Accepted, to appear IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007

BM^3E : Discriminative Density Propagation for Visual Tracking

Cristian Sminchisescu Atul Kanaujia Dimitris N. Metaxas

Contact author:

Cristian Sminchisescu

TTI-C

University of Chicago Press, 2nd floor

1427 East 60th Street

Chicago, IL, 60637, USA

Email: *crismin@nagoya.uchicago.edu*

Web: *http://ttic.uchicago.edu/˜crismin*

Phone: + 1 773 834 2622

Fax: + 1 773 834 2557

BM^3E : Discriminative Density Propagation for Visual Tracking

Abstract

We introduce BM^3E*, a Conditional Bayesian Mixture of Experts Markov Model, for consistent probabilistic estimates in discriminative visual tracking. The model applies to problems of temporal and uncertain inference and represents the unexplored bottom-up counterpart of pervasive generative models estimated with Kalman filtering or particle filtering. Instead of inverting a non-linear generative observation model at run-time, we learn to cooperatively predict complex state distributions directly from descriptors that encode image observations – typically bag-of-feature global image histograms or descriptors computed over regular spatial grids. These are integrated in a* conditional graphical model *in order to enforce temporal smoothness constraints and allow a principled management of uncertainty. The algorithms combine sparsity, mixture modeling, and non-linear dimensionality reduction for efficient computation in high-dimensional continuous state spaces. The combined system automatically self-initializes and recovers from failure. The research has three contributions: (1) We establish the density propagation rules for* discriminative inference *in continuous, temporal chain models; (2) We propose flexible supervised and unsupervised algorithms for* learning *feedforward, multivalued contextual mappings (multimodal state distributions) based on compact, conditional Bayesian mixture of experts models; (3) We validate the framework empirically for the* reconstruction of 3d human motion in monocular video sequences*. Our tests on both real and motion capture-based sequences show significant performance gains with respect to competing nearest-neighbor, regression, and structured prediction methods.*

Keywords: *computer vision, statistical models, video analysis, motion, tracking.*

1 Introduction and Motivation

We consider the problem of probabilistic state inference in feedforward, conditional chain models, based on temporal observation sequences. For demonstration, we concentrate on the tracking and 3d reconstruction of articulated human motion in monocular video. This is a challenging research topic with a broad set of applications for scene understanding, but we emphasize that our framework applies generally, to *continuous and uncertain temporal state estimation* problems.

Two general classes of strategies exist for visual modeling and inference:*(i) Generative (feedback) methods* optimize 3d kinematic and appearance models for good alignment with image features. The objective is encoded as an observation likelihood or cost function with optima (ideally!) centered at correct pose hypotheses; *(ii) Conditional (feedforward) methods* – also referred as discriminative, diagnostic, or recognition-based – predict human poses directly from images features. Both approaches require a state representation, $\mathbf{x}$ say, here a 3d human model with kinematics or shape (joint angles, surfaces or joint positions), and both use a set of image feature observations, $\mathbf{r}$, for state inference. A training set, $\mathcal{T} = \{(\mathbf{r}_i, \mathbf{x}_i) \mid i = 1 \ldots N\}$, sampled from the *joint distribution* is usually available. The computational goal is common: the conditional distribution, or a model state point estimate, given observations. The state and the observation descriptors are important components of modeling. The state needs dimensionality adequate for the variability in the task, and the observation descriptor needs to be specific enough to capture not only strong image dependencies but also discriminative detail. Typically, these are obtained by combining a-priori design and off-line unsupervised learning. Once selected, the representation (model state & observation descriptor) is known for later learning and inference stages. This currently holds for both generative and discriminative models.

Generative algorithms model the joint distribution using a constructive form of the the observer: the observation likelihood or cost function. Complex sampling or non-linear optimization methods are used to infer the likelihood peaks, and Bayes' rule is used to compute the state conditional from the observation conditional and the state prior. Both supervised and unsupervised procedures are used for model learning – either to obtain state priors [9, 19, 14, 38, 40, 54, 21] or to tune the parameters of the observation model, *e.g.* texture, ridge or edge distributions, using problem-dependent, natural image statistics [37, 34, 49]. Tracking is framed in a clear probabilistic and computational framework based on mixture filters or particle filters [20, 14, 11, 47, 48, 50, 38].

It has been argued that generative models can flexibly reconstruct complex unknown motions and can naturally handle problem constraints. It has been counter-argued that both flexibility and modeling difficulties lead to expensive, uncertain inference [14, 37, 48, 41], and a constructive form of the observation (*i.e.* image appearance) is both difficult to build and indirect with respect to the task – primarily conditional state estimation, *not* conditional observation modeling.

The generative counter-argument motivates the complementary study of **discriminative algorithms** [33, 32, 36, 52, 3, 16], that predict state distributions directly from image features. This approach is not without its own difficulties: background clutter, occlusion or depth ambiguities make the observations-to-state mapping multi-valued and not amenable to simple functional prediction. Although, in principle, single hypothesis methods are not expected to be sufficient in this context, several authors demonstrated good practical performance [36, 32, 52, 3, 16]. The methods differ in the organization of the training set and in the runtime hypothesis selection method: some construct data structures for fast nearest-neighbor retrieval [36, 52, 32], others learn regression models [3, 16]. Inference involves either indexing for the nearest-neighbors of the observation and using their state for locally weighted prediction, direct prediction using the learned regression model [3, 16], or affine reconstruction from joint centers [32].

