. (2009). Rank priors for continuous non-linear dimensionality reduction. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Hou, S., Galata, A., Caillette, F., Thacker, N., & Bromiley, P. (2007). Real-time body tracking using a Gaussian process latent variable model. In *Proceedings international conference on computer vision*.

Husz, Z. L., Wallace, A. M., & Green, P. R. (2011) Behavioural analysis with movement cluster model for concurrent actions. *EURASIP Journal on Image and Video Processing*.

Jaeggli, T., Koller-Meier, E., & Van Gool, L. (2009). Learning generative models for multi-activity body pose estimation. *International Journal of Computer Vision*, *83*(2), 121–134.

Jenkins, O. C., Serrano, G. G., & Loper, M. M. (2007). Interactive human pose and action recognition using dynamical motion primitives. *International Journal of Humanoid Robotics*, *4*(2), 365–385.

Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007). A biologically inspired system for action recognition. In *Proceedings international conference on computer vision*.

Kittler, J., Hatef, M., Duin, R., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 226–239.

Kläser, A., Marszałek, M., Schmid, C., & Zisserman, A. (2010). Human focused action localization in video. In *International workshop on sign, gesture, and activity*.

Kovar, L., & Gleicher, M. (2004). Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics*, *23*, 559–568.

Laptev, I., & Lindeberg, T. (2003). Space-time interest points. In *Proceedings international conference on computer vision*.

Laptev, I., Marszałek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, *6*, 1783–1816.

Lee, C., & Elgammal, A. (2010). Coupled visual and kinematic manifold models for tracking. *International Journal of Computer Vision*, *87*, 118–139.

Li, R., Tian, T., & Sclaroff, S. (2007). Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In *Proceedings international conference on computer vision*.

Li, R., Tian, T., Sclaroff, S., & Yang, M. (2010). 3d human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, *87*, 170–190.

Lin, R., Liu, C., Yang, M., Ahja, N., & Levinson, S. (2006). Learning nonlinear manifolds from time series. In *Proceedings European conference on computer vision*.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos 'in the wild'. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Lv, F., & Nevatia, R. (2007). Single view human action recognition using key pose matching and Viterbi path searching. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Maji, S., Bourdev, L., & Malik, J. (2011). Action recognition from a distributed representation of pose and appearance. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Mitra, S., & Acharya, T. (2007). Gesture recognition: a survey. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, *37*(3), 311–324.

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

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

Müller, M., Röder, T., & Clausen, M. (2005). Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics*, *24*, 677–685.

Natarajan, P., Singh, V., & Nevatia, R. (2010). Learning 3d action models from a few 2d videos for view invariant action recognition. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Pavlovic, V., Rehg, J., & Maccormick, J. (2000). Learning switching linear models of human motion. In *Neural information processing systems* (pp. 981–987).

Peursum, P., Venkatesh, S., & West, G. (2010). A study on smoothing for particle-filtered 3d human body tracking. *International Journal of Computer Vision*, *87*, 53–74.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*.

Rao, C., Yilmaz, A., & Shah, M. (2002). View-invariant representation and recognition of actions. *International Journal of Computer Vision*, *50*(2), 203–226.

Raskin, L., Rudzsky, M., & Rivlin, E. (2011). Dimensionality reduction using a Gaussian process annealed particle filter for tracking and classification of articulated body motions. *Computer Vision and Image Understanding*, *115*(4), 503–519.

Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge: MIT Press.

Rodriguez, M., Ahmed, J., & Shah, M. (2008). Action Mach: a spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Rosales, R., & Sclaroff, S. (2001). Learning body pose via specialized maps. In *Neural information processing systems*.

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

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally Linear embedding. *Science*, *290*(5500), 2323–2326.

Schindler, K., & Van Gool, L. (2008). Action snippets: how many frames does human action recognition require. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Schmaltz, C., Rosenhahn, B., Brox, T., & Weickert, J. (2011). Region-based pose tracking with occlusions using 3d models. In *Machine vision and applications* (pp. 1–21).

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: a local svm approach. In *Proceedings international conference on pattern recognition*.

Shaheen, M., Gall, J., Strzodka, R., Van Gool, L., & Seidel, H. P. (2009). A comparison of 3d model-based tracking approaches for human motion capture in uncontrolled environments. In *IEEE workshop on applications of computer vision*.

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

Sidenbladh, H., Black, M., & Sigal, L. (2002). Implicit probabilistic models of human motion for synthesis and tracking. In *Proceedings European conference on computer vision* (pp. 784–800).

Sigal, L., Balan, A., & Black, M. (2010). Humaneva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, *87*(1–2), 4–27.

Sminchisescu, C., & Jepson, A. (2004). Generative modeling for continuous non-linearly embedded visual inference. In *Proceedings international conference on machine learning*.

Sminchisescu, C., Kanaujia, A., & Metaxas, D. (2007). Bm3e: discriminative density propagation for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(11), 2030–2044.

Taycher, L., Demirdjian, D., Darrell, T., & Shakhnarovich, G. (2006). Conditional random people: tracking humans with crfs and grid filters. In *Proceedings IEEE conference on computer vision and pattern recognition* (pp. 222–229).

Taylor, G., Sigal, L., Fleet, D., & Hinton, G. (2010). Dynamical binary latent variable models for 3d human pose tracking. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Tenenbaum, J., de Silva, V., & Langford, J. (2000). *A global geometric framework for nonlinear dimensionality reduction*. Chicago: Science.

Tenorth, M., Bandouch, J., & Beetz, M. (2009). The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *IEEE workshop on tracking humans for the evaluation of their motion in image sequences*.

Thurau, C., & Hlavac, V. (2008). Pose primitive based human action recognition in videos or still images. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Ukita, N., Hirai, M., & Kidode, M. (2009). Complex volume and pose tracking with probabilistic dynamical model and visual hull constraint. In *Proceedings international conference on computer vision*.

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

Urtasun, R., Fleet, D., Hertzman, A., & Fua, P. (2005). Priors for people tracking from small training sets. In *Proceedings international conference on computer vision*.

Wang, J., Fleet, D., & Hertzmann, A. (2008). Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 283–298.

Weinland, D., & Boyer, E. (2008). Action recognition using exemplar-based embedding. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Weinland, D., Boyer, E., & Ronfard, R. (2007). Action recognition from arbitrary views using 3d exemplars. In *Proceedings international conference on computer vision*.

Willems, G., Becker, J., Tuytelaars, T., & Van Gool, L. (2009). Exemplar-based action recognition in video. In *Proceedings British machine vision conference*.

Yacoob, Y., & Black, M. (1999). Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, *73*(2), 232–247.

Yang, W., Wang, Y., & Mori, G. (2010). Recognizing human actions from still images with latent poses. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Yao, A., Gall, J., & Van Gool, L. (2010). A hough transform-based voting framework for action recognition. In *Proceedings IEEE conference on computer vision and pattern recognition*.

Yao, A., Gall, J., Fanelli, G., & Van Gool, L. (2011). Does human action recognition benefit from pose estimation. In *Proceedings British machine vision conference*.

Yilmaz, A., & Shah, M. (2005). Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Proceedings international conference on computer vision*.