Tracking human pose with multiple activity models

John Darby*, Baihua Li, Nicholas Costen

Department of Computing and Mathematics, Manchester Metropolitan University, John Dalton Building, Chester Street, Manchester, M1 5GD, UK

A R T I C L E   I N F O

Article history:
Received 25 June 2009
Received in revised form 20 January 2010
Accepted 20 March 2010

Keywords:
Human motion analysis
3D human tracking
Generative model
Particle filter
Annealing

A B S T R A C T

Tracking unknown human motions using generative tracking techniques requires the exploration of a high-dimensional pose space which is both difficult and computationally expensive. Alternatively, if the type of activity is known and training data is available, a low-dimensional latent pose space may be learned and the difficulty and cost of the estimation task reduced. In this paper we attempt to combine the competing benefits—flexibility and efficiency—of these two generative tracking scenarios within a single approach. We define a number of “activity models”, each composed of a pose space with unique dimensionality and an associated dynamical model, and each designed for use in the recovery of a particular class of activity. We then propose a method for the fair combination of these activity models for use in particle dispersion by an annealed particle filter. The resulting algorithm, which we term multiple activity model annealed particle filtering (MAM-APF), is able to dynamically vary the scope of its search effort, using a small number of particles to explore latent pose spaces and a large number of particles to explore the full pose space. We present quantitative results on the HumanEva-I and HumanEva-II datasets, demonstrating robust 3D tracking of known and unknown activities from fewer than four cameras.

1. Introduction

In this paper we present a 3D generative tracking approach in which the configuration of a volumetric body model is optimised to coincide with observations. This is in contrast to the group of discriminative approaches that predict pose configurations directly from observations. Following the taxonomy of Poppe [1], the generative approach can be further subdivided into a modelling and estimation stage. Modelling requires the specification of a cost function for comparison of the body model with observations, and estimation requires the recovery of the optimal pose given the cost function.

A body model consisting of a simple kinematic tree requires around 30 parameters to specify fully. Even with carefully formulated modelling techniques [2,3], the estimation problem over a 3D space given a single observation has been found to contain a large number of local optima [4]. For this reason probabilistic inference has been favoured by researchers and particle filtering methods [5] have become perhaps the most widely adopted approach to estimation in generative tracking. By maintaining multiple simultaneous hypotheses about the true pose configuration, particle filters are able to account for a multimodal cost surface during estimation.

By deploying a particle filter in a body model’s full pose space and permitting each configuration parameter to vary independently, no restrictions are placed on the type of motions that can be tracked. However, such approaches have relied on large particle numbers to sample the pose space with sufficient density [6]; well constrained dynamical models [7]; and observations from a minimum of four synchronised cameras to minimise ambiguity in the modelling step [8]. Reducing reliance on any one of these factors is desirable, but tends to come at the expense of increased dependence on another.

Learning a low-dimensional latent pose space from training data is an effective method for constraining the estimation task. Projection of training poses onto a low-dimensional manifold encodes correlations between body model parameters, and particle filtering in the resulting subspace has permitted reductions of both particle numbers and camera numbers, e.g. [9–11]. The central drawback of such an approach is that it limits the classes of activity that can be tracked to known activities—that is, those present in the training set.

In this work we argue that the properties of these high- and low-dimensional generative tracking scenarios are complementary and warrant combination. In earlier work we have introduced a particle-based method for the inexpensive recovery of walking poses from a latent pose space [12], and proposed a tentative approach for its extension to two known activities [13]. In this paper we combine and extend our previous efforts to present a tracking approach able to dynamically adjust the scope of its estimation task and additionally recover unknown activities from...
the full pose space. To achieve this we make the following contributions:

- Construction of activity models for known activities. We use PCA to recover a latent pose space from MoCap training data (Section 4.2), and then learn a dynamical model by training a hidden Markov model (HMM) from the resulting distribution of latent variables (Section 4.4). We refer to the combination of pose space and dynamical model as an activity model.
- Integration of the activity model into an annealed particle filtering (APF) [14] framework for particle dispersion during estimation (Section 5.2). We introduce a reversed transition matrix to allow the synthesis of past and future poses during refinement of the pose estimate (Section 4.5), and a new cost function for the comparison of pose candidates with image evidence (Section 7). The resulting algorithm (HMM-APF) is able to track known activities from fewer than four cameras using a small number of particles.
- Evaluation of the proposed HMM-APF scheme on HumanEva-I data. We make a quantitative comparison between the use of HMM-APF and a simpler alternative activity model using Gaussian noise as a dynamical model to explore the same latent pose space (Section 8.1). Our results on HumanEva-I sequences indicate the importance of the HMM for robust tracking.
- Definition of two further complementary activity models. We model known activity transitions by permitting particles to flow between activity manifolds in a joint-activity latent pose space (Section 5.3). Unknown activities are modelled using Gaussian noise to propagate particles in the high-dimensional full pose space (Section 5.1).
- Proposal of a multiple activity model APF (MAM-APF) scheme to unify separate search strategies under the APF framework (Section 6). Our particle stacking approach allows for the simultaneous consideration of multiple activity models described by different dynamical models spanning pose spaces of different dimensionality. A variable number of particles are resampled at each annealing layer, allowing for the recovery of known activities using only a small number of particles in a latent pose space, and unknown activities using a large number of particles in the full pose space (Section 6.2).
- Evaluation of the proposed MAM-APF scheme on HumanEva-II data. We demonstrate robust tracking and classification of the HumanEva-II Combo sequences, which contain known activities, known activity transitions and unknown activity (Section 8.2). MAM-APF allows for a reduction of over 50% in the number of cost function evaluations required during known activity tracking.

2. Related work

Existing generative tracking approaches can be broadly divided between two groups: those that attempt to solve an estimation problem in the body model’s full pose space, e.g., [14,7,6], and those that attempt it in a low-dimensional embedding of the full pose space learned from training data, e.g., [15,16,9]. High-dimensional applications of particle-based estimation—including particle filtering [5], annealed particle filtering [14], adaptive diffusion [17], and partitioned sampling [18]—have required large particle numbers and a minimum of four cameras [8,19,6]. While such approaches are computationally demanding, requiring a large number of cost function evaluations against each camera observation, they have been successful in recovering freeform motions without restriction on activity class.

An alternative to searching the full pose space is to learn a low-dimensional latent pose space from training data. The estimation task has been attempted in linear PCA spaces recovered from MoCap data using both particle filtering [15] and deterministic optimisation [20]. Similar techniques have also been applied in nonlinear latent pose spaces recovered using “piecewise linear” PCA [21], locally linear coordination (LLC) [16], the Laplacian eigenmaps latent variable model (LELVM) [10], and the Gaussian process latent variable model (GP-LVM) [9]. In contrast to high-dimensional approaches, the use of a latent pose space has allowed for robust tracking from fewer cameras, at reduced computational expense. The main drawback of these approaches is their inability to generalise. Although some pose spaces have been shown to account for intra-activity variations in style [22], none are able to account for new activities not featured in the training set.

The exploration of any pose space—be it low- or high-dimensional—by particle filtering also requires the specification of a dynamical model for particle dispersion. First-, second- and higher-order models such as Gaussian random variables [14], autoregressive processes [23] and the variable length Markov model [7] have all been adopted in the full pose space. More recently, each has also been used for particle dispersion in latent pose spaces [9,24,25]. Here we propose an approach to learning dynamical models from latent variables using a hidden Markov model (HMM). The HMM provides a framework for modelling noisy observations of a stochastic process that has been widely adopted by work on gesture representation and recognition, e.g., [21,26]. HMMs offer the capacity to classify the current pose configuration as well as the synthesis of future pose configurations [27].