Among (dominantly) discriminative methods, Rosales & Sclaroff [33] take a notably different approach, by accurately modeling the joint distribution using a mixture of perceptrons. Their system combines multiple image-based state prediction with hypothesis selection based on a rendering (feedback) model. A related method has been suggested by Grauman *et al* [18], who model the joint distribution of multi-view silhouettes-pose using a mixture of probabilistic PCA. The problem has been independently studied by Agarwal & Triggs [2] who use joint models based on random regression in a Condensation-based generative tracking framework. There is an important difference between working with the joint distribution [13, 33] and working only with the conditional state distribution – even when using mixture of feedforward state models (as opposed to generative observation models). In a joint model based on multiple components, the reliability of each state predictor has to be ranked at run-time – a non-trivial operation because the state is missing. The problem can be solved either by conditioning and marginalization (the application of Bayes rule) in the joint model or by verification, using an ad-hoc, external observation model. Depending on the assumed modeling details, the computations can be difficult to perform or may not be probabilistically consistent. An alternative is to use a conditionally parameterized model. Details on models and computations for both directly parameterized conditionals, and for models based on random regression and joint density appear in our earlier work [42].

To summarize, discriminative models provide fast inference, interpolate flexibly in the trained region, but can fail on non-typical inputs, especially if trained using small datasets. Large training set and complex motions increase the image-to-pose ambiguity which manifests as multivalued image-to-pose relations or, probabilistically, as multimodal conditional state distributions. Learning multivalued models is inherently difficult. Moreover, existing discriminative methods lack a probabilistic temporal estimation framework that has been so fruitful with generative models [20, 14, 48]. Existing tracking algorithms [52, 3, 16] involve per-frame state inference, often using estimates at previous timesteps [52, 3, 16], but do rely on an set of independence assumptions or propagation rules. *What distributions should be modeled, how should they be modeled, and how should they be temporally combined for optimal solutions?*

The research we present addresses these questions formally. We introduce BM^3E, a Conditional $\underline{B}$ayesian $\underline{M}$ixture of $\underline{E}$xperts $\underline{M}$arkov $\underline{M}$odel for consistent probabilistic estimates in discriminative visual tracking. This represents the unexplored, feedforward counterpart of temporal generative models estimated with Kalman filtering or particle filtering. Instead of inverting a generative observation model at run-time, we learn to cooperatively predict complex state distributions directly from image descriptors. These are integrated in a conditional graphical model in order to enforce temporal smoothness constraints, and allow a principled management of uncertainty.[1]

[1] This model should not be confused with a Maximum Entropy Markov Model, MEMM [31], designed for discrete state variables, and based on a different, maximum entropy representation of conditional distributions.

The algorithm combines sparsity, mixture modeling, and non-linear dimensionality reduction for efficient computation in high-dimensional continuous state spaces [45, 44, 42, 39]. The combined system automatically initializes and recovers from failure – it can be used either stand-alone, or as a component to bootstrap generative inference algorithms. This research has three technical contributions:

(1) We establish the density propagation rules for *discriminative inference* in continuous, temporal chain models. The ingredients of the approach are: *(a)* the structure of the graphical model (see fig. 1 and §2.1); *(b)* the representation of local, per-node conditional state distributions (see **(2)** below and §2.2); *(c)* the belief propagation (chain inference) procedure (§2.1). We work parametrically and analytically, to predict and propagate Gaussian mixtures [41], but non-parametric belief propagation methods [50, 38] can also be used to solve *(c)*.

(2) We propose flexible algorithms for *learning* to contextually predict feedforward multimodal state distributions based on compact, conditional Bayesian mixture of experts. (An expert is any functional approximator, *e.g.* a perceptron or regressor.) These are based on hierarchical mixtures of experts [24, 55, 53, 7], an elaborated version of clusterwise or switching regression [13, 33], where the expert mixture proportions, called gates, are themselves observation-sensitive predictors, synchronized across experts to give properly normalized conditional state distributions for any input observation. Our learning algorithm is different from the one of [55] in that we use sparse greedy approximations, and differs from [7] in that we use type-II maximum likelihood Bayesian approximations [30, 29, 51, 26], not structured variational ones.

(3) We validate the framework empirically on the problem of *reconstructing 3d human motion in monocular video sequences*. Our tests on both real and motion capture-based sequences show important robustness and performance gains compared to nearest-neighbor, regression, and structured prediction methods.

Paper Organization: We introduce the discriminative density propagation framework, referred as BM^3E, in §2 as follows: §2.1 reviews the structure of the graphical model and the equations used for temporal density propagation (precise derivations are given in the Appendix), §2.2

describes the Conditional Bayesian Mixture of Experts Model (BME) and explains its parameter learning algorithm; §2.3 shows how to construct structured predictors and restrict inference to low-dimensional kernel-induced state spaces (kBME). In §3 we describe experiments on both synthetic and real image sequences, and evaluate both high-dimensional and low-dimensional models. We conclude and discuss future research directions in §4. The work is based on our previous results in [42, 41, 45, 44].

Terminology: We refer to the full modeling framework in §2, consisting of a conditional Markov Model with local distributions represented as conditional Bayesian Mixture of Experts (BME) as BM^3E. Its low-dimensional version based on local kBME conditionals is referred as kBM^3E.

2 Formulation of the BM^3E Model

We work with a conditional graphical model with chain structure, shown in fig. 1a. This has continuous temporal states $\mathbf{x}_t$ and observations $\mathbf{r}_t$, $t = 1 \ldots T$. For notational compactness, we write joint states as $\mathbf{X}_t = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t)$, and joint observations as $\mathbf{R}_t = (\mathbf{r}_1, \ldots, \mathbf{r}_t)$. For learning and inference we model local conditionals: $p(\mathbf{x}_t|\mathbf{r}_t)$, and $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$.

