Editorial Manager(tm) for International Journal of Computer Vision
Manuscript Draft

Manuscript Number: VISI762R2

Title: A Study on Smoothing for Particle-Filtered 3D Human Body Tracking

Article Type: Special Issue: EHuM

Keywords: articulated human body tracking; particle filtering; smoothing

Corresponding Author: Dr Patrick Peursum, Ph.D.

Corresponding Author's Institution: Curtin University of Technology

First Author: Patrick Peursum, PhD

Order of Authors: Patrick Peursum, PhD; Svetha Venkatesh, PhD; Geoff West, PhD

Manuscript Region of Origin: AUSTRALIA

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

A Study on Smoothing for Particle-Filtered 3D Human Body Tracking

Patrick Peursum, Svetha Venkatesh and Geoff West
{P.Peursum, S.Venkatesh, G.West}@curtin.edu.au

the date of receipt and acceptance should be inserted later

Abstract Stochastic models have become the dominant means of approaching the problem of articulated 3D human body tracking, where approximate inference is employed to tractably estimate the high-dimensional (∼30D) posture space. Of these approximate inference techniques, particle filtering is the most commonly used approach. However filtering only takes into account past observations – almost no body tracking research employs smoothing to improve the filtered inference estimate, despite the fact that smoothing considers both past and future evidence and so should be more accurate. In an effort to objectively determine the worth of existing smoothing algorithms when applied to human body tracking, this paper investigates three approximate smoothed-inference techniques: particle-filtered backwards smoothing, variational approximation and Gibbs sampling. Results are quantitatively evaluated on both the HUMANEVA dataset as well as a scene containing occluding clutter. Surprisingly, it is found that existing smoothing techniques are unable to provide much improvement on the filtered estimate, and possible reasons as to why are explored and discussed.

1 Introduction

In time-series data with noisy observations, filtering is the process of estimating (or tracking) the true state at time t given all the observations $\{y_1, \ldots, y_t\}$ that lead up to t, and smoothing is the process of using all future observations $\{y_{t+1}, \ldots, y_T\}$ to correct the filtering estimate in light of the future evidence. Consequently, smoothing should provide a better estimate than filtering since it takes all available evidence into account. Hence it is common practice to use smoothed estimates in many fields such as signal processing and speech recognition. In contrast, research into articulated human body tracking is dominated by filtering. In generative (top-down) tracking where the observation is viewed as 'caused' by the true state, the most prevalent approach is particle filtering (Moeslund *et al.* 2006) which approximates the state with a set of weighted Monte Carlo samples called particles (Doucet *et al.* 2000). However, research employing smoothed inference for body tracking is almost non-existent despite the existence of several smoothing algorithms for particle filters that have been shown to benefit other tracking fields (Doucet *et al.* 2002; Godsill *et al.* 2004; Klaas *et al.* 2006), as well as alternative efficient approximate smoothed inference techniques such as variational and Gibbs sampling (Ghahramani and Jordan 1997).

This paper investigates approximate smoothing techniques in order to ascertain their worth for 3D multi-view articulated human body tracking in both controlled and realistic environments, where the latter contains occluding objects such as tables and chairs. Such realistic scenes are rarely considered in human body tracking since occlusions produce observation 'errors' and thus often cause filtered tracking to fail for the duration of the occlusion. Our previous work (Peursum *et al.* 2007) showed that a strong motion model can minimise such failures, but this restricts tracking to modelled motions. In contrast, smoothing is applicable to any motion dynamics and has been reported to improve tracking estimates over filtering in other, lower-dimensional, tracking fields (Doucet *et al.* 2002; Godsill *et al.* 2004; Klaas *et al.* 2006). This paper investigates smoothing in both 'clean' and cluttered environments to establish the conditions where smoothing is and isn't beneficial for high-dimensional human body tracking. Focus is given to smoothing in generative models rather than discriminative (bottom-up) models since although generative approaches are usually slower, they generalise well to different people and naturally handle missing/occluded observations, properties that are important in realistic scenes. In brief, this paper:

Dept of Computing, Curtin University of Technology GPO Box U1987, Perth, Western Australia

– Examines the issues of applying existing smoothing algorithms to generative articulated tracking and proposes the use of a mixture approximation to overcome these issues whilst retaining modest computational costs (*i.e.* no greater than filtering).
– Quantitatively evaluates the performance of three popular smoothing algorithms based on three different filtering models and using two datasets (HUMANEVA-I/II and our own CLUTTER dataset) in multi-view environments.
– Finds that, contrary to expectations and results in lower-dimensional problems (Doucet *et al.* 2002; Godsill *et al.* 2004; Klaas *et al.* 2006), smoothing does not provide much benefit to high-dimensional articulated tracking. Follow-up experiments indicate that dimensionality is the cause of the poor smoothing performance.

The three smoothed inference techniques investigated include forwards-backwards smoothing (FBS), variational approximation and Gibbs sampling. FBS is a natural choice for particle-filtered inference but one which has rarely been employed in articulated tracking (to our knowledge, the only other example is Sminchisescu and Jepson (2004), who use a dynamic programming approach that is similar to FBS). Variational and Gibbs sampling have seen some use in body tracking but not as a means to smooth *across* time – for example, mean-field Monte Carlo proposed by Hua and Wu (2007) optimises the observation likelihood at each time t, but still uses a particle filter to propagate the posture across time. Moreover, in a generative model with a complex image-based observation likelihood that is costly to evaluate, it is computationally impractical to directly implement variational or Gibbs sampling for smoothed inference. To overcome this, we approximate the observation function $P(y_t|x_t)$ with a more manageable mixture of Gaussians based on a 'pre-processing' particle filter. This differs from Sminchisescu and Jepson (2004), who approximated a handful of the maxima in the particle-filtered *posterior* $P(x_{1:T}|y_{1:T})$ with a Gaussian mixture using gradient ascent optimisations involving costly evaluations of the true $P(y_t|x_t)$.

Figure 1 gives an overview of this paper, depicting the algorithms investigated and their relationships. Tracking is evaluated on both the HUMANEVA-I and -II datasets (Sigal and Black 2006) and more difficult videos of meandering walking sequences in a realistic indoors scene containing occluding tables and chairs (henceforth referred to as the CLUTTER dataset). The latter videos are difficult for filtering-only approaches to handle due to the sub-optimal observations caused by frequent occlusions. This paper focuses on walking since it will lead to repeated occlusions in the CLUTTER dataset – although HUMANEVA contains other motions (*e.g.* throwing, boxing), this paper seeks to contrast the results of the two datasets (arising from the differences in the observing conditions) and so requires similar motions in both. A loose-fitting body model is employed for both datasets to minimise any reliance on *a priori* knowledge of the tracked person's shape and ensure the tracker generalises well to different people and clothing. To establish the effect of motion models on smoothing, three models are employed for the pre-processing filter, two using a 'generic' motion model and the third using a learned (motion-specific) motion model – in this case, of walking. The two generic models differ in that one is filtered with the standard particle filter (Doucet *et al.* 2000) and other with the annealed particle filter (Deutscher and Reid 2005). For the third, a motion-specific model of walking is learned using the factored-state hierarchical hidden Markov model (FS-HHMM) of Peursum *et al.* (2007). The three smoothed-inference algorithms (FBS, variational, Gibbs) are then executed based on these three pre-processing filters.

The results of each technique are evaluated quantitatively and compared with one another as well as with filtered inference. Evaluation is based on the ground-truth position of critical points (head, elbows, hands, knees, feet, etc). Although the HUMANEVA dataset provides the ground truth of these points via motion capture markers, most video sequences typically have no associated motion capture data, including our CLUTTER dataset. Ground-truthing posture in such videos by manually defining 'virtual markers' (Bălan *et al.* 2005) is a labour-intensive and time-consuming task. To minimise the tedium, a small Matlab GUI utility was developed for hand-labelling virtual markers from video, accelerating the task so that each marker takes only 5–10 minutes to label in 500 frames (Peursum 2008). The source code for this utility is available for download.[1]

This paper is organised as follows. Section 2 summarises recent work in the field of body tracking to place this paper in context. Sections 3 and 4 describe the filtering and smoothing algorithms evaluated in this paper, followed by a description of the experimental setup in Section 5 and a discussion of the results and follow-up experiments in Sections 6 and 7. Finally, Section 8 presents the conclusions.

2 Background and Related Work

Articulated human body tracking has received significant research attention over the past few years – a survey of work in the field up to early 2006 is provided by Moeslund *et al.* (2006). This paper is concerned with fully-articulated 3D body tracking, where articulation covers all of the major body parts including the feet to produce a body model totalling 28 degrees of freedom. Most contemporary approaches to such 3D body tracking are in terms of a stochastic time-series framework where a human body model is explicitly defined as a kinematic tree of body parts whose joint angles

[1] Download the Matlab source code from http://impca.cs.curtin.edu.au/downloads/software.php

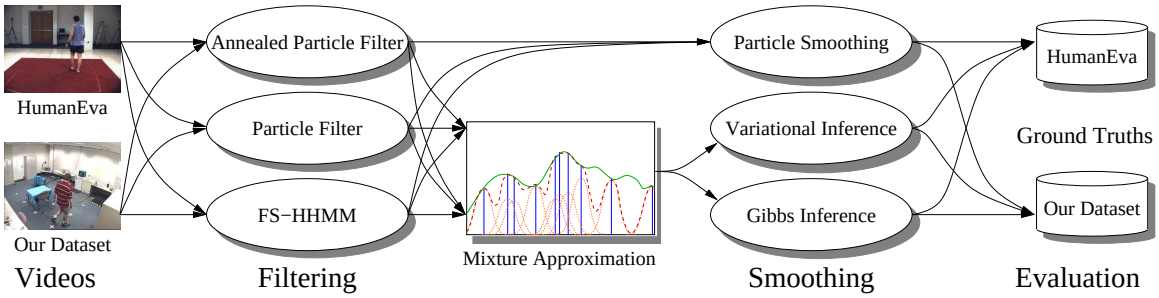


Fig. 1 Overview of the algorithms and data sets investigated in this paper. Both datasets are filtered with three filtering algorithms; each filtered result is passed on to the three smoothing algorithms to produce nine smoothing results, which are evaluated along with the three filtering results.

evolve over time according to some motion dynamics system. The goal is to recover an estimate of the posture distribution via inference on the time-series probability model. The high dimensionality of the posture space means that approximate inference is necessary, and this usually takes the form of a sampling approach. Strategies for sampling and evaluating postures can be grouped into two broad categories: bottom-up (discriminative) models and top-down (generative) models.

Discriminative Approaches Discriminative models are typically more efficient than generative models since they employ 'limb detectors' to search for candidate body parts in the observed images and use these candidates in conjunction with the previous posture and kinematic constraints to infer the next posture. Thus sampling is strongly guided by the limb detectors towards good matches. Many discriminative approaches also mix in generative aspects, using the limb detector to define where a generative tracker should sample from. Lee *et al.* (2002), Lee and Nevatia (2005) and Gupta *et al.* (2007) detect the face and torso before determining in a top-down manner the rest of the body's structure, whereas Sigal *et al.* (2004) detects all limbs and draw samples in the neighbourhood of these detections for a generative tracker. In addition, Sigal *et al.* (2004) models the dependencies among limbs so that the position of one limb can provide useful evidence for the position of another. For example, a person's arms will usually swing in synchronisation with their legs during walking. Thus the position of the legs can imply the likely position of the arms and vice-versa.

Pure discriminative approaches have also been taken. Elgammal and Lee (2004) learn a non-linear mapping between observed silhouettes and their equivalent 3D body postures. They first learn a mapping from silhouettes to a low-dimensional manifold that represents the 'path' that a given activity (*e.g.* walking) takes through the high-dimensional space of human posture. This embedded manifold is then mapped to 3D body postures. An observed silhouette can then be efficiently mapped to its body posture via the manifold, resulting in fast pose estimation. However, each learned manifold is specific to a particular viewpoint of an activity, so multiple manifolds must be learned to handle different viewpoints. A different approach is taken by Mündermann *et al.* (2007) and Cheng and Trivedi (2007), who align a 3D visual hull body model to an observed visual hull constructed from silhouettes seen in multiple viewpoints in order to achieve viewpoint independence. Other researchers (Taycher *et al.* 2006; Sminchisescu *et al.* 2006) utilise statistical time-series models such as conditional random fields (CRFs) and maximum entropy Markov models (MEMMs) to perform tracking. However, failures in detecting the true limb or the full silhouette and the need to train limb detectors specific to the person being tracked means that discriminative approaches have difficulty with observation 'errors' (*e.g.* occlusion by scene objects) and do not generalise well to different people without retraining (Kanaujia *et al.* 2007).

Generative Approaches In contrast to discriminative methods, generative approaches evaluate 'guesses' of the state against the true observation in a predict-then-evaluate cycle, a method that is almost always implemented with a particle filter (Doucet *et al.* 2000). Such an approach can generalise well to different people and is better able to handle poor observations than discriminative approaches. On the other hand, generative models require evaluating an observation likelihood which in many cases is an expensive projection of each 3D posture 'guess' onto the 2D image and evaluating the difference in a pixel-wise manner (Deutscher and Reid 2005; Peursum *et al.* 2007). An alternative approach is to calculate a 3D representation of the observation, typically a visual hull (Mikić *et al.* 2001; Caillette *et al.* 2005). This can facilitate a faster observation evaluation but is offset by the visual hull's need for accurate full-body silhouettes from multiple views, which can be sensitive to errors in any one view.

