Dynamical Binary Latent Variable Models for 3D Human Pose Tracking

Graham W. Taylor
New York University
New York, USA
gwtaylor@cs.nyu.edu

Leonid Sigal
Disney Research
Pittsburgh, USA
lsigal@disneyresearch.com

David J. Fleet and Geoffrey E. Hinton
University of Toronto
Toronto, Canada
{fleet,hinton}@cs.toronto.edu

Abstract

We introduce a new class of probabilistic latent variable model called the Implicit Mixture of Conditional Restricted Boltzmann Machines (imCRBM) for use in human pose tracking. Key properties of the imCRBM are as follows: (1) learning is linear in the number of training exemplars so it can be learned from large datasets; (2) it learns coherent models of multiple activities; (3) it automatically discovers atomic "movemes"; and (4) it can infer transitions between activities, even when such transitions are not present in the training set. We describe the model and how it is learned and we demonstrate its use in the context of Bayesian filtering for multi-view and monocular pose tracking. The model handles difficult scenarios including multiple activities and transitions among activities. We report state-of-the-art results on the HumanEva dataset.

1. Introduction

Prior models of human pose and motion play a key role in state-of-the-art techniques for monocular pose tracking. Prior models constrain what is otherwise a difficult estimation problem because of its high dimensionality, intrinsic ambiguities, and noisy or missing measurements. Most successful prior models are activity specific and hence implicitly rely on activity detection before allowing pose tracking. Indeed, models that deal with multiple motions and transitions between them are scarce, in part because of the computational complexity that limits the size of training corpora, the lack of training data labeled with transitions, and the lack of a clear definition of atomic motion primitives.

This paper advocates the use of the Conditional Restricted Boltzmann Machine (CRBM) as a latent variable model for human pose tracking, and introduces a new class of models called the *Implicit Mixture of Conditional Restricted Boltzmann Machines* (imCRBM) for handling multiple activities. Inference and learning with the imCRBM are efficient. Learning is linear in the number of training exemplars, so one can use large training corpora, and it can handle multiple activities. We demonstrate that it can also infer transitions between activities even when such transitions do not occur in the training data. Finally, training can either be supervised (when labels are available) or unsupervised. Unsupervised, the learning algorithm automatically segments motions into statistically salient atomic parts.

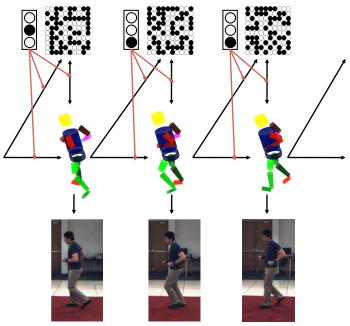

Figure 1. **Bayesian filtering with the imCRBM.** Each pose maps to a distribution over a discrete K-state vector and many binary latent features. The discrete component modulates the interaction weights among observed variables and latent features.

In this paper the CRBM and imCRBM models are used to learn human motion models for human pose tracking. We demonstrate learning and inference for single activities and for multiple activities (with transitions). These models are applied to both multi-view and monocular tracking.

2. Related Work

The literature on human pose estimation and tracking is vast, so a complete overview is beyond the scope of this paper. We refer the reader to [5] for a more complete overview. Below we focus on the most relevant body of work, that is, generative models of human pose and motion.

Early dynamical models were formulated as smooth, linear Markov models [2, 3, 18], but such models do not capture nonlinear properties of human motion and were found insufficient for monocular 3D pose tracking. Switching linear dynamical systems (SLDS) are more expressive, but they have not been used extensively (*e.g.* [17]). Learning with SLDS requires large training datasets given the high state dimensionality, and with SLDS it is difficult to ensure consistency in the latent variables when switching from one LDS to another. Modeling multiple activities with a *dis-*

tributed representation that uses many latent features, like the CRBM and imCRBM, rather than an explicit mixture, provides a more natural model of transitions.

One obvious way to manage the high dimensionality of human pose data is to use dimensionality reduction, or a latent variable model, with the dynamical model formulated on the latent variables. The earliest such models employed nonlinear dimensionality reduction, followed by a combination of density estimation and regression to learn generative mappings [11, 15, 20]. One problem with low-dimensional representations is that real data occasionally departs radically from the manifold. A regular walk may be low-dimensional, but if the walker occasionally scratches his nose or kicks a pebble, an *explicit* low-dimensional representation might be inadequate. To cope with such variability one can use *implicit* dimensionality reduction, as in the CRBM; *i.e.*, the latent representation remains high-dimensional, but the model learns to construct energy ravines in the latent space. In doing so, one is biased toward motions in the training set, but large deviations from the training set, while implausible, are not impossible.

Perhaps the most prominent current latent variable models are derived from the Gaussian Process Latent Variable Model [8, 23] and the Gaussian Process Dynamical Model [24]. Such models can serve as effective priors for tracking [23, 24] and can be learned with small training corpora [23]. However, larger corpora are problematic since learning and inference are $O(N^3)$ and $O(N^2)$, where N is the number of training exemplars. While sparse approximations to GPs exist [10], sparsification is not always straightforward and effective. Recent additions to the GP family include the topologically-constrained GPLVM [25], Multifactor GPLVM [26], and Hierarchical GPLVM [9]. Such models permit stylistic diversity and multiple motions (unlike the GPLVM and GPDM), but to date these models have not been used for tracking, and complexity remains an issue.