2.1 Discriminative Density Propagation

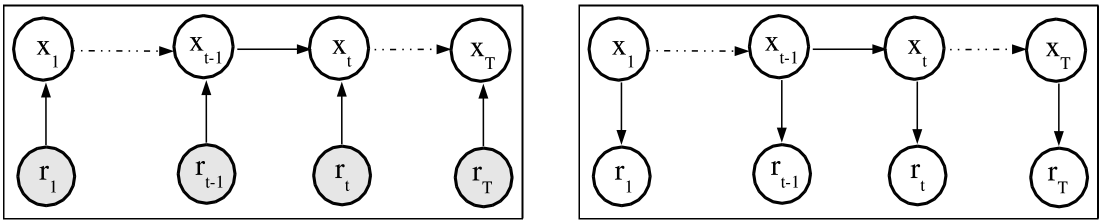

Figure 1: A conditional temporal chain model *(a, left)* reverses the direction of the arrows that link the state and the observation (shaded nodes indicate variables that are not modeled – only instantiated) compared with a generative one *(b, right)*. The state conditionals $p(\mathbf{x}_t|\mathbf{r}_t)$ or $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ can be learned using supervised methods and predicted during inference. Instead, a generative approach *(b)* will model and learn $p(\mathbf{r}_t|\mathbf{x}_t)$ and do more complex probabilistic inference to invert it to $p(\mathbf{x}_t|\mathbf{r}_t)$ using Bayes' rule.

For filtering, we compute the optimal state distribution $p(\mathbf{x}_t|\mathbf{R}_t)$, conditioned by observations

$\mathbf{R}_t$ up to time t. The filtered density can be derived using the conditional independence assumptions implied by the graphical model in fig. 1a, as follows:

$$p(\mathbf{x}_t|\mathbf{R}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) \mathbf{d}\mathbf{x}_{t-1} \tag{1}$$

Similarly, the joint distribution:

$$p(\mathbf{X}_T|\mathbf{R}_T) = p(\mathbf{x}_1|\mathbf{r}_1) \prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t) \tag{2}$$

The detailed derivations of (1) and (2) are given in the Appendix.[2]

In practice, we model $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ as a conditional Bayesian mixture of M experts (*c.f.* §2.2). The prior $p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$ is also represented as a Gaussian mixture with M components. To compute the filtered posterior, we integrate M^2 pairwise products of Gaussians analytically [39]. The means of the M^2-size posterior and used to initialize a fixed M-size component Kullback-Leibler approximation, refined using variational optimization [41].

Remark: A conditional $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ can be in practice more sensitive to incorrect previous state estimates than 'memoryless' models $p(\mathbf{x}_t|\mathbf{r}_t)$. We assume, as in any probabilistic approach, that training and testing data are representative samples of the underlying distributions in the domain. To improve robustness, it is straightforward to include an importance sampler based on $p(\mathbf{x}_t|\mathbf{r}_t)$ to eq. (1), effectively sampling from a mixture of observation-based and dynamics-observation based state conditionals – as we also use for initialization (see §3).[3] It is also useful to correct out-of-sample observations $\mathbf{r}_t$ (caused *e.g.* by inaccurate silhouettes due to shadows) by projecting onto $p(\mathbf{r})$. Out of sample inputs or high entropy filtered posteriors can be indicative heuristics for the

[2] Eqs. (1) and (2) can be derived more generally, based on a predictive conditional dependent on a longer window of observations up to time t [42]. The advantage of these models has to be contrasted to: *(i)* Increased amount of data required for training due to higher dimensionality. *(ii)* Increased difficulty to generalize due to sensitivity to timescale and / or alignment with a long sequence of past observations.

[3] For the directed conditional model in fig. 1a), the filtered posterior is equal to the joint posterior, hence the influence of future observations on past state estimates is eliminated. In certain directed, discrete conditional models used in text processing, *e.g.* MEMMs [31], this model can encounter effects caused 'label-bias'. In BM^3E, these would only occur in conjunction with incorrectly learned conditionals, but such failures would be harmful anyway, in any model. In MEMMs [31], 'label-bias' occurs in models with sparse (as opposed to dense) state space transitions matrices, whenever critical inter-state paths are absent, arguably, primarily a local conditional design and training problem.

loss of track, or absence of the target from scene.

2.2 Conditional Bayesian Mixture of Experts Model (BME)

This section describes models to represent multimodal conditional distributions and algorithms for learning their parameters. We model $p(\mathbf{x}_t|\mathbf{r}_t)$ for initialization or recovery from failure, and $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ for density propagation, *c.f.* (1).

Representation: To accurately model multivalued image-state relations, we use several 'experts' that are simple function approximators. The experts process their inputs[4] and produce state predictions based on their parameters. Predictions from different experts are combined in a probabilistic Gaussian mixture with centers at predicted values. The model is consistent across experts and inputs, *i.e.* the mixing proportions of the experts reflect the distribution of the outputs in the training set and they sum to 1 for every input. Certain input domains are predicted competitively by multiple experts and have multimodal state conditionals. Other 'unambiguous' input regions are predicted by a single expert, with the others effectively switched-off, having negligible probability (see fig. 3). This is the rationale behind a conditional Bayesian mixture of experts, a powerful model for representing complex multimodal state distributions contextually. Formally, the model is:

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}, \mathbf{\Omega}, \boldsymbol{\lambda}) = \sum_{i=1}^{M} g(\mathbf{r}|\boldsymbol{\lambda}_i) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1}) \tag{3}$$

with:

$$g(\mathbf{r}|\boldsymbol{\lambda}_i) = \frac{f(\mathbf{r}|\boldsymbol{\lambda}_i)}{\sum_{k=1}^{M} f(\mathbf{r}|\boldsymbol{\lambda}_k)} \tag{4}$$

