

Accepted Manuscript

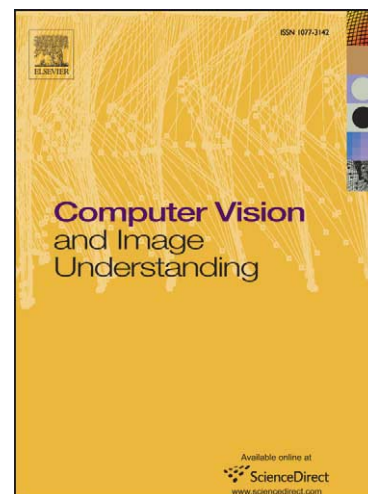
Real-Time 3-D Human Body Tracking using Learnt Models of Behaviour

Fabrice Caillette, Aphrodite Galata, Toby Howard

PII: S1077-3142(07)00083-5
DOI: [10.1016/j.cviu.2007.05.005](https://doi.org/10.1016/j.cviu.2007.05.005)
Reference: YCVIU 1372

To appear in: *Computer Vision and Image Understanding*

Received Date: 14 September 2005
Revised Date: 8 July 2006
Accepted Date: 16 May 2007



Please cite this article as: F. Caillette, A. Galata, T. Howard, Real-Time 3-D Human Body Tracking using Learnt Models of Behaviour, *Computer Vision and Image Understanding* (2007), doi: [10.1016/j.cviu.2007.05.005](https://doi.org/10.1016/j.cviu.2007.05.005)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Real-Time 3-D Human Body Tracking using Learnt Models of Behaviour

Fabrice Caillette^a Aphrodite Galata^{a,*} and Toby Howard^a

^a*Advanced Interfaces Group, School of Computer Science,
University of Manchester, Manchester M13 9PL, UK*

Abstract

In this paper, we introduce a 3-D human-body tracker capable of handling fast and complex motions in real-time. We build upon the Monte-Carlo Bayesian framework, and propose novel prediction and evaluation methods improving the robustness and efficiency of the tracker. The parameter space, augmented with first order derivatives, is automatically partitioned into Gaussian clusters each representing an elementary motion: hypothesis propagation inside each cluster is therefore accurate and efficient. The transitions between clusters use the predictions of a variable length Markov model which can explain high-level behaviours over a long history. Using Monte-Carlo methods, evaluation of model candidates is critical for both speed and robustness. We present a new evaluation scheme based on hierarchical 3-D reconstruction and blob-fitting, where appearance models and image evidences are represented by mixtures of Gaussian blobs. Our tracker is also capable of automatic initialisation and self-recovery. We demonstrate the application of our tracker to long video sequences exhibiting rapid and diverse movements.

Key words: Real-Time, Human-Body Tracking, Variable Length Markov Models, Bayesian, Monte-Carlo, Volumetric Reconstruction, Visual-Hull, Blobs, Entropy, Kullback-Leibler

1 Introduction

Full human-body tracking has a wide and promising range of applications. Movements and gestures are essential vehicles of communication that can be used to interact with computers in a more natural and expressive way than current computer-centred devices. The domain of computer interfaces could be reshaped by gesture-based interactions, allowing users to interact freely with virtual objects. Video games are an obvious example of application which would greatly benefit from body tracking to enhance the immersion of players. Likewise, tracking motions can be used to control realistic avatars in virtual environments.

With motion analysis, computers could assess the recovery of patients and help sportsmen improve their performances. Computers could also become virtual teachers in activities such as dancing or sign-language, capable of both instructing students and correcting their errors. Last but not the least, the film industry has been pioneering the need for motion capture since the emergence of realistic computer graphics. The capture and re-targeting of the movements of actors towards animated characters is a very important application, used not only in films, but also in video games and in live broadcasts.

Tracking people is difficult because of the high dimensionality of full body kinematics, fast movements and frequent self-occlusions. Moreover, loose cloth-

* Corresponding author.

Email address: a.galata@cs.man.ac.uk (Aphrodite Galata).

ing, shadows or camera noise may further complicate the inference problem. Despite a very high level of interest in the computer-vision community, the general human-body tracking problem remains largely unsolved and current markerless trackers still cannot compete in accuracy and robustness with commercial motion capture systems.

In this paper, we present a full body tracker based on a Monte-Carlo Bayesian framework. Real-time tracking of challenging human motions is made possible by novel prediction and evaluation schemes. We use a high-order temporal model (which we learn automatically) for propagating particles. Our novel prediction scheme is based on Variable length Markov models (VLMs) that can efficiently encode local dynamics as well as long temporal dependencies. Our novel evaluation scheme is based on volumetric reconstruction and blob-fitting and allows a large number of model candidates to be tested in a very efficient manner. The tracker is also capable of self-initialisation and recovery from tracking failures by using the motion prototypes as new starting points.

After reviewing related work and motivating our approach (Section 2), we introduce in Section 3 our human body model and the Monte-Carlo Bayesian tracking framework. In Section 4, we show how complex movements are decomposed into clusters of elementary motions, and how high-order behaviours are learnt over these clusters. A predictive scheme which utilises the learnt behaviour model to efficiently propagate particles within the Bayesian framework is presented in Section 5. In Section 6, we introduce a method for fast evaluation of the particles, and finally, Sections 7 and 8 respectively present some results and discussion.

2 Related Work

Tracking is a global optimisation process. Because of kinematic constraints, even relatively independent limbs must compete to fit onto their own detected features (image evidence). Hierarchical methods [6,28] fit the torso in a first stage and then optimise each limb independently. The parameter space is then partitioned, which drastically reduces the complexity of inference. However, problems occur when the torso cannot accurately be located on its own. This can be the case in human body tracking because of self-occlusions, or simply measurement noise.

One approach to tracking as a global optimisation problem is to start from image data, trying to detect features independently in each frame. The configuration of the model is then recovered from the “bottom-up” [30], using nonparametric belief propagation techniques. Since the feature detectors will inevitably return many false positives, the configuration of the model is globally optimised by iterating belief propagation in a graph with strong kinematic and temporal priors [35]. While these techniques are theoretically appealing, they rely on the detection of specific features, which is not always possible because of occlusions or loose clothing. Additionally, the computational complexity of the method is currently too high for real-time applications.

Alternatively, one can use the body configuration in the current frame and a dynamic model to predict the next configuration candidates (*motion prior*). These candidates are then tested against image evidence to find the most likely configuration. Tracking with particle filters works along those lines, approximating the *posterior* distribution by a set of representative elements, and up-

dating these particles with Monte Carlo importance sampling [20]. However, in full body tracking problems, the dimensionality of the parameter space is far too high to represent accurately the true posterior distribution everywhere. Instead, particles tend to concentrate in only a few of the most significant modes, leading to possible failures when too few particles are propagated to represent a new peak. A common solution is to reposition particles based on some importance function. Sullivan and Rittscher [37] and Sminchisescu and Triggs [36] used a deterministic local search to localise the set of particles around significant maxima of the importance function. Deutscher et al. proposed annealed particle filtering [13] which is a coarse to fine approach that can help focusing the particles on the global maxima of the posterior, at the price of multiple iterations per frame. Alternatively, sophisticated motion prior models have been proposed [34], trying to predict the subject's dynamics and propagating particles around the next expected peaks of the posterior.

Prediction is hard because human dynamics are complex and highly non-linear. Models of linear dynamics such as Kalman filters suffice to predict simple linear motions, but for faster, more complex non-linear movements, a better predictive model is required. Projecting the parameters of the model onto a lower dimensionality manifold [21] encodes implicitly the correlations between parameters, and makes linear prediction methods efficient again. Such methods have shown to predict successfully the walking cycle using Autoregressive Models [1]. However, problems reappear with long sequences of complex motions, where the parameters are not sufficiently correlated to give good predictions under projection.

The main performance bottleneck when using Monte-Carlo methods is the evaluation of the likelihood function. For each particle, it usually involves

generating a 3-D appearance model from the particle state, projecting this appearance model onto the available image planes, and finally comparing it with some extracted image features such as silhouettes or edges. Various simplifications and optimisations [7] have been attempted, but none of them were able to make full use of image information in real-time.