Most generative priors do not address the issue of explicit inference over activity labels. While latent variable models can be constructed from data that contains multiple activities [20], knowledge about the activities and transitions between them is typically only implicit in training data. As a result, training prior models to capture transitions, especially when they do not occur in training data, is challenging and often requires that one constrain the model explicitly (*e.g.* [25]). In [12] a coordinated mixture of factor analyzers was used to facilitate model selection, but to our knowledge, this model has not been used for tracking multiple activities and transitions. Another way to handle transitions is to to build a discriminative classifier for activities, and then use corresponding activity-specific priors to bootstrap the pose inference [1]. The proposed imCRBM model bridges the gap between pose and activity inference within a single coherent and efficient generative framework.

3. Conditional Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) [21] is a bipartite Markov Random Field consisting of a layer of stochastic "visible" variables connected to a layer of stochastic latent variables. The lack of direct connections among the latent variables, $\mathbf{z}$, ensures that they are conditionally independent given a setting of the visible variables, $\mathbf{x}$, which simplifies inference and learning. RBMs typically use binary visible and latent variables, but for real-valued data (*e.g.* pose) we can use a modified RBM with Gaussian, real-valued variables and binary latent variables [27].

The RBM can be extended to capture temporal dependencies by making its latent and visible variables receive additional input from previous states of the visible variables (Fig. 2, left). This model is called a Conditional RBM (CRBM) [22]. Conditioning on past data does not change the model's most important computational properties: simple, exact inference and efficient approximate learning.

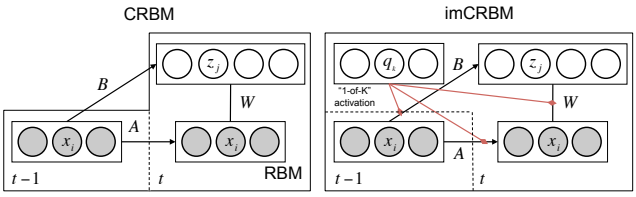

Figure 2. **Models.** Left: first-order CRBM. We typically use first-order models but also experiment with higher-order models. Right: the imCRBM. The discrete component variable $\mathbf{q}$ sets the "effective" CRBM. Bias parameters (C, D) are not shown.

The CRBM defines a joint probability distribution over a real-valued representation of current pose, $\mathbf{x}_t$, and a collection of binary latent variables, $\mathbf{z}_t, z \in \{0, 1\}$:

$$p(\mathbf{x}_t, \mathbf{z}_t | \mathbf{x}_{h_t}) = \exp\left(-E(\mathbf{x}_t, \mathbf{z}_t | \mathbf{x}_{h_t})\right) / Z(\mathbf{x}_{h_t}). \quad (1)$$

The distribution is conditional on the history of past N poses, $\mathbf{x}_{h_t}$, where $h_t \equiv t\text{–}N\text{:}t\text{–}1$, and normalized by constant Z which is intractable to compute exactly[1]. The joint distribution is characterized by an "energy function":

$$E = \sum_i \frac{1}{2}(x_{it} - \hat{c}_{it})^2 - \sum_j z_{jt}\hat{d}_{jt} - \sum_{ij} W_{ij} x_{it} z_{jt} \quad (2)$$

which captures the pairwise interactions between variables, assigning high scores to improbable configurations and low scores to probable configurations. Each visible variable contributes a quadratic offset to E (first term) that dominates Eq. 2 when it deviates too far from a "dynamical mean" that is a linear function of the previous poses: $\hat{c}_{it} = c_i + \sum_l A_{il}\mathbf{x}_{lh_t}$. The dynamical mean is much like a prediction from an autoregressive model of order N with

[1] To compute Z exactly we would need to integrate over the joint space of all possible poses and all settings of the binary latent variables.

constant offsets c_i. Each latent variable contributes a linear offset to E (second term) which is also a function of the past pose: $\hat{d}_{jt} = d_j + \sum_l B_{jl}\mathbf{x}_{lh_t}$. The third term of E is a bilinear constraint on the interaction between (current) visible and latent variables, characterized by weights W. A large value of W_{ij} means that x_i and z_j are strongly correlated.

3.1. Learning and prediction

Ideally we would like to maximize the marginal conditional likelihood, $p(\mathbf{x}_t|\mathbf{x}_{h_t})$, over parameters $\theta = \{W, A, B, \mathbf{c}, \mathbf{d}\}$ but this is difficult for all but the smallest models due to the intractability of computing Z. Learning, however, still works well if we approximately follow the gradient of another function called the contrastive divergence (CD) [6]. This learning method is simply called CD.

For sake of brevity, we refer the reader to [22] for details of learning a CRBM by CD. In short, learning relies on two main operations: 1) sampling the latent variables, given a window of training data, $\{\mathbf{x}_t, \mathbf{x}_{h_t}\}$:

$$p(z_{jt} = 1|\mathbf{x}_t, \mathbf{x}_{h_t}) = \left(1 + \exp(-\sum_i W_{ij}x_{it} - \hat{d}_{jt})\right)^{-1}, \quad (3)$$

and 2) reconstructing[2] the data, given the latent variables:

$$x_{it} \sim \mathcal{N}\left(x_{it}; \sum_j W_{ij}z_{jt} + \hat{c}_{it}, 1\right). \quad (4)$$

Both Eq. 3 and 4 follow from Eq. 1. Note that we always *condition* on the past, $\mathbf{x}_{h_t}$, it is never updated. Typically this process is repeated M times, giving rise to the term CD-M learning. Details of the weight updates are given in the supplementary material.