Our knowledge there have been no previous attempts to unite low- and high-dimensional approaches to generative human motion tracking. The approach we present is partly inspired by the use of mixed-state particle filters to track with multiple dynamical models [28], but additionally adapts the number of particles needed. Variable particle numbers have previously been adopted to minimise an error estimate between the true posterior and the sample-based approximation [29]. However, here we vary their numbers based on the difficulty of the estimation task given a particular activity model. Our approach is similar in style to the variable-mass particle filter for vehicle tracking [30], where variable particle numbers may be allocated to competing dynamical models based on arbitrary criteria.

3. Estimation

Generative approaches to tracking human motion must be able to cope with both nonlinear motions, and non-Gaussian observation functions caused, for example, by background clutter. Particle filtering supports both these requirements, maintaining a finite number of weighted samples to approximate a conditional probability density for the pose configuration given observed data. We briefly review particle filtering below, before describing the annealing extension proposed by Deutscher and Reid [14].

3.1. Particle filtering

Human motion tracking problems can be formulated as the evolution of a system state $x_t$ over time, $t=1,2,...,T$, described by a Markov process and observed by some sensor providing independent observations given $x_t$, [5]. The state density $p_t(x_t)$,
given by \( p(x_t | z_t) \), where \( z_t = (z_1, \ldots, z_t) \) is the set of all observations up until time \( t \), may be propagated over time with the following rule:

\[
p(x_t | z_t) \propto p(z_t | x_t) \int p(x_t | x_{t-1}) p(x_{t-1} | z_{t-1}) \, dx_{t-1}.
\]

(1)

The sequential importance resampling (SIR), or condensation algorithm [5,31], allows for the representation of a multimodal posterior, \( p(x_t | z_t) \), via a finite set of \( B \) weighted particles,

\[
S_t^* = \left\{ (x_t^{(1)}, \pi_t^{(1)}), \ldots, (x_t^{(B)}, \pi_t^{(B)}) \right\}.
\]

(2)

After initialization of the particle set at \( x_t \) (usually with ground truth), \( B \) particles are randomly sampled and dispersed using a dynamical model, \( p(x_t | x_{t-1}) \). Each new point in the state space \( x_t^{(b)} \) is evaluated using a cost function \( w(z_t | x_t^{(b)}) \) and assigned a weighting \( \pi_t^{(b)} \) approximating the observation likelihood \( p(z_t | x_t^{(b)}) \). Resampling then takes place, with \( B \) particles randomly sampled from the existing distribution for dispersion, with likelihood proportional to their weighting, and with replacement. In this way, the particle set may be propagated over time to maintain a representation of \( p(x_t | z_t) \). The expected pose at each time \( t \) can be found by

\[
\mathcal{E}(x_t) = \sum_{b=1}^{B} \pi_t^{(b)} x_t^{(b)}.
\]

(3)

### 3.2. Annealed particle filtering

Given an observation, annealed particle filtering (APF) [14] attempts only to recover the single pose that maximises the cost function. This is done by “cooling” the weighting distribution at each time step and then gradually “warming” it over a number of successive resampling stages, or layers. The result is a slow transition from a broad and inclusive distribution over the pose space to a narrow and discriminative one. This causes resampled particles to concentrate gradually into the globally optimal mode of the cost function. The posterior distribution is not fully represented—a departure from the formal Bayesian framework—but APF has been found to give good results on human motion tracking problems, outperforming SIR [19,8].

Resampling takes place at \( r = R - 1, \ldots, 0 \) separate resampling layers at each time step \( t \), where

\[
w_t(z_t, x_t) = w(z_t | x_t)^{\beta_t},
\]

(4)

with \( \beta_0 > \beta_1 > \cdots > \beta_R \). Setting the exponents too high risks particles becoming distracted by other local optima. Setting them too low means a large number of layers are required to recover an optimal pose. Deutscher and Reid [14,17] proposed a method for the automatic selection of these parameters based on achieving a desired particle survival rate at each layer. The survival rate [18] is an approximation of the fraction of particles that will be resampled from a distribution for inclusion in the next layer,

\[
x_r = D_r / B,
\]

(5)

where \( D_r \) is an estimate of the number of particles resampled,

\[
D_r = \left( \sum_{b=1}^{B} \pi_t^{(b)} \right)^{-1}.
\]

(6)

A high survival rate results in an evenly spread weighting distribution, while a low survival rate concentrates weights into just a few particles. Quantitative investigations into human motion tracking using APF [19,8] have found good tracking poses can be reliably recovered using a constant survival rate of 0.5

\[
\alpha_2 = \cdots = \alpha_0 = 0.5.
\]

(7)

APF is used for the estimation step in the generative approach we present here. To aid our exposition we summarise the steps described in [14] for a single annealing run in Fig. 1.

The original APF implementation proposes a quite general first-order dynamical model \( \text{func}_d(x_{t-1}) \), using the addition of Gaussian noise to approximate \( p(x_t | x_{t-1}) \). Finite differencing of training data is used to find the maximum change in each body model parameter between consecutive time steps. These values

\[
\text{1. At each time step } t \text{ annealing begins at layer } r = R.
\]

\[
\text{2. The annealing run is initialised by a set of unweighted particles, } S_{r,1} = \left\{ (x_t^{(1)}), \ldots, (x_t^{(B)}) \right\}. \text{ These may be the result of a previous annealing run, or the manually initialised particle set } S_{r,1}.
\]

\[
\text{3. Each particle is then assigned a weight based on the evaluation of a cost function,}
\]

\[
\pi_t^{(b)} = w_t(z_t, x_t^{(b)})
\]

(9)

\[
\text{and the results normalised, so that } \sum_{b=1}^{B} \pi_t^{(b)} = 1. \text{ This forms the weighted particle set,}
\]

\[
S_t^* = \left\{ (x_t^{(1)}, \pi_t^{(1)}), \ldots, (x_t^{(B)}, \pi_t^{(B)}) \right\}.
\]

(10)

\[
\text{4. The weighted particle set } S_t^* \text{ is then resampled to give } B \text{ particles randomly drawn with a probability equal to their weighting } \pi_t^{(b)} \text{ and with replacement. As the } b^{th} \text{ particle is drawn, it is dispersed to produce a new unweighted particle using}
\]

\[
x_t^{(b)} = \text{func}_o(x_t^{(b)})
\]

(11)

\[
\text{where } \text{func}_o \text{ represents an arbitrary dynamical model.}
\]

\[
\text{5. A new set } S_{r+1,1} \text{ has now been recovered and is used to initialise the layer } r + 1. \text{ Steps 3 and 4 are repeated until the set } S_{r, \bar{B}} \text{ is produced.}
\]

\[
\text{6. The set } S_{r, \bar{B}} \text{ can be used to calculate the expected tracking pose by}
\]

\[
\mathcal{E}(x_t) = \sum_{b=1}^{B} \pi_t^{(b)} x_t^{(b)}
\]

(12)

\[
\text{7. A new unweighted set } S_{r,1, \bar{R}} \text{ used to initialise the first layer } r = R \text{ of the next annealing run at } t + 1 \text{ is then found by}
\]

\[
x_t^{(b)} = \text{func}_o(x_t^{(b)}).
\]

(13)

Fig. 1. Standard APF particle set dispersion, as proposed by Deutscher and Reid [14].
form the diagonal covariance matrix $P_0$ of a multivariate Gaussian random variable with zero mean $n_0 \sim N(0, P_0)$, that is used for the dispersion of particles. The magnitude of the dynamical model is rescaled at each annealing layer (denoted by func(_)) in Fig. 1) by multiplication of the covariance matrix by the particle survival rate $z_\tau$, to give

$$P_t = z_\tau \times \cdots \times z_\tau \times P_0$$

and $n_\tau \sim N(0, P_\tau)$. This is in order to decrease particle diffusion at the same rate the particle set density increases. We refer to the use during tracking of this activity model—full pose space plus Gaussian noise—as standard APF.