Tracking using a particle filter in a Bayesian framework has a number of advantages when compared to earlier systems [31,32,18] that perform tracking through the optimization of a cost function. The formal statistical foundations of a bayesian approach ensure that the estimate of the current pose is optimal given all available observations. When confronted with ambiguous image evidences, a particle-filter based tracker naturally tracks multiple hypothesis until disambiguation. Finally, particle filters are very flexible and scalable: the number of particles can be adjusted to match the requirements of the system, and the computation is easily parallelized over multi-core CPUs, or different computers.

In this work we address two of the pitfalls commonly associated with particle filters, namely the high computational cost and the lack of recovery process. We also show that robustness and accuracy benefit from the predictions of a learnt high-order dynamic model.

3 Tracking Framework

In this section we describe the parametrisation of the human body model as well as the features we use to learn the predictive model that will constrain the search within the Bayesian tracking framework.

3.1 Kinematic Tree and Constraints

The model of the human body is based on a kinematic tree consisting of 14 segments, as seen in Figure 1. Each pose is represented by a 25-dimensional vector \mathbf{C}_t which consists of the joint angles, as well as the position and orientation of the root of the kinematic tree.

Constraints are placed on joint rotations (expressed as Euler angles) in the form of bounding values. Redundant configurations and singularities are eliminated by limiting each joint to two degrees of freedom. The constraints restrict the number of impossible poses, but are insufficient to capture the complexity of human morphological constraints. More advanced constraints schemes have been proposed [24,12]. In our case, however, a high level behaviour model learnt from training sets of 3-D human motions (e.g., joint angles over time) will implicitly play the same role.

3.2 Feature space representation

In order to learn a concise probabilistic model of 3-D human motion, we need to choose an appropriate feature space. For each body pose, we define a corresponding feature vector $\mathbf{X}_t = (\mathbf{x}_t, \dot{\mathbf{x}}_t)$ consisting of the joint angle vector \mathbf{x}_t and its first derivative $\dot{\mathbf{x}}_t$. Global position and orientation are omitted from the chosen feature representation as we do not wish the learnt behaviour model to be sensitive to them. The inclusion of derivatives helps resolve ambiguities in configuration space. Moreover, it facilitates the use of models in performing generative tasks using local dynamics (see Section 5.2).

Human body behaviour may be viewed as a smooth trajectory within the feature space that is sampled at frame rate, generating a sequence of feature vectors \mathbf{X}_t . Each sequence describes the temporal evolution of human body poses, augmented by the first derivatives of the joint angles: $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m\}$.

3.3 Bayesian Tracking Framework

Using Bayes' rule, the filtered probability of a model configuration \mathbf{C}_t given all available measurements $\mathbf{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t\}$ up to time t is:

$$\underbrace{P(\mathbf{C}_t|\mathbf{Z}_t)}_{\text{Posterior}} = \kappa \cdot \underbrace{P(\mathbf{z}_t|\mathbf{C}_t)}_{\text{Likelihood}} \cdot \int \underbrace{P(\mathbf{C}_t|\mathbf{C}_{t-1})}_{\text{Motion Prior}} \cdot \underbrace{P(\mathbf{C}_{t-1}|\mathbf{Z}_{t-1})}_{\text{Previous posterior}} d\mathbf{C}_{t-1} \quad (1)$$

where $\kappa = \frac{1}{P(\mathbf{z}_t|\mathbf{Z}_{t-1})}$ is a normalising constant. The *posterior* distribution is approximated by a set of discrete particles, each representing a body configuration, as illustrated by Figure 1.

The whole set of particles is resampled at each frame using the Sample Importance Resampling (SIR) algorithm [20], which prevents sample impoverishment, and has a linear complexity with respect to the number of particles. While the distribution of the posterior may well be multimodal, the tracked position at each frame is defined as the global maximum of the posterior (approximated by the particle with highest weight).

In Sections 4 and 5, we shall describe a behaviour-based *motion prior* using VLMMs for prediction. The choice of a VLMM as a mathematical framework for modelling complex non-linear human activities is based on its ability to capture dependencies at variable temporal scales in a simple and efficient manner. Unlike previous work [26,23,8,29] that use first order dynamic models

for prediction, our *motion prior* is a high-order predictive model. The size and complexity of our model is automatically learnt from the training data. We also present a fast way of evaluating the likelihood using volumetric reconstruction and blob-fitting in Section 6.

4 Learning Dynamics

4.1 Clustering the Feature Space

Due to the complexity of human dynamics, we break down complex behaviours into elementary movements for which local dynamic models are easier to infer. The problem is then to automatically find, isolate and model these elementary movements from the training data. We achieve this by clustering the feature space into Gaussian clusters using a variant of the EM algorithm proposed by Figueiredo and Jain [14]. Their proposed method automatically addresses the main pitfalls of traditional EM, that is, the delicate initialisation, the arbitrary choice of the number of components, and the possibility of singularities. Body configurations sampled from a few clusters of ballet-dancing data are shown in Figure 2.

4.2 Learning High-Level Behaviour with VLMMs

Complex human activities such as dancing (or even simpler ones such as walking) can be viewed as a sequence of primitive movements with a high level structure controlling the temporal ordering.

By incorporating probabilistic knowledge of the underlying behavioural structure in the way we propagate our particles, we can explore only the plausible directions of the parameter space. Accurate predictions can drastically reduce the number of required particles. An informed predictive model is also critical for robustness, as poor image evidence can then be disambiguated. A suitable way to obtain such knowledge is variable length Markov models (VLMMs) [33].

Variable length Markov models deal with a class of random processes in which the memory length varies, in contrast to an n -th order Markov models. They have been previously used in data compression [10] and language modelling domains [33,22]. More recently, they have been successfully introduced in the computer vision domain for learning stochastic models of human activities, with applications to behaviour recognition and behaviour synthesis [17,16,3,15].

In this paper we extend our previous work [17,16,15] and utilise the generative capabilities of variable length Markov models for the purpose of improving the robustness and efficiency of object tracking systems. In particular, we integrate annealed particle filtering with a VLMM in such a way that both continuous movements and a discrete representation of structured behaviour are jointly represented. The VLMM in the particle filter effectively constraints the space of plausible body postures and body posture transitions for specific activities.

The advantage of VLMMs over a fixed memory Markov model is their ability to locally optimise the length of memory required for prediction. This results in a more flexible and efficient representation, which is particularly attractive in cases where we need to capture higher-order temporal dependencies in some parts of the behaviour and lower-order dependencies elsewhere. A detailed

description on building and training variable length Markov models is given by Ron *et al.* [33].

A VLMM can be thought of as a probabilistic finite state automaton (PFSA) $\mathcal{M} = (Q, \mathcal{K}, \tau, \gamma, s)$, where \mathcal{K} is a set of tokens that represent the finite alphabet of the VLMM, and Q is a finite set of model states. Each state corresponds to a string in \mathcal{K} of length at most $N_{\mathcal{M}}$ ($N_{\mathcal{M}} \geq 0$), representing the memory for a conditional transition of the VLMM. The transition function τ , the output probability function γ for a particular state, and the probability distribution s over the start states are defined as:

$$\tau : Q \times \mathcal{K} \rightarrow Q \quad \gamma : Q \times \mathcal{K} \rightarrow [0, 1] \quad s : Q \rightarrow [0, 1]$$

An illustration is given in Figure 3(a). The VLMM is a generative probabilistic model: by traversing the model’s automaton \mathcal{M} we can generate sequences of the tokens in \mathcal{K} . By using the set of Gaussian clusters as the alphabet, we can capture the temporal ordering and space constraints associated with the primitive movements. Consequently, traversing \mathcal{M} will generate statistically plausible examples of the behaviour.