Strong motion models are becoming an increasingly common method of focusing sampled predictions onto good areas of the posture space so as to reduce the number of particles needed to achieve accurate tracking. Such models also learn the conditional dependencies between limbs for a given

motion, for similar reasons to the discriminative approach of Sigal *et al.* (2004) described earlier. Many methods involve learning a model of human motion dynamics in terms of transitions of the ∼30D state. Caillette *et al.* (2005) learn the transitions of a variable-length Markov model where each state defines a Gaussian subset of possible postures. Similarly, Peursum *et al.* (2007) employed a two-level factored-state hierarchical HMM where the upper level defines the 'phase' (sub-sequence) of motion and the lower level defines the motion dynamics of the posture for each phase. Husz and Wallace (2007) proposed a hierarchical partitioned particle filter, a variant on the annealed particle filter of Deutscher and Reid (2005), in conjunction with 'action primitives', which are motion sub-sequences similar to the phases of Peursum *et al.* (2007). These action primitives are clustered with EM and PCA and new sub-sequences are compared against training primitives to determine which action is the best match in order to draw samples for the next posture. Along slightly different lines, other researchers incorporate the physics of walking (foot collisions with the ground, stride cycle length, etc) (Brubaker *et al.* 2006, 2007; Vondrak *et al.* 2008). This is used to strongly guide sampling as well as achieve a more aesthetically believable tracking result. Another way to incorporate *a priori* information on motion is through dimensionality-reduction methods, which attempt to find a low-dimensional manifold in the circa-30D body-motion space that represents most of the information of a given action. One of the earliest examples is work by Sidenbladh *et al.* (2000), who learned a multi-variate PCA model of walking and showed that this could significantly outperform linear-Gaussian models. Urtasun *et al.* (2006) also uses PCA and later (Urtasun *et al.* 2005) a Gaussian process latent variable model (GPLVM) to find a mapping of walking and a golf swing onto a simpler manifold. Lee and Elgammal (2006) do a similar mapping onto a low-dimensional torus, then sample particles (representing silhouettes) from this torus and compare these samples against the observed silhouette via a similarity measure.

Smoothed Body Tracking Given that this paper is concerned with realistic scenes containing occluding clutter, we take the path of generative models with a projection-based observation likelihood. One of the few to consider smoothing for articulated tracking in a generative setting is Sminchisescu and Jepson (2004). They use a complicated mix of particle filtering, second-order gradient ascent and variational methods to estimate the posture. Their system proceeds by extracting the eight most-likely (in terms of maximum a-posterior) particle trajectories from an initial particle filter. These trajectories are then optimised via Hessian-based (second-order) gradient ascent over the entire distribution $P(x_{1:T}, y_{1:T})$ to produce a Gaussian mixture. This is then the input to a variational step that further refines the mixture. The final output is a Gaussian mixture that represents several modes of $P(x_{1:T}|y_{1:T})$. The authors demonstrate tracking in a monocular view, a difficult task given the lack of depth information. However, the resulting optimised trajectories differ noticeably from one another even in their 2D projections, and it is not clear how to determine which is the best trajectory since the gradient ascent has ensured that all trajectories have high image likelihood. In addition, the algorithm's running time is not reported, although the complexity of the algorithm and the need to optimise over a projection-based observation function $P(y_t|x_t)$ suggests it is computationally expensive. Finally, given that the ground-truth was not available for comparison, it is also uncertain as to what extent the system provides for more accurate tracking (as opposed to *smoother* tracking, which the authors demonstrate).

3 Filtered Articulated Tracking

This paper employs three particle-based filtering models whose outputs will later be smoothed in Section 4. The three vary in their motion dynamics models and particle algorithms in order to investigate the effect of such differences on smoothing. Two of the three (Simple-PF and Simple-APF) use generic motion models in that the next posture is assumed to be distributed according to Gaussian diffusion of the current posture's joint rotations. They differ in that one model uses the standard particle filter (Doucet *et al.* 2000) whilst the other uses the annealed particle filter of Deutscher and Reid (2005). The third filter (FSHHMM-PF) employs a motion-specific model that is learned from training data, with filtering via a standard particle filter. The motion model is built on a factored-state hierarchical hidden Markov model (FS-HHMM) to facilitate tractably modelling the non-linear dynamics of human motion. All three utilise the same body model and observation likelihood function.

3.1 Particle Filter with a Simple Model (Simple-PF)

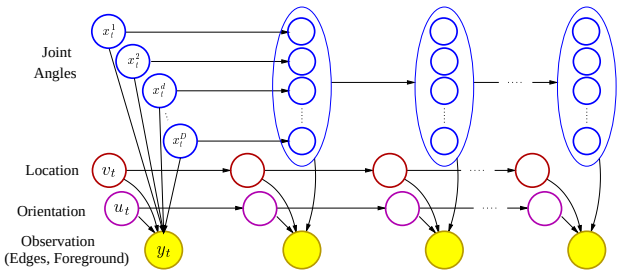


Fig. 2 Bayesian network of the model for generic (motion-agnostic) articulated tracking. The body pose $\{x_t, v_t, u_t\}$ is fully factored.

Overview Particle filtering, also known as sequential Monte Carlo sampling (Doucet *et al.* 2000) is a popular technique for approximate *filtered* inference in a generative model due to its algorithmic simplicity and ability to model non-linear, non-Gaussian dynamics systems. Indeed, several special cases of the technique have been independently developed and introduced under various names, including the bootstrap filter in signal processing (Gordon *et al.* 1993) and CONDENSATION (Isard and Blake 1998) in computer vision.

Given a true, though unobservable, state x_t, observations y_t ($t = \{1..T\}$), first-order Markov dynamics $P(x_t|x_{1:t-1}) = P(x_t|x_{t-1})$ and observation probability $P(y_t|x_{1:t}, y_{1:t-1}) = P(y_t|x_t)$, the posterior distribution $P(x_t|y_{1:t})$ is approximated with N weighted samples (called particles) $\{x_t^{(i)}, w_t^{(i)}\}$, $i = \{1..N\}$. Particles (i) are independently propagated forward in time by sampling from an arbitrary proposal distribution $Q(x_t|x_{t-1}^{(i)}, y_t)$ and updating the weights:

$$x_t^{(i)} \sim Q(x_t|x_{t-1}^{(i)}, y_t) \tag{1}$$

$$w^{*(i)}_t = w^{(i)}_{t-1} \frac{P(y_t|x_t^{(i)})P(x_t^{(i)}|x_{t-1}^{(i)})}{Q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)} \tag{2}$$

$$w_t^{(i)} = \frac{w^{*(i)}_t}{\sum_{j=1}^{N} w^{*(j)}_t} \tag{3}$$

where $\sim$ means 'sampled from'. The algorithm has time complexity $\mathcal{O}(NT)$, but the particles only approximate the *filtering* distribution since future observations have not been taken into account. The proposal distribution Q controls how efficient the particle filter is with its samples – a good Q will return samples in highly-weighted areas of the state space at $t+1$. Setting $Q = P(x_t^{(i)}|x_{t-1}^{(i)}, y_t)$ is the optimal choice, ensuring that samples are selected based on knowledge from both the previous state and the current observation. Selecting this optimal Q reduces Eq (2) to:

$$\begin{aligned} w^{*(i)}_t &= w^{(i)}_{t-1} \frac{P(y_t|x_t^{(i)})P(x_t^{(i)}|x_{t-1}^{(i)})}{P(x_t^{(i)}|x_{t-1}^{(i)}, y_t)} = w^{(i)}_{t-1} \frac{P(y_t, x_t^{(i)}|x_{t-1}^{(i)})}{P(x_t^{(i)}|x_{t-1}^{(i)}, y_t)} \\ &= w^{(i)}_{t-1} P(y_t|x_{t-1}^{(i)}) \end{aligned} \tag{4}$$

One issue with the optimal Q is that it is often difficult to directly evaluate $P(y_t|x_{t-1}^{(i)})$ and one must instead evaluate $\int P(y_t, x_t|x_{t-1}^{(i)})dx_t$. However, in the case of the articulated models of this paper, the integration over x_t is computationally intractable since x_t is a 24-dimensional variable. Moreover, it can be difficult to sample from the optimal Q, especially given the multiple modality engendered by the image-based observation. Thus in this paper Q is set to the transition probability $P(x_t^{(i)}|x_{t-1}^{(i)})$ for all models (Simple-PF, Simple-APF and FSHHMM-PF). Although this Q is sub-optimal since the observation y_t is not taken into account for sampling, it has the advantage of being easy to sample from and reduces Eq (2) to a simple evaluation:

$$\begin{aligned} w^{*(i)}_t &= w^{(i)}_{t-1} \frac{P(y_t|x_t^{(i)})P(x_t^{(i)}|x_{t-1}^{(i)})}{P(x_t^{(i)}|x_{t-1}^{(i)})} \\ &= w^{(i)}_{t-1} P(y_t|x_t^{(i)}) \end{aligned} \tag{5}$$

One issue for particle filters is that of *degeneracy*, where the weights of all but a few particles tend towards zero after a few transitions. This occurs since only a few particles will be consistently sampled from highly-weighted areas of the state space. Although the problem of degeneracy can be minimised by utilising the optimal Q, degeneracy cannot be completely avoided. Thus a common strategy is to regularly resample particles when the effective sample size (Doucet *et al.* 2000) drops below some threshold in order to multiply high-weight particles and discard low-weight particles. Resampling at every time t yields the Sequential Importance Resampler (SIR) algorithm.

Articulated Tracking with the Simple-PF For articulated tracking, this paper uses the state-space model of Figure 2, where $x_t^{\{d\}}$ is the rotation for joint angle ($d = \{1..D\}$), v_t is the tracked person's global position (pos_x, pos_y, pos_z), u_t is the person's orientation in the scene and y_t is the observed image. $P(x_t|x_{t-1})$ is modelled with generic motion dynamics where posture transitions are assumed to be Gaussian distributed about the previous posture x_{t-1} (*i.e.* Gaussian diffusion: $x_t = x_{t-1} + \varepsilon$, where ε is zero-mean Gaussian noise). Note that it is more usual in the general tracking literature to include velocity $\dot{x}_t$ into the state and implement a second-order (constant velocity) motion model so that predictions utilise the current velocity of the tracked object (*i.e.* $x_t = x_{t-1} + \dot{x}_{t-1} + \varepsilon_x$ and $\dot{x}_t = \dot{x}_{t-1} + \varepsilon_v$). Indeed, Sidenbladh *et al.* (2000) and Poon and Fleet (2002) utilised such a second-order model for human body tracking. However, later work by Bălan *et al.* (2005) showed that second-order models actually perform worse in human body tracking than first-order (diffusion) approaches due to the highly non-linear nature of human motion. Such issues could be overcome by particle-filtered inference on a non-linear, non-Gaussian model of human motion since the particle filtering framework is not restricted to linear Gaussian models. However, such a model (*e.g.* as implemented by this paper in Section 3.3) requires significantly more effort to construct than the Simple-PF. Hence this paper employs first-order Gaussian diffusion transition dynamics for the Simple-PF.

The Simple-PF also assumes that x_t fully factorises into its component degrees of freedom (*i.e.* the covariance of each joint's rotations is diagonal), with variances set based on the maximum change in rotation over one frame. This greatly simplifies the task of manually specifying rotation variances and allows the model to generically represent any

human motion, at the cost of a weak motion model and hence poor proposal distribution $Q(x_t|x_{t-1})$. To offset this weakness filtering is performed with 10,000 particles, a relatively large amount for a generative body tracker. More sophisticated generic approaches could have been used to make the model more efficient, such as dynamically adjusting the covariance (Sminchisescu and Triggs 2001). However, this paper deals with articulated tracking in the presence of occluding objects, and the simpler model makes fewer assumptions that may prove to be invalid during times of occlusions and other problematic observations.

3.2 Annealed Particle Filter w/Simple Model (Simple-APF)

Overview The annealed particle filter (APF), first proposed by Deutscher and Reid (2005), is a variant of the SIR particle filter where initial particles generated from a particle filter prediction step at time t are iteratively perturbed and resampled based on an annealing schedule. The annealing causes the system to gradually cluster particles into peaks of the observation likelihood by weighting the likelihood:

$$P_\ell(y_t|x_t) = P(y_t|x_t)^{\lambda_\ell} \quad 0 < \lambda_1 < \cdots < \lambda_L \tag{6}$$

where $P_\ell(\cdot)$ is the annealed likelihood at the ℓ-th annealing iteration ($\ell = \{1..L\}$). The monotonically increasing values of the annealing powers λ_ℓ cause $P_\ell(\cdot)$ to become more peaked as the schedule progresses, thereby placing increasing emphasis on particles in more-likely parts of the observation. At each iteration ℓ, the particles are evaluated with Eq (6), resampled to proliferate the best particles and then perturbed via Gaussian diffusion to search the neighbourhood around these best particles. The process is repeated until the annealing schedule is completed. In this way, the APF gradually focuses its search in the peaks of the observation likelihood at time t.

Articulated Tracking with the Simple-APF The Simple-APF uses the same generic motion model with independent (fully-factorised) joint rotations and Gaussian diffusion for posture transitions that the Simple-PF employs, (Figure 2). Similarly, 10,000 particles are used for APF inference but these are empirically split into 10 annealing layers of 1,000 particles each. In comparison with the Simple-PF, the Simple-APF will typically produce particle sets that are densely packed in the observation likelihood peaks and sparser elsewhere due to the iterative annealing.

3.3 Particle Filter with a Factored-State Hierarchical Hidden Markov Model (FSHHMM-PF)

Overview The FS-HHMM (Figure 3) is a two-level hierarchy (Peursum *et al.* 2007) that addresses the problem of

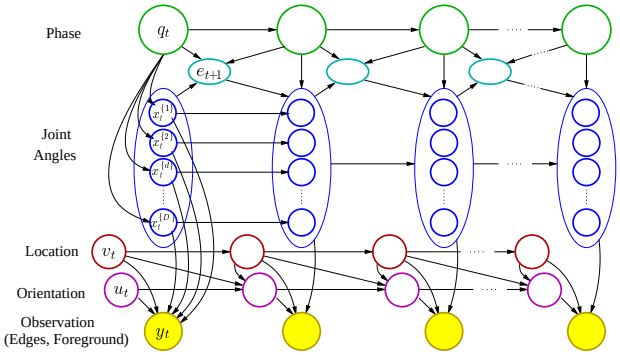


Fig. 3 Bayesian network of the FS-HHMM for learning-based articulated tracking. Source: Peursum *et al.* (2007), © 2007 IEEE.

compactly representing the non-linear dynamics of articulated human motion in a Bayesian setting. The model is parameterised as follows:

$$C_{mn} \triangleq P(q_{t,n}|q_{t-1,m}) \tag{7a}$$
$$A^{\{d\}}_{nij} \triangleq P(x^{\{d\}}_{t,j}|x^{\{d\}}_{t-1,\dot{\nu}}\, q_{t,n}, e^{\{d\}}_t{=}0) \tag{7b}$$
$$\Lambda^{\{d\}}_{nj} \triangleq P(x^{\{d\}}_{t,j}|q_{t,n}, e^{\{d\}}_t{=}1) \tag{7c}$$
$$\phi_m \triangleq P(q_{1,m}) \tag{7d}$$
$$\varphi^{\{d\}}_{mi} \triangleq P(x^{\{d\}}_{1,i}|q_{1,m}) \tag{7e}$$
$$\Omega^{\{d\}}_{nmi} \triangleq P(e^{\{d\}}_t|x^{\{d\}}_{t-1,i}, q_{t-1,m}, q_{t,n}) \tag{7f}$$
$$\Psi^{\{g\}} \triangleq P(v^{\{g\}}_t|v^{\{g\}}_{t-1}) \tag{7g}$$
$$\Upsilon_t \triangleq \omega^{u|u}_t P(u_t|u_{t-1}) + \omega^{u|v}_t P(u_t|v_{t-1}, v_t) \tag{7h}$$

where $\{x^{\{1:D\}}_t, v^{\{1:G\}}_t, u_t\}$ represents the posture, position and orientation of the person's body and q_t is the phase of the motion (described below). Omitted is the observation function $P(y_t|x^{\{1:D\}}_t, v^{\{1:G\}}_t, u_t)$, since it is a fixed heuristic that evaluates the posture $\{x^{\{1:D\}}_t, v^{\{1:G\}}_t, u_t\}$ against the observed image, as described in Section 3.4. For more details on the FS-HHMM see Peursum *et al.* (2007).