4. Learning activity models

In this section we discuss the steps taken to build activity models in this work. Each activity model consists of a pose space (Sections 4.1 and 4.2) augmented by a dynamical model (Sections 4.3 and 4.5), and can be used for particle dispersion during tracking. To learn activity models we use the Training partition of the HumanEva-I dataset, which provides a number of MoCap activity sequences for use in learning motion priors [8].

4.1. High-dimensional “Full” pose space

The state vector $x_t$ must completely describe the configuration of a geometric body model which can be projected into the image plane for comparison with observations. We adopt the body model of Bălan et al. [19] which consists of a kinematic tree containing ten truncated cones, the sizes of which are taken from HumanEva-I subject measurements.

We define the body model’s configuration using a set of position parameters giving the global translation and rotation of the pelvis, $v = (v_1^1, \ldots, v_3^1)^T$ and a set of pose parameters giving the relative 3DOF joint rotations between limbs, $e = (e^{11}, \ldots, e^{36})^T$. We use the notation $\Omega = (\omega_1, \ldots, \omega_M)$ to denote a set of $M$ pose vectors, and $E = (\xi_1, \ldots, \xi_M)$ to denote a set of $M$ pose vectors. These parameters can be calculated from the MoCap data in the Training partition of the HumanEva-I dataset, see Fig. 2(a).

In the event that we search for a pose solution in the body model’s full pose space, recovering $x_t = (x_1^1, \ldots, x_3^1)$ requires the use of enough particles to sample a 42D space with sufficiently high density. The high-dimensional approach to estimation places no restrictions on pose but is both challenging and computationally expensive, e.g. [8,19,14].

4.2. Low-dimensional “Latent” pose space

To reduce the difficulty of the high-dimensional estimation task, a low-dimensional latent pose space can be recovered from training data. Two particular dimensionality reduction techniques have been widely adopted for generative tracking: (linear) principal components analysis (PCA), e.g. [20,15], and the (nonlinear) Gaussian process latent variable model (GP-LVM), e.g. [9,25]. Each approach may be formulated as a latent variable model, where a matrix of concatenated low-dimensional pose vectors, $F = [f_1, \ldots, f_M]$ are related to a matrix of high-dimensional pose vectors, $E = [\xi_1, \ldots, \xi_M]$ through a set of mapping parameters $W$.

In the probabilistic latent variable formulation of PCA [32], the mapping from latent to full space is a linear one, corrupted by Gaussian noise. Assuming spherical noise covariances and

![Figure 2](image-url)

**Fig. 2.** Construction of latent pose spaces using PCA for Walk and Jog (top and bottom, respectively). (a) 36D pose vectors for known activities. Vertical lines denote the omission of bad MoCap data. (b) 3D latent pose spaces and latent variables, found by PCA. (c) Average 3D absolute error between 500 original and reconstructed pose vectors using a range of PCs. Body dimensions of S4 were used for comparison.
independence between pose vectors, an expression for their conditional probability given the latent space can be derived, \( p(E|F, W) \). The latent variables \( F \) are then marginalised as “nuisance parameters” (using a Gaussian prior), and the mapping parameters \( W \) optimised by maximisation of \( p(E|W) \).

In contrast, derivation of the GP-LVM [33] begins with a dual probabilistic interpretation of PCA in which the mapping parameters \( W \) are marginalised (also using a Gaussian prior), and the latent variables \( F \) optimised by maximisation of \( p(E|F) \). The key contribution of the GP-LVM is to recognise the resulting expression as a product of Gaussian processes (GPs) with linear kernels. By substituting nonlinear kernels, such as the radial basis function, a nonlinear mapping from the latent to the full space is achieved.

PCA and the GP-LVM have competing advantages in terms of data reconstruction accuracy and computational cost. Typically it is not necessary to effect PCA by expectation-maximisation [32] and the principal axes can be recovered by singular value decomposition of the pose vectors’ covariance matrix, requiring negligible computation time. In contrast, GP-LVMs have training requirements with complexity cubic in the number of training points, and it is generally necessary to employ a sparse representation of activity training data. However, the resulting GP-LVM can be used to give lower activity reconstruction errors than PCA; see Quirion et al. [34] for a comprehensive comparison. Perhaps more importantly however, PCA and the GP-LVM share a common limitation: regardless of the particular choice of dimensionality reduction technique, the resulting latent pose space has only a limited capacity to generalise beyond the training data.

To evaluate the ability of these models to generalise we processed a set of Walk pose vectors to recover both a 2D back constrained GP-LVM (BC-GPLVM) [35] and a 2D PCA subspace. We then mapped a single pose from an unknown Box activity into each latent space, see Fig. 3, and reconstructed the pose by mapping back to the full pose space. Both spaces give high reconstruction errors: 271 mm for BC-GPLVM and 267 mm for PCA. Similarly, an exhaustive sampling-based approach will not result in a Box pose being found. This result is representative, and neither the linear nor the nonlinear latent space is able to generalise to substantially novel unknown poses.

The focus of this work is to address this problem by combining latent space estimation for known activity with full space estimation of unknown activity. To this end we choose to use PCA for dimension reduction. This is because particles must be free to flow between full and latent pose spaces during tracking, and the inexpensive bi-directional mapping offered by PCA is ideal for this purpose. For GP-LVMs, calculation of the GP mapping between new points in the latent pose space and the full high-dimensional pose space has complexity quadratic in the number of training points. Further, the GP mapping is not bi-directional, and additional steps, such as the use of “back constraints” [35], must be taken to enable mapping from new points in the full space to new points in the latent space.

In Fig. 2(c) we plot errors for the Walk and Jog pose vectors of HumanEva subjects S1–S3 reconstructed from PCA pose spaces with a range of different dimensionalities, \( \eta \). The error measure is the same used to evaluate tracking performance and is calculated from the average distance between 15 joint centres in the original and reconstructed poses, as defined by Sigal et al. [8]. Although PCA produces higher reconstruction errors than its nonlinear alternatives at a given dimensionality, these errors are still below the state of the art in generative tracking from four cameras [36] by \( \eta = 4 \) (around 20 mm). They are therefore unlikely to present a performance barrier to our approach, which uses fewer cameras. For the construction of all latent pose spaces in this work we fix \( \eta = 4 \) and so full body model configurations can be recovered by sampling a 10D space to find \( x_0 = (x_1^1, \ldots, x_{10}^1)' = (0, \ldots, \alpha_0, f_1, \ldots, f_5)' \). For work on generalising to substantially novel unknown poses using latent variable models, we refer the interested reader to recent work on topological constraints [37] and hierarchies of latent variables [38].