Recall that we break down complex behaviours into elementary movements by clustering the feature space into Gaussian clusters (Section 4.1). Next, for each frame in a particular training sequence, we identify which gaussian cluster k_{r_i} the observed model configuration \mathbf{X}_t belongs to. An image sequence is thus represented as a sequence gaussian cluster labels. These sequences are then used for training the VLMM.

The memory size $w_{\mathcal{M}}$ and threshold ε parameters are used to control the actual VLMM construction, depending on the nature of the training data. The

choice of values for these parameters can be based on a measure of how well the learned model describes the training data. One such measure, traditionally used in text compression [2] and language modelling [22,25], is the *model cross-entropy rate* (or *model entropy*) [11,25] \hat{H}_M .

In our case, given a sequence of length n of gaussian cluster labels, an estimate of model entropy for the learnt VLMM model is given by [15]:

$$\hat{H}_M = -\frac{1}{n} \sum_{i=1}^n \log P(k_{r_i} | q_{r_i}). \quad (2)$$

where $P(k_{r_i} | q_{r_i}) = \gamma(q_{r_i}, k_{r_i})$.

Calculating the model entropy (Eq. 2) over the training data gives a measure of how well the learned VLMM describes the training data. A good approximation of observed behaviour is indicated by a low model entropy value. In the general case, an increase in the value of ε will result in an increase of the model's entropy whereas an increase on the model's maximum memory w_M will decrease the model entropy (however, increasing w_M does not always guarantee lower model entropy, see [22,25] and [15] for details.)

The performance of the model can then be measured by the model entropy over the test data. A model that has achieved a good generalisation of the observed behaviour, will have very close model entropy values over both the training and test data. Otherwise, significantly different values indicate a model that is overfitted to the training data and thus more training is required.

5 Predictions using the Dynamic Model

Particles are accurately propagated using both the VLMM for high-level predictions and the Gaussian clusters for local dynamics. Figure 3 gives a simplified overview of this prediction scheme.

5.1 Particles Transitions Between Clusters using the learnt VLMM

The particles are augmented with their current VLMM state q_t , from which the cluster k_t they belong to is easily deduced. Transitions (or jumps) between clusters are conditional on the particle's feature vector \mathbf{X}_t as well as the transition probabilities γ in the VLMM. The probability of transition towards a new Gaussian cluster k_{t+1} of mean $\mu_{k_{t+1}}$ and covariance $\Sigma_{k_{t+1}}$ is:

$$\begin{aligned}
 P(k_{t+1} \mid \mathbf{X}_t, q_t) &\propto P(\mathbf{X}_t \mid k_{t+1}) \cdot P(k_{t+1} \mid q_t) \\
 &= \frac{1}{\sqrt{(2\pi)^d |\Sigma_{k_{t+1}}|}} \cdot e^{-\frac{1}{2} \cdot (\mathbf{X}_t - \mu_{k_{t+1}})^T \cdot \Sigma_{k_{t+1}}^{-1} \cdot (\mathbf{X}_t - \mu_{k_{t+1}})} \cdot \gamma(q_t, k_{t+1})
 \end{aligned} \tag{3}$$

At each frame, the state transition is chosen according to the above probabilities for each neighbouring cluster. In practice, only a few transitions are encoded in the VLMM, making the evaluation efficient. If the same cluster is chosen ($k_{t+1} = k_t$), the particle is propagated using local dynamics, as formulated in the next section. If a new cluster is selected, the particle's parameters are re-sampled from the new Gaussian cluster.

5.2 Local Dynamics

Inside each Gaussian cluster, a new model configuration can be stochastically predicted from the previous feature vector \mathbf{X}_t . Since the Gaussian clusters include derivatives, the prediction effectively behaves like a second-order model. Let us consider a Gaussian cluster of mean $\mu = \begin{pmatrix} \mu_X \\ \mu_{\dot{X}} \end{pmatrix}$ and covariance matrix $\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{X\dot{X}} \\ \Sigma_{X\dot{X}}^T & \Sigma_{\dot{X}\dot{X}} \end{pmatrix}$. The noise vector is directly sampled from the cluster's covariance matrix with an attenuation coefficient λ , leading to the formulation:

$$\begin{aligned} \dot{\mathbf{x}}_t &= \dot{\mathbf{x}}_{t-1} + \lambda \cdot d\dot{\mathbf{x}}_t \\ \mathbf{x}_t &= \mathbf{x}_{t-1} + \dot{\mathbf{x}}_t + \lambda \cdot d\mathbf{x}_t \end{aligned} \quad \text{with} \quad \begin{pmatrix} d\mathbf{x}_t \\ d\dot{\mathbf{x}}_t \end{pmatrix} \sim \mathcal{N}(0, \Sigma) \quad (4)$$

The random noise vector is drawn as $\begin{pmatrix} d\mathbf{x}_t \\ d\dot{\mathbf{x}}_t \end{pmatrix} = \sqrt{\Sigma} \cdot X$ with $X \sim \mathcal{N}(0, I)$. The square-root of the covariance matrix is computed by performing the eigenvalue decomposition, $\Sigma = V \cdot D \cdot V^T$, and taking the square root of the eigenvalues on the diagonal of D , so that $\sqrt{\Sigma} = V \cdot \sqrt{D} \cdot V^T$.

This predictive model should be understood in the context of Monte-Carlo sampling, where noise is introduced to model uncertainty in the prediction: the properties of the noise vector are therefore almost as important as the dynamics themselves. The covariance matrix of the current cluster provides a good approximation of this uncertainty, and sampling the noise vector from the cluster itself makes propagation of uncertainty much closer to the training data than uniform Gaussian noise. Kinematic constraints are also implicitly encoded because particles are only propagated in valid directions of the parameter space, as learnt from the training data.

To keep the behaviour model independent of the global position and orientation of the subject, the six global parameters are not modelled by the Gaussian clusters, and are therefore propagated with uniform Gaussian noise.

6 Fast Evaluation of the Likelihood

Using the dynamic model introduced in Section 5, the particles are propagated in the parameter space. As new observations \mathbf{z}_t become available at each new frame, the particles are re-weighted according to the likelihood function. This is usually done by generating an appearance model for each particle, and comparing it to image evidence. When dealing with large amounts of particles, the computational cost involved with this evaluation process can easily become prohibitive.

This section introduces an efficient scheme to evaluate the particles against image evidence, by adopting a compact common representation for both the candidate model configurations and the current observation \mathbf{z}_t . As illustrated by Figure 4, we first perform a hierarchical volumetric reconstruction to merge information from all views and resolve spatial ambiguities. The data is then summarised by fitting a mixture of Gaussian blobs, using the predictions of the dynamic model. Finally, as a candidate appearance model is generated for each particle, it is evaluated against the current observation. Both the appearance model generated for the particle and the current observation are mixtures of non-overlapping Gaussian blobs. The two mixtures are efficiently compared using a closed form solution of their relative entropy.

6.1 Volumetric Reconstruction

Our volumetric reconstruction method follows the shape-from-silhouette paradigm where the visual-hull of the subject is defined as the maximal volume explained by all silhouettes. Background subtraction is a central piece of all shape-from-silhouette methods, but is often performed as a binary segmentation on each camera view. Our contribution consists in merging silhouette extraction and volumetric reconstruction into a single hierarchical algorithm. Our algorithm uses robust pixel statistics on sets of pixel samples, allowing the use of unconstrained (cluttered) environments, and improving computational efficiency compared to previously reported reconstruction methods [9,28,4]. We also recover colour information, making the reconstructed volume an appropriate basis for tracking.

As illustrated in Figure 5, the tracking space is initially subdivided into a coarse voxel grid (typically $16 \times 16 \times 16$). Each voxel is successively projected onto the available image planes, and sets of pixels are uniformly sampled from the corresponding projected areas. In our implementation, the number of pixel samples k is the square root of the number of pixels covered by the projected area.

Our algorithm involves an initial stage during which a model of the background is acquired. Background pixels are individually modelled by a full Gaussian distribution in YUV colour-space. Shadows are handled by creating a second Gaussian distribution for each pixel, with a mean shifted by 10% in the luminance (Y) channel. Either the shadowed or unshadowed Gaussian model can therefore be selected by a simple test on the luminance channel of the current

pixel.