Given a trained CRBM and a N-step history of poses, we can obtain a joint sample from $p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{h_t})$ by alternating Gibbs sampling. This means starting at some reasonable initialization of $\mathbf{x}_t$ (we use $\mathbf{x}_{t-1}$) then alternating between Eq. 3 and 4 for some fixed number of steps (we use 100).

4. Implicit Mixtures of CRBMs

The capacity of the CRBM can always be increased by increasing the number of latent variables. Nonetheless, for data that contains several distinct modes (*e.g.* walking and running) it may be more desirable to use a mixture of CRBMs, where each component specializes to an activity. Compared to the density models employed by standard mixtures (*e.g.* Gaussians) it is intractable to exactly compute the normalized density under a CRBM and therefore it appears that learning a mixture of CRBMs is also intractable. Nair and Hinton [16] showed that a type of mixture model where each component was an RBM could be learned efficiently using contrastive divergence as long as the number of components was reasonably small (say, less than 100). The key was to parameterize the model as a type of third-order Boltzmann machine where the energy function captures three-way interactions between visible variables, binary latent variables and discrete "component" variables. However, this model treats each observation as *i.i.d.* and thus would ignore the temporal structure in time series data.

This paper proposes a new type of *dynamical* mixture model using three-way interactions (Fig. 2, right). We extend the CRBM by introducing a discrete variable, $\mathbf{q}$, with K possible states. For convenience, we define $\mathbf{q}$ to be a K-element vector, constrained such that only one element can be active. Our new model is defined by a joint distribution:

$$p(\mathbf{x}_t, \mathbf{z}_t, \mathbf{q_t}|\mathbf{x}_{h_t}) = \exp\left(-E(\mathbf{x}_t, \mathbf{z}_t, \mathbf{q_t}|\mathbf{x}_{h_t})\right)/Z(\mathbf{x}_{h_t}) \quad (5)$$

where the energy function, $E(\mathbf{x}_t, \mathbf{z}_t, \mathbf{q_t}|\mathbf{x}_{h_t})$, is given by:

$$E(\mathbf{x}_t, \mathbf{z}_t, \mathbf{q_t}|\mathbf{x}_{h_t}) = \frac{1}{2}\sum_i (x_{it} - \hat{c}_{it})^2 - \sum_j z_{jt}\hat{d}_{jt} - \sum_k q_{kt}\sum_{ij} W_{ijk}x_{it}z_{jt} \quad (6)$$

and the dynamical terms, $\hat{c}_{it}$, $\hat{d}_{jt}$, are given by:

$$\hat{c}_{it} = \sum_k q_{kt}\left(C_{ik} + \sum_l A_{ilk}\mathbf{x}_{lh_t}\right), \quad (7)$$

$$\hat{d}_{jt} = \sum_k q_{kt}\left(D_{jk} + \sum_l B_{ilk}\mathbf{x}_{lh_t}\right). \quad (8)$$

What were previously weight matrices, $\{W, A, B\}$, now become weight tensors, where each slice along the $\mathbf{q}$ dimension corresponds to the parameters of the K component CRBMs. Similarly, the static biases, $\{\mathbf{c}, \mathbf{d}\}$ become matrices $\{C, D\}$. Since at each time step t only one element of $\mathbf{q}_t$ is active, we can see from Equations 6-8 that $\mathbf{q}$ has the effect of "activating" a particular CRBM.

We can write the model in a traditional "mixture" form by marginalizing over the latent variables:

$$p(\mathbf{x}_t|\mathbf{x}_{h_t}) = \sum_{\mathbf{z}_t, \mathbf{q}_t} p(\mathbf{x}_t, \mathbf{z}_t, \mathbf{q_t}|\mathbf{x}_{h_t}) = \sum_{k=1}^{K} p(q_{kt} = 1)\sum_{\mathbf{z}_t} p(\mathbf{x}_t, \mathbf{z}_t|q_{kt} = 1, \mathbf{x}_{h_t}). \quad (9)$$

Compared to other mixture models, however, our model is unusual in that the mixing proportion is not a model parameter but implicitly defined by the energy function in Eq. 6. Thus we refer to it as an implicit mixture of CRBMs.

4.1. Learning and prediction

Like the CRBM, our mixture model can by trained by contrastive divergence. This, however, relies on sampling the conditional distributions $p(\mathbf{z}_t, \mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ and

[2] In practice, we sample the hidden state but set the updated visible state to the mean. This suppresses noise and learns slightly faster.

$p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{q}_t, \mathbf{x}_{h_t})$ which are not as straightforward as the case of the standard CRBM (Eq. 3 and 4). Details of sampling from these distributions are provided in the Appendix.

Given a trained imCRBM and a N-step history of poses, we can obtain a joint sample from $p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{h_t})$ by alternating Gibbs sampling in a way almost identical to that of a standard CRBM. The only difference is we first compute the posterior distribution over components $p(\mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ and then pick a component, k, before sampling the latent variables under the k^{th} CRBM using $p_k(\mathbf{z}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ and updating the visible variables using $p_k(\mathbf{x}_t|\mathbf{z}_t, \mathbf{x}_{h_t})$.