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i) = \mathcal{N}(\mathbf{x}|\mathbf{W}_i \mathbf{\Phi}(\mathbf{r}), \mathbf{\Omega}_i^{-1}) \tag{5}$$

[4] The 'inputs' can be either observations $\mathbf{r}_t$, when modeling $p(\mathbf{x}_t|\mathbf{r}_t)$ or observation-state pairs $(\mathbf{x}_{t-1}, \mathbf{r}_t)$ for $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. The 'output' is the state throughout. Temporal information is used to learn $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$.

where $\mathbf{r}$ are input or predictor variables, $\mathbf{x}$ are outputs or responses, g are *input dependent* positive gates, computed using functions $f(\mathbf{r}|\boldsymbol{\lambda}_i)$, parameterized by $\boldsymbol{\lambda}_i$. f has to produce gates g within $[0, 1]$, the exponential and the softmax functions are typical:

$$g(\mathbf{r}|\boldsymbol{\lambda}_i) = \frac{e^{\boldsymbol{\lambda}_i^\top \mathbf{r}}}{\sum_k e^{\boldsymbol{\lambda}_k^\top \mathbf{r}}} \tag{6}$$

Notice how g are normalized to sum to 1 for consistency, by construction, for any given input $\mathbf{r}$. In the model, p are Gaussian distributions (5) with covariances $\boldsymbol{\Omega}_i^{-1}$, centered at different 'expert' predictions, here kernel ($\boldsymbol{\Phi}$) regressors with weights $\mathbf{W}_i$. We work in a Bayesian setting [29, 51, 7], where the weights $\mathbf{W}_i$ (and the gates $\boldsymbol{\lambda}_i$), are controlled by hierarchical priors, typically Gaussians with 0 mean, and having inverse variance hyperparameters $\boldsymbol{\alpha}_i$ (and $\boldsymbol{\beta}_i$) controlled by a second level of Gamma distributions. This gives an automatic relevance determination mechanism [29, 51] which avoids overfitting and encourages compact models with a small number of non-zero weights for efficient prediction. The parameters of the model, including experts and gates are collectively stored in $\boldsymbol{\theta} = \{(\mathbf{W}_i, \boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) \mid i = 1 \ldots M\}$. The graphical model at two different levels of detail is shown in fig. 2.

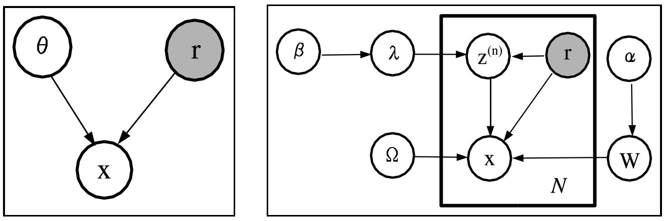

Figure 2: The graphical model of a conditional Bayesian mixture of experts. *(a) Left* shows the model block; *(b) Right* gives a detail with the parameters and the hidden variables included (see text). Shadowed nodes indicate variables that are not modeled, but conditioned upon (instantiated).

Inference (state or output prediction) directly uses (3). The result is a conditional mixture distribution with input-dependent components and mixing proportions. In fig. 3 we explain the model using an illustrative toy example, and show its relation with clusterwise and univalued regression.

Learning the conditional mixture of experts involves two levels of optimization. We describe the general procedure, and refer the reader to [42] for additional derivations and discussion on models

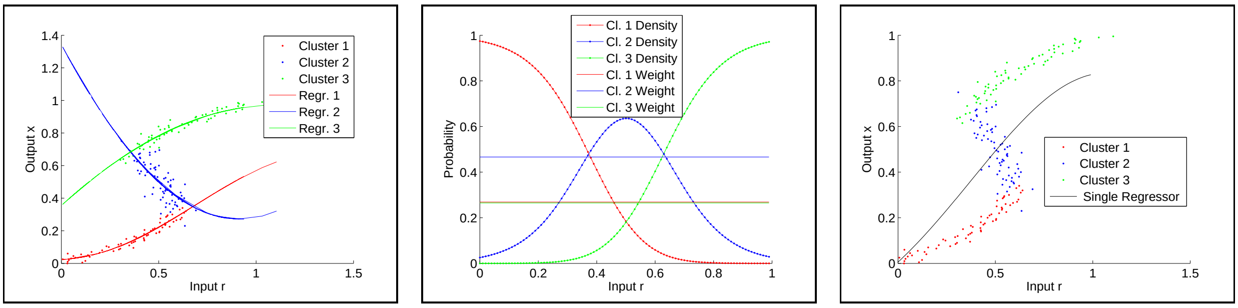

Figure 3: An illustrative dataset [7] consists of about 250 values of x generated uniformly in $(0, 1)$ and evaluated as $r = x + 0.3\sin(2\pi x) + \epsilon$, with ϵ drawn from a zero mean Gaussian with standard deviation 0.05. Notice that $p(x|r)$ is multimodal. *(a) Left* shows the data colored by the posterior membership probability h (7) of three expert kernel regressors. *(b) Middle* shows the gates g (6), as a function of the input, but also the three uniform probabilities (of the joint distribution) that are computed by a clusterwise regressor [13, 33]. *(c) Right* shows how a single kernel regressor cannot represent a multimodal distribution – it may either average the data or zig-zag through its multiple branches, depending on the kernel parameters.