The Mahalanobis distance $d_M(s_i)$ between a pixel and the selected Gaussian model follows a Chi-Square (χ^2) distribution with 3 degrees of freedom (dimensionality of the colour-space). If all the k pixels sampled from the projected area of a given voxel are part of the background, the sum of their Mahalanobis distances also follows a χ^2 distribution, but with $3.k$ degrees of freedom. We define a classification threshold $T_k(\alpha)$ as:

$$\int_0^{T_k(\alpha)} \chi_{3k}^2(t) dt = \alpha \quad \Rightarrow \quad P\left(\sum_{i=1}^k d_M(s_i) \leq T_k(\alpha)\right) = \alpha \quad (5)$$

where α is the confidence level wanted for classification (in practice, we choose $\alpha = 0.99$). At runtime, we approximate the thresholds T_k by fitting a second order polynomial in k . The projected area is then classified according to Equation 6, where edges are distinguished from foreground areas using per-sample thresholds.

$$\mathbf{if} \sum_{i=1}^k d_M(s_i) \leq T_k(\alpha) \left\{ \begin{array}{l} \mathbf{then} \quad \mathit{background} \\ \mathbf{else} \quad \mathbf{if} \forall i, d_M(s_i) > T_1(\alpha) \left\{ \begin{array}{l} \mathbf{then} \quad \mathit{foreground} \\ \mathbf{else} \quad \mathit{edge} \end{array} \right. \end{array} \right. \quad (6)$$

The whole voxel is discarded if it is classified as *background* in at least one view, and reconstructed if classified as *foreground* from all views. Otherwise, the voxel is subdivided and the algorithm is recursively applied on each octant until a maximal depth is reached. When a voxel is reconstructed, colour information is added for each view as the mean colour of the pixel samples. Results of reconstruction are shown in Figure 6.

6.2 *Acquiring and Generating Gaussian Blobs*

A coloured Gaussian blob is fully defined by a 6-dimensional mean and a 6×6 covariance matrix. We attach such blobs to the bones of the skeletal model. A mixture of blobs is then readily generated from any set of kinematic parameters using forward kinematics. Blobs are generated in the local coordinate system of each body part, therefore only four free spatial parameters need to be retained: a single offset value which summarises the mean along the first axis of the bone on which the blob is attached, and the three eigenvalues which fully describe the spatial part of the covariance matrix.

The attributes of the blobs are incrementally learnt from the voxel data during the first frames of the tracking process. Since the colour of each blob is unimodal, clothing with multiple colours must be handled by a mixture of blobs. Starting with a single blob for each body-part, a “split and merge” process ensures an optimal description of the data (see Figure 7). The criterion used to decide whether a blob should be split is the colour variance along the main spatial axis of the blob. This measurement is obtained by projecting the 3×3 mixed covariance matrix between spatial and colour information onto the direction of the current bone in the kinematic model, and taking the norm of the resulting vector.

6.3 *Fitting the Gaussian Blobs onto the Voxels with K-Means*

Following the volumetric reconstruction, the large amount of voxel data still prevents the efficient evaluation of the particles. In order to get a representation of the image observations that be can efficiently compared to the model

parameters, we summarise these image observations by fitting the Gaussian blobs introduced in the previous section onto the reconstructed voxels. As illustrated in Figure 8, during the E-step each voxel is first assigned to the closest blob using the Mahalanobis distance on both position and colour. In the M-step, the attributes of the blobs are then re-evaluated from the set of voxels that were assigned to them.

Initialising the blob-fitting from the last tracked model configuration can prove insufficient for fast movements, causing some blobs to converge (“snap”) to incorrect body parts. Also, if the mixture of blobs was simply left to converge from a single initial position, all the benefits of the Bayesian framework would be diminished at this stage, reducing the system to a somewhat complex directed search.

In order to take full advantage of our probabilistic framework, we exploit the learnt behaviour model to initialise the blob-fitting with mixtures of blobs generated from predicted body configurations. To achieve this, we determine the support of each cluster after propagation of the particles by computing the ratio of particles they contain. The means of the clusters with significant support are used as candidate initialisation for the blob-fitting. In practice, only a few clusters are selected so that real-time performance is not threatened. After the blob-fitting from all selected motion prototypes, the mixture of blobs maximising the likelihood of the voxels is retained as the new “image evidence”.

This blob-fitting procedure has the important advantage of detecting tracking failures: if the best mixture of blobs provides a poor likelihood, the tracker is lost and needs re-initialisation. Unlike most other trackers, automatic re-

covery from failures is then possible because the parameter space is clustered in motion prototypes. The VLMM state of all particles is then reset, which has the effect of spreading the particles across the clusters. Performing the blob-fitting from all clusters might provoke a noticeable lag, depending on the total number of motion prototypes, but has a high chance to return a good fit.

Further details about the volumetric reconstruction and the blob fitting process can be found in earlier work [6,5].

6.4 Particle Evaluation with Relative Entropy Measure

A model configuration (particle) is evaluated by first generating an appearance model from the particle state, and then comparing the produced blobs with those obtained from the image evidence. Let us note $F = \sum_i \alpha_i f_i$ the mixture generated from the model and $G = \sum_i \beta_i g_i$ the one corresponding to image evidences. The Kullback-Leibler (KL) divergence can be used to measure the cross-entropy between the two mixtures:

$$D_{KL}(F\|G) = \int F \ln \frac{F}{G} = \sum_i \alpha_i \int f_i \ln F - \sum_i \alpha_i \int f_i \ln G \quad (7)$$

Using the approximation proposed by [19] for non-overlapping clusters:

$$\begin{aligned} D_{KL}(F\|G) &\simeq \sum_i \alpha_i \int f_i \ln \alpha_i f_i - \sum_i \alpha_i \max_j \int f_i \ln \beta_j g_j \\ &= \sum_i \alpha_i \min_j (D_{KL}(f_i\|g_j) + \ln \frac{\alpha_i}{\beta_j}) \end{aligned} \quad (8)$$

Correspondence between blobs is maintained under the form $f_i \leftrightarrow g_{\pi(i)}$, so that the complexity of the run-time evaluation function is linear with respect

to the number of blobs:

$$D_{KL}(F\|G) \simeq \sum_{i=1}^n \alpha_i \left(D_{KL}(f_i\|g_{\pi(i)}) + \ln \frac{\alpha_i}{\beta_{\pi(i)}} \right) \quad (9)$$

This last formulation can be efficiently computed using the closed form solution of the KL divergence between two Gaussian blobs $f \sim \mathcal{N}(\mu_f, \Sigma_f)$ and $g \sim \mathcal{N}(\mu_g, \Sigma_g)$:

$$\begin{aligned} D_{KL}(f\|g) &= \int_{x \in \mathbb{R}^d} P(x|f) \ln \frac{P(x|f)}{P(x|g)} dx \\ &= \frac{1}{2} \left(\ln \frac{|\Sigma_f|}{|\Sigma_g|} - d + \text{tr}(\Sigma_f^{-1} \Sigma_g) + (\mu_g - \mu_f)^T \Sigma_f^{-1} (\mu_g - \mu_f) \right) \end{aligned} \quad (10)$$

where d is the dimensionality of the Gaussian blobs f and g . The weighting of a particle is proportional to the inverse of the relative entropy $D_{KL}(F\|G)$ between the particle and the mixture of blobs corresponding to image evidence. The proportionality factor is unimportant since the weights are normalised before resampling.

7 Results

7.1 Description of the Training and Test Data

Ballet dancing is an interesting application for the evaluation of human-body tracking algorithms because of its diverse body postures, and its fast and challenging choreographies. Ballet dancing is also a structured activity, allowing some predictability in the succession of the movements, therefore making the learning of behaviours patterns possible.

We evaluated our tracking algorithms on sequences performed by ballet dancing students. Our setup was composed of 5 firewire webcams, capturing images

at 30fps in a resolution of 320×240 . The dancers performed 2 exercises, composed of approximately 700 frames each. Our test sequence features the full choreography (2 exercises), while the training sequences consist of 3 repetitions of each dance exercise.

The data used respectively for training and quantitative evaluation were obtained by manual annotation of the video sequences. The 2-D locations of 12 body parts were first annotated for each frame of the sequence. The 3-D locations of the body parts were then computed as a linear optimisation problem, minimising the re-projection error. The trajectories of the body parts were smoothed with a Gaussian kernel and interpolated from with cubic splines. The joint angles parameters were finally recovered using inverse kinematics. By oversampling and varying the amount of smoothing for each training sequence, we obtained a total of 13,000 frames for training.

We automatically clustered the parameter space into clusters of elementary movements, as described in Section 4.1. The optimal number of clusters was found to be 122 for the full sequence (both dance exercises), which can seem quite high but actually reflects the underlying complexity of the motions. As a comparison, the same clustering on a simpler “arms pointing” sequence returned only 5 clusters. We then learnt a VLMM over the Gaussian clusters using various maximal memory lengths. Using a maximal memory length of 10, the VLMM learnt 948 distinct states. This number of states rose to 2890 with a maximal memory length of 30.

7.2 *Qualitative results*

Figure 9 shows the tracking of the first dance exercise, superimposed on one of the 5 input views. Despite poor image evidences, the tracking was successful over the whole sequence.

A subject performing the second dance exercise is tracked in Figure 10. This second dance exercise is particularly challenging because the limbs tend to stay close to the body during fast rotations (pirouette). In this case, important self-occlusions combined with motion blur are taking place, and the reconstructed volume provides poor image evidence. The learnt motion model is fully exploited, providing good initialisations for the blob-fitting procedure, and constraining the poses of the model to the learnt configurations. The ability of the particle filter to keep track of multiple hypothesis is also important for automatic recovery after short periods of ambiguous likelihood function.

7.3 *Accuracy and Robustness*

Using manually annotated test sequences, we present comparative error measurements between our method and other standard algorithms based on particle filters. The CONDENSATION [27] algorithm propagates particles with a Gaussian noise, while Annealing [13] iterates a propagation-evaluation loop over multiple layers, in a “coarse to fine” manner. Having no informed (as opposed to random) predictive model, these two methods are unable to provide a good initialisation for the blob-fitting procedure. Even with 5000 particles evaluations, they both quickly loose track when used in our “blobs evaluation” framework, as illustrated in Figure 11. Our algorithm, however, maintains a

good overall accuracy with only 1000 particles. A momentary tracking failure around frame 420 is automatically detected and recovered from by reinitialising the VLMM.

To keep the comparison focused on the dynamic models, we use the same likelihood distribution for all three algorithms (CONDENSATION, annealing and our method). At each frame, the blob-fitting procedure is initialised from the annotated ground-truth pose of the model. This provides a good, but also realistically noisy, likelihood function for all three algorithms. Results are reported in Figure 12. Even using 5000 particles, CONDENSATION is unable to explore the parameter-space in all appropriate directions, resulting in a poor overall accuracy. The Annealed particle filter uses only 1000 particles, but because of the 5 layers of annealing, the computational cost remains equivalent to CONDENSATION. Annealing produces relatively accurate results in most of the test sequence, although it is still distracted by the noisy likelihood function. Annealing also tends to focus particles on a single mode of the posterior, limiting the ability of the tracker to recover from ambiguous situations. We tested our propagation method with only 200 particles. Despite having 25 times less particle-evaluations than the two other methods, accuracy and robustness were maintained throughout the sequence.

Figure 13 compares the prediction accuracy using various memory lengths for the VLMM and only 200 particles. A memory of 1 frame (first order Markov model) is insufficient to capture the complexity of the succession of movements, and wastes particles by propagating them to the wrong clusters. With a longer memory, the propagation of the particles is more focused, and the overall accuracy is improved.

7.4 *Performance*

Table 1 reports performance measurements on a 2GHz Pentium 4. The maximal recursive depth of the 3-D reconstruction has a strong influence on the processing time because it conditions directly the amount of voxels generated. The performance, however, is not significantly influenced by the number of camera-views used for reconstruction. This is due to the per-sample image segmentation scheme, which leaves large portions of the input images uninspected when voxels are already discarded from a previous view. As expected, the number of particles has a linear influence on performance, but even with 1000 particles, real-time performance is achieved.

7.5 *Scalability Issues*

Although our test sequences exhibit diverse and challenging movements, more tests are needed to confirm the generality and applicability of our method to different types of behaviour. Atomic and cyclic activities, such as walking, should be straightforward to learn because of their simple dynamics. A more interesting and challenging task, however, would be to test the performance of our tracking system when learning multiple and diverse classes of motion; further research is needed to investigate this scenario. In this section, we discuss the potential limitations and benefits of our predictive framework, when confronted with larger sets of motions.

Clustering the parameter space into atomic behaviours will become more computationally expensive as the training data set grows. However, since some atomic movements will be common and recurrent between different types of

behaviour, the number of clusters will not necessarily grow linearly with the size of the data-set. The process of learning transitions between clusters should also scale naturally to a larger dataset. The variable length Markov models are designed to locally optimise their memory length, so the size of the VLMM remains manageable (please refer to Section 4.2 for details).

As the number of clusters increases, the computational cost of learning a VLMM would increase. However, acquiring a VLMM of diverse and complex activities would have an advantage over a corresponding first-order learnt Markov model since the longer history utilised for prediction by the VLMM allows it to differentiate between behaviours and give more accurate predictions. Also, it is worth noting that although the computational cost of learning a VLMM would increase, the computational cost of predicting using the more complex (with respect to the number of states and state connectivity) VLMM would remain largely unaffected. This is due to the fact that the VLMM is conveniently represented by a probabilistic finite state automaton (PFSA): the longest (and optimal) history of atomic behaviours needed for prediction at each time step is actually pre-computed into the states of the PFSA [33,22]. Predicting forward simply involves traversing from one state of the PFSA to the other.

Problems will still appear for movements previously unseen in the training data. In the current implementation of the system, nothing is done to handle these cases, and the tracker has to be re-initialised. As long as the total number of clusters remains significantly lower than the number of particles, this simple re-initialisation of the VLMM works well. However, for larger pools of diverse movements, a simple re-initialisation can prove both computationally expensive and inefficient. Unfortunately, switching back to a stochastic prop-

agation method is not a viable option, as we demonstrated in this chapter that such methods are incapable of exploring the whole parameter space. This difficult problem is left open for future research.

8 Discussion

The main challenge in human-body tracking is the high dimensionality of the parameter space, making the search for the correct pose a hard problem. Using Monte-Carlo methods, the number of required particles tends to become very large, and even if methods such as annealing improve convergence, the computational cost remains too high for real-time applications.

In this paper, we have demonstrated an algorithm using high-level behaviours to track challenging movements in real-time. Contributions reside in the prediction scheme which uses VLMMs and in a fast evaluation method based on volumetric reconstruction and blob-fitting. By focusing the propagation of particles towards predicted directions, the number of particles required for robust tracking is kept low, and in conjunction with a fast evaluation scheme, real-time performance is achieved on commodity hardware.

The work presented in this paper was primarily targeted at human-computer interaction and motion capture setups. Tracking multiple subjects simultaneously has not yet been fully investigated. Given that our current volumetric reconstruction method is based on a shape-from-silhouette algorithm, we expect that partial occlusions and occlusions between multiple objects will pose problems to the current implementation of our tracking system. Increasing the number of cameras would limit the generated occlusions, but in the general

case, further research is needed to extend our system to handle these kind of scenarios.

As future research directions, we intend to investigate and evaluate various dimensionality reduction methods, in an effort to make the learning of clusters more efficient. Online learning, where unseen sequences are incrementally integrated into the behaviour model, would also represent a worthy contribution.

References

- [1] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *Proc. ECCV*, volume 3023, pages 54–65, 2004.
- [2] T. Bell, J. Cleary, and I. Witten. *Text Compression*. Prentice Hall, 1990.
- [3] F. Bettinger and T.F. Cootes. A model of facial behaviour. In *Proc. Int. Conference on Automatic Face and Gesture Recognition*, pages 123–128, 2004.
- [4] E. Borovikov, A. Sussman, and L. Davis. A high performance multi-perspective vision studio. In *Proc. Annual ACM Int. Conf. on Supercomputing*, 2003.
- [5] F. Caillette and T. Howard. Real-Time Markerless Human Body Tracking Using Colored Voxels and 3-D Blobs. In *Proc. ISMAR*, pages 266–267, Nov. 2004.
- [6] F. Caillette and T. Howard. Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction. In *Proc. BMVC*, volume 2, pages 597–606, 2004.
- [7] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel. Free-viewpoint video of human actors. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 569–577,

2003.

- [8] T. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. Computer Vision and Pattern Recognition (CVPR'99)*, pages 239–245, 1999.
- [9] K. M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. CVPR*, volume 2, pages 714–720, 2000.
- [10] G. Cormack and R. Horspool. Data Compression using Dynamic Markov Modelling. *Computer Journal*, 30(6):541–550, 1987.
- [11] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.
- [12] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proc. ICCV*, pages 1071–1078, 2003.
- [13] J. Deutscher, A. Blake, and I. D. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. CVPR*, volume 2, pages 126–133, 2000.
- [14] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on PAMI*, 24(3):381–396, 2002.
- [15] A. Galata, A. G. Cohn, D. Magee, and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *Proc. European Conference on Artificial Intelligence (ECAI'02)*, pages 741–745, 2002.
- [16] A. Galata, N. Johnson, and D. Hogg. Learning Variable Length Markov Models of Behaviour. *Computer Vision and Image Understanding*, 81(3):398–413, 2001.
- [17] A. Galata, N. Johnson, and D. Hogg. Learning structured behaviour models using variable length Markov models. In *IEEE International Workshop on Modelling People*, pages 92–102, 1999.

- [18] D.M. Gavrila and L.S. Davis. 3-d model-based tracking of humans in action : a multi-view approach. In *Proc. Computer Vision and Pattern Recognition (CVPR'96)*, pages 73–80, 1996.
- [19] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. In *Proc. ICCV*, pages 487–493, 2003.
- [20] N. Gordon, J. Salmond, and A. Smith. Novel approach to non-linear/non-gaussian bayesian state estimation. In *Radar and Signal Processing*, pages 107–113, 1994.
- [21] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):522–531, 2004.
- [22] I. Guyon and F. Pereira. Design of a Linguistic Postprocessor using Variable Memory Length Markov Models. In *ICDAR*, pages 454–457, 1995.
- [23] T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. *Proc. 6th Int. Conf. on Computer Vision*, pages 344–349, 1998.
- [24] L. Herda, R. Urtasun, and P. Fua. Hierarchical implicit surface joint limits to constrain video-based motion capture. In *Proc. ECCV*, volume 2, pages 405–418, 2004.
- [25] J. Hu, W. Turin, and M Brown. Language Modelling using Stochastic Automata with Variable Length Contexts. *Computer Speech and Language*, 11(1):1–16, 1997.
- [26] M. Isard and A. Blake. A mixed-state Condensation tracker with automatic model-switching. *Proc. 6th Int. Conf. on Computer Vision*, pages 107–113, 1998.

- [27] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. In *International Journal of Computer Vision*, volume 29(1), pages 5–28, 1998.
- [28] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *IJCV*, 53(3):199–223, 2003.
- [29] V. Pavlovic, J. M. Rehg, T. Cham, and K. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamical models. In *Proc. International Conference on Computer Vision*, 1999.
- [30] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *Proc. CVPR*, volume 2, pages 467–475, 2003.
- [31] J. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. *Third European Conf. on Computer Vision*, pages 35–46, 1994.
- [32] J. Rehg and T. Kanade. Model-based tracking of self occluding articulated objects. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 612–617, 1995.
- [33] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25(2–3):117–149, 1996.
- [34] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV*, volume 1, pages 784–800, 2002.
- [35] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Proc. CVPR*, volume 1, pages 421–428, 2004.
- [36] C. Sminchisescu and B. Triggs. Covariance-scaled sampling for monocular 3d body tracking. In *Proc. Computer Vision and Pattern Recognition (CVPR'01)*, pages 447–454, 2001.

- [37] J. Sullivan and J. Rittscher. Guiding random particles by deterministic search.
In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 323–330, 2001.

ACCEPTED MANUSCRIPT

	3-D Reconstruction			Blobs	Particle Filter	
	3 views	4 views	5 views	Fitting	500	1000
Max. Depth 6	11.0ms	11.1ms	12.8ms	3.1ms		
Max. Depth 7	24.1ms	22.8ms	28.1ms	14.8ms	20.7ms	45.3ms
Max. Depth 8	73.5ms	66.0ms	78.6ms	74.1ms		

Table 1

Performance measurements

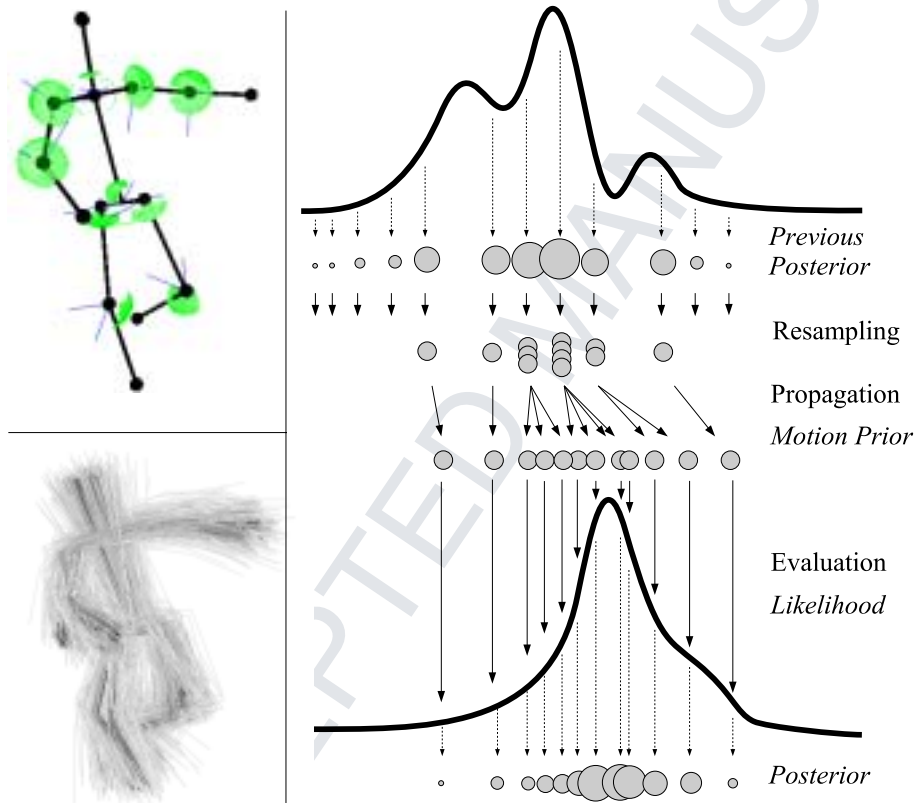


Fig. 1. Kinematic model and Bayesian tracking framework. (top-left) The kinematic model with the joint constraints and (bottom-left) the set of weighted particles approximating the posterior. (right) The particles approximating the distribution of the posterior in the previous frame are successively resampled, propagated and evaluated. The new set of particles approximates the posterior for the new frame.

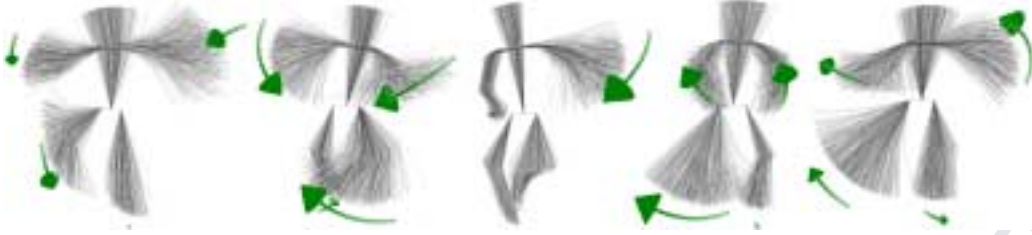


Fig. 2. Model configurations sampled from selected learnt Gaussian clusters. The mean of the first derivatives is represented with a green arrow at the hands and the feet. Note that the training data for head movements were not available.

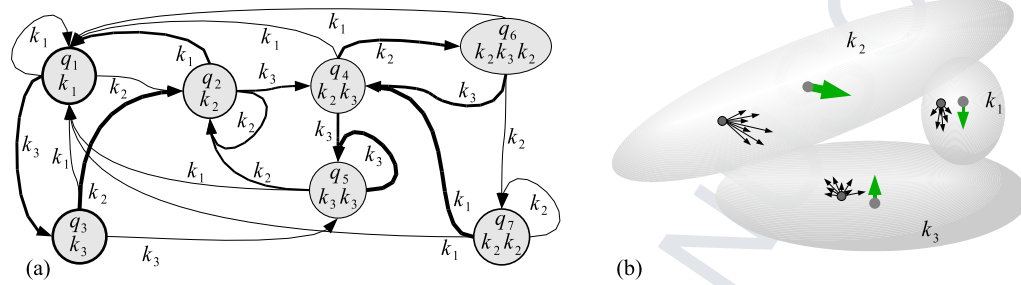


Fig. 3. (a) Example of VLMM over an alphabet $\mathcal{K} = \{k_1, k_2, k_3\}$ with maximal memory $N_{\mathcal{M}} = 3$. The transition probabilities γ and initial probabilities s are represented by the width of the arcs. (b) Local dynamics inside the Gaussian clusters (corresponding to the ones visualised in Figure 2) where particles are propagated with the uncertainty modelled by the cluster itself.

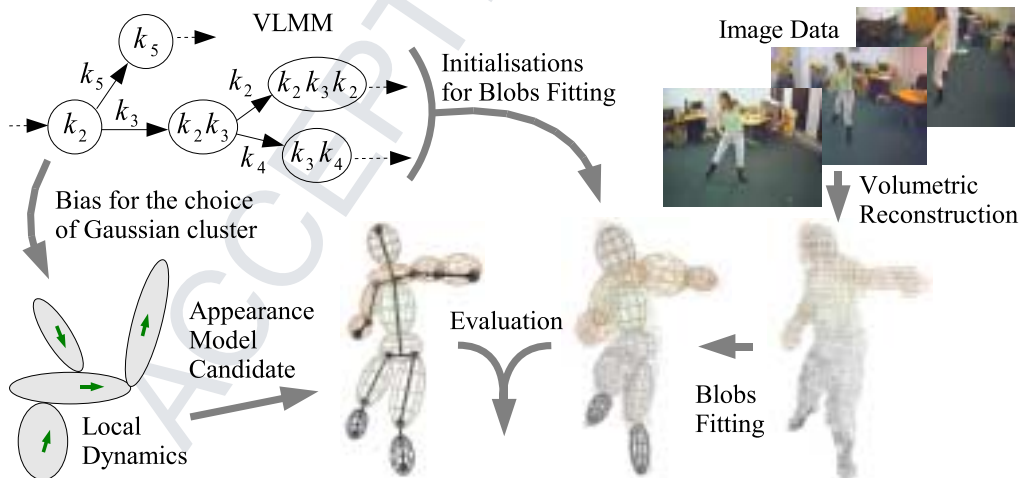


Fig. 4. Overview of the evaluation process.

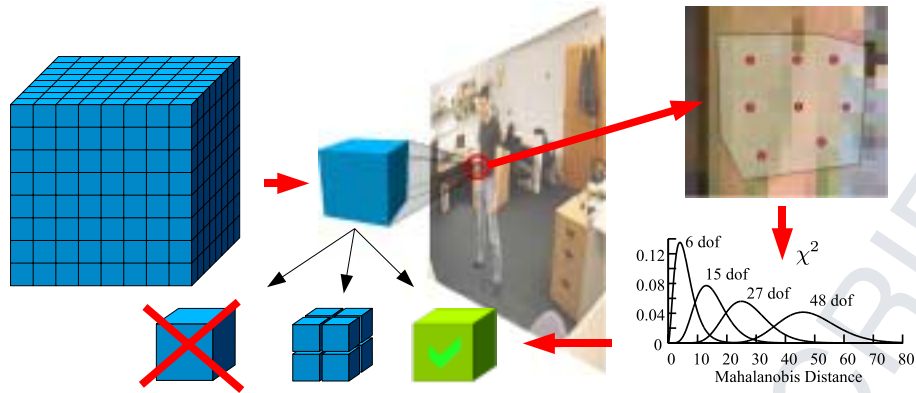


Fig. 5. Volumetric reconstruction: starting from a coarse subdivision of the tracking space, each voxel is projected onto the image planes and pixel samples are uniformly sampled from the projected area. The voxel is then robustly classified using a statistical distance for sets of samples.

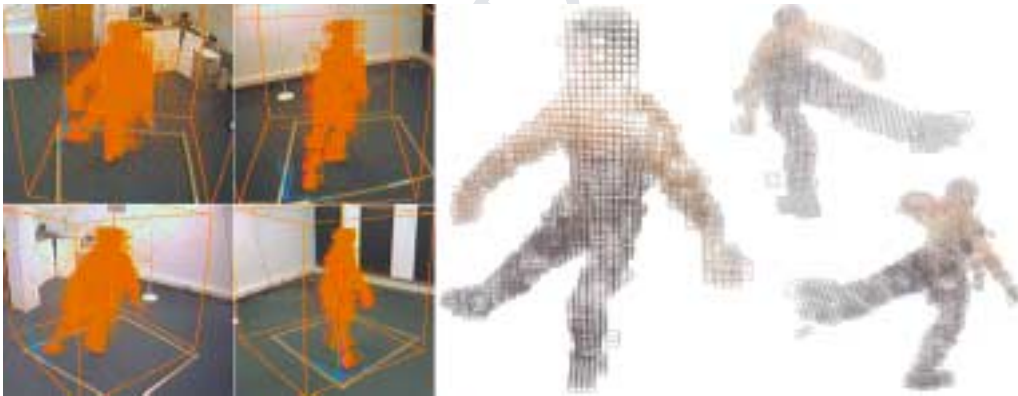


Fig. 6. Volumetric reconstruction from 4 camera views with (left) the pixel samples used during the reconstruction process and (right) the voxel-based volume from 3 arbitrary viewpoints.



Fig. 7. Automatic acquisition of the blobs models over 20 frames. Note that the initial global position and orientation of the model are computed from the mean and the principal axes of the voxels distribution.

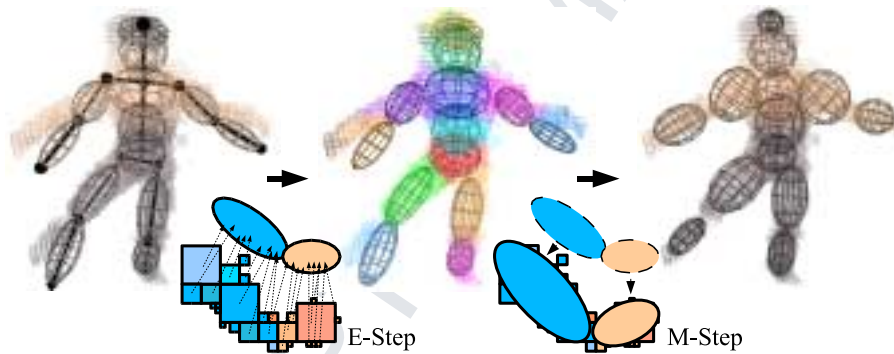


Fig. 8. Fitting Gaussian blobs onto the voxels data with K-Means. The algorithm is initialised with a predicted model configuration, for which a mixture of blobs is generated. In the E-step, voxels are assigned to the nearest blob using Mahalanobis distances between blobs and voxels on both colour and position. The means and covariances of the blobs are re-evaluated in the M-Step from the set of voxels previously assigned to them.

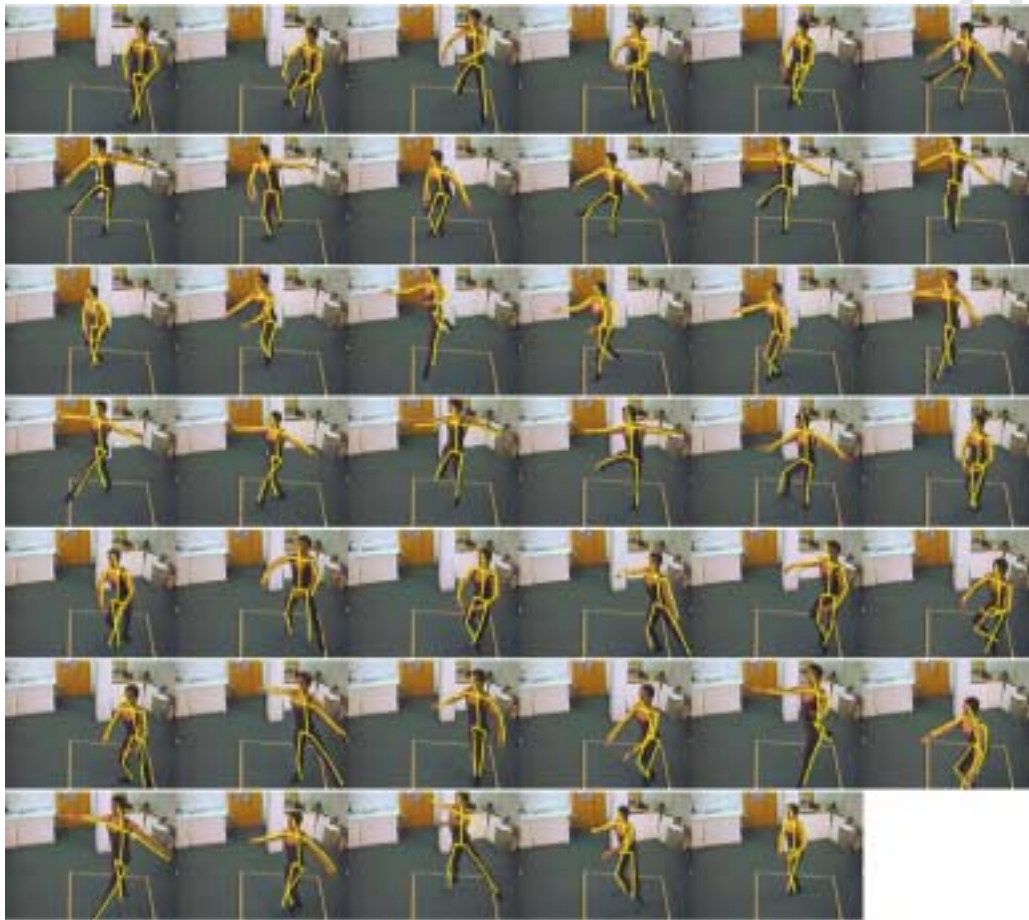


Fig. 9. Tracking the first dance exercise using a VLMM with a maximal history of 20 frames and 1000 particles. The images shown are sampled approximately every 300ms.



Fig. 10. Tracking the second dance exercise using a VLMM with a maximal history of 20 frames and 1000 particles.

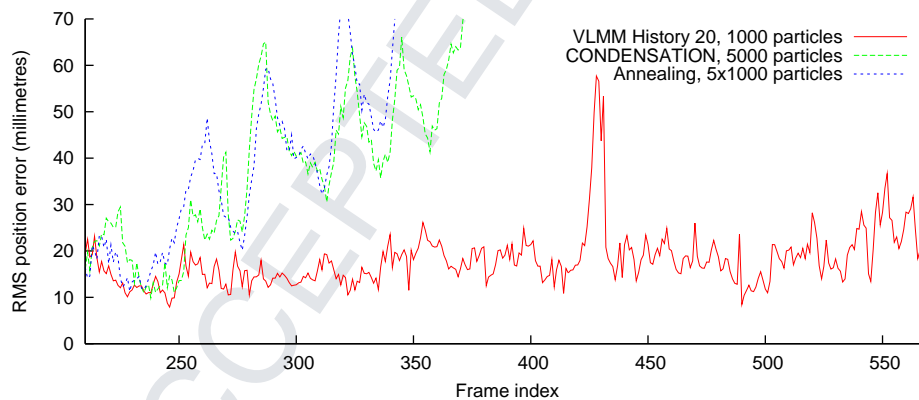


Fig. 11. Accuracy comparison between the particles propagation schemes of CONDENSATION, annealing, and our method. The RMS joint position error with the manually annotated ground truth is shown for the dance first exercise.

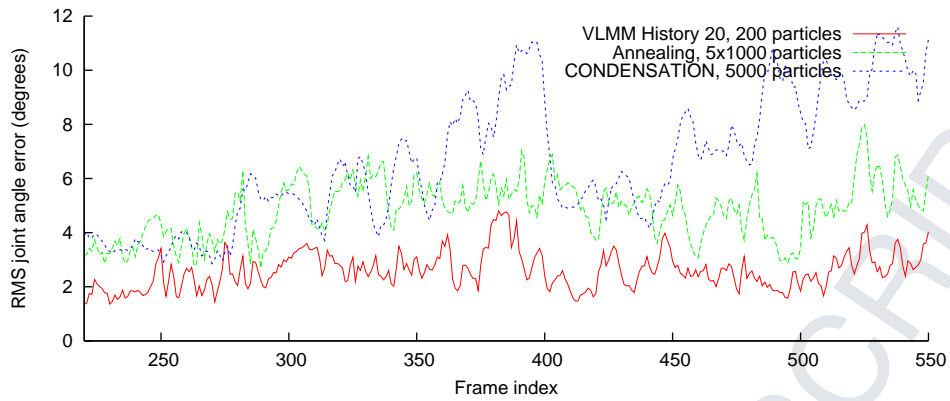


Fig. 12. Accuracy comparison between the particles propagation schemes of CONDENSATION, annealing, and our method. The RMS joint angle error with the manually annotated ground truth is shown for the dance first exercise.

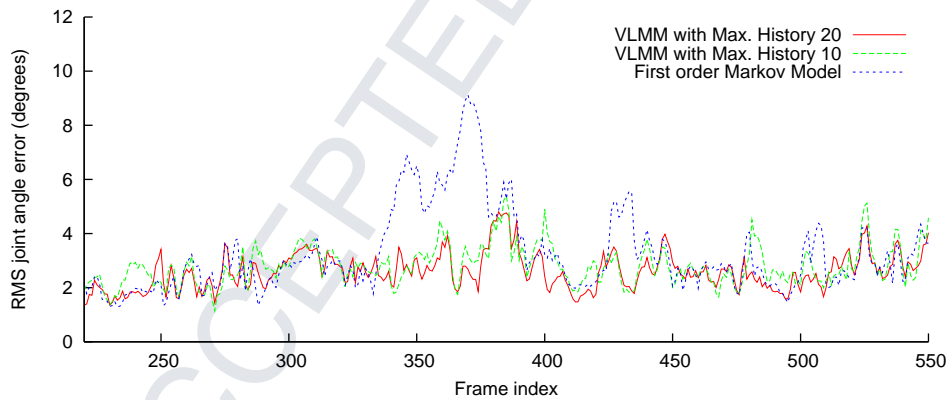


Fig. 13. Prediction accuracy for various history lengths of the Markov model, using 200 particles.