5. Bayesian Filtering with CRBM-type models

In tracking one is generally interested in approximating the filtering distribution, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, the distribution over the pose of the body at time t, $\mathbf{x}_t$, conditioned on past image observations $\mathbf{y}_{1:t} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t]$. Assuming conditional independence of observations (given the state) the posterior above can be written as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}), \tag{10}$$

where $p(\mathbf{y}_t|\mathbf{x}_t)$ is the *likelihood*, which measures consistency of the state with image observations, and $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ is the *predictive distribution*, which predicts the state at time t given image observations up to but not including time t. Making a 1st order Markov assumption on the state evolution, the predictive distribution can be written as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\ d\mathbf{x}_{t-1}, \tag{11}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the *transition density* and $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ is the posterior at time $t-1$.

If the temporal prior is an implicit mixture of first-order CRBMs, we introduce latent variables, $\mathbf{z}_t$ and $\mathbf{q}_t$:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{\mathbf{x}_{t-1}} \int_{\mathbf{z}_t, \mathbf{q}_t} p(\mathbf{x}_t, \mathbf{z}_t, \mathbf{q}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \, d\mathbf{z}_t d\mathbf{q}_t d\mathbf{x}_{t-1}, \tag{12}$$

that need to be integrated out either implicitly using sampling or in closed form.

The first integral in Eq. 12 is often estimated using Monte Carlo methods [4] that approximate the posterior by a set of P weighted samples, $\{\mathbf{x}_t^{(p)}, \pi_t^{(p)}\}_{p=1}^P$. The simplest among such approaches is Sequential Importance Resampling (SIR). At every time instant SIR samples from the predictive distribution[3] and then assigns them weights based on the likelihood:

$$\begin{aligned} \mathbf{x}_t^{(p)} &\sim p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \\ \pi_t^{(p)} &\propto p(\mathbf{y}_t|\mathbf{x}_t^{(p)}). \end{aligned} \tag{13}$$

[3]In practice we add small amount of Gaussian noise to the predictive distribution to allow for a diffusion of samples that can account for noise in the inference or un-modeled correlations.

Higher-order CRBM models, that incorporate history over the past N frames, impose a N-th order Markov dependency among the states. This, however, can easily be addressed within the context of a particle filter, by defining an augmented state [7] $\hat{\mathbf{x}}_t = \mathbf{x}_{t-N:t}$, and a transition density that (1) temporally shifts the elements of augmented state by one (dropping the oldest state in the sequence) and (2) predicts the most recent state according to the N-th order CRBM conditioned on the past augmented state.

We utilize the freely available SIR implementation of [2], but augment it to maintain a sample-based representation of the history over the past N frames, $\{\mathbf{x}_{h_t}^{(p)}, \pi_t^{(p)}\}_{p=1}^P$.

5.1. Modeling the body

As is common in the literature, we model the body as a 3D kinematic chain with limbs represented by truncated elliptical cross-section cones. Our body model consists of 15 segments: 2 torso segments (pelvic region and torso), lower and upper arms and legs, feet, hands, and a head. The lengths, widths and cross-sectional scaling for all segments is assumed fixed and known. Inference consists of finding the pose of the body over time, $\mathbf{x}_t \in \mathbb{R}^{40}$. The pose consists of global position and orientation of the body in the world (6 DoF), orientation of the hips, shoulders, head and abdomen joints (3 DoF each), the clavicle, elbow, and knee joints (2 DoF) and wrist and ankle joints (assumed to be 1 DoF).

To achieve invariance, instead of modeling the absolute position in the ground plane, we model velocity along the anterioposterior and lateral axes. We also model the velocity of rotation about the vertical instead of absolute orientation. Velocities are expressed as the difference between the current and previous frame. For learning, each dimension is also scaled to have zero mean and unit variance. During tracking, predictions are made in the normalized, invariant space and then converted back to the global representation.

5.2. Likelihood

In general, for tracking, one prefers rich likelihoods that are robust to lighting variations, occlusions and image noise. For simplicity, and to fairly evaluate our approach with respect to prior art, we utilize relatively weak and generic likelihoods based on silhouette[4] and edge information (for details, see [3]), but admit that better results can be obtained using richer likelihood models (*e.g.* optical flow or adaptive appearance regions [23, 24]).

6. Experiments

We conducted a series of experiments to measure the effectiveness of our prior models in real multi-view and monocular 3D settings on a variety of sequences from the HumanEva dataset [19]. Since the imCRBM is a generalization of the CRBM, we begin by illustrating the efficacy

[4]For some of the experiments we utilize a more robust silhouette-based likelihood described in [19].

of both models for tracking of atomic motions. Then we demonstrate how the imCRBM can further improve performance through better modeling of transitions.

Datasets. HumanEva consists of a set of multi-view sequences with synchronized motion capture data to allow quantitative evaluation of performance. The HumanEva dataset consists of six different motions, performed by four different subjects (S1-S4); we utilize sequences of walking, jogging, boxing and combo (walk transitioning to a jog) for our experiments. We also utilize the earlier synchronized walking sequence from a different subject [2], denoted S5.

Evaluation. To quantitatively evaluate the performance we adopt the measure proposed in [19], which computes an average Euclidean distance between 15 virtual markers on the body (corresponding to joints and end points of segments). The use of the dataset and this measure allows us to easily compare our performance with prior methods.