The FS-HHMM models a single human action (*e.g.* walking) by breaking it down into phases (sub-motions) that define a set of valid possible body configurations (postures) and their transitions. The discrete nature of the HHMM allows for learning arbitrary non-linear motion, an important factor since human motion is not well-modelled with linear dynamics (Bălan *et al.* 2005). The phase q_t facilitates factorising the body joint rotations – by assuming rotations are conditionally independent given the phase, a particular phase defines a transition regime for each rotation and collectively these regimes define the coordinated motion of the limbs for the sub-motion represented by the phase. Note that the FS-HHMM models body joint rotations with a discrete x_t rather than the continuous x_t of the Simple-PF and Simple-APF. As mentioned, this allows for learning arbitrary non-linear motion transition distributions (A_{nij} and Λ_{nj}), at the cost of some loss in accuracy due to discretisation.

Articulated Tracking with the FSHHMM-PF As with the Simple-PF and Simple-APF, the FSHHMM-PF utilises a particle filter for approximate inference,with a proposal distribution $Q \triangleq P(x_t|x_{t-1})$. This Q is learned from training data of the motion being represented (*e.g.* walking) and so will channel particle-filtered sampling down good areas of the predicted posture space (assuming that the person is performing the modelled motion). Hence only 1,000 particles are needed to provide reliable body tracking, far fewer than the generic Simple-PF and Simple-APF models. For this paper, an FS-HHMM is trained with a single walking sequence of four steps along a straight line. This training data is sufficient to model and track most walking motions (including turns and pivoting).

3.4 Body Model and Observation Function

All three filtering models employ the same body model and observation function. This paper focuses on human pose tracking with generative Bayesian models where the observation function is projection-based to improve robustness to observation errors such as occlusions by scene objects (Peursum *et al.* 2007). A loose-fitting body model is used to avoid the need to manually tune it to the specific physiques of the people being tracked.

Body Model This paper employs a 28-dimensional model of the human body (Figure 4), rooted at the pelvis and parameterised by 24 joint rotations and four global variables (x, y, z, orientation – body pitch is modelled at the pelvis so that it can be learned by the FSHHMM-PF) as well as a fixed scale. Scale applies to the entire body model since the relative length of each limb is fixed. Each body part is modelled with a cylinder whose sides are projected onto the 2D image and then joined with lines to produce the cardboard look for efficient projection (Sidenbladh *et al.* 2000). The model is fairly loose-fitting so that any tracker based on it should generalise well to different people. Broad limits on joint rotations are enforced to constrain postures to those that are feasible for most human motions, but these limits are not specific to any particular motion. No effort is made to prevent body part intersection in 3D space.

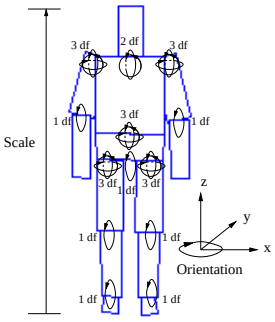


Fig. 4 28D 'cardboard' body model. Source: Peursum *et al.* (2007), © 2007 IEEE.

Observation Likelihood Function This uses an observation function $P(y_t|x_t)$ based on projecting x_t onto the image y_t and evaluating the difference between the two. Here, y_t is a tuple consisting of the edges and foreground images. Foreground is extracted using a mixture of Gaussians background subtraction (Stauffer and Grimson 2000) and edges are extracted with a thresholded Sobel detector, with the foreground used as a mask on the edges. A modified version of the cost function of Deutscher and Reid (2005) is employed:

$$D = Dist\Big(y_t, Proj(x_t)\Big) \tag{8}$$

which calculates the 'distance' between y_t and the foreground / edges projections of x_t (see Peursum (2006) for details). For the particle-filtered models in this paper (*i.e.* Simple-PF and FSHHMM-PF), the observation probability $P(y_t|x_t)$ is then calculated via the exponential distribution and D:

$$P(y_t|x_t) = \lambda e^{-\lambda D} \tag{9}$$

where $\lambda > 0$ and $D \geqslant 0$. The value of λ controls how sharply the distribution drops off with increasing values of D (*i.e.* increasing distance), hence a larger λ will more heavily penalise slightly incorrect particles. This is important since most particles will return similar values for D due to the high dimensionality of the state space. For example, if two particles only differ significantly in their elbow angles, 95% (27 of 28D) of the body model is still much the same, plus the forearm is not a large body part in the projection and may even be occluded in some views.

For the APF, the observation probability is defined as follows (according to Deutscher and Reid (2005)):

$$f(y_t|x_t) = \big(e^{-D}\big)^{\lambda} \tag{10}$$

where the APF's algorithm automatically sets λ, usually such that $\lambda \gg 1$ so that slightly incorrect particles are heavily penalised. Notice that, unlike Eq (9), this function is not a probability density since it does not integrate to 1.0 unless λ=1. However, note also that since particles in this paper are weighted by $P(y_t|x_t)$ (see Eq (5)) and then normalised, the proportionality constant multiplier λ in Eq (9) is redundant since it will be normalised away. With this insight in mind, it is possible to see that (9) and (10) are effectively equivalent:

$$\begin{aligned} P(y_t|x_t) &= \lambda e^{-\lambda D} \\ &= \lambda\big(e^{-D}\big)^{\lambda} \\ &\propto \big(e^{-D}\big)^{\lambda} \\ &\propto f(y_t|x_t) \end{aligned} \tag{11}$$

In other words, applying the annealing factor λ of the APF as a power is merely another way of writing the general form of the exponential distribution (disregarding the proportionality constant, which is redundant in the particle filter due to normalisation). Thus a suitable $\lambda \gg 1$ could be chosen for the Simple-PF to define its observation probability and similar performance to the Simple-APF should result – this is precisely what is found in Section 6.1. Due to this equivalence,

this paper will henceforth refer to λ as the annealing factor (rather than a more cumbersome term like 'exponential coefficient').

However, there is still an important difference between the particle filter and the APF in that the APF selects the value of λ *dynamically* for each annealing layer. This means that the APF is continually altering its definition of $P(y_t|x_t)$ between annealing layers and time instants – one can argue that this means the Simple-APF is not strictly a Bayesian model. Indeed, the authors themselves make note of this in (Deutscher *et al.* 2000). The Simple-PF and FSHHMM-PF do not suffer from this problem since λ in Eq (9) is fixed across all time instants, with values chosen empirically as λ=8 and λ=10 respectively (the Simple-PF's weak motion model necessitates more aggressive 'annealing').

Although the observation probability is heuristic in that it is constructed around the heuristic function D, most generative human body trackers are forced to utilise such heuristics due to the difficulty of constructing and learning an algebraic function that will produce a usable measure of similarity between the state and observation in arbitrary scenes. The consequence of using heuristics is that it creates an observation probability that cannot be trained and so will return erroneous results during unforeseen circumstances such as partial occlusions. Hence what this paper describes as 'observation errors' are in fact failures of D to properly account for the observation.

4 Smoothed Articulated Tracking

4.1 Issues of Existing Smoothing Techniques

As has been discussed, smoothed inference is rare in articulated tracking. Part of the reason for this is that the high dimensionality of the posture space means that approximate filtering is already a computationally expensive task, and smoothing only adds to this cost. However, over the past decade there has been increasing use in tracking and signal processing of efficient approximate smoothed-inference techniques such as variational approximations, Markov Chain Monte Carlo (MCMC) methods and particle smoothing.

A variational approach to smoothed inference has great potential for articulated tracking since it has been shown to scale well to high dimensionality (Ghahramani and Jordan 1997). MCMC methods such as Gibbs sampling (Andrieu *et al.* 2003) are also worthwhile exploring given their flexibility and typically polynomial (though difficult-to-measure) convergence. Finally, the forward-filter backward-smoother (Doucet *et al.* 2000) or two-filter smoother (Klaas *et al.* 2006) for particle filters would be obvious smoothing choices for existing particle-filtered methods. However, all of these approaches face obstacles when applied to articulated tracking in a generative model:

- *Variational* methods require the ability to take expectations and differentials of the joint probability parameters. However, the observation likelihood function in a generative tracker is often implemented as a complex heuristic function. Such functions are difficult to express algebraically (precluding analytical solutions) and are computationally expensive to evaluate (making numerical methods such as gradient descent and MCMC integration impractical).
- *MCMC / Gibbs sampling* also face difficulties with the high computational cost of the heuristic observation function. Efforts to obtain a faster observation function (*e.g.* Caillette *et al.* (2005)) rely on extracting a 3D visual hull for the observation, but this requires multiple views and is sensitive to observation errors (*e.g.* occlusions by scene objects), thus robustness will suffer accordingly.
- *Particle smoothing* algorithms do not require evaluations of the observation function but are limited to adjusting particle weights and so will not explore new parts of the posture space during smoothing. In addition, these algorithms have $\mathcal{O}(N^2T)$ complexity where N is the number of particles, which may be computationally impractical since N is usually quite large.

Most of the issues centre around the computational cost of executing the various smoothing algorithms. This paper proposes to facilitate efficient execution of variational and Gibbs methods by approximating the computationally-costly observation function with a mixture of Gaussians derived from a pre-processing particle filter. Particle smoothing is also implemented and shown to have a reasonable computational cost with respect to filtering for $N \leqslant$ 10,000 due to the high overheads that the projection-based observation likelihood adds to filtering. Note that this paper considers a smoothing algorithm to be "computationally feasible" if its runtime is comparable to, or less than, that of the preprocessing particle filter run on the same sequence. Complexity analysis (O-notation) is not suitable here since it does not indicate constant overheads and there is no way of comparing algorithms whose complexity terms differ.

4.2 Particle Smoothing

There are several methods that have been proposed to provide smoothed inference from a forward pass by a particle filter: the forwards-backwards smoother (FBS; Doucet *et al.* (2000)), smoothed distribution sampling, (SS; Doucet *et al.* (2002); Godsill *et al.* (2004)), maximum a-posteriori smoother (MAP; Doucet *et al.* (2002); Klaas *et al.* (2006)) and two-filter smoother (TFS; Klaas *et al.* (2006)). While their details vary, all of these smoothing algorithms are essentially methods to re-weight the particles of the initial filtering pass to take into account the future data. The parti-

cles themselves are not adjusted towards better areas of the state space. Moreover, all are $\mathcal{O}(N^2T)$ complexity, although Klaas *et al.* (2006) describes an approximate technique using KD-trees that can reduce this to $\mathcal{O}\big((N\log N)T\big)$.

The FBS calculates new weights for the particles at each time t by considering the level of 'support' that each particle has in the future, where support is the mass of particles at time t+1 that a particle $x_t^{(i)}$ could (hypothetically) transition to, weighted by the probability of those transitions. This iterates from T–1 . . . 1, carrying the smoothing backwards until all weights are smoothed. The SS approach is substantively similar to FBS, differing mainly in that the FBS re-weights the particles to estimate the smoothed distribution whereas SS *samples* particles trajectories from this smoothed distribution. Thus SS can be loosely viewed as a resampled version of FBS, consequently losing some of the smoothed distribution's information. The MAP smoother also resamples, but differs from SS in that it computes a Viterbi-like state path through the particle trellis and samples only the *single most likely particle trajectory* (in a MAP sense), and so discards even more of the distribution than SS.

In contrast to the other methods, the TFS approach involves a second, independent, particle filter that is run in reverse (from T to 1). The smoothed particle weights are calculated based on the mutual support between the two filters' particle sets, much like the FBS weight update. Unlike the other particle smoothing methods, the reverse run allows the TFS to explore parts of the state space not represented by the forward filter. However, the trajectories of the particles in the two filters must overlap somewhat in order for there to be reasonable support between particles in the two filters. Given that the reverse filter evolves independently of the forward filter, this overlap can be difficult to guarantee in high-dimensional state spaces. In 28D human posture tracking it is possible that the reverse filter explores a local maxima of the state space that is entirely isolated from the forward filter at a given time t. In such a case, the mutual support between the two particle sets may not be very meaningful. Finally, the reverse filter is itself a significant processing overhead for generative trackers where the observation function is slow to evaluate. Due to the reduced information of SS and MAP and the potential issues of TFS, this paper employs FBS.

4.3 Mixture Approximation of $P(y_t|x_t)$

4.3.1 Motivation and Overview

The fact that particle smoothers do not shift the position of the filtering particles given the future evidence from smoothing is their main drawback. Ideally, smoothing would explore new areas of the state space that both past and future evidence indicates is promising (the TFS reverse filter ignores the past and so is no better than the forward filter). This is particularly important for posture tracking in realistic scenes, where posture failures caused by (say) an occluding chair tend to 'stick' until the occlusion ceases or the error becomes large enough to force the tracker to correct itself (Peursum *et al.* 2007). For particle smoothers, the gap between the failure and the correction is a void that cannot be filled since no particles exist in this space. This motivates the search for other smoothed inference algorithms that are not so restrictive.