### 4.3. Dynamical modelling by finite differencing

Given a particular pose space we must specify a dynamical model \( f_{c_0}(x_{n-1}) \) in order to conduct estimation by APF. One option is to use finite differencing of training data, just as in the original APF algorithm [14]. By finite differencing the original position vectors \( \Omega = (\alpha_1, \ldots, \alpha_{10}) \), and pose vectors \( E = (e_1, \ldots, e_{10}) \), we can recover associated Gaussian random variables

\[
\begin{align*}
\mathbf{n}_p^0 &\sim N(0, \mathbf{P}_n^0) \\
\mathbf{n}_v^0 &\sim N(0, \mathbf{P}_v^0)
\end{align*}
\]

Similarly, given a set of latent variables \( \mathbf{F} = (f_1, \ldots, f_{10}) \), (e.g., see Fig. 2(b)) we can compute

\[
\mathbf{n}_f^0 \sim N(0, \mathbf{P}_f^0)
\]

![Fig. 3. Reconstructions of an unknown Box pose from Walk latent pose spaces. (a) PCA latent pose space. (b) BC-GPLVM latent pose space using a multi-layer perceptron with 15 hidden nodes for back constraints. Note that neither space is able to generalise to the unknown pose. The BC-GPLVM and PCA spaces give reconstruction errors of 271 and 267 mm, respectively.](image-url)
A single anomalous jump of large magnitude in any one pose space dimension, for example due to inaccuracies in the MoCap data, can result in an overly noisy dynamical model that makes estimation difficult. We chose to use the 95th percentile of delta values for estimation of covariance matrices. Results for a range of different percentiles in latent and full pose spaces are shown in Figs. 4(a) and (c), respectively.

We make use of Gaussian random variables for particle dispersion throughout the work presented here. For example, we use \( n^a \) to disperse position parameters in all experiments, and \( n^f \) to disperse pose parameters when tracking unknown activities in the full pose space (details are presented in Section 5). Additionally, we refer to the use of \( n^f \) to disperse latent parameters as latent APF and adopt it as a baseline in later experiments. Ultimately we find that latent APF is unable to provide robust tracking of known activity, and in Section 4.4 we describe a method to recover better constrained dynamical models using HMMs.

### 4.4. Dynamical modelling by HMMs

The latent variables \( F=\{f_1, ..., f_M\} \) recovered from pose vector training data form smooth trajectories: points that are nearby in the full pose space are also nearby in the latent pose space. The resulting distributions (see Fig. 2(b)) have two important properties: dynamics vary depending on the current location within the training manifold, and the latent space away from the training manifold may contain unrelated, even impossible, pose configurations. Given these properties, we advocate the use of HMMs to recover a better constrained dynamical model than the simple addition of Gaussian noise. The HMM’s construction is ideal for modelling dimensionally reduced activity data. The performance of human activity is an inherently stochastic process, and the latent coordinates \( F=\{f_1, ..., f_M\} \) constitute noisy observations of that process. HMMs allow us to describe such a doubly stochastic system, e.g. [26].

An HMM \( \lambda \) is specified by the parameters \( \{S, A, a, p_f(f)\} \), where \( S=\{s_1, ..., s_N\} \) is the set of states; \( A \) is a transition matrix where \( A_{ij} \) gives the probability of a transition from state \( s_i \) to state \( s_j \); \( a \) is a vector holding the probabilities of a sequence starting in each state, and \( p_f(f) \) is the probability of observing latent vector \( f \) while in state \( s_i \). In this work the emission probability is modelled by a single spherical multivariate Gaussian \( p_f(f) = N(\mu, \Sigma) \) with mean \( \mu_f \) and covariance matrix \( \Sigma \). We initialise 15 states by k-means clustering of \( F \) and set \( A \) randomly (with rows normalised) and \( a \) with every value equal to 1/N. Each HMM parameter is then reestimated using no more than 50 iterations of the Baum–Welch algorithm. In order that synthesis may begin from any point in the activity cycle without penalty, we do not reestimate \( a \). For further detail on HMM training the reader is referred to [27].

Fig. 4(b) shows HMMs learned from the HumanEva-I activity data of subject S2. Using the addition of \( n_f \) for particle dispersion will result in a random walk through the latent pose space, while traversing an HMM will restrict pose estimates to lie close to the training data and provide a spatially sensitive dynamical model.

### 4.5. Time reversal

If an HMM is used for particle dispersion, it will synthesise “future” activity poses as implied by the ordering of the training data. However, if the estimation step recovers an incorrect future pose, dispersion at succeeding time steps has no mechanism to explore “past” activity poses and recover track. We propose the incorporation of a second time reversed transition matrix \( \tilde{A} \) for use in particle dispersion.

An equilibrium distribution \( \psi \) where each element \( \psi_i \) gives the probability of being in state \( s_i \) at any time \( t \) can be estimated from the transition matrix \( A \) by generating a number of transitions and...
recording the current state index. Given $\psi$, the elements of a time reversed transition matrix $\hat{A}$ may be calculated by,

$$\hat{A}_{ji} = \frac{\psi_j}{\psi_i} A_{ij}. \quad (21)$$

We use $\hat{A}$ to provide a second dynamical model for the time reversed activity. Note that for the activities studied here, the reversed transition matrix defines a statistically distinguishable process and so $A$ is not said to be “reversible” [39].

5. Defining activity models

In this section we combine the various techniques described in Section 4 to define three activity models. These are intended for use in particle dispersion during three different scenarios: (i) unknown activities, (ii) known activities, (iii) known activity transitions.

Bălan et al. [19] have observed that if standard APF becomes stuck in the incorrect mode of the cost function, even if only for a few time steps, tracking may never be recovered. This is because the magnitude of the jump through pose space required to recapture track quickly becomes larger than that which is permitted by the dynamical model. For this reason it can be beneficial to exaggerate the levels of diffusion produced by the dynamical model [2,40]. In anticipation of this fact, each of the following subsections describes how the dynamical model may be used to produce $p(x_t | C_0, x_{t-1})$ when creating a new particle set for the next frame with Eq. (13). In line with the APF dispersion scaling in Eq. (8), we rescale the number of

1. The position of the $(b)$th particle in the $r$th layer is given by the position and pose parameters,

$$x^{(b)}_{r} = [\omega_{r}, \eta_{r}] = (\omega_{r, 0}^{1}, ..., \omega_{r, 36}^{1}, \eta_{r, 0}^{1}, ..., \eta_{r, 36}^{1})^T. \quad (17)$$

2. The particle's position parameters $\omega_{r}$ are updated $T_0$ times by the addition of the Gaussian random variable $n_{r}^{\omega}$,

$$\omega_{r+1} = \omega_{r} + \sum_{i=0}^{T_0} n_{r}^{\omega}. \quad (18)$$

3. Similarly, the particle's pose parameters $\eta_{r}$ are updated $T_0$ times by the addition of the Gaussian random variable $n_{r}^{\eta}$,

$$\eta_{r+1} = \eta_{r} + \sum_{i=0}^{T_0} n_{r}^{\eta}. \quad (19)$$

4. The new particle location is then given by

$$x^{(b)}_{r+1} = [\omega_{r+1}, \eta_{r+1}]. \quad (20)$$

Fig. 5. Particle dispersion for unknown activity tracking.