and learning algorithms. As in many prediction problems, we optimize the parameters $\boldsymbol{\theta}$, to maximize the log-likelihood of a data set, $\mathcal{T} = \{(\mathbf{r}_i, \mathbf{x}_i) \mid i = 1 \ldots N\}$, *i.e.* the accuracy of predicting $\mathbf{x}$ given $\mathbf{r}$, averaged over the data distribution. For learning, a full Bayesian treatment requires the computation of posterior distributions over parameters and hyperparameters. Because exact computations are intractable, we design iterative, approximate Bayesian EM algorithms, based on type-II maximum likelihood [29, 51]. These use Laplace approximation for the hyperparameters and analytical integrate the weights, which in this setting become Gaussian [29, 51]. The algorithm proceed as follows. In the E-step we estimate the posterior:

$$h(\mathbf{x}, \mathbf{r}|\mathbf{W}_i, \boldsymbol{\Omega}_i, \boldsymbol{\lambda}_i) = \frac{g(\mathbf{r}|\boldsymbol{\lambda}_i) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \boldsymbol{\Omega}_i^{-1})}{\sum_{j=1}^{M} g(\mathbf{r}|\boldsymbol{\lambda}_j) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_j, \boldsymbol{\Omega}_j^{-1})} \tag{7}$$

This computes the probability that expert i has generated the datapoint n, and requires knowledge of both inputs and outputs (there is one $h_i^{(n)}$ variable for each expert-training pair). The data generation process assumes N datapoints are produced by one of M experts, selected in a stochastic manner. This is modeled by indicator (hidden) variables which are turned-on if the datapoint $\mathbf{x}^{(n)}$ has been produced by expert i and turned-off otherwise. In the M-step we solve two optimization problems, one for each expert and one for its gate. The first learns the expert parameters

$(\mathbf{W}_i, \mathbf{\Omega}_i)$, based on training data $\mathcal{T}$, weighted according to the current membership estimates h (the covariances $\mathbf{\Omega}_i$ are estimated from expert prediction errors [55]). The second optimization teaches the gates g how to predict h.[5] The solutions are based on ML-II, with greedy (expert weight) subset selection. This strategy aggressively sparsifies the experts by eliminating features[6] with small weights after each iteration [51, 57, 26]. This computation can be viewed as a limiting series of variational approximations (Gaussians with decreasing variances), based on dual forms in weight space [57]. The double-loop algorithm is summarized below [24, 55, 42]:

1. **E-step:** For each data pair $\{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \,|\, n = 1 \dots N\}$ compute posteriors $h_i^{(n)}$ for each expert $i = 1 \dots M$, using the current value of parameters $(\mathbf{W}_i, \boldsymbol{\lambda}_i, \mathbf{\Omega}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$.

2. **M-step:** For each expert, solve weighted regression problem with data $\{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \mid n = 1 \dots N\}$ and weights $h_i^{(n)}$ to update $(\mathbf{W}_i, \boldsymbol{\alpha}_i, \mathbf{\Omega}_i)$. This uses Laplace approximation for the hyperparameters and analytical integration for the weights, and optimization with greedy weight subset selection [51, 26].

3. **M-step:** For each gating network i, solve regression problem with data $(\mathbf{r}^{(n)}, h_i^{(n)})$ to update $(\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)$. This maximizes the cross-entropy between g and h, with sparse gate weight priors, and greedy subset selection [51, 26]. We use Laplace approximation for both the hyperparameters and the weights.

4. Iterate using the updated parameter values $\boldsymbol{\theta} = \{(\mathbf{W}_i, \boldsymbol{\alpha}_i, \mathbf{\Omega}_i, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) \mid i = 1 \dots M\}$.

[5] Prediction based on the input *only* is essential for runtime state inference, when membership probabilities (7) cannot be computed as during learning, because the output is missing.

[6] The selected ‘features’ are either specific examples for kernel-based predictors or components of the observation descriptor for linear predictors. Sparse kernel predictors eliminate samples in the training set but leave the input feature vector unchanged, whereas linear predictors work with the entire training set, but eliminate entries in the input.

2.3 Learning Bayesian Mixtures in Kernel Induced State Spaces (kBME)

In this section we introduce low-dimensional extentions to the original BM^3E model in order to improve its computational efficiency in certain visual tasks. As introduced, BM^3E operates in the selected state and observation spaces. Because these can be task generic, therefore redundant, and often high-dimensional, temporal inference can be more expensive or less robust. For many visual tracking taks, low-dimensional models are appropriate. *E.g.* , the components of the joint angle state and the image observation vector are correlated in many human activities with repetitive structure like walking or running. The low intrinsic dimensionality makes a high-dimensional model of 50+ human joint angles non-economical.

In order to model conditional mappings between high-dimensional spaces with strongly correlated dimensions, we rely on kernel non-linear dimensionality reduction and conditional mixture prediction, as introduced in §2.2. Earlier research by Weston *et al* [56] introduced Kernel Dependency Estimation (KDE), a powerful univalued structured predictor. This decorrelates the output using kernel PCA and learns a ridge regressor between the input and each decorrelated output dimension. Our procedure is also based on nonlinear methods like kernel PCA [35], but takes into

$$\begin{tikzcd}
\mathbf{z} \in \mathcal{P}(\mathcal{F}_r) \arrow[r, "p(\mathbf{y}|\mathbf{z})"] & \mathbf{y} \in \mathcal{P}(\mathcal{F}_x) \arrow[dr] & \\
\mathbf{\Phi}_r(\mathbf{r}) \subset \mathcal{F}_r \arrow[u, "PCA"] & \mathbf{\Phi}_x(\mathbf{x}) \subset \mathcal{F}_x \arrow[u, "PCA"] & \mathbf{x} \approx \text{PreImage}(\mathbf{y}) \arrow[d] \\
\mathbf{r} \in \mathcal{R} \subset \mathbb{R}^r \arrow[u, "\mathbf{\Phi}_r"] & \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^x \arrow[u, "\mathbf{\Phi}_x"] & p(\mathbf{x}|\mathbf{r}) \approx p(\mathbf{x}|\mathbf{y})
\end{tikzcd}$$