Unfortunately, as described in Section 4.1, existing approximate inference techniques cannot be applied directly to generative pose tracking, mostly due to the computational cost of the observation function $P(y_t|x_t)$. This paper resolves this by approximating $P(y_t|x_t)$ with a more manageable function $\hat{P}(y_t|x_t)$. Since the observation function employed in this paper is based on heuristic edge and foreground comparisons between the projected body model and the observation, approximating the function in general (*i.e.* for any observation-model pairing) is not an easy task. However, a discrete approximation of $P(y_t|x_t)$ *is* available from particle filtering – to evaluate a particle i, $P(y_t|x_t^{(i)})$ must be calculated. In effect, each particle can be thought of as sampling from $P(y_t|x_t)$ with a weight equal to the function's probability at $x_t^{(i)}$. $P(y_t|x_t)$ could then be approximated with a Gaussian mixture $\hat{P}(y_t|x_t^{(i)})$ that is learned from these weighted samples:

$$P(y_t|x_t) \approx \hat{P}(y_t|x_t) = \sum_{k=1}^{K} \eta_k \, \mathcal{N}(x_t|\mu_k, \Sigma_k) \tag{12}$$

where $\mathcal{N}$ is the Gaussian distribution and η_k, μ_k and Σ_k are the weight, mean and covariance for component k. In this paper, the covariance is held constant for all mixture components, hence the mixture is quite similar to kernel density estimation (KDE, also known as Parzen window density estimation). The main difference from KDE is that the mixture adjusts the weights for each component via learning. This is crucial since otherwise the approximation will not faithfully reflect the true observation probability distribution $P(y_t|x_t)$. A lesser difference is that mixture learning also shifts the component means about, although in practice the shifts are small and if one fixes the means (as in KDE) a very similar approximation is produced.

To illustrate the need for adjusting the component/kernel weights, consider the case of kernel density estimation with Gaussian kernels, where each particle is the kernel for a component (*i.e.* $\mu_k = x_t^{(i)}$, $K = N$). If each kernel is weighted by the observation probability of the particle that generated it, the probability of any given $x_t = \widetilde{x}_t$ is then:

$$\hat{P}(y_t|\widetilde{x}_t) = \sum_{i=1}^{N} \rho^{(i)} \mathcal{N}(\widetilde{x}_t|x_t^{(i)}, \Sigma) \tag{13}$$

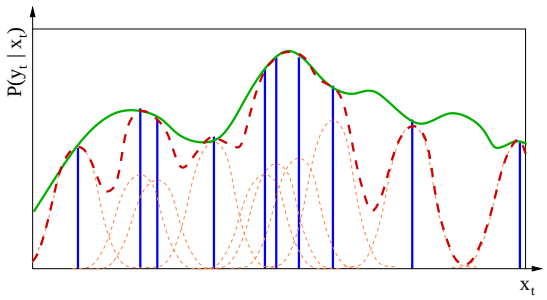


Fig. 5 Example mixture approximation generated from particle filter samples. Solid green curve is the true function $P(y_t|x_t)$; vertical blue bars indicate particles; dashed red curve is the mixture approximation $\hat{P}(y_t|x_t)$ produced from the Gaussian components (thin dashed orange).

where $\rho^{(i)} = P(y_t|x_t^{(i)})$. The problem with this definition is that the weights of closely-spaced components (*e.g.* components less than one standard deviation apart) will add up, causing $\hat{P}(y_t|x_t^{(i)}) \gg P(y_t|x_t^{(i)})$ at these closely-packed points and thus misrepresenting $P(y_t|x_t)$. Hence it is necessary to adjust the weights (and optionally shift the means) via Expectation-Maximisation (EM) in order to faithfully replicate the values of $P(y_t|x_t^{(i)})$ with the mixture.

In comparison to the discrete particles, the continuous nature of the Gaussian mixture should be a better representation of the similarly-continuous $P(y_t|x_t)$, as depicted in Figure 5. In particular, the mixture will partially 'fill in' the voids between the discrete particles, providing a reasonable representation of the observation function's behaviour in the vicinity of the particles. The idea is that the mixture will facilitate a range of inference techniques and provide them with some flexibility in exploring the state space whilst remaining acceptably accurate to the true $P(y_t|x_t)$. Subsequent inference is then effectively a smoothing of the original particle filter used to generate the mixture.

4.3.2 Learning the Gaussian Mixture

Learning a Gaussian mixture model (GMM) via EM is a well-known procedure. Given data x_i, $i = \{1..N\}$ and Gaussians with K components whose means, covariances and weights are $\{\mu_k, \Sigma_k, \eta_k\}$, $k = \{1..K\}$, the EM update equations for estimating a GMM are:

$$\tau_{i,k} = \frac{\eta_k \, \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \eta_j \, \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{14}$$

$$\hat{\eta}_k = \frac{1}{N} \sum_{i=1}^{N} \tau_{i,k} \tag{15}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \tau_{i,k} \, x_i}{\sum_{i=1}^{N} \tau_{i,k}} \tag{16}$$

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{N} \tau_{i,k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \tau_{i,k}} \tag{17}$$

where it is preferred that $K \ll N$ for the sake of efficiency. For the particle-filtered samples, each $x_i \triangleq x_t^{(i)}$ has an associated weight $\rho_i = P(y_t|x_t^{(i)})$. This can be incorporated into the EM equations by reinterpreting the weights as representing the relative number of samples at each location x_i. For example, the update for η_k becomes $\hat{\eta}'_k = \frac{1}{Z} \sum_{i=1}^{N} (\rho_i \, \tau_{i,k})$, as if there were ρ_i-worth of data points at x_i (where $Z = N \times \sum_j \rho_j$). Since $\tau_{i,k}$ is the weight for the assignment of x_i to Gaussian k and ρ_i is the weight of each x_i, one will find that $\tau_{i,k}$ always occurs together with ρ_i. Hence it is convenient to define $\tau'_{i,k} = \rho_i \, \tau_{i,k}$. The update equations for the GMM approximation $\hat{P}(y_t|x_t)$ at a given t thus become:

$$\tau'_{i,k} = \rho_i \, \tau_{i,k} \quad \triangleq \rho_i \, \frac{\eta'_k \, \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \eta'_j \, \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{18}$$

$$\hat{\eta}'_k = \frac{1}{N \sum_j \rho_j} \sum_{i=1}^{N} \tau'_{i,k} \tag{19}$$

$$\hat{\mu}'_k = \frac{\sum_{i=1}^{N} \tau'_{i,k} \, x_i}{\sum_{i=1}^{N} \tau'_{i,k}} \tag{20}$$

$$\hat{\Sigma}'_k = \frac{\sum_{i=1}^{N} \tau'_{i,k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \tau'_{i,k}} \tag{21}$$

The usual practical issues arise in the approximation, including deciding how many components K to use, the initial value of each component's parameters and whether to place any constraints on the EM updates. Due to the high dimensionality of $P(y_t|x_t)$ (28D), this paper is fairly conservative in its choices to avoid causing $\hat{P}(y_t|x_t)$ to become unrepresentative of the true $P(y_t|x_t)$. There is however a tradeoff between speed and faithfully representing $P(y_t|x_t)$. Hence Gaussian modes are chosen based on the distribution of samples (by weight), selecting all samples that are above-average (*i.e.* $\rho_i \geqslant \frac{1}{N} \sum_j \rho_j$) and using the corresponding value of $x_t^{(i)}$ as the initial mean for each mode. The assumption is that below-average particles are in uninteresting areas of $P(y_t|x_t)$ and so can be safely ignored as seeds for mixture components. K is roughly the same as the filter's effective sample size at each time t, which in this paper is between $0.05N$ and $0.2N$. As with particle resampling, the approach retains more mixture components during times of problematic observations such as when occlusions occur. These cause the effective sample size to increase (*i.e.* the distribution becomes more uniform) since all particles are somewhat in error according to $P(y_t|x_t)$ due to the occlusion. Conversely, fewer samples are retained with clean observations. This behaviour is desirable – during an occlusion, particles that are more probable according to $P(y_t|x_t)$ are often *less* accurate in truth since they have latched onto spurious edges and foreground (Peursum *et al.* 2007). Initialising EM with more components will thus include a broader range of particles, giving smoothing

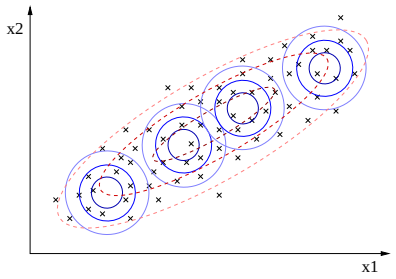


Fig. 6 Example Gaussian (dashed red) and mixture of Gaussians (blue circles) fitted to a set of samples (crosses). Although each component of the mixture assumes independence among dimensions, the *overall* mixture still captures the covariance of the samples.

the chance to override the erroneous observation information with future data. Although there are fewer components than particles, the low weights of the below-average particles means that they can be modelled with the tails of the Gaussian components spawned from the above-average particles.

Another practical issue is the relative isolation of many samples in the high-dimensional (28D) space given that there are only 1,000–10,000 particles. Gaussian components that are assigned to relatively isolated samples by EM will have their covariance collapse towards zero. Even fairly densely-sampled areas are unlikely to always contain enough particles to properly characterise the covariance of the neighbourhood in the true $P(y_t|x_t)$. Hence this paper fixes the covariance of all components to a diagonal 29×29 matrix (*i.e.* each dimension within a component is independent) with each diagonal variance manually set to a reasonable value for the posture dimension it represents. Specifically, the 24 joint rotation variances are all set to 9 (in degrees, *i.e.* standard deviation is 3°) global position variances of $\{x=400, y=400, z=100\}$ (millimetres) and global orientation variance set to 25 (degrees). Note that although the 28 posture dimensions are assumed to be independent *within* each component of the mixture, dependencies between dimensions are still modelled by the *overall* mixture. To illustrate how this is possible, consider Figure 6, where each Gaussian component is diagonal but together they form a strongly covariant mixture.

4.4 Variational Smoothing

Replacing the heuristic $P(y_t|x_t)$ with its mixture approximation $\hat{P}(y_t|x_t)$ facilitates the derivation of a variational approximation for Figure 3. Note that the Bayesian model for the FSHHMM-PF (Figure 3) reduces to that of the Simple-APF and Simple-PF (Figure 2) when there is only one phase q_t. The difference between the two is that the Simple-PF / Simple-APF posture is continuous and has Gaussian diffusion transitions, whereas the FSHHMM-PF is discrete to allow for modelling arbitrary non-linear/non-Gaussian motion in its transition distributions. Therefore, rather than derive two sets of variational update equations for what is essentially the same model, we derive the approximation for the discrete FSHHMM-PF and reuse this for the Simple-PF/Simple-APF by quantising their filtered postures and hand-crafting a discrete generic transition model that replicates their Gaussian diffusion dynamics. This entails some loss of accuracy on the part of the continuous Simple-PF and Simple-APF, but the quantisation error is small when compared to errors caused by tracking failures due to occlusions and poor observations (Peursum *et al.* 2007). Moreover, the manual ground-truth obtained with virtual markers is only accurate to within about $\pm$50mm. The remainder of this section describes the changes made to the FSHHMM-PF model of Figure 3 for the purposes of variational and Gibbs approximation.

4.4.1 Graphical Model with the Mixture

Figure 7a depicts the adjusted FS-HHMM used for variational approximation of posture, where $P(y_t|x_t)$ has been substituted with the mixture approximation $\hat{P}(y_t|x_t)$. The model parameters differ slightly from that of the original FS-HHMM in Figure 3. Eqs (7a)–(7e) remain unchanged. Additional parameters are:

$$\eta_t^{(k)} \triangleq P(s_t^{(k)}) \tag{22a}$$

$$P(y_{x,t}^{\{d\}}|x_t^{\{d\}}) = \sum_{k=1}^{K} \eta_t^{(k)} \mathcal{N}(x_t^{\{d\}} |\, \bar{y}_{x,t}^{\{d\}(k)}, R_{y_x}^{\{d\}}) \tag{22b}$$

$$P(y_{v,t}^{\{g\}}|v_t^{\{g\}}) = \sum_{k=1}^{K} \eta_t^{(k)} \mathcal{N}(v_t^{\{g\}} |\, \bar{y}_{v,t}^{\{g\}(k)}, R_{y_v}^{\{g\}}) \tag{22c}$$

$$P(y_{u,t}|u_t) = \sum_{k=1}^{K} \eta_t^{(k)} \mathcal{N}(u_t |\, \bar{y}_{u,t}^{(k)}, R_{y_u}) \tag{22d}$$

and Eqs (7g) and (7h) are changed to:

$$\Psi_t \triangleq P(v_t^{\{g\}}) = \mathcal{N}(v_t |\, \mu_{v,t}^{\{g\}}, \Sigma_{v,t}^{\{g\}}) \tag{22e}$$

$$\Upsilon_t \triangleq P(u_t) \quad = \mathcal{N}(u_t |\, \mu_{u,t}, \Sigma_{u,t}) \tag{22f}$$

where s_t in Eq. (22a) is the mixture component 'selector' expressed as a Boolean vector $s_t^{(k)}$ $k = \{1..K\}$ and $\eta_t^{(k)}$ is the mixture weights (k is indexed as a superscript to reinforce the fact that mixture components (k) arise from particles (i)). The means for the 28 dimensions of each component k in the observation mixture are split across body joint angles $x_t^{\{d\}}$, global body position $v_t^{\{g\}}$, $g \in \{pos_x, pos_y, pos_z\}$ (the g factors are not explicitly shown in the figure) and global body orientation u_t. Observation mixture means are thus defined as $\bar{y}_{x,t}^{\{d\}(k)}$, $\bar{y}_{v,t}^{\{g\}(k)}$ and $\bar{y}_{u,t}^{\{g\}(k)}$ respectively, with associated empirically-specified variances $R_{y_x}^{\{d\}}$, $R_{y_v}^{\{g\}}$ and R_{y_u}.

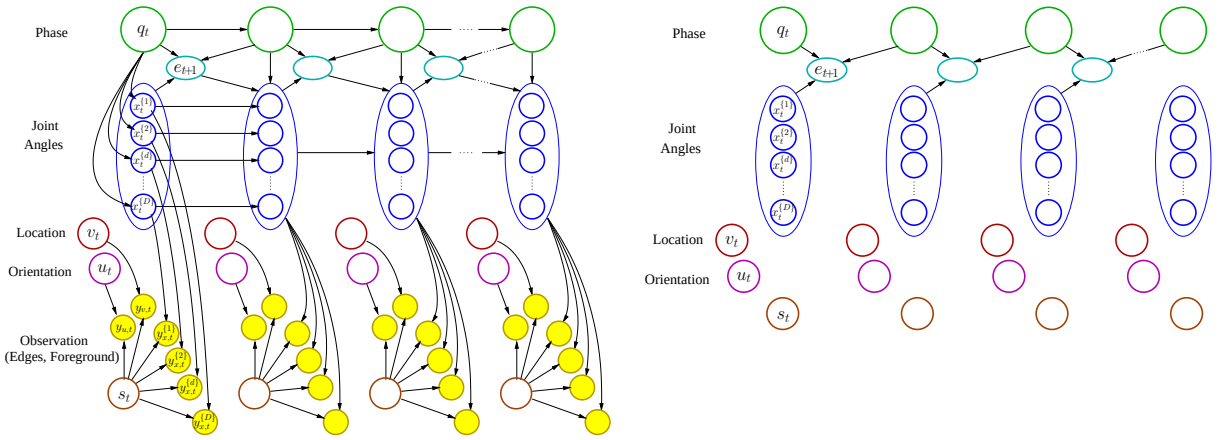


Fig. 7 (a) FS-HHMM with a mixture approximation $\hat{P}(y_t|x_t)$ for the observation likelihood. (b) Quasi-mean-field variational representation of (a) – the dependency structure for $P(e_t^{\{d\}}|q_t, q_{t+1}, x_t^{\{d\}})$ is retained since it is deterministic.

Note the peculiarity that the observations $\bar{y}_{x,t}$ are in fact the means of the Gaussian mixture and x_t is the data variable, rather than vice-versa as $P(y_t|x_t)$ would imply. This isn't a problem since Gaussians are symmetric about their mean, hence viewing either variable as the mean is equivalent.

$\{\mu_{v,t}^{\{g\}}, \Sigma_{v,t}^{\{g\}}\}$ and $\{\mu_{u,t}, \Sigma_{u,t}\}$ are the means and variances for the priors on v_t and u_t. Note that the dynamics dependencies $P(v_t|v_{t-1})$ and $P(u_t|u_{t-1})$ have been dropped; this has been facilitated by the mixture's properties. Specifically, the mixture has low variance for the dimensions representing the person's global 3D position and orientation $\{\bar{y}_{v,t}^{\{g\}}, \bar{y}_{u,t}\}$. This is due to the fact that these dimensions lie at the very root of the body model's kinematic tree (see Figure 4) and so do not accumulate the uncertainties of nodes higher up in the tree. In other words, $P(y_t|v_t, u_t)$ is non-zero only in a small area and so the dynamics dependencies are largely redundant. In fact, the assumption of the linear dynamics model for $P(v_t|v_{t-1})$ is actually detrimental since people do not move in a strictly linear fashion, but the dynamics will erroneously bias a variational approximation towards such linear motion. Instead, $P(v_t)$ and $P(u_t)$ are characterised by fitting a Gaussian to the particles of the pre-processing filter:

$$\mu_{v,t}^{\{g\}} = \sum_i w_t^{(i)} v_t^{\{g\}^{(i)}} \tag{23a}$$

$$\Sigma_{v,t}^{\{g\}} = \sum_i \left(w_t^{(i)} v_t^{\{g\}^{(i)}}\right)^2 - \left(\mu_{v,t}^{\{g\}}\right)^2 \tag{23b}$$

$$\mu_{u,t} = \sum_i w_t^{(i)} u_t^{(i)} \tag{23c}$$

$$\Sigma_{u,t} = \sum_i \left(w_t^{(i)} u_t^{(i)}\right)^2 - \left(\mu_{u,t}\right)^2 \tag{23d}$$

4.4.2 Variational Equations

Variational approximation proceeds by obtaining a lower bound on the log-likelihood of the true posterior $\mathcal{P} = P(x_{1:T}|y_{1:T})$ using a more tractable distribution $\mathcal{Q}(x_{1:T})$. This lower bound is achieved by varying $\mathcal{Q}$ to minimise the Kullback-Liebler (KL) divergence between $\mathcal{P}$ and $\mathcal{Q}$, where KL is defined as:

$$\begin{aligned}
\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) &= \int \mathcal{Q}(x_{1:T}) \log\left(\frac{\mathcal{Q}(x_{1:T})}{\mathcal{P}(x_{1:T}|y_{1:T})}\right) dx_{1:T} \\
&= \int \mathcal{Q}(x_{1:T}) \log \mathcal{Q}(x_{1:T})\, dx_{1:T} \\
&\quad - \int \mathcal{Q}(x_{1:T}) \log \mathcal{P}(x_{1:T}|y_{1:T})\, dx_{1:T} \\
&= \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{Q} \right\rangle - \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P} \right\rangle
\end{aligned} \tag{24}$$

where $x_{1:T} \triangleq \{x_{1:T}^{\{1:D\}}, v_{1:T}^{\{1:G\}}, u_{1:T}, s_{1:T}\}$ in the FS-HHMM and $\mathbb{E}_P\langle f \rangle$ is the expectation of f with respect to the distribution P. For a more detailed introduction to variational approximation in the context of factored HMMs, the reader is referred to Ghahramani and Jordan (1997). Note that the KL divergence can be interchanged with the variational free energy $\mathrm{KL}_{\mathcal{F}}$ (Sminchisescu and Jepson 2004), which differs from KL only in that $\mathcal{P}_{\mathcal{F}} = P(x_{1:T}, y_{1:T}) \propto \mathcal{P}$. This is useful since it is usually algebraically easier to work with the joint probability and $\mathrm{KL}_{\mathcal{F}}$ does not change the minimum of the KL divergence. This can be shown by expanding the second term of Eq (24):

$$\begin{aligned}
\mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P} \right\rangle &= \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P}(x_{1:T}|y_{1:T}) \right\rangle \\
&= \mathbb{E}_{\mathcal{Q}}\left\langle \log \frac{\mathcal{P}(x_{1:T}, y_{1:T})}{\mathcal{P}(y_{1:T})} \right\rangle \\
&= \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P}(x_{1:T}, y_{1:T}) \right\rangle - \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P}(y_{1:T}) \right\rangle \\
\therefore \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P} \right\rangle &= \mathbb{E}_{\mathcal{Q}}\left\langle \log \mathcal{P}_{\mathcal{F}} \right\rangle - K
\end{aligned} \tag{25}$$

where $K = \mathcal{P}(y_{1:T})$, a constant with respect to Q and hence it has no effect on the position of the KL minimum.

In this paper, $\mathcal{Q}$ is essentially a mean-field version of $\mathcal{P}$ where almost all of the hidden states are assumed to be independent, as shown in Figure 7b. However, the model does retain some structure from Figure 7a, specifically the conditional dependencies for $P(e_t^{\{d\}}|q_t, q_{t+1}, x_t^{\{d\}})$. This simplifies variational inference because not only does $P(e_t^{\{d\}}|q_t, q_{t+1}, x_t^{\{d\}})$ cancel out in the KL divergence (since it exists in both $\mathcal{P}$ and $\mathcal{Q}$), it also does not require an inference calculation since it is deterministic. From Peursum *et al.* (2007), e_t is:

$$e_t^{\{d\}} = \begin{cases} 1 & \text{if } \forall j, P(x_t^{\{d\}}{=}j, q_t | x_{t-1}^{\{d\}}, q_{t-1}) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

The definition of the variational distribution $\mathcal{Q}$ is thus:

$$\begin{aligned} \mathcal{Q} = & \prod_{t=1}^{T} \prod_{m=1}^{M} \Big(\theta(q)_{t,m}\Big)^{q_{t,m}} \times \prod_{t=1}^{T} \prod_{d=1}^{D} \prod_{i=1}^{120} \Big(\theta(x)_{t,i}^{\{d\}}\Big)^{x_{t,i}^{\{d\}}} \\ & \times \prod_{t=1}^{T} \prod_{g=1}^{G} \mathcal{N}\big(v_t^{\{g\}} | \theta(v_\mu)_t^{\{g\}}, \theta(v_\Sigma)_t^{\{g\}}\big) \\ & \times \prod_{t=1}^{T} \mathcal{N}\big(u_t | \theta(u_\mu)_t, \theta(u_\Sigma)_t\big) \times \prod_{t=1}^{T} \prod_{k=1}^{K} \Big(\theta(s)_t^{(k)}\Big)^{s_t^{(k)}} \\ & \times \prod_{t=2}^{T} \prod_{d=1}^{D} P(e_t^{\{d\}} | q_t, q_{t-1}, x_{t-1}^{\{d\}}) \end{aligned} \tag{27}$$

where $P(e_t^{\{d\}}|q_t, q_{t-1}, x_{t-1}^{\{d\}})$ is not further expanded into its parametric form $\Omega_{mni}^{\{d\}}$ since it will end up cancelling out with the equivalent term in $\mathcal{P}$ when calculating $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$. Here, the parameters $\theta(q)_{t,m}$, $\theta(x)_{t,i}^{\{d\}}$, $\theta(s)_t$, $\theta(v_\mu)_t^{\{g\}}$, $\theta(v_\Sigma)_t^{\{g\}}$, $\theta(u_\mu)_t$ and $\theta(u_\Sigma)_t$ are the *variational parameters* which will be used to approximate the original parameters $q_{t,m}$, $x_{t,i}^{\{d\}}$, s_t, $v_t^{\{g\}}$ and u_t in $\mathcal{P}$. Note that $i = \{1..120\}$ for x_t since this paper quantises the joint rotations at 3° intervals (3×120=360). Also note that the distributions for v_t and u_t are similar to the priors in $\mathcal{P}$ (Eqs (22e)–(22f)), differing in that they are parameterised by the variational $\theta(\cdot_\mu)$ and $\theta(\cdot_\Sigma)$ rather than Eqs (23a)–(23d).

For purpose-built models such as the FS-HHMM the derivation of the variational approximation is quite lengthy since the most basic 'building block' is the entire structure at a single time-slice t. Hence for the sake of brevity, the reader is referred to Peursum (2008) for details on deriving the variational update equations of the FS-HHMM model. Instead, the remainder of this section will describe the process of utilising the update equations found in (Peursum 2008). Briefly, to derive the update equations one must plug Eq (27) and the equivalent equation for the joint probability of $\mathcal{P}$ into the KL divergence (24), then take derivatives with respect to the various $\theta(\cdot)$ parameters and solve for zero. This leads to a set of fixed-point update equations. Note that to do this, the dynamic Bayesian networks of Figure 7 are being implicitly 'unrolled' across time to match the length of the observed sequence $y_{1:T}$, hence the list of $\theta(\cdot)$ variables is also fixed (*i.e.* the variational updates are occurring on a fixed network).

These $\theta(\cdot)$ are then optimised by iteratively evaluating all the fixed-point equations for each $\theta(\cdot)$ one round at a time. Specifically, a single iteration round involves calculating the new value of each $\theta(\cdot)$ in turn using the latest version of all the other $\theta(\cdot)$'s (*i.e.* use the new values of $\theta(\cdot)$'s that were updated earlier in the current iteration round). The order of updating the $\theta(\cdot)$ variables is in terms of moving along the Bayesian network from top-to-bottom (q_t before x_t) and left-to-right ($t = 1$ before $t = 2$, etc), although convergence should occur regardless of the chosen ordering. Iterating these update rounds will then converge to a locally optimal solution, where convergence is monitored by calculating the KL divergence after each update iteration, and comparing this to the previous iteration's KL divergence.

4.5 Gibbs Smoothing

As with the variational approximation, Gibbs inference (Andrieu *et al.* 2003; Ghahramani and Jordan 1997) uses the model described in Section 4.4.1, with the Bayesian network of Figure 7a. Gibbs sampling is one of the simplest Markov Chain Monte Carlo methods, and proceeds by implicitly unrolling the dynamic Bayesian network to a fixed network of length T to match the observed sequence $y_{1:T}$. Hidden states are then set to an initial value before repeatedly sampling new values for each state given its Markov blanket until N samples of the full joint distribution are drawn. Sampling occurs in rounds, where during round p a sample is drawn for each hidden state given the current value of the other states (whose value may have been updated earlier in round p, depending on the order of processing). Again, the order in which the states are processed is from top-to-bottom and left-to-right along the Bayesian network. Once all states have been sampled the process is repeated for round $p+1$, continuing until $p = N$. The full set of sampled values over all rounds N then provides the sufficient statistics for the Gibbs estimate of each state – in the case of a discrete distribution this is the histogram of the samples and for a Gaussian it is the sample mean and covariance. See Peursum (2008) for details on the Gibbs sampling distributions necessary for Figure 7a. Intuitively, Gibbs sampling works by drawing a new sample that is 'consistent' with the current values of states that can influence it (*i.e.* its Markov blanket). This consistency is then propagated along the network one link at a time, resulting in states concentrating their sampling in areas of the state space that 'make sense' given the state's location in the unrolled Bayesian network. Eventually this leads to the samples effectively being drawn from the

true posterior $P(x_t|y_{1:T})$. Since the initial values may not be very consistent with each other, the Markov sampling chain usually requires time (called burn-in) to converge to sampling from $P(x_{1:T}|y_{1:T})$. Thus the first 5%–10% of samples are typically excluded from the final Gibbs estimate. The number of burn-in samples needed to achieve convergence depends on the initial state values and the distributions being sampled from, and is difficult to estimate in advance.

In this paper, the number of Gibbs samples is set to 2,000 for the 1,000-particle FSHHMM-PF. Empirical tests showed that values larger than 2,000 do not significantly improve the Gibbs estimate, indicating that convergence has been reached. For the Simple-PF / Simple-APF (10,000 particles), 5,000 Gibbs samples are used. Although one would expect that the 10× difference in filtering particles would suggest the use of 20,000 Gibbs samples, the lower value of 5,000 is chosen so as to keep the computational runtime of Simple-PF/Simple-APF Gibbs smoothing in the same ballpark as the computational time of Simple-PF/Simple-APF filtering. Burn-in time for all filters is set to the first 5% of Gibbs samples. See Section 5.2 for details on initialisation.