Baseline. As a baseline we compare performance against the standard particle filter with smooth zero-order dynamics (*i.e.* $\mathbf{x}_{t+1} = \mathbf{x}_t$ up to additive noise). For fairness, we always utilize the same number of samples and the same likelihoods between the Baseline and the proposed approach.

Learning. Except where noted, all CRBM models were trained as follows: Each training case was a window of $N+1$ consecutive frames and the order of the training cases was randomly permuted. The training cases were presented to the model as "mini-batches" of size 100 and the weights were updated after each mini-batch. Typically we use models with 100 latent variables (chosen based on the relatively small amount of available training data) and we make no attempt to optimize this number. Models were trained using CD-10 (see Sec. 3.1) for 5000 complete passes through the data. All parameters used a learning rate of $\lambda = 10^{-3}$, except for the autoregressive weights which used $\lambda_A = 10^{-5}$. A momentum term was also used: 0.9 of the previous accumulated gradient was added to the current gradient.

Initialization. To initialize the tracker we use the first two frames from the provided motion capture data (converted into our representation). The use of two frames, as opposed to one is required to calculate the velocities described in Sec. 5.1). Where higher order models are used, we first grow the trajectory using a lower order CRBM.

6.1. Multi-view tracking

Generalization across subjects. We repeat the experiments proposed in [28] to demonstrate the insensitivity of the CRBM prior to the identity of the test subject. To compare with previously published results, we maintain the same experimental setup as in [28] but in place of the motion correlation (MoCorr) model, we use a first-order CRBM prior. In all cases, we track subject S5 (first 150 frames of the validation sequence), but train with three different datasets: (1) S5 walking, (2) S1 walking, and (3) the combined walking motions of subjects S1, S2 and S3. We use 4 camera views and 1000 particles for all three sequences. To the best of our knowledge, we utilize the same edge and silhouette-based likelihood, the same basic inference architecture, the same number of samples, and the same test and training sets as [28]. Results (averaged over 10 runs) are shown in Table 1 (the plot of performance over time is provided in the supplementary material).

Train on	Baseline	MoCorr [28]	CRBM
S5	91.37±6.29	48.98	**41.97±3.57**
S1		51.66	**38.45±0.80**
S1+S2+S3		55.30	**48.03±0.29**

Table 1. **Generalization across subjects.** Tracking of subject S5 using a 1st order CRBM prior model learned from S5, S1, and S1+S2+S3 training data. ± indicates standard deviation over runs.

In each case, our method outperforms standard particle filtering and particle filtering using the MoCorr model. Surprisingly, we perform slightly better on subject S5 using a prior model trained on subject S1. This may be due to the much smaller S5 training dataset (921 frames compared to 2232 for S1).

HumanEva-I walking. Next, we apply both a CRBM and 10-component imCRBM to the walking sequences in the HumanEva-I dataset. As in [28], we track each of subjects S1, S2 and S3 (walking validation) using a first-order dynamical model trained on the combined walking data of all three subjects. We repeat the experiment using a subject-specific prior, to compare to Li *et al.* [13].

Li *et al.* [12] proposed a nonlinear dynamical model, the Coordinated Mixture of Factor Analyzers (CMFA), later extended to include variational inference in [13] (CMFA-VB). They integrated the CMFA prior into a tracking architecture (significantly different from particle filtering) and achieved favorable performance compared to a particle filter with a Gaussian Process Latent Variable Model (GPLVM) prior. We compare to the performance of both methods reported in [13] (see Table 2).

For this experiment we utilize 3 color views to be consistent with [28] and [13]. In all cases the performance of our method is considerably better than other methods, with significantly lower variance as compared to the GPLVM and CMFA [13]. The imCRBM outperforms the CRBM in the case of the (S1+S2+S3) training set, but overfits relative to the CRBM on the smaller, subject-specific training set.

HumanEva-I boxing. To illustrate that the CRBM prior model is not specific to cyclic motions, and at the same time explore the effect of history on performance we illustrate first, third and sixth-order CRBM priors on the validation boxing sequence of S1. For this experiment we only use 200 particles. The choice of the sequence and setup is moti-

Training	Test	Baseline	MoCorr [28]	GPLVM [13]	CMFA-VB [13]	CRBM	imCRBM-10
S1+S2+S3	S1	129.18±19.47	140.35	-	-	55.43±0.79	**54.27±0.49**
S1	S1		-	-	-	**48.75±3.72**	58.62±3.87
S1+S2+S3	S2	162.75±15.36	149.37	-	-	99.13±22.98	**69.28±3.30**
S2	S2		-	88.35±25.66	68.67±24.66	**47.43±2.86**	67.02±0.70
S1+S2+S3	S3	180.11±24.02	156.30	-	-	70.89±2.10	**43.40±4.12**
S3	S3		-	87.39±21.69	69.59±22.22	**49.81±2.19**	51.43±0.92

Table 2. **HumanEva-I walking performance.** Tracking of subjects S1, S2, S3 using various prior models. For the implementation of the GPLVM in [13] an annealed particle filter with 5 layers of annealing and 500 particles per layer was used.

vated by comparison to the original results reported in [12]. Since in [12] Li *et al.* also compare to the performance of a Switching Linear Dynamical System (SLDS) and the Dynamic Global Coordination Model (DGCM) of [14], we include those results for completeness. The tracking performance is summarized in Table 3; we report the average over 10 runs using each order of CRBM. Again our method shows significant improvement over the other approaches considered. While the order of the CRBM model does not seem to improve performance in the multi-view scenario, it does reduce the variance.

Model	Order 1	Order 3	Order 6
SLDS [12]	569.90±209.18	-	-
DGCM [12]	380.02±74.97	-	-
CMFA [12]	187.50±39.73	-	-
Baseline	116.95±5.54	-	-
CRBM	**75.35±9.71**	**82.40±8.26**	**82.91±5.15**

Table 3. **HumanEva-I boxing performance.** Subject S1.

6.2. Tracking over transitions

Tracking over transitions presents a considerable challenge even when using a dynamical prior. Here we illustrate the benefits of using the imCRBM to capture the discrete nature of "walking" and "jogging" motions and include transitions. In these experiments we track the first 700 frames of the S3 "combo" sequence which consists of walking transitioning to jogging. Each dynamical model is trained on all walking and jogging training sequences associated with S3. These contain no transitions. We train the imCRBM under two settings. In the first, imCRBM-2L, we use the labeled training data to fix the components of a two-component imCRBM during the positive phase of learning (the components are inferred during the negative phase of learning and at test time). The second setting, imCRBM-10U, is completely unsupervised, where we have trained a 10-component mixture on the same mocap data but without labels. We also compare to the performance of the baseline and standard CRBM. All dynamical models are first-order. Results are shown in Figure 3 and summarized in Table 4.

With the imCRBM we can compute an approximate posterior distribution over component labels by counting and normalizing the assignment of particles at each time step.

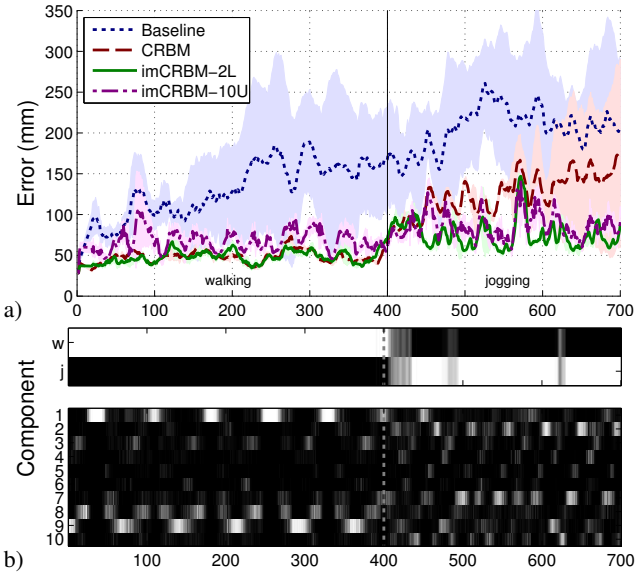

Figure 3. **Transitions between activities.** a) Mean prediction error on the S3 "combo" test sequence using various prior models. Shading indicates ±1 std. dev. b) Approximate posterior distribution over activities using the imCRBM-2L (top) and imCRBM-10U (bottom). For all plots, the horizontal axis is frame number.

Model	Walking	Jogging	All
Baseline	132.06±48.49	205.64±11.39	164.24±25.03
CRBM	48.09±0.55	125.36±28.62	81.88±12.41
imCRBM-2L	48.12±0.80	75.67±2.18	**60.17±1.24**
imCRBM-2L*	61.84±1.51	93.05±4.72	75.48±1.77
imCRBM-10U	67.48±2.63	86.44±2.00	**75.77±1.74**
imCRBM-10U*	80.72±1.78	89.90±1.16	84.74±1.13

Table 4. Mean predictive error over walking frames (1-400), jogging frames (401-700), and mean over frames 1-700 of S3 "combo" sequence. Note that the boundary is approximate. * We also repeated the imCRBM runs with a non-subject-specific prior (trained on S1,S2,S3 data) to demonstrate generalization ability.

When the imCRBM is trained supervised, these components correspond to activity labels (Figure 3b, top). In the unsupervised case, they correspond to *movemes*, or atomic segments automatically discovered by the model (Figure 3b, bottom). Here, the components still seem to be action-specific and correspond to parts of the gait cycle. Note that the posterior for frames near the transition – dashed line in Fig. 3b, marked roughly around frame 400 – is much softer.

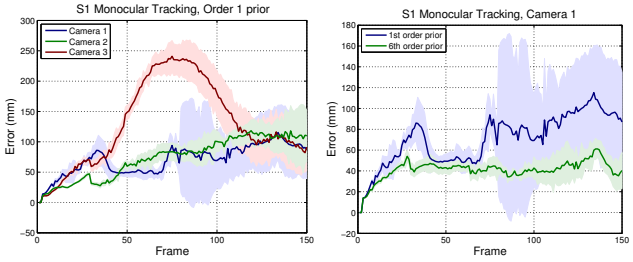

Figure 4. **Monocular tracking quantitative performance.** Left: Monocular tracking of subject S1, all cameras. Right: Using a sixth order model improved the results for cameras 1 and 2 (camera 1 shown). Results are averaged over 5 runs per camera (per-frame standard deviation is shaded).