![Fig. 5. Particle dispersion for unknown activity tracking.](image)

Fig. 6. 2D view of particle dispersion over 5 layers for $T_0=4$ in the latent pose space. Layer $r=2$ is omitted to maximise figure size. Initial particle locations are shown in black, intermediate locations in green and final particle locations in red. The Gaussian covariances from which samples are drawn are shown as blue ellipses. (a) Latent APF: particle dispersion by Gaussian noise produces a random walk through the latent pose space that ignores the path of training data. (b) HMM-APF: particle dispersion by the HMM causes particles to move along an “activity axis” ensuring relevant pose hypotheses. Cyan lines depict significant HMM state transition probabilities and shifted state observation densities are depicted with dashed ellipses (see also item 4(c) in Fig. 7). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
1. The position of the \((b)\)th particle in the \(r\)th layer is given by the position and latent parameters,
\[
x^{(b)}_{r} = [\omega_{r}, f_{r}] = (\omega_{r}, \ldots, \omega_{r}, f_{r}, \ldots, f_{r})^T.
\]
(23)

2. The particle’s position parameters \(\omega_{r}\) are updated \(T_r\) times by the addition of the Gaussian random variable \(\xi_{r}\),
\[
\omega_{r+1} = \omega_{r} + \xi_{r}.
\]
(24)

3. For \(r = 0\), the particle’s latent parameters \(f_{r}\) are updated using an HMM \(A\) trained from the distribution of latent variables in a pose space recovered from activity training data (Section 4.4). The current latent vector estimate is assigned to the state \(s_j\) most likely to have emitted it as an observable via \(p_j(f)\).

4. After maximal dispersion at \(r = 0\), the particle’s latent parameters continue to be reestimated as described in point 3 (above) but with the following steps taken to aid refinement for \(r > 0\):

(a) The transition matrix is randomly selected as either \(A\) or the time reversed compliment \(A^T\) in order to allow for the synthesis of past and future poses.

(b) The dispersion rescaling procedure (Eq. 8) is extended to the observation densities at each HMM state to give \(p_{r}(f) = N(f_j, \Sigma_{j})\), where
\[
\Sigma_{j} = \sigma_{x} \times \sigma_{y} \times \sigma_{z} \times \Sigma_{c}.
\]
(25)

(c) Where \(s_j = s_i\), we uncouple the dispersion from \(\mu_j\) and sample \(f_{r+1}\) using the scaled version of the parent state’s covariance matrix \(\Sigma_{j}\), but replacing \(\mu_j\) with the particle’s current latent parameter estimate, \(f_{r}\). This stops the training data from dominating the choice of new pose hypotheses, allowing the cost function scores to guide final refinement.

5. The new particle location is then given by
\[
x^{(b)}_{r+1} = [\omega_{r+1}, f_{r+1}].
\]
(26)

Fig. 7. Particle dispersion for known activity tracking.
The density $p(x_t|Z_t)$ is propagated between time steps—the annealing process recovers a set of particles that are densely concentrated about a particular pose solution. To produce the initialising particle set for the next time step, a dynamical model is used to disperse particles with maximum levels of diffusion, see Eq. (13).

Each of the three activity models described in Section 5 is a candidate for the performance of this dispersion step. Their competing predictive properties are well summarised with reference to the “streetlight effect” [41]. Given a fixed allocation of $B$ particles, the activity models for known activity and known activity transitions are analogous to narrow and bright
streetlights illuminating small regions of the full pose space (via the latent pose space) with high sample density. The number of particles required to recover a solution is small, but if the true solution lies outside this region then the search is a futile endeavour. Alternatively, the activity model for unknown activities is analogous to a wide and dim streetlight illuminating a high-dimensional volume of the full pose space with low sample density. This streetlight should guarantee that we illuminate the true solution, but the number of particles used must be large in order to ensure that we successfully recover it.

In the remainder of this section we propose a multiple activity model annealed particle filtering (MAM-APF) method for the simultaneous consideration of complementary activity models. This is achieved by assigning each activity model its own unique quota of particles when re-initialising the particle set between frames. In each of the annealing layers that follow, a variable number of particles are drawn during resampling based on how well populated each activity model becomes. This approach ensures that enough particles are available to recover unknown activity via the full pose space, but that where known activities occur we do not oversample the latent pose space.

6.1. Simultaneous activity models

As a first step to supporting multiple activity classes during tracking, we propose that each of the three activity models described in Section 5 receive an equal allocation of $B$ particles upon creation of the new (maximum dispersion) particle set at each frame. This constitutes an equal prior on each activity class. $3B$ particles are resampled from the particle set recovered at the previous time step $\mathcal{S}_{t-1}^0$ and divided randomly between each of the three activity models to produce equal quotas of $B$ particles. The activity models are then used to disperse their particle allocations, producing $\mathcal{S}_{t+1}^0$. The result is a maximally diffused particle set that represents the predictions of all three activity models, which may be evaluated and resampled over successive annealing layers to recover a pose that maximises the cost function.

Particles are augmented by their activity model index $\mathcal{a} \in \{1,2,3\}$ and are fully specified by the three parameters $(x^\mathcal{a}_{r,t}, \pi^\mathcal{a}_{r,t}, b^\mathcal{a}_{r,t})$. This index persists throughout the annealing run at each time step and ensures the particle is dispersed using its corresponding activity model at each layer. At each subsequent resampling stage we draw just $B$ particles. These particles may belong to any activity model and no quotas are enforced. By setting the particle number low, we are able to reliably and efficiently track known activity and known activity transitions in the latent pose space. However, we risk losing track where unknown activity occurs and the true pose can only be found by searching the full pose space. Conversely, by choosing $B$ large enough to support full pose space search, we must oversample the latent pose space during known activity thus sacrificing any potential gain in efficiency.

6.2. Variable particle numbers

In order to increase the efficiency of our search, we modify the approach described in Section 6.1 and allow differently sized particle quotas to be allocated to each activity model. The activity models for known activities and known activity transitions (whose dynamical models span a low-dimensional latent pose space) are assigned a quota of $B_1 = B_2 = B_{\text{min}}$ particles each. The activity model for unknown activity (whose dynamical models span the high-dimensional full pose space) is assigned a quota of $B_3 = B_{\text{max}}$ particles. The quotas reflect how many particles are required for each scheme to assume complete responsibility for tracking.

For creation of the new (maximum dispersion) particle set at each time step, every particle is dispersed by its respective activity model. The result is (as in Section 6.1) a maximally diffused particle set that represents the predictions of all three activity models. However, the equal prior on activity classes no longer holds, and the particle set is not suitable for resampling. For example, take the case where after dispersion takes place, every particle achieves the same cost function score. If the distribution of particles between dynamical models is uneven due to the quota allocation, then the resampled particle set will contain the same disparity. This is despite the fact that each model’s predictions explained the current observation equally well.

In order to negate this problem we distinguish between an effective particle number and a unique particle number. We “stack” $B_{\text{max}}/B_{\text{min}}$ particles at each of the $2B_{\text{min}}$ unique particle locations in

$$
B_{\text{max}} / B_{\text{min}}
$$

**Fig. 11.** Modelling transitions between known activities. Poses reconstructed from the transition lines in Fig. 9(b). Individual-subject space (left) and joint-subject space (right). Each set of poses is shown from two rotated views.

**Fig. 10.** 3D view of MAM-APF particle dispersion over 5 layers for $T_0=4$ in the latent pose space. Layer $r=2$ is omitted to maximise figure size. The observed pose is a Walk pose. Multiple activity models are employed for each of known activity (red pluses), unknown activity (green points) and transitions (black points). Unknown activity hypotheses are projected into the latent pose space for visualisation. The numbers of resampled particles from each activity model are shown in the legends, with the counter contribution in brackets. For final layer $r=0$ (maximal dispersion) the unique particle numbers are shown with effective particle numbers in brackets. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
the latent pose space to give an equal number of effective particles in each activity model. By placing multiple particles at the same point, we are effectively returned to the approach described in Section 6.1, but only one cost function evaluation is required per stack. Resampling from this new particle set is no longer biased in favour of schemes with larger quota allocations, and subsequently resampled particle sets do not require stacking.