5 Experimental Setup

The filtering and smoothing algorithms described in this paper were evaluated against twelve video sequences – seven HUMANEVA-I videos, two HUMANEVA-II sequences (Sigal and Black 2006) and three CLUTTER videos captured in scenes containing occluding tables and chairs. Each sequence is processed with 12 algorithms (Table 1) to produce 144 tracking results in total.

Type	Algorithms		
Filtering	Simple-PF	Simple-APF	FSHHMM-PF
FBS	Simple-PF+FBS	Simple-APF+FBS	FSHHMM-PF+FBS
Variational	Simple-PF+Vartnl	Simple-APF+Vartnl	FSHHMM-PF+Vartnl
Gibbs	Simple-PF+Gibbs	Simple-APF+Gibbs	FSHHMM-PF+Gibbs

Table 1 Filtering and smoothing combinations (12 in total) employed for tracking in each video.

5.1 Test Scenes and Ground-Truth

Datasets Twelve video sequences of walking are used for test data, seven from the HUMANEVA-I dataset, two from the HUMANEVA-II dataset and three from CLUTTER. The HUMANEVA sequences (Figure 8) consists of several videos captured in tandem with marker-based motion capture, hence actors are restricted to moving on a 3m×3m mat. Several actors are used, each with different physiques. Videos are captured at 640×480 resolution and 60 frames per second (fps) – HUMANEVA-I uses three colour cameras whereas HUMANEVA-II employs four colour cameras (this paper does not use the greyscale HUMANEVA videos). All views were

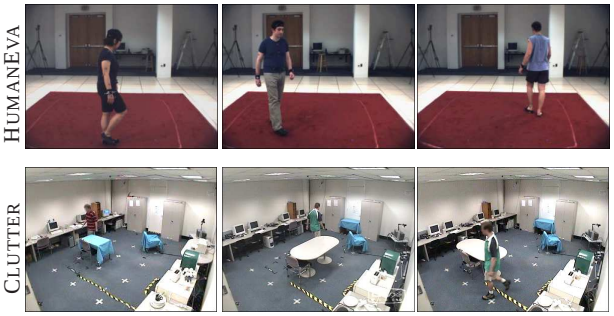


Fig. 8 Example screenshots of the two datasets used in this paper.

used for tracking in this paper. Although the dataset contains various actions (walking, boxing, gesturing), this paper only considers the sections of video which contain walking motion where the actor walks in a circle for up to a minute. Ground-truth is provided by the marker-based motion capture, and this paper evaluates accuracy with most of the available joints (13 of 15 unique 3D joints – upper leg proximals were ignored due to difficulties in defining corresponding points on the body model that matched well).

The CLUTTER data set (Figure 8) is captured in a 7m×6m room monitored by four ceiling-mounted colour cameras, one in each corner. All views are used for tracking. The room contains a variety of furnishings and whitegoods. A table and chairs were placed in the center of the room to produce a reasonably cluttered home-like scene and three video sequences were captured where the placement of the occluding tables and chairs was changed for each sequence. Videos are captured at 384×288 resolution and 25fps, with the room initially empty for background learning before the actor enters and walks through the cluttered scene for about a minute. No motion-captured ground-truth exists, hence the ground-truth was manually labelled using a GUI utility developed to minimise the tedium of the task (Peursum 2008).[1]

Evaluation of Error Although the various filtering and smoothing algorithms return a distribution of postures, for simplicity the mean posture is taken and compared against the ground-truth. In the case of particle filtering and FBS, this mean is the weighted mean of the particle set. For the variational and Gibbs approximations the mean of each joint rotation is calculated independently (due to the factorisation assumptions of the models in Figures 2 and 3).

In order to be consistent with other research based on the HUMANEVA dataset, the difference between the mean posture and ground-truth for a sequence is described in terms of the mean and variance of the 3D Euclidean error (in millimetres)[2], as defined in Equations (3)–(6) of Sigal and Black (2006). Since multiple views are employed the absolute 3D

[2] This implies a Gaussian distribution – although a Rayleigh distribution may be better suited to modelling Euclidean error since the error is in the range $[0 \ldots \infty)$, the Gaussian is a reasonable approximation and its mean and variance parameters are far more intuitive.

		Filtering Error (mm)											
Sequence →		HE-I 1.1	HE-I 1.2	HE-I 2.1	HE-I 2.2	HE-I 3.1	HE-I 3.2	HE-I 4.2	HE-II 2.1	HE-II 4.4	Clu 1	Clu 2	Clu 3
	Frames	6-590	6–605	6–438	6–605	6–448	6–605	6–605	1–380	2–380	265–865	300–1300	270–1270
Simple-PF	Mean	79.9	120.4	100.4	103.2	86.3	120.9	103.6	104.7	83.2	131.5	188.5	184.7
	StdDev	±21.6	±16.2	±17.8	±19.3	±29.4	±21.9	±27.5	±12.9	±23.6	±50.1	±66.2	±56.3
Simple-APF	Mean	81.3	113.0	112.1	97.8	81.1	87.9	100.3	118.2	95.6	143.8	218.9	193.9
	StdDev	±24.7	±29.7	±16.3	±33.9	±38.3	±43.5	±20.0	±29.6	±22.5	±50.3	±60.4	±61.1
FSHHMM-PF	Mean	85.5	89.6	116.9	87.8	84.7	98.2	87.5	106.6	92.0	102.2	105.4	123.8
	StdDev	±35.7	±9.7	±11.9	±10.1	±22.3	±10.9	±9.9	±8.2	±21.0	±29.5	±25.2	±34.2

Table 2 Mean error and standard deviation of error for each of the filtering algorithms on the seven HUMANEVA-I (HE-I), two HUMANEVA-II (HE-II) and three CLUTTER (Clu) sequences. All sequences are of walking (for HUMANEVA-II, only the frames with walking were used). HUMANEVA sequences are abbreviated as follows: HE-I *A.S* = HUMANEVA-I, Actor *A*, Walking Sequence *S*.

error is calculated. Note that for the CLUTTER dataset, the ground-truth obtained via virtual markers is itself uncertain due to the manual nature of the labelling – for the CLUTTER dataset this uncertainty is approximately ±50mm.

5.2 Training and Initialisation

Filtering Both the Simple-PF and Simple-APF have their Gaussian diffusion parameters empirically defined. For 25fps video, rotation variances are $\{6, 8, 4\}$ (in degrees) of each joint's azimuth, elevation and roll respectively, position variances are $\{x = 300, y = 300, z = 100\}$ and orientation variance 20 (also in degrees). Variances are scaled to accommodate the HUMANEVA dataset's 60fps. The initial posture is also manually defined by the user.

For the FSHHMM-PF, the ground-truth posture from a single video sequence of a person taking four steps in a straight line is used to train a walking model. This is sufficient for the FSHHMM-PF to track a person through turns even though the training data does not contain these movements (Peursum *et al.* 2007). Training data is captured at 50fps, and although the test data is captured at 25fps the FSHHMM-PF can handle the difference in frame rates. The FSHHMM-PF also estimates the initial posture without human intervention.

Smoothing The variational and Gibbs inference approximations both require initialisation of their state. Rather than initialise randomly, both initialisations are extracted from the particles from the filtering step. For variational this involves calculating the distribution of each hidden state based on the particles, whereas for Gibbs sampling each state is set to the value of the most-likely particle at each time t. Good initialisation is important for both smoothing techniques, but for different reasons. In particular, the optimisation surface in variational approximation can have multiple local maxima (Corduneanu and Bishop 2001; Winn and Bishop 2005) and so initialisation will determine which local maxima is selected. This is very similar to the Expectation-Maximisation (EM) algorithm, given that both EM and variational can be viewed as optimising the KL divergence (Barber and Bishop 1998; Neal and Hinton 1998). For Gibbs sampling, good initial values are important in order to minimise the time taken for the Gibbs MCMC chain to reach convergence.

6 Results and Analysis

6.1 Filtering Performance

Table 2 summarises the accuracy of each pre-processing filter (Simple-PF, Simple-APF and FSHHMM-PF) with respect to the ten sequences. Note that some of the Simple-APF sequences (HUMANEVA-I 2.1 and all CLUTTER) suffered from severe tracking failures where the person's body orientation becomes reversed over the course of about 10 frames (Figure 10). This brittleness of the Simple-APF arises because annealing discards much of the particle distribution to focus on one or two modes and selection of a poor mode can lead to severe tracking failures. Conversely, the Simple-PF does not suffer from the same issue. To avoid biasing the subsequent smoothing results against the Simple-APF, the frames during which the failures occur were excluded from the error calculation.

HumanEva Dataset For the HUMANEVA-I and -II sequences, the average filtering error across all sequences is {100mm, 98mm, 94mm} for the Simple-PF, Simple-APF and FSHHMM-PF respectively. Of interest is that the FSHHMM-PF (with only 1,000 particles, an added error due to its discrete nodes, and trained on a walking model that contains no turns) is on par with the 10,000-particle Simple-PF and Simple-APF despite the HUMANEVA actors continuously walking in a tight circle. Overall, the tracking accuracy of the three filters is comparable to the results of other recent body tracking systems tested with the HUMANEVA walking sequences.

Clutter Dataset As expected, accuracy is worse in the CLUTTER dataset, with the average error rising to {168mm, 185mm, 110mm}. Here, the motion model of the FSHHMM-PF minimises the degradation in tracking by guiding tracking when occlusions, poor contrast and/or low resolution cause the observations to become unreliable. In contrast, the generic

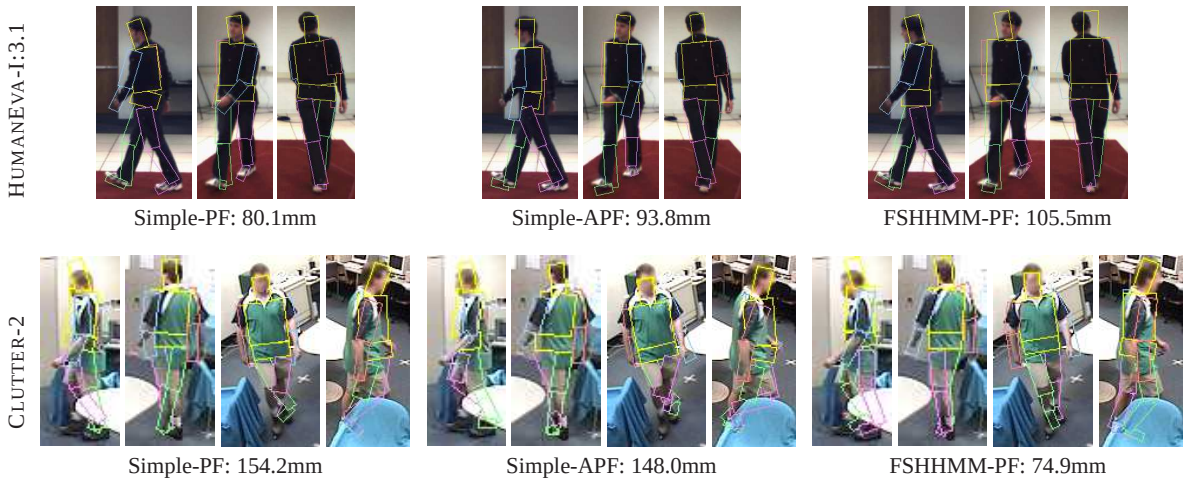


Fig. 9 Examples of tracking for each filter with corresponding error. All camera views are shown.

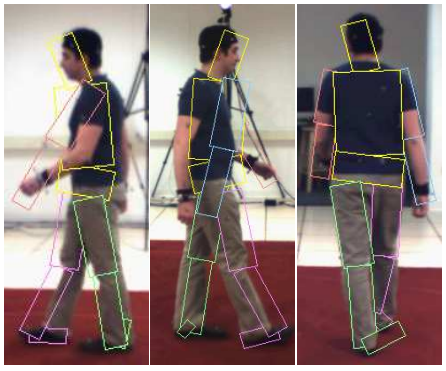


Fig. 10 The Simple-APF experiences severe tracking failures in several sequences, such as this complete reversal of body orientation for HumanEva sequence 2.1.

models of the Simple-PF and Simple-APF cannot provide similar guidance and so erratic tracking and failures ensue.

Note that the Simple-PF and Simple-APF perform similarly to each other, confirming the findings of Bălan *et al.* (2005) that the APF does not perform much better than a standard PF with the same motion model. This seemingly contradicts Deutscher and Reid (2005), who found that the PF was far less accurate. However, Deutscher did not 'anneal' the PF observation likelihood as this paper does – if λ is set to 1 in Eq (9), the Simple-PF does indeed fail very quickly as Deutscher found. In contrast, given a λ=10 Table 2 shows that the Simple-PF returns similar results to the Simple-APF. The APF's strategy of focusing on the peaks of the observation likelihood modes while discarding the rest of the distribution means that when observing conditions are poor it can follow a poor mode and have difficulty recovering when observations improve, hence the Simple-APF performs slightly better than the Simple-PF in the HumanEva sequences and worse in the Clutter sequences, in addition to occasionally experiencing the aforementioned body reversal tracking failures.

6.2 Smoothing Performance

Figure 12 shows the relative performance of the three smoothing algorithms for all three pre-processing filters. Contrary to expectations that incorporating additional evidence should lead to a better estimate, most of the smoothing results are often *less* accurate than the pre-processing filter that they are based upon. This is particularly noteworthy in the case of the Clutter sequences where the poor observation conditions cause temporary ambiguities in filtering that smoothing should have been able to improve on.

To gain an insight into why smoothing performs so unexpectedly a closer look was taken at the tracking sequences that were output by the various algorithms. This inspection found that there are certain times when smoothing *does* improve tracking accuracy, but this is offset by other occasions where smoothing performs more poorly than filtering.

Fig. 11 Examples of smoothing (light yellow) correcting a temporary lag error in filtering (dark blue).