6.3. Monocular tracking

Tracking with a single camera presents a significant challenge to current methods. Balan et al. [2] report that monocular tracking using an Annealed Particle Filter (5 layers, 200 particles per layer) with edge and silhouette-based likelihoods fails, on average, after 40 frames of tracking the S5 validation sequence. They report an error of 263±60mm tracking the first 150 frames of the sequence. Because we track a single activity, we applied standard particle filtering with a CRBM motion prior trained on S5 training data to the same validation sequence. We used an identical likelihood, and used the same total number of particles (1000). Averaging errors over all 4 cameras, and 5 runs per camera, using a first-order CRBM gives a result of 133.93±55.62mm. If we apply a sixth-order CRBM, our results improve to 112.25±79.52mm. For each camera, at least one run successfully tracks the entire sequence. All camera 1 runs are successfully tracked for the entire sequence.

We also applied the tracker to S1. Using a first order CRBM, and a bi-directional silhouette likelihood term, we obtain an error of 90.98±32.70 averaged over 5 runs per camera (Figure 4). When tracking is reasonably successful, using a higher-order model helps considerably. For example, using camera 1 gives an error of 47.29±4.95mm using a sixth-order model (compared to 70.50±24.19 for the first-order model).

Monocular tracking with transitions. We applied the imCRBM trained with activity labels (imCRBM-2L) to track the S3 "combo" sequence. This is a difficult task at which both the baseline and standard CRBM fail. With the new model we are able to successfully track the entire sequence, including the transitions. A single run is shown in Figure 5. See the supplementary material for more details.

7. Discussion

We have demonstrated that binary latent variable models work effectively as a prior in Bayesian filtering, allowing 3D tracking of people from multi-view and monocular observations. Our models use a high-dimensional, non-linear representation which captures low-dimensional structure by learning energy ravines. This allows one to learn models from many different types of motion and subjects using the same set of latent variables. We have also introduced a new type of dynamical prior that can capture both discrete and continuous dynamics. The imCRBM should be useful for time series analysis beyond the tracking domain.

Acknowledgments

The authors thank NSERC and CIFAR for financial support. This work was primarily conducted while the first two authors were at the University of Toronto.

References

[1] A. Baak, B. Rosenhahn, M. Mueller, and H.-P. Seidel. Stabilizing motion tracking using retrieved motion priors. *ICCV*, 2009.

[2] A. Balan, L. Sigal, and M. Black. A quantitative evaluation of video-based 3D person tracking. *IEEE Workshop on Visual Surveillance and PETS*, pp. 349-356, 2005.

[3] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *IJCV*, 61(2):185-205, 2005.

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

[5] D. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. *Found. Trends Comp. Graphics and Vision*, 1(2/3), 2006.

[6] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput*, 14(8):1771–1800, 2002.

[7] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[8] N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Machine Learning Res.*, 6:1783–1816, Nov. 2005.

[9] N. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. *ICML*, 2007.

[10] N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *NIPS*, 15:625–632, 2003.

[11] C. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. *ICCV*, 2007.

[12] R. Li, T. Tian, and S. Sclaroff. Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. *ICCV*, 2007.

[13] R. Li, T. Tian, S. Sclaroff, and M.-H. Yang. 3D human motion tracking with coordinated mixture of factor analyzers. *IJCV*, 87(1–2):170–190, 2010.

[14] R.-S. Lin, C.-B. Liu, M.-H. Yang, N. Ahuja, and S. Levinson. Learning nonlinear manifolds from time series. *ECCV*, pp. 245-256, 2006.

[15] Z. Lu, M. Carreira-Perpinan, and C. Sminchisescu. People tracking with the Laplacian eigenmaps latent variable model. *NIPS*, 2007.

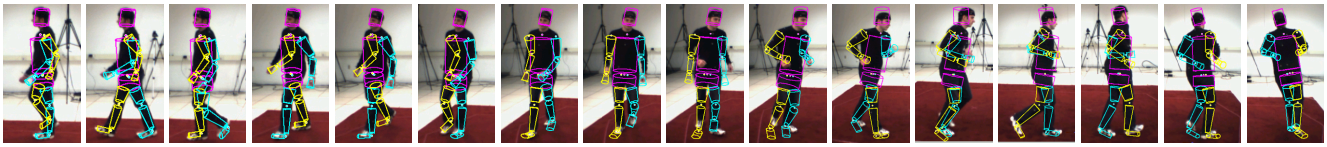

Figure 5. **Monocular tracking with transitions (S3 combo, Camera 2) with the imCRBM-2L.** Frames 280-505 (every 15 frames) are shown to demonstrate the transition from walking to jogging. Quantitative results (for all cameras) are given in the supplementary material.

[16] V. Nair and G. Hinton. Implicit mixtures of restricted Boltzmann machines. *NIPS*, pp. 1145-1152, 2009.

[17] V. Pavolvic, J. Rehg, T. Cham, and K. Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models. *ICCV*, pp. 94-101, 1999.

[18] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion. *ECCV*, 2:702-718, 2000.

[19] L. Sigal, A. Balan, and M. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 2010.

[20] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. *ICML*, pp. 759-766, 2004.

[21] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing: Vol 1*, pp. 194–281. MIT Press, Cambridge, 1986.

[22] G. Taylor, G. Hinton, and S. Roweis. Modeling human motion using binary latent variables. *NIPS*, 19, 2007.

[23] R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. *ICCV*, 1:403-410, 2005.

[24] R. Urtasun, D. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. *CVPR*, 238-245, 2006.

[25] R. Urtasun, D. Fleet, A. Geiger, J. Popovic, T. Darrell, and N. Lawrence. Topologically-constrained latent variable models. *ICML*, 2008.