Figure 4: The learned low-dimensional predictor, kBME, for computing $p(\mathbf{x}|\mathbf{r}) \equiv p(\mathbf{x}_t|\mathbf{r}_t), \forall t$. (We similarly learn $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$, with input $(\mathbf{x}, \mathbf{r})$ instead of $\mathbf{r}$ – here we illustrate only $p(\mathbf{x}|\mathbf{r})$ for clarity.) The input $\mathbf{r}$ and the output $\mathbf{x}$ are decorrelated using Kernel PCA to obtain $\mathbf{z}$ and $\mathbf{y}$ respectively. The kernels used for the input and output are $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_x$, with induced feature spaces $\mathcal{F}_r$ and $\mathcal{F}_x$, respectively. Their principal subspaces obtained by kernel PCA are $\mathcal{P}(\mathcal{F}_r)$ and $\mathcal{P}(\mathcal{F}_x)$. A conditional Bayesian mixture of experts $p(\mathbf{y}|\mathbf{z})$ is learned using the low-dimensional representation $(\mathbf{z}, \mathbf{y})$. Using learned local conditionals of the form $p(\mathbf{y}_t|\mathbf{z}_t)$ or $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)$, temporal inference can be efficiently performed in a *low-dimensional kernel induced state space*. This uses (1) with $\mathbf{y} \leftarrow \mathbf{x}$ and $\mathbf{z} \leftarrow \mathbf{r}$. For visualization and error measurement, the filtered density $p(\mathbf{y}_t|\mathbf{Z}_t)$ is transferred to $p(\mathbf{x}_t|\mathbf{R}_t)$ using the pre-image, *c.f.* (9).

account the structure of our monocular visual perception problem, where both the inputs and the outputs may be low-dimensional and the mapping between them multivalued. The output variables $\mathbf{x}_i$ are projected onto the column vectors of the principal space in order to obtain their principal coordinates $\mathbf{y}_i$. A similar procedure is performed on the inputs $\mathbf{r}_i$ to obtain $\mathbf{z}_i$. In order to relate the reduced feature spaces of $\mathbf{z}$ and $\mathbf{y}$ ($\mathcal{P}(\mathcal{F}_r)$ and $\mathcal{P}(\mathcal{F}_x)$), we estimate a probability distribution over mappings from training pairs $(\mathbf{z}_i, \mathbf{y}_i)$. As in §2.2, we use a conditional Bayesian mixture of experts (BME) in order to account for ambiguity when mapping similar, possibly identical reduced feature inputs to distant feature outputs, as common in our problem. This gives a *conditional Bayesian mixture of low-dimensional kernel-induced experts (kBME)*:

$$p(\mathbf{y}|\mathbf{z}) = \sum_{i=1}^{M} g(\mathbf{z}|\boldsymbol{\lambda}_i)\mathcal{N}(\mathbf{y}|\mathbf{W}_i\boldsymbol{\Phi}(\mathbf{z}), \boldsymbol{\Omega}_i^{-1}) \tag{8}$$

where $g(\mathbf{z}|\boldsymbol{\lambda}_i)$ is a softmax function parameterized by $\boldsymbol{\lambda}_i$ and $(\mathbf{W}_i, \boldsymbol{\Omega}_i^{-1})$ are the parameters and output covariance of expert i, here a kernel regressor, as before (3).

The kernel-induced kBME model requires the computation of pre-images in order to recover the state distribution $\mathbf{x}$ from its image $\mathbf{y} \in \mathcal{P}(\mathcal{F}_x)$. This is a closed form computation for polynomial kernels of odd degree. In general, for other kernels, optimization or learning (regression based) methods are necessary [5]. Following [5, 56], we use a sparse Bayesian kernel regressor to learn the pre-image. This is based on training data $(\mathbf{x}_i, \mathbf{y}_i)$:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\mathbf{A}\boldsymbol{\Phi}_y(\mathbf{y}), \boldsymbol{\Sigma}^{-1}) \tag{9}$$

with parameters and covariances $(\mathbf{A}, \boldsymbol{\Sigma}^{-1})$. Since temporal inference is performed in the low-dimensional kernel induced state space, the pre-image has to be calculated only for visualization or error reporting. The solution is transferred from the reduced feature space $\mathcal{P}(\mathcal{F}_x)$ to the output $\mathcal{X}$ by covariance propagation. This gives a Gaussian mixture with M elements, coefficients $g(\mathbf{z}|\boldsymbol{\lambda}_i)$ and components $\mathcal{N}(\mathbf{x}|\mathbf{A}\boldsymbol{\Phi}_y(\mathbf{W}_i\boldsymbol{\Phi}(\mathbf{z})), \mathbf{A}\mathbf{J}_{\boldsymbol{\Phi}_y}\boldsymbol{\Omega}_i^{-1}\mathbf{J}_{\boldsymbol{\Phi}_y}^{\top}\mathbf{A}^{\top} + \boldsymbol{\Sigma}^{-1})$, where $\mathbf{J}_{\boldsymbol{\Phi}_y}$ is the Jacobian of the mapping $\boldsymbol{\Phi}_y$.