Where Smoothing Works Although in each sequence there are many frames in which smoothing outperforms filtering, most of these seem to be random fluctuations in accuracy. However, there are circumstances in which smoothing consistently improves upon the accuracy of filtering. In particular, smoothing is able to correct filtering 'lag'. This is where the filter lags behind the true motion when the person moves faster than the motion model predicts (*e.g.* as the person's

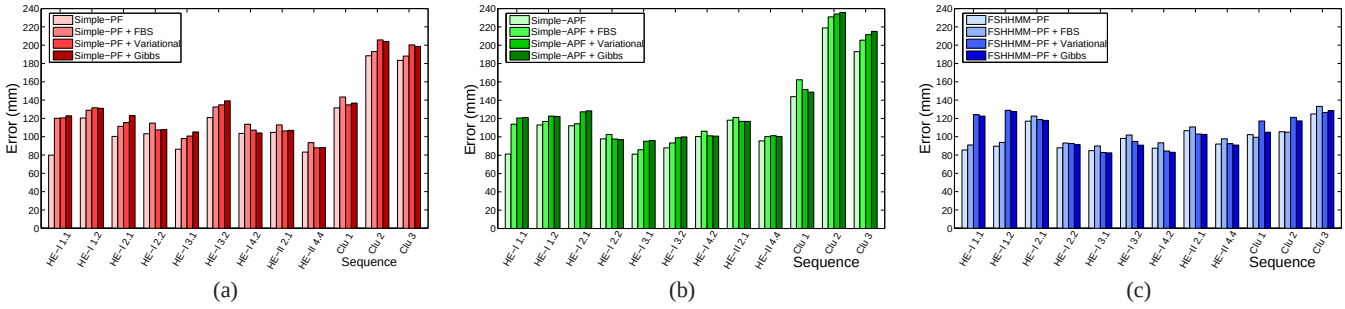


Fig. 12 Comparison of filtering and smoothing algorithm results. Each sub-figure depicts the filtering and smoothing error against every sequence for a given pre-processing filter: (a) Simple-PF, (b) Simple-APF, (c) FSHHMM-PF.
Datasets are labelled as follows: HE-I *A.S* = HUMANEVA-I (Actor *A*, Walking Sequence *S*); HE-II *A.S* = HUMANEVA-II; Clu = CLUTTER.

leg swings forward during a stride). When such a lag occurs, only a few particles are able to keep up with the motion. Although these particles are highly-weighted they are not numerous enough to dominate the mean body posture used in error evaluation. This creates a lag effect in the filtered track that is often corrected by the filter a few frames later due to resampling and/or the motion slowing down enough to allow the particles to catch up (*e.g.* at the end of the leg swing).

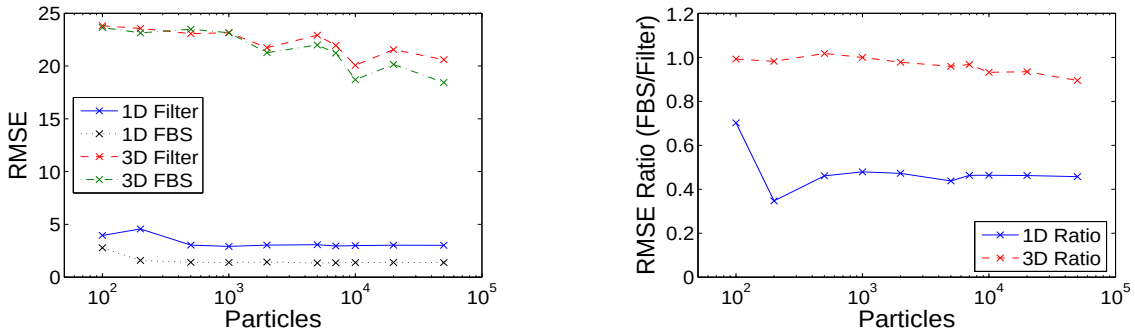
Smoothing is able to overcome this lag since the filter has sampled at least some particles along the correct trajectory and so smoothing can downgrade the contribution of lagging particles since they have a low probability of transitioning to future states. In particular, FBS downgrades the weight of lagging particle whereas variational and Gibbs approximations never visit the lagging mixture components since they are unlikely transition targets given the future states. Figure 11 shows examples of this lag correction.

Where Smoothing Fails The positive smoothing effect described above is only prevalent with the FSHHMM-PF when tracking in the CLUTTER sequences. Tracking in the HUMANEVA sequences does not see as much benefit because filtering rarely lags behind the true motion given that the HUMANEVA frame rate is comparable to the FSHHMM-PF's training frame rate (60fps vs 50fps). In contrast, the CLUTTER sequences are at 25fps and so motion often outpaces the motion model's predictions. However, if no particle keeps up with the motion, the lag cannot be overcome since particles attempting to 'bridge the gap' will have a low weight and so are discarded during resampling. Since the frame-to-frame effective sample size is quite low at around 0.05–0.2, resampling is unavoidably frequent and will eliminate all but a few particles. In fact, in a 1,000-particle filter only one particle at time t can expect to have surviving 'descendents' at $t+3$! This rate of degeneracy means that mode exploration is limited, hence errors lasting more than a couple of frames are not resolvable by smoothing. Unfortunately, such aggressive resampling is necessary since otherwise the filter will lose coherence and become inaccurate as particles explore the multitudes of distracting modes in the high-dimensional space.

In terms of filtering algorithms, the Simple-PF and Simple-APF do not realise smoothing benefits in the CLUTTER sequences because the frequent occlusions produce observation ambiguities that result in the filters being unable to maintain a good tracking lock for very long. These same ambiguities also affect the FSHHMM-PF, but its strong motion model minimises the duration and magnitude of the resulting tracking failures. In fact, if filtered tracking fails with any of the three filtering algorithms and the filter does not quickly correct itself, resampling will ensure that the error becomes permanent. Smoothing will then often result in a *lower* accuracy than filtering because the smoother adjusts the filtered trajectory towards an *erroneous* future state and so propagates the error backwards in time. Whether or not this causes the smoother to produce poorer accuracy depends on the nature of the error.

Fig. 13 Example of variational smoothing (light yellow) causing an odd limb posture during a lag error in filtering (dark blue). At frame 558 the smoothed calf rotation is carried over from Frame 557, whereas the thigh rotation better matches that of Frame 559. Considered independently, both joint angles have good support from the mixture approximation $\hat{P}(y_t|x_t)$, but together they create an odd limb posture.

In comparison to FBS, variational and Gibbs smoothed inference are less adept at correcting lag. This is not too surprising given that the former two are approximate smoothers based on another approximation – the observation mixture $\hat{P}(y_t|x_t)$, which is in turn based on a particle filter approximation. In cases where FBS is able to smooth a lag, the vari-



(a) (b)

Fig. 14 Smoothing with a synthetic non-linear, multi-modal model. Figure (a) shows the root mean squared error (RMSE) whereas Figure (b) depicts the ratio between smoothing and filtering.

ational and Gibbs approximations instead may return postures with odd limb positioning (see Figure 13). Such postures do not exist in the original filtered particle set; they arise due to the independence assumptions of the mixture approximation, which means that the joint rotations in variational and Gibbs inference are estimated independently. Thus odd joint combinations may occur, especially when the filter is shifting from one mode to another such as during lag. In contrast, FBS smooths by adjusting the weights of entire particles, where the weights reflect the full body posture, and thus joint dependencies are enforced given the observation. Despite these handicaps, the variational and Gibbs approximations actually provide an improvement in accuracy for three FSHHMM-PF HUMANEVA sequences and is only significantly worse than FBS in two cases (also in the FSHHMM-PF; HE-I 1.1 and HE-I 1.2). Given the variability of the variational and Gibbs approximations' performances, it is likely that they are being limited by how well the mixture approximates the important areas of the true observation likelihood, which in turn is limited by the performance of the particle filter. Since the image-based observation likelihood will contain a high number of local maxima (Smith and Lovell 2006), it is difficult to approximate $\hat{P}(y_t|x_t)$ with a reasonable number of mixture components even if the particle filter visits several modes.

7 Follow-up Experiments

7.1 Effect of Dimensionality

In Section 6 smoothing only provided consistent improvements when there were at least some particles in the filter that follow the correct (though less-likely) mode and the ambiguity does not last so long that the particles are lost during resampling. Thus it is probable that the ineffectiveness of smoothing stems from an insufficient number of particles for the size of the 28D state space – the filter is simply unable to keep track of enough modes, especially given that the number of modes grows exponentially with dimensionality. Under this hypothesis, the FSHHMM-PF is able to gain more benefit from smoothing than the Simple-PF or Simple-APF since the former employs a strong motion model to sample particles from good areas of the state space. Such a motion model is implicitly channelling its particle sampling along a lower-dimensional manifold according to the model's training data, effectively performing a type of dimensionality reduction.

Follow-up experiments were thus conducted to test the hypothesis that the 'curse of dimensionality' is the main factor behind the unimpressive smoothing results seen in this paper. The experiments are based on the same non-linear multi-modal synthetic model that Godsill *et al.* (2004) employed to demonstrate the worth of smoothing (a similar model was also used by Klaas *et al.* (2006)). The model's dynamics equation and observation emission equation is as follows:

$$x_t = \frac{1}{2}x_{t-1} + \frac{25x_{t-1}}{1+(x_{t-1})^2} + 8\cos(1.2t) + \nu_t \tag{28}$$

$$y_t = \frac{(x_t)^2}{20} + \omega_t \tag{29}$$

where ν_t and ω_t are zero-mean Gaussian noise with covariances 10 and 1 respectively, and the initial x_1 is sampled from a zero-mean Gaussian with covariance 10. The square term in the observation equation (29) creates a dual modality where the observed y_t fits $\pm x_t$ equally well, and the strongly non-linear dynamics of Eq (28) ensures that particles which are slightly incorrect can easily select the wrong mode.

Filtering and FBS smoothing was performed with between 100 and 50,000 particles. The ground-truth $x_{1:T}$ and sampled observations $y_{1:T}$ with $T = 100$ are generated according to equations (28) and (29) for a 1D case as well as an analogous 3D case where x_t and y_t are three-element vectors. For simplicity the 3D case assumes a diagonal covariance with all variances set to 10. Variational and Gibbs sampling were not tested against this synthetic model since

their ability to scale to high dimensionality given a well-behaved observation function with simple Gaussian noise has been previously demonstrated (see Ghahramani and Jordan (1997)).

Figure 14a plots the root mean squared error (RMSE) of the filtered and FBS-smoothed trajectories for both the 1D and 3D cases. Only one run was conducted for each particular particle count hence the trend as particles increase is somewhat noisy, especially for the 3D case. Figure 14b uses the same data but depicts the ratio between the smoothed and filtered cases to indicate the relative improvement that smoothing provides. The behaviour of the 1D case is similar to that reported by Klaas *et al.* (2006) in that FBS is able to significantly improve upon filtering, with diminishing returns starting to occur beyond a certain number of particles (in this case, at around 500). For the 3D case smoothing is unable to provide any real improvement on filtering (and is sometimes *worse* than filtering) until at least 2,000 particles are used, at which point smoothing begins to increasingly improve upon filtering. However, this improvement is limited – for the 3D case smoothing is only 10% better than filtering at 50,000 particles, in comparison to the more than 50% benefit that smoothing brings to the 1D case with just 500 particles.

This discrepancy can be explained by considering the difference in size of the two state spaces; if around 200 particles gives good results in a 1D case, one would expect that 200^3 = 8,000,000 particles is necessary to see a similar performance in the 3D case. Figure 14 seems to bear this rule-of-thumb out for both the particle filter and the associated smoother. Thus in Section 6 it is likely that smoothing is unable to benefit tracking since an impractically large number of particles would be necessary to properly explore the 28D state space. It would be interesting to see whether smoothing would be more effective with trackers that explicitly reduce dimensionality, such as that of Elgammal and Lee (2004) or Urtasun *et al.* (2006).

7.2 Effect of Multi-Modality

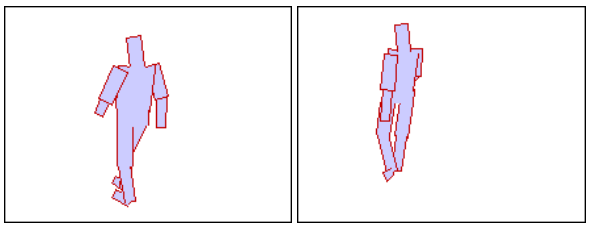


Fig. 15 Example screenshots of the monocular video generated by animating a known ground-truth with the body model of Figure 4.

Another possible explanation for the poor smoothing performance of Section 6 is that the experiments either had too few modes due to the use of multiple views, or the multi-modality was too 'messy' since it was being produced by occlusions. In either case it could be argued that smoothing is presented with few viable opportunities to overcome poor filtering, hence the weak smoothing results. Therefore, a quasi-synthetic video was generated in which only a single monocular view is used to capture a person walking in a straight line at an angle away from the camera. Such a scenario will experience strong depth ambiguities since much of the person's motion (*e.g.* leg and arm swinging) will be along the depth axis. In order to eliminate other distracting sources of ambiguity, a real video was captured of a walking person and ground-truthed. This was then used to generate a synthetic video by animating the body model of Figure 4. This ensures that the body model can perfectly fit the observation (avoiding ambiguities caused by a loose fit) as well as removing background clutter. Moreover, the ground-truth was in fact the training data of the FSHHMM-PF downsampled to 25fps (see Figure 15), further reducing ambiguity caused by deviations from the expected motion model of the FSHHMM-PF. Thus the only significant sources of ambiguity should be due to depth and self-occlusions.

	Relative Error (mm)			
Algorithm	Filter	FBS	Vartnl	Gibbs
Simple-PF	111.1	112.1	120.1	111.7
FSHHMM-PF	54.1	47.6	57.3	57.1

Table 3 Results of monocular tracking with the simulated video.

The Simple-PF and FSHHMM-PF were run against the animated sequence, along with all smoothing algorithms. The Simple-APF was not used since it would perform similarly to the Simple-PF. Table 3 lists the RMSE across all joints of the body model in terms of relative 3D error. Overall the trends are similar to Section 6, indicating that smoothing does not provide added benefit to monocular views with the algorithms employed. Note that the error of the Simple-PF is disappointingly high given the perfect observations and use of relative error.

The variational and Gibbs approximations are no more accurate than filtering, probably because the mixture $\hat{P}(y_t|x_t)$ is still attempting to approximate a 28D space with relatively few particles. These methods may find more success in lower-dimensional problems. The FSHHMM-PF does benefit from FBS smoothing, but as with Section 6 this is mostly in terms of correcting filtering lag (although the lag is more pronounced here due to the self-occlusions in a single view). Depth ambiguities in the filtering step are minimal since the strong motion model of the FSHHMM-PF reduces the number of observation modes that the filter can visit.