Rather than resampling a fixed number of particles from the maximally diffused particle set, we resample a variable number of particles based on activity model membership. With every particle that is resampled from a particular activity model \(a = 1, 2, 3\), the value \(B_{\text{max}}/B_{a}\) is added to a counter parameter. Sampling continues to take place until the counter value reaches \(B_{\text{max}}\). The set of survival rates (see Eq. (7)) are used to reshape the weighting function at each layer, just as in standard APF. The difference is that resampling of the distribution may terminate early: a maximum of \(B_{\text{min}}\) particles can be resampled from the latent pose space, or \(B_{\text{max}}\) particles from the full pose space. In general, a mixture of particles from the competing activity models are resampled, see Fig. 10.

7. Cost function

For the modelling step we require a cost function that allows for the comparison of synthesised poses with observations and the calculation of particle weights using \(w(z, x_\tau)\). In standard APF [14] the cost function is based on a sum-squared difference (SSD) \(\Sigma^s\) between a binary observation foreground mask \(V\) found by background subtraction of the observation image, and a set of points \(\{\xi\}\) drawn from the surfaces of the cones in the body model hypothesis projected into the image.

\[
\Sigma^s = \frac{1}{|V|} \sum_{\xi} (1 - V(\xi))^2.
\] (33)

A similar measure is also used for a comparison of edge features calculated between a binary observation foreground mask \(V\) by \(\Sigma^e\). Here \(V\) is replaced by an observation edge mask calculated by convolution of the observation with a gradient-based edge detection mask, and \(\{\xi\}\) is a set of points drawn from the edges of the cones in the body model hypothesis projected into the image. Two SSDs can then be combined and exponentiated to give a single cost value for a particular pose,

\[
w(z, x_\tau) = \exp\left[-(\Sigma^s + \Sigma^e)^2\right].
\] (34)

Baláš et al. [19] found only a small quantitative improvement in tracking accuracy from the additional consideration of edge images. In this work we exchange the use of edge cues for a complementary silhouette-based measure. Our approach could be described as a sampling-based version of the symmetrical pixel-based cost function used by Sigal et al. [8]. The problem with the use of \(\Sigma^e\) in isolation is that no consideration is given to observation foreground that remains unaccounted for by the body model. For example, see the observation foreground masks for a series of “punch” poses in Figs. 12(b)-(e), where an erroneous “low guard” pose hypothesis is largely subsumed by observation foreground and therefore low cost in terms of \(\Sigma^s\). To address this problem, we propose sampling the observation foreground for comparison with the body model. A measure of agreement \(\Sigma^g\) is calculated between a binary hypothesis foreground mask \(W\), and a set of points \(\{v\}\) drawn uniformly from the foreground region of the observation foreground mask \(V\).

\[
\Sigma^g = \frac{1}{|W|} \sum_{v} (1 - W(v))^2.
\] (35)

To account more accurately for the observation foreground masks cast by subjects, we create the binary hypothesis foreground mask from a set of truncated “clothes” cylinders with the subjects’ limb widths scaled by a factor between 1.0–1.5. The hypotheses masks in Figs. 12(b)-(e) use a scaling of 1.5 for each limb, but in practice we set these parameters manually based on a subject’s clothing, e.g. 1.5 for a trousered lower leg, 1.0 for an unclothed head or forearm.

When the measure \(\Sigma^g\) is combined with the standard silhouette comparison \(\Sigma^s\) (by substituting \(\Sigma^g\) for \(\Sigma^e\) in Eq. (34)), synthesised poses are required to satisfy two criteria: the body model should not lie over observation background nor leave observation foreground unaccounted for. We demonstrate the usefulness of this measure by considering the consequences of inducing known, artificial errors in the pose derived from a HumanEva-I Box sequence. As shown in Fig. 12(b)-(e), while the 500 frames of fragment run, the pose extracted stays motionless.

![Fig. 12. Silhouette features for a fixed ”low guard” pose hypothesis with hands held against the torso during a 500 frame box sequence. (a) 3D absolute error scores and corresponding SSD scores. Dashed vertical lines denote punches, with the four bold vertical lines corresponding to the image pairs (b)-(e). These show the sampling strategy for \(\Sigma^s\) (left) and \(\Sigma^g\) (right), with nonmatching samples plotted in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)](image-url)
and is compared on the one hand with the cost scores extracted from the images, and on the other with the true pose obtained from motion capture. This arrangement allows us to consider the relationship between the image-based SSD terms and true pose inaccuracies for a wide range of desired poses; when the algorithm is in fact running, a wide range of possible poses will be tested against a single frame. As can be seen from Fig. 12(a), both cost scores relate quite closely to the true pose difference. However, the revised cost measure \( \sum_{i} \sum_{j} (x_{ij} - \hat{x}_{ij})^2 \) is in fact significantly more strongly associated with the true pose difference, as assessed by a Spearman rank correlation analysis. This shows that while the correlation between the old measure and the pose difference has \( r = 0.267 \), the new measure has \( r = 0.677 \). Due to the large number of frames, \( d.f. = 498 \), and so both of these values are significant far beyond \( P = 0.05 \). Similarly, the probability of the difference between these correlations being generated by chance is too small to be calculated.

8. Experiments

In Section 8.1 we investigate HMM-APF known activity tracking for a range of \( T_0 \) values. We draw comparison with latent APF which uses Gaussian noise as a dynamical model in the same latent pose space. In Section 8.2 we unite each of the three activity models described in Section 5 within an MAM-APF framework to recover the HumanEva-II Combo sequences which contain both known and unknown activity with transitions. We draw comparison with the use of standard APF [14] using the same cost function.

In all experiments, we calculate the 3D absolute error between the expected tracking pose (calculated in the full pose space, see Eq. (12)) and a ground truth MoCap pose at each frame, using the error metric defined by Sigal et al. [8]. This comprises the average 3D distance between a set of 15 virtual markers located at the joint positions of the body model, and the corresponding points on the joints of the subject as defined by a set of MoCap markers.

8.1. Known activity using HMM-APF

In our first experiment we investigated the performance of the single known activity tracking scheme, HMM-APF (see Section 5.2) for a range of \( T_0 \) values. Setting the number of time steps synthesised by the dynamical model to \( T_0 > 1 \) exaggerates the level of particle diffusion produced and can facilitate recovery from errors, making tracking more robust. We also investigate the performance of latent APF using the Gaussian random variable \( \mathbf{n}_t^f \) (see Section 4.3) as a dynamical model in the same pose space.

We tested each approach on a Walk and Jog sequence from the HumanEva-I Validation partition. Walk and Jog activities are of particular interest as they are the two known activities in the HumanEva-II Combo sequences (see Section 8.2). We chose the Walk sequence of subject S1 and the Jog sequence of subject S3 as they are the longest in the HumanEva-I dataset. Pose and position vectors were extracted from S2 and S3’s portions of the HumanEva-I Training partition and finite differencing used to estimate the Gaussian random variable \( \mathbf{n}_t^r \), and PCA applied to recover latent pose spaces and associated HMMs.

For every value of \( T_0 \), we tracked the whole of each sequence ten times using the two cameras C1 and C2. We used 50 particles over five annealing layers, the lowest number tested by Sigal et al. [8]. In Fig. 13 we present average 3D absolute error results for each sequence using both HMM-APF (both with and without the time reversed matrix, \( \hat{A} \)) and latent APF. Error bars show the standard deviation in average absolute 3D error across the ten runs at each \( T_0 \) value.

Average latent APF errors decrease with the number of time steps up to around \( T_0 = 4 \). However tracking failures still take place, as evidenced by the large standard deviations in error. Without the use of the time reversed transition matrix \( \hat{A} \), HMM-APF slowly degrades with increased \( T_0 \). This is particularly noticeable for the faster Jog activity. In contrast, HMM-APF with \( \hat{A} \) correctly tracks both of the sequences across the range of \( T_0 \) values, producing low average errors and low standard deviations in error across each batch of ten runs. These results suggest that, in addition to the latent pose space, the choice of dynamical model is important for producing robust tracking. Further discussion is given in Section 9.

8.2. Known and unknown activity using MAM-APF

In the second experiment we tracked the HumanEva-II Combo sequence for subject S2 using MAM-APF (see Section 6). Pose and position vectors for Walk and Jog were extracted from S2’s portion of the HumanEva-I Training partition. Finite differencing was used to estimate the Gaussian random variables \( \mathbf{n}_t^p \) and \( \mathbf{n}_t^r \) used in unknown activity tracking, and PCA applied to recover a joint-activity latent pose space and associated HMMs for known activity tracking, see top of Fig. 9(b). We assigned \( B_{\text{min}} = 50 \).
particles to each of the latent pose space activity models (known activity, known activity transitions) and \( B_{\text{max}} = 250 \) particles to the full pose space activity model (unknown activity).

S2’s Combo sequence was tracked five times from cameras C1 and C2 with \( T_0 = 3 \). The 3D absolute tracking errors were calculated for each run using the online evaluation system [8]. We also used the final weighted particle set at each frame, \( S_{p,t} \) to perform a simple classification task. First we looked at each particle’s activity model index and classified the current pose as Unknown if more than half belonged to the unknown-activity model. Otherwise we found every particle’s parent HMM state in the latent pose space and classified the activity as Walking if more than half were assigned to the Walk HMM and Jogging if more than half were assigned to the Jog HMM.

Fig. 14 shows the average tracking error across the five runs at each frame, with the colour set according to the mode classification result across the five runs, and the total number of cost function evaluations made at each frame. The number of cost function evaluations is also equivalent to the number of unique particles used per frame (see Section 6.2). Note that cost function evaluations remain low throughout the known activities before rising to recover the unknown activity. Images showing the expected tracking pose superimposed on the image observations of HumanEva-II camera C1 are shown in Fig. 17.

8.3. Unknown subjects

Finally we considered how the MAM-APF approach might be extended to track an unknown subject. A joint-activity joint-subject space was recovered from the training data of all three HumanEva-I subjects and a separate HMM trained for each subject’s performance of each activity. By capturing the variation between subjects’ performances, we aimed to maximise the ability of the latent pose space to generalise to new styles of known activity. The resulting activity model—shown at the bottom of Fig. 9(b)—was then used to track the HumanEva-II Combo sequence for the unknown subject S4, for whom no training data is available.

Using the same parameters as in Section 8.2, known activities were consistently and accurately recovered. However, the unknown Balance segment proved more difficult. A failure mode—in which the subject’s legs switch places—was regularly recovered at around frame 950. We believe this is caused by strong shadows cast onto the floor by the subject’s lower legs and (incorrectly) included in the observation foreground mask. V. Neither increases in \( T_0 \) nor doubling of the unknown particle quota to \( B_{\text{max}} = 500 \) enabled us to consistently recover the correct pose and we moved to using three cameras (C1–C3) to obtain robust results. Fig. 15 shows the average tracking error across the five runs at each frame. Just as for the known subject in

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
& \text{Walk} & \text{Jog} & \text{Balance} \\
\hline
3D absolute errors & 127 \pm 32 & 199 \pm 26 & 194 \pm 12 \\
\hline
\text{Cost evaluations} & 1250 & 1250 & 1250 \\
\hline
\end{array}
\]

Fig. 14. MAM-APF tracking results for S2’s HumanEva-II Combo sequence. (top left) 3D absolute error results averaged over five separate runs and colour coded by mode activity classification result. (top right) Average number of cost function evaluations per frame and Matlab processing time. Note the increase during the unknown Balance activity as more particles are used in order to recover poses from the full (high-dimensional) pose space. The subject’s posture as they prepare to balance on one foot around frame 750 is indeed well described by a Walking pose.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
& \text{Walk} & \text{Jog} & \text{Balance} \\
\hline
3D absolute errors & 103 \pm 24 & 173 \pm 14 & 203 \pm 28 \\
\hline
\text{Cost evaluations} & 1250 & 1250 & 1250 \\
\hline
\end{array}
\]

Fig. 15. MAM-APF tracking results for S4’s HumanEva-II Combo sequence. (top left) 3D absolute error results averaged over five separate runs and colour coded by mode activity classification result. Frames 208–335 are ignored as accurate ground truth is not available. (top right) Average number of cost function evaluations per frame and Matlab processing time in seconds. The subject does indeed stop balancing and start walking out of the capture area at around frame 1200. See Section 9.5 and Fig. 16 for further work on activity classification.
Section 8.2, MAM-APF consistently outperforms the standard APF baseline and uses only half as many particles during the known activities. The final Walking segment is a correct known activity classification, as the subject leaves their balance pose (around frame 1200) and starts walking out of the capture area. Images showing the expected tracking pose superimposed on the image observations of HumanEva-II camera C1 are shown in Fig. 18.

9. Discussion and conclusions

We have introduced an approach that gives equal consideration to the predictions of multiple activity models at each frame. The difficulty of the associated estimation tasks is quite different and this has allowed the recovery of known and unknown activities using a variable number of particles. Here we give further discussion of some specific aspects of the work presented.

9.1. Tracking performance

For known activity tracking, HMM-APF using the time reversed transition matrix \( \hat{A} \) is accurate, robust and relatively insensitive to the magnitude of particle dispersion, governed by \( T_0 \) (see Fig. 13). Raising \( T_0 \) does allow us to ensure optimal performance from the latent APF baseline. This occurs around \( T_0 = 4 \), but the average error and standard deviation of HMM-APF remains lower. This result highlights the need for a good activity-specific dynamical model in addition to an activity-specific latent pose space. We recommend the use of \( T_0 > 1 \) for HMM-APF. This is because spreading particles further through the HMMs (both forwards and backwards in time) does not degrade tracking performance, but is likely to increase the chances of recovering from serious errors (e.g. 180° rotations) if they were to occur.

MAM-APF is able to reliably recover known activities from the HumanEva-II Combo sequences with fewer than four cameras and only a small number of particles. This is in contrast to standard APF, see the quantitative comparisons in Figs. 14 and 15 and the investigation by Sigal et al. [8]. MAM-APF is also able to increase particle numbers to recover the Balance phases with its unknown activity model. Estimating freeform motion in the high-dimensional full pose space with a generic dynamical model is inherently more challenging, and the average 3D absolute error rises by 30–50 mm. In general, however, a good track is maintained throughout the sequences, see Figs. 17 and 18.

The recovery of a failure mode when using two cameras to track S4's Balance phase illustrates a potential danger of the annealing methodology where image evidence is ambiguous: if an incorrect mode is recovered, tracking may never be regained. However, particle filtering has been found to perform significantly worse than standard APF on the Combo sequences [8], despite its capacity to approximate a multi-modal posterior over time. Furthermore, we believe that robust two-camera tracking of S4’s Balance phase is likely to be possible if some consideration is given to the effects of shadows cast by the lower legs. The addition of feet to the body model may be helpful e.g. [8], or more sophisticated background subtraction methods could be adopted [42].

9.2. Classification

The MAM-APF approach naturally lends itself to sequence classification based on the activity model membership of particles. Figs. 14 and 15 show the algorithm is able to correctly classify Combo frames into their particular activity classes with reasonably few exceptions. The dashed vertical lines in these figures represent the ground truth activity segmentations defined by Sigal et al. [8], and used in the calculation of the error tables. Misclassifications are generally due to the unknown activity model recovering a known activity pose. Sigal et al. [8] note that S4's Jog phase displays a greater variation in performance style. This may explain why we found a slightly higher number of S4's Jog poses were recovered by the unknown activity model than for S2. We experienced no problems with false known activity transitions, e.g. [13].

Rather than clear and instantaneous changes between activities, the Combo sequences feature a number of slow activity transitions (relative to sampling rate) where intermediate poses do not feature in the HumanEva-I Training dataset (in which activities are segmented). S2's transition from Walk to Jog takes place over a period of approximately 1 s, starting with an abrupt rise in the forward swing of the left forearm that appears to increase vertical displacement (frames 380–400) before the subject eventually settles into a jog by around frame 440. In the absence of training data, it is the transition activity model that facilitates tracking, and permits the recovery of intermediate poses from the space in between the two activity manifolds (e.g. Fig. 11). During this period however, the mode classification result of MAM-APF remains as Walking up until the jog gait is fully established at around frame 440. More accurate identification and recovery of activity transitions themselves is a potentially interesting future research topic.

9.3. Computational cost

The computational cost of generic particle filtering is proportional to the number of particles used. For APF it is proportional to the number of particles used across all annealing layers. Total runtime is dominated by the evaluation of the cost function for each particle. As the cost function must be evaluated for each observation, computation time is also proportional to the number of cameras. For the standard APF baseline computation times are constant at around 25 and 40 s per frame for two and three cameras, respectively.

This work has addressed the high and fixed computational cost of particle-based inference by varying the number of particles depending on their activity class membership. For Combo sequences, the number of cost function evaluations remains low throughout the known activities of Walk and Jog as poses are recovered from the latent pose space. The number of evaluations then rises as the full pose space is explored to recover the unknown Balance activity, see tables in Figs. 14 and 15. Computation times fall by around 50% to 15 and 20 s per frame when tracking known activity from two and three cameras, respectively.

9.4. Many known activities

In this work we have adopted a single joint-activity pose space to recover known activity transitions, and a joint-subject pose...
space to generalise to unknown subjects. The HumanEva-II data set has permitted quantitative investigation, but it would be interesting to extend the approach to support larger numbers of known activities in future. Where more activities are used to create a joint pose space, a particle-based approach is well placed to explore activity manifolds that lie in close proximity. Given a junction in latent space—due to two or more activities sharing a similar component pose—noisy dispersion means that particles naturally divide between competing HMM states for propagation. For example, in Section 8.3 the proximity of the three subjects’ latent data (see bottom of Fig. 9(b)) leads particles to flow between HMMs during tracking. Where HMMs represent substantially different but partially overlapping activities in a joint space, the correct HMM will “win out” at such time as its pose hypotheses begin to diverge from those of the others.
The dimensionality of a joint-activity space must grow with the number of activities modelled. Alternatively a set of individual low-dimensional latent pose spaces could be used, one for each known activity. Here the use of transition lines is no longer possible. This is the approach taken in earlier work [13], but forcing an equal number of particles into each space means that computational cost increases with the number of activities. Here we have found it is not necessary to saturate every HMM with particles, only to select a single parent HMM state for each particle in the quota.

An infinity of points in the high dimensional pose space describe unrelated unknown poses but in fact project to latent coordinates close to known activity training data. This is because pose variations are concentrated in the orthogonal complement to the PCA subspace [43]. If multiple latent pose spaces are used, it may be insufficient simply to find the most likely parent state in order to determine which known activity model contains the “closest” pose. A low cost solution would be to reconstruct an unknown particle’s pose from its latent coordinate in each space, and select the activity that gives the lowest projection-reconstruction error.

9.5. Activity class transitions

The calculation of projection-reconstruction (P-R) errors also has potential applications in the approach presented here. It is possible for the unknown activity model to do the work of the known activity model by recovering known poses from the full space, see the occasional unknown (blue) frames during the Walk and Jog phases in Figs. 14 and 15. P-R errors can be used to identify such “overlap” between activity models. We are able to automatically correct misclassifications by requiring that unknown poses exceed a lower bound on P-R error given the latent pose space, see Fig. 16. Alternatively, P-R error thresholding could be used as a prior on particle dispersion by the unknown activity model, resampling until all full pose space configurations are “novel” given the latent pose space.

Where the projection-reconstruction error is consistently high given the latent pose space, it is natural to ask whether the projection is appropriate at all. At all frames we place an equal prior on all activity models, anticipating the commencement of any class of activity with equal probability. This can be interpreted as a “flat” Markovian activity model transition matrix, e.g. see those used for dynamical model transitions by Isard and Blake [28]. While it is prudent to continually cater for the possibility that known activity will start to transform into unknown activity, the reverse does not always hold. If the unknown activity model (correctly) permits particles to move through the full pose space until they are “far” from all known activity poses, it may no longer be appropriate to force equal particle quotas into the known activity model.

The projection-reconstruction error could be used to dynamically adjust the prior on activity model transitions. In practice this would mean making adjustments to the probability of unknown-to-known activity class transitions based on how accurately the latent pose space is able to reconstruct the last expected pose. The potential computational saving is relatively modest—$2B_{\text{run}}$ unnecessary cost function evaluations at the first layer during unknown activity (around 5% of computation time per frame)—but the practice may also help to guard against false transitions.

9.6. Conclusions

In generative tracking, the use of high-dimensional activity models has previously allowed the recovery of freeform human motions without limitations on activity class. Drawbacks have included the need for sufficiently rich observations from multiple cameras, and a high and fixed computational cost during tracking. As an alternative, many approaches have adopted a low-dimensional activity model to recover certain classes of activity from fewer cameras and at reduced computational cost. The drawback being that training data must be available for every activity one wishes to track. To address these limitations we have introduced a generative tracking approach that gives equal consideration to the predictions of both low- and high-dimensional activity models at each frame. The associated estimation tasks are quite different in terms of difficulty, and we assign to them differently sized particle quotas to reflect this. The resulting MAM-APF algorithm is an attempt to combine the best aspects of both approaches: faster recovery of known activity with few particles where possible, but the flexibility to work for longer with more particles to recover unknown activities where necessary.

Acknowledgements

This work was supported by a Dalaton Research Institute research studentship and EPSRC Grant EP/D054818/1. We would like to thank the authors of [19] for making their APF code freely available, and the creators of the HumanEva datasets for sharing their data and running the online submission system.

References


About the Author—JOHN DARBY received a B.Sc in Computational Physics from the University of Edinburgh and an M.Sc in Mobile and Distributed Computer Networks from Leeds Metropolitan University. He is currently a PhD candidate at Manchester Metropolitan University, working on 3D human motion synthesis and tracking.

About the Author—BAIHUA LI received B.S. and M.S. degrees in Electronic Engineering from Tianjin University and PhD degree in Computer Science from University of Wales, Aberystwyth. She is a Senior Lecturer at Manchester Metropolitan University. Her research interests include computer vision, human motion analysis, 3D modelling and animation.

About the Author—NICHOLAS COSTEN received a B.A in Experimental Psychology from the University of Oxford and PhD degree in Mathematics and Psychology from the University of Aberdeen. He has undertaken research at the Advanced Telecommunications Research Laboratory, Kyoto and at the Division of Imaging Science and Biomedical Engineering, University of Manchester. He is a Senior Lecturer at Manchester Metropolitan University where his interests include face recognition and human motion analysis.