[26] J. Wang, D. Fleet, and A. Hertzmann. Multifactor Gaussian process models for style-content separation. *ICML*, 2007.

[27] M. Welling, M. Rosen-Zvi, and G. Hinton. Exponential family harmoniums with an application to information retrieval. *NIPS*, 2005.

[28] X. Xu and B. Li. Learning motion correlation for tracking articulated human body with a Rao-Blackwellised particle filter. *ICCV*, pp. 1-8, 2007.

Appendix: imCRBM inference and learning

Note: this Appendix assumes familiarity with learning a CRBM using contrastive divergence (CD). For a review, see [22]. Learning an imCRBM with CD requires us to sample from two conditional distributions: $p(\mathbf{z}_t, \mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ and $p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{q}_t, \mathbf{x}_{h_t})$. Recall that $\mathbf{q}$ is 1-of-K encoded, and so $p(\mathbf{z}_t, \mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t}) \equiv p(\mathbf{z}_t, q_{kt} = 1|\mathbf{x}_t, \mathbf{x}_{h_t})$. Sampling from the second distribution is straightforward: given $q_{tk} = 1$, we simply sample from $p_k(\mathbf{x}_t|\mathbf{z}_t, \mathbf{x}_{h_t})$ defined by the k^{th} component CRBM. Sampling from $p(\mathbf{z}_t, \mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ is performed in two steps. We first sample $\mathbf{q}_t$ using $p(q_{kt} = 1|\mathbf{x}_t, \mathbf{x}_{h_t})$ and then sample from $p_k(\mathbf{z}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ defined by the k^{th} component CRBM corresponding to our draw. Computing $p(\mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ relies on the fact that

$$p(q_{kt} = 1|\mathbf{x}_t, \mathbf{x}_{h_t}) \propto \exp\left(-F(\mathbf{x}_t, q_{kt} = 1|\mathbf{x}_{h_t})\right), \qquad (14)$$

where the *free energy*, F, is given by

$$F(\mathbf{x}_t, q_{kt} = 1|\mathbf{x}_{h_t}) = \frac{1}{2}\sum_i (x_{it} - \hat{c}_{it})^2 - \sum_j \log\left(1 + \exp(\sum_i W_{ijk}x_{it} + \hat{d}_{jt})\right). \qquad (15)$$

F is the negative log probability of an observation plus $\log Z$. As long as K is reasonably small, we can evaluate Eq. 15 for each setting of k, and renormalize such that

$$p(q_{kt} = 1|\mathbf{x}_t, \mathbf{x}_{h_t}) = \frac{\exp\left(-F(\mathbf{x}_t, q_{kt} = 1|\mathbf{x}_{h_t})/\tau\right)}{\sum_l \exp\left(-F(\mathbf{x}_t, q_{lt} = 1|\mathbf{x}_{h_t})/\tau\right)}, \qquad (16)$$

where τ is a temperature parameter which ensures that random scale differences in initialization and learning do not cause the model to collapse to a single component. We used fixed $\tau = 100$.

Now that we have a well-defined sampling procedure for the conditional distributions $p(\mathbf{z}_t, \mathbf{q}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$ and $p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{q}_t, \mathbf{x}_{h_t})$ we can train the model with contrastive divergence. The algorithm for one iteration of learning is:

1. Given a history of observations, $\mathbf{x}_{h_t}$, and a training vector, $\mathbf{x}_t^+$, compute $p(q_{kt} = 1|\mathbf{x}_t^+, \mathbf{x}_{h_t})\ \forall k \in K$. Pick a component by sampling. Let k^+ be the index of the selected component.
2. Sample $\mathbf{z}_t^+ \sim p_{k^+}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$.
3. Compute the *positive phase* statistics (see the supplementary material): $\{W_k^+, A_k^+, B_k^+, \mathbf{c}_k^+, \mathbf{d}_k^+\}$ using the $k^{+\text{th}}$ CRBM.
4. Sample $\mathbf{x}_t^- \sim p_{k^-}(\mathbf{x}_t|\mathbf{z}_t^+, \mathbf{x}_{h_t})$.
5. Compute $p(q_{kt} = 1|\mathbf{x}_t^-, \mathbf{x}_{h_t})\ \forall k \in K$. Pick a component by sampling. Let k^- be the index of the selected component.
6. Sample $\mathbf{z}_t^- \sim p_{k^-}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{x}_{h_t})$.
7. Repeat steps 4-6 above $M - 1$ times for CD-($M > 1$), substituting $\mathbf{z}_t^-$ for $\mathbf{z}_t^+$ in step 4.
8. Compute the *negative phase* statistics: $\{W_k^-, A_k^-, B_k^-, \mathbf{c}_k^-, \mathbf{d}_k^-\}$ using the $k^{-\text{th}}$ CRBM.
9. Update weights $\{W_k, A_k, B_k, \mathbf{c}_k, \mathbf{d}_k\}, \forall k \in \{k^+, k^-\}$.

In practice, parameter updates are performed after each presentation of a mini-batch consisting of several $\{\mathbf{x}_{h_t}, \mathbf{x}_t^+\}$ pairs. The update to each parameter of a component CRBM is proportional to the difference of summed positive phase statistics and summed negative phase statistics assigned to that component (for details, see the supplementary material).

Furthermore, if we have labeled training data, we can fix the component in the positive phase to match the label (step 1), but still sample the component in the negative phase (step 5). We can then perform inference over the component given unlabeled data.