In contrast to the FSHHMM-PF, the Simple-PF experiences significant depth ambiguities since the generic motion model places few constraints on the filter's evolution. The

filter therefore has little difficulty finding highly-weighted – though often inaccurate – postures. Vondrak *et al.* (2008) found a similar effect when filtering with the APF in monocular views. Since the person does not turn and there is no other viewpoint, little evidence exists for smoothing to correct the filtering trajectories. Thus filtering and smoothing return similar results, with smoothing slightly less accurate.

One potential way to overcome this lack of evidence is the kinematic jump sampling of Sminchisescu and Triggs (2003). This involves sampling particles from a predicted posture *and* the prediction's depth-ambiguous mirrors, thereby maintaining multi-modality for longer periods and hence providing smoothing with the chance to choose between competing modes. However, ambiguities caused by factors other than depth (*e.g.* occlusions, loose-fitting body models) are unlikely to be resolved by such an approach. General-purpose methods for maintaining multi-modality (Vermaak *et al.* 2003) in conjunction with a strong motion model may be a worthwhile alternative to explore.

7.3 Time Efficiency of Algorithms

		Runtime per frame (sec.)			
Filter	Smoother	GMM	Smooth	Total	w/Filter
HUMANEVA-I Dataset:					
Simple-PF	-	-	-	**240s**	**240s**
	FBS	-	138s	**138s**	**378s**
	Variational	30s	40s	**70s**	**310s**
	Gibbs	30s	200s	**230s**	**470s**
Simple-APF	-	-	-	**245s**	**245s**
	FBS	-	140s	**140s**	**385s**
	Variational	14s	15s	**29s**	**274s**
	Gibbs	14s	93s	**107s**	**352s**
FSHHMM-PF	-	-	-	**25s**	**25.0s**
	FBS	-	0.8s	**0.8s**	**25.8s**
	Variational	0.5s	1.4s	**1.9s**	**26.9s**
	Gibbs	0.5s	1.7s	**2.3s**	**27.3s**
CLUTTER Dataset:					
Simple-PF	-	-	-	**95s**	**95s**
	FBS	-	133s	**133s**	**228s**
	Variational	30s	34s	**64s**	**159s**
	Gibbs	30s	108s	**138s**	**233s**
Simple-APF	-	-	-	**98s**	**98s**
	FBS	-	134s	**134s**	**232s**
	Variational	15s	12s	**27s**	**125s**
	Gibbs	15s	88s	**103s**	**201s**
FSHHMM-PF	-	-	-	**10s**	**10.0s**
	FBS	-	0.8s	**0.8s**	**10.8s**
	Variational	0.4s	0.8s	**1.2s**	**11.2s**
	Gibbs	0.4s	1.5s	**1.9s**	**11.9s**

Table 4 Average runtimes per frame for filtering and smoothing algorithms. HUMANEVA-I video consists of three camera views at 640×480 whereas CLUTTER video comprises four views 384×284 (this only affects filtering runtime since smoothing does not use the video). Experiments were run on an Intel Core-2 6420 2.13GHz.

Table 4 shows the runtime of the experiments from Sections 5 and 6. All algorithms are implemented in C++ and share a common set of video and image processing routines. No particular effort was made to optimise the code to increase performance. For example, object-oriented principles (information hiding, polymorphism), assertion checks and garbage collection were all extensively used, and no assembler or multi-threading was employed outside of the video decoding functions.

Note that the Simple-APF is faster than the Simple-PF for the variational and Gibbs approximations. This is due to the fact that duplicate particles are ignored during the construction of the mixture approximation $\hat{P}(y_t|x_t)$, which significantly reduces the number of components for the Simple-APF since annealing crowds many particles together that become identical when the state space is quantised to 3° intervals (see Section 4.4). The CLUTTER videos also process slightly faster than the equivalent HUMANEVA-I videos since the variance of particle weights is higher in CLUTTER due to it having more difficult scenes, hence fewer particles are retained as the seeds for components in the mixture.

For the 1,000-particle FSHHMM-PF, the overhead of smoothing is low for all algorithms (5%–20% extra computation time on top of filtering) since the constant cost of evaluating the observation likelihood during filtering dominates processing time. However, this gap largely disappears with the 10,000-particle Simple-PF and Simple-APF. Even the variational approximation becomes a significant fraction of the filtering time due to the increased number of mixture components that the 10,000 particles generate. Although the smoothing efficiency is better than one would expect given that particle filtering is generally seen as being quite efficient (Klaas *et al.* 2006), the fact that smoothing does not improve accuracy makes it difficult to recommend, at least with the filtering algorithms explored in this paper.

8 Conclusions

This paper has presented a study of three smoothed-inference algorithms (forward-backward smoothing, variational and Gibbs sampling) for 3D human body tracking in generative models. As part of the study, a method was introduced to facilitate the efficient execution of the latter two algorithms by approximating the observation likelihood with a mixture of Gaussians. Although it is generally expected that smoothing will improve upon the filtering estimate since smoothing incorporates future evidence, results in a multi-view environment show that the smoothing algorithms provide no real improvement on the tracking accuracy of filtering and in fact can produce an *increased* error. Simulations with monocular views indicate that the same issues exist in single views despite depth ambiguities being a classic case of multi-modality that smoothing should be well suited to handling. The cause of the poor performance of smoothing can be traced to the high-dimensional nature of body tracking – as with earlier studies, smoothing on a low-dimensional signal is shown to significantly improve upon filtering, but when the dimensionality is increased then exponentially more particles are needed to support effective

smoothing. In the high-dimensional state space of human posture this would require an impractically large number of particles. Thus for the filtering algorithms explored in this paper, smoothing does not significantly improve body tracking accuracy and the time spent on smoothing would be better spent on increasing the number of particles for filtering. Methods to overcome the difficulties facing smoothing in high-dimensional human body tracking are needed in order to take advantage of the benefits that smoothing can give during times of poor or ambiguous observations. In addition to investigating new approaches to smoothing and methods to maintain multi-modality in the filter, a more immediate possibility is to apply forward-backward particle smoothing to dimensionality-reducing body trackers, sidestepping the problem of high dimensionality altogether.

References

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, **50**, 5–43.

Barber, D. and Bishop, C. (1998). Ensemble learning in Bayesian neural networks. In M. Jordan, M. Kearns, and S. Solla, editors, *Neural Networks and Machine Learning*, pages 215–237. Springer.

Brubaker, M., Fleet, D. J., and Hertzmann, A. (2006). Physics-based human pose tracking. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM)*.

Brubaker, M. A., Fleet, D. J., and Hertzmann, A. (2007). Physics-based person tracking using simplified lower-body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Bălan, A., Sigal, L., and Black, M. J. (2005). A quantitative evaluation of video-based 3D person tracking. In *Proceedings of the Joint Workshop on Visual Surveillance and Performance and Evaluation of Tracking Systems (VS-PETS)*, pages 349–356.

Caillette, F., Galata, A., and Howard, T. (2005). Real-time 3-D human body tracking using variable length Markov models. In *Proceedings of the British Machine Vision Conference*, pages 469–478.

Cheng, S. Y. and Trivedi, M. (2007). Articulated human body pose inference from voxel data using a kinematically constrained Gaussian mixture model. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM)*.

Corduneanu, A. and Bishop, C. M. (2001). Variational Bayesian model selection for mixture distributions. In T. Richardson and T. Jaakkola, editors, *International Conference on Artificial Intelligence and Statistics*, pages 27–34.

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

Deutscher, J., Blake, A., and Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte-Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, **10**(3), 197–208.

Doucet, A., Godsill, S. J., and West, M. (2002). Monte Carlo filtering and smoothing with application to time-varying spectral estimation. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, volume 2, pages 701–704.

Elgammal, A. and Lee, C.-S. (2004). Inferring 3D body pose from silhouettes using activity manifold learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 681–688.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial Hidden Markov Models. *Machine Learning*, **29**, 245–273.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, **99**(465), 156–168.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings-F*, **140**(2), 107–113.

Gupta, A., Mittal, A., and Davis, L. S. (2007). Constraint integration for efficient multiview pose estimation with self-occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(3), 493–506.

Hua, G. and Wu, Y. (2007). A decentralized probabilistic approach to articulated body tracking. *Computer Vision and Image Understanding*, **108**(2), 272–283.

Husz, Z. and Wallace, A. (2007). Evaluation of a hierarchical partitioned particle filter with action primitives. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM)*.

Isard, M. and Blake, A. (1998). CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, **29**(1), 5–28.

Kanaujia, A., Sminchisescu, C., and Metaxas, D. (2007). Semi-supervised hierarchical models for 3D human pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Klaas, M., Briers, M., de Freitas, N., Doucet, A., Maskell, S., and Lang, D. (2006). Fast particle smoothing: If I had a million particles. In *Proceedings of the International Conference on Machine Learning*, pages 481–488.

Lee, C.-S. and Elgammal, A. (2006). Body pose tracking from uncalibrated camera using supervised manifold learning. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM)*.

Lee, M. W. and Nevatia, R. (2005). Integrating component cues for human pose tracking. In *Proceedings of the Joint Workshop on Visual Surveillance and Performance and Evaluation of Tracking Systems (VS-PETS)*.

Lee, M. W., Cohen, I., and Jung, S. K. (2002). Particle filter with analytical inference for human body tracking. In *IEEE Workshop on Motion and Video Computing*, pages 159–165.

Mikić, I., Trivedi, M., Hunter, E., and Cosman, P. (2001). Articulated body posture estimation from multi-camera voxel data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 455–460.

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

Mündermann, L., Corazza, S., and Andriacchi, T. (2007). Accurately measuring human movement using articulated ICP with soft-joint constraints and a repository of articulated models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Neal, R. and Hinton, G. (1998). A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Press.

Peursum, P. (2006). On the behaviour of the annealed particle filter in realistic conditions. Technical report, Curtin University of Technology. http://impca.cs.curtin.edu.au/pubs/reports.php.

Peursum, P. (2008). Variational and Gibbs inference for generative human body tracking. Technical report, Curtin University of Technology. http://impca.cs.curtin.edu.au/pubs/reports.php.

Peursum, P., Venkatesh, S., and West, G. (2007). Tracking-as-recognition for articulated full-body human motion analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Poon, E. and Fleet, D. J. (2002). Hybrid Monte Carlo filtering: Edge-based people tracking. In *IEEE Workshop on Motion and Video Computing*.

Sidenbladh, H., Black, M. J., and Fleet, D. J. (2000). Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of the European Conference on Computer Vision*, pages 702–718.

Sigal, L. and Black, M. J. (2006). HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University.

Sigal, L., Bhatia, S., Roth, S., Black, M. J., and Isard, M. (2004). Tracking loose-limbed people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 421–428.

Sminchisescu, C. and Jepson, A. (2004). Variational mixture smoothing for non-linear dynamical systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 608–615.

Sminchisescu, C. and Triggs, B. (2001). Covariance scaled sampling for monocular 3D body tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 447–454.

Sminchisescu, C. and Triggs, B. (2003). Kinematic jump processes for monocular 3D human body tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 69–76.

Sminichisescu, C., Kanaujia, A., and Metaxas, D. (2006). Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding*, **104**, 210–220.

Smith, A. W. and Lovell, B. C. (2006). Measurement function design for visual tracking applications. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 789–792.

Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 747–757.

Taycher, L., Shakhnarovich, G., Demirdjian, D., and Darrell, T. (2006). Conditional random people: Tracking humans with CRFs and grid filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–229.

Urtasun, R., Fleet, D. J., Hertzmann, A., and Fua, P. (2005). Priors for people tracking from small training sets. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 403–410.

Urtasun, R., Fleet, D. J., and Fua, P. (2006). Temporal motion models for monocular and multiview 3D human body tracking. *Computer Vision and Image Understanding*, **104**, 157–177.

Vermaak, J., Doucet, A., and Pérez, P. (2003). Maintaining multi-modality through mixture tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1110–1116.

Vondrak, M., Sigal, L., and Jenkins, O. (2008). Physical simulation for probabilistic motion tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Winn, J. and Bishop, C. (2005). Variational message passing. *Journal of Machine Learning Research*, **6**, 661–694.

Acknowledgements This research is supported by Australian Research Council grant LP0561867. We would also like to thank the anonymous reviewers for their valuable comments and suggestions.

REVIEWER 1 COMMENTS

In section 3.1 at the end of the overview of particle filtering the authors correctly note that resampling at every time yields the SIR algorithm. It may also be worth noting that in the tracking literature, this algorithm was introduced as CONDENSATION by Isard and Blake (IJCV, 1998)

Response We have included a sentence mentioning and citing some of the special cases of sequential Monte Carlo that have been independently developed over the years, including CONDENSATION and the bootstrap filter. This discussion is located at the beginning of Section 3.1.

The only particularly egregious problem is in section 4.5 on Gibb's sampling. On lines 38-41 the authors claim that "The order in which states are processed is not an issue since sampling is based only on the previous round's samples..." However, this is not true of Gibb's sampling in general, see for instance the description of Gibbs sampling in Andrieu et al, 2003

Response We have corrected the description since it was definitely incorrect, and we thank the reviewer for pointing it out.

Finally, one comment concerning the effectiveness of smoothing in general and the FBS method in particular. ... Frequent resampling causes the approximation of $P(x_t|y_{1:t+p})$ computed by the FBS to quickly collapse to a single point as p increases. In practice this can happen even with seemingly modest values of p, say 5-10. One interesting statistic to monitor is the survival rate of particles over a period of time, i.e., how many unique particles at time t have survived until time $t + p$

Response This is a valid point, and we have accordingly added a discussion of the role that degeneracy and resampling plays in limiting smoothing's performance (Section 6.2, heading "Where Smoothing Fails"). This also required describing the effective sample size earlier in the paper (Section 3.1, last paragraph of Overview).

OTHER CHANGES

In addition to the above comments, we have also made the following minor changes to the manuscript:

- Improved the clarity of the bullet points summarising the contributions of the paper (Introduction, top of page 2).
- Corrected the URL in footnote 1.
- Added IEEE copyright notices to Figures 3 and 4.
- Improved the description of how we generated the synthetic video in Section 7.2.
- Corrected various minor grammar and typo errors.