3 Experiments

This section describes our experiments, as well as the training sets and the image features we use. We show results on real and artificially rendered motion capture-based test sequences, and compare with existing methods: nearest neighbor, regression, KDE, both high-dimensional and low-dimensional. The prediction error is reported in degrees (for mixture of experts, this is w.r.t. the most probable one – but see also fig. 9 and fig. 15b) – and normalized per joint angle, per frame. We also report maximum average estimates which are static or temporal averages of the maximum error among all joint angles at particular timestep. The models are learned using standard cross-validation. For kBM^3E, pre-images are learned using kernel regressors with average error 1.7^o.

Database and Model Representation: It is difficult to obtain ground truth for human motion and difficult to train using many viewpoints or lighting conditions. To gather data, we use, as others authors [33, 36, 16, 3, 52], packages like Maya (Alias Wavefront), with realistically rendered computer graphics human surface models, animated using human motion capture [1]. Our human representation ($\mathbf{x}$) is based on an articulated skeleton with spherical joints, and has 56 d.o.f. including global translation (the same model is shown in fig. 5 and used for all reconstructions). The database consists of 8262 individual pose samples obtain from motion sequence clips of different human activities including walking, running, turns, jumps, gestures in conversations, quarreling and pantomime. The training set contains pairs of either states and observations, when learning $p(\mathbf{x}_t|\mathbf{r}_t)$, or states at two succesive timesteps and observations at one of them, when learning $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. Fig. 6 shows data analysis for the database. The data is whithened (this is the format used to train models) and we cluster the input features – either $\mathbf{r}_t$ or $(\mathbf{x}_{t-1}, \mathbf{r}_t)$, and the joint angle vectors – $\mathbf{x}_t$, independently, using k-means. For every sample in the database, its input (either $\mathbf{r}_t$ or $\mathbf{x}_{t-1}, \mathbf{r}_t$) is assigned to the closest input cluster, and its output is assigned to the closest joint angle cluster. Each input cluster stores the *maxium number of different joint angle clusters* selected by samples assigned to it, and we build histograms of the maximum values across all input clusters. The use of many clusters models input perturbations, *e.g.* caused by shadows or different body proportions. The number of joint angle clusters is selected in the order of the expected number of forward-backward ambiguous 'sheets' for monocular human pose – $2^{\#joints} \approx 1000 - 2000$ [48] for a fully sampled pose space. Working with the previous state and the current observation (fig. 6b) and c) does not eliminate uncertainty but this shifts by 2-3 units and peaks higher in the low-mode domain. The ambiguity is severe enough to cause tracking failure or significant errors during initialization. This is shown quantitatively in table 1 and fig. 9. *Ambiguity always increases* with larger training sets, subject body and clothing variability and complex motions. A two-level clustering strategy similar to the one used for the database analysis (fig. 6), is used to initialize the learning of BME models. We initially cluster based on the inputs and then separately cluster the samples within each 'input' cluster, based on the outputs. This tends to avoid cases when single experts would inconsistently represent multiple branches of the inverse pose mapping (see fig. 3), leading to poor models and likelihood optima.

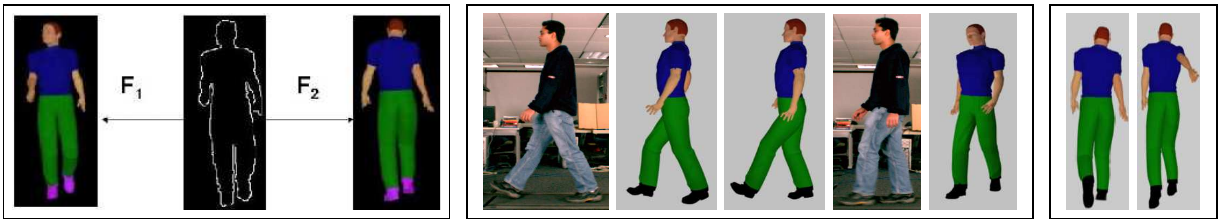

Figure 5: Various ambiguities that make the observation-to-state mapping multivalued. *(a) Left:* Example of 180^o ambiguity in predicting 3D human poses from silhouette image features (center). It is essential that multiple plausible solutions (F_1 and F_2) are correctly represented and tracked over time. A single state predictor will either average the distant solutions or zig-zag between them, see also tables 1 and 2. *(b) Middle:* first three images show leg assignment ambiguities; last two images show a global rotation ambiguity around vertical axis. *(c) Right:* shows two reflective ambiguities obtained by flipping the left and the right knee joints and the right arm shoulder joint.

Image Feature Descriptors: Our choice of image features is based on previously developed methods for texture modeling and object recognition [12, 32, 6, 28]. We mostly work with silhouettes having internal edges, and we assume that in real settings these can be obtained using statistical background subtraction – we use one based on separately built foreground and background models, using non-parametric density estimation [15] and motion segmentation [8]. We use shape context features extracted on the silhouette [6, 32, 3] (5 radial bins, 12 angular bins, with bin size range 1 / 8

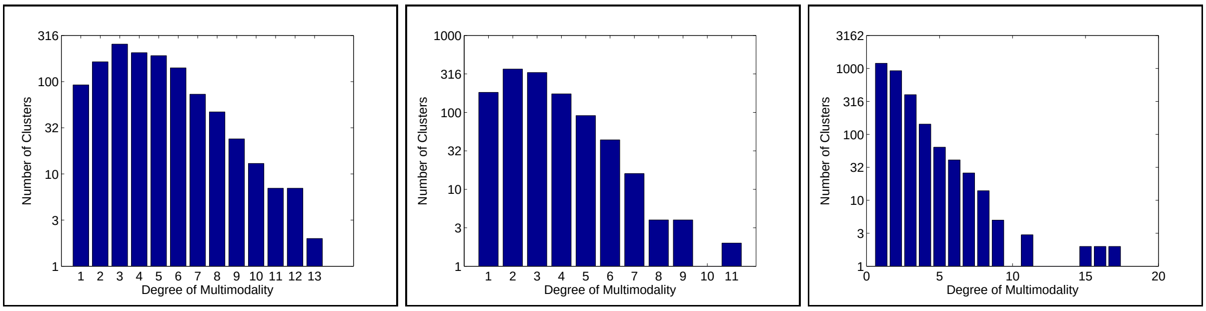

Figure 6: Data analysis for a 8262 sample human motion database. The ‘number of clusters’ axis is shown on a logscale, the input and output vectors have been whitened, as for model training. *(a) Left:* The $\mathbf{x}|\mathbf{r}$ dependency (1209 clusters). *(b) Middle:* Analysis of $\mathbf{x}_t|(\mathbf{x}_{t-1}, \mathbf{r}_t)$ (1203 clusters). *(c) Right:* Same as *(b)* for 2808 clusters. We cluster the input features – either $\mathbf{r}_t$ or $(\mathbf{x}_{t-1}, \mathbf{r}_t)$, and the joint angle vectors $\mathbf{x}_t$, independently, in a large number of clusters using k-means. For every database sample, its input – either $\mathbf{r}_t$ or $(\mathbf{x}_{t-1}, \mathbf{r}_t)$ – is assigned to the closest input cluster, and its output is assigned to the closest joint angle cluster. For each input cluster, we store the maxium number of different joint angle clusters accessed and histogram these across all input clusters.

to 3 on log scale). We compute shape context histograms by sampling features at a variety of scales and sizes on the silhouette. To work in a common coordinate system, we cluster the image features in a representative subset of the images in the training set into k = 60 clusters, using k-means. To compute the representation of a new shape feature (a point on the silhouette), we ‘project’ onto the common basis (vector quantize w.r.t. the codebook) by inverse distance weighted voting into the cluster centers. To obtain the representation $\mathbf{r}$, of a new silhouette we regularly sample about 100-200 points on it and accumulate the feature vectors in a feature histogram. This representation is semi-local, rich and has been effectively demonstrated in many applications, including texture and object recognition [12] or pose prediction [32, 36, 3]. We also experiment with descriptors based on pairwise edge angle and distance histograms [4] and with block SIFT descriptors [28] extracted on a regular image grid and concatenated in a descriptor vector. These are used to demonstrate our method’s ability to produce reliable human pose estimates in images with cluttered backgrounds, when silhouettes are not available. All image descriptors (histogram-based or block-based) are extracted over partially intersecting neighborhoods – hence they are based on *overlapping features of the observation* and have strongly dependent components. In a conditional framework (fig. 1a), this representation is consistent and tractable – differently from the generative case, the observation distribution is not modeled and no simplifying assumptions are necessary.

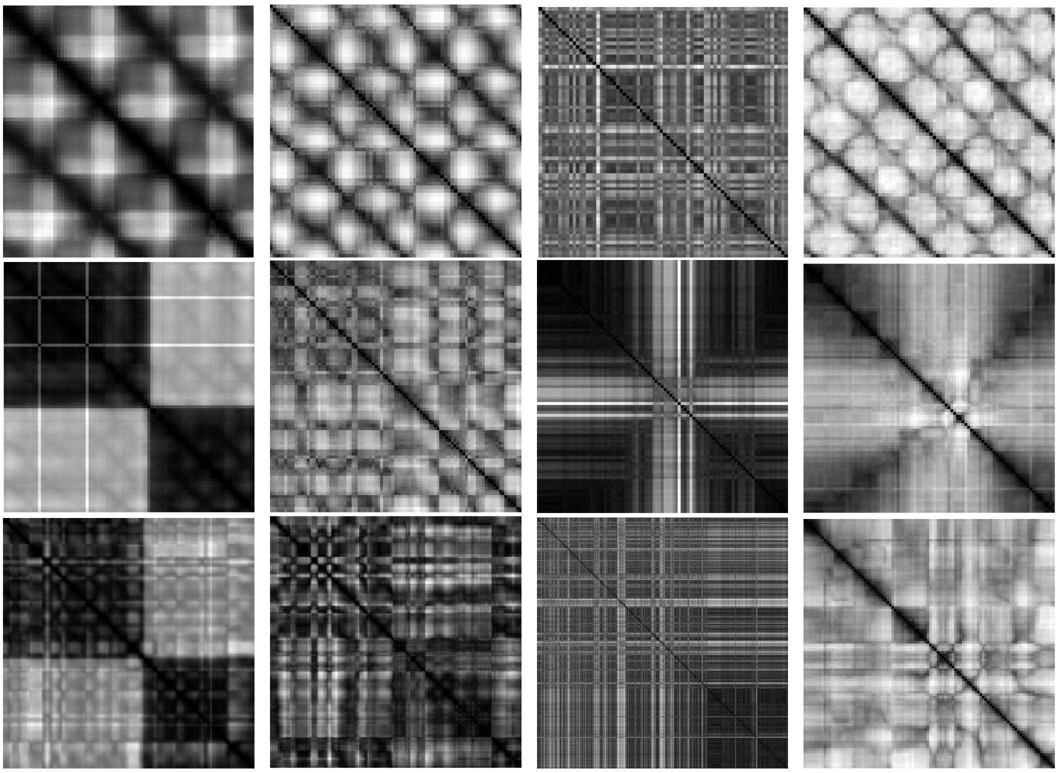

Figure 7: Affinity matrices based on Euclidean distances for temporally ordered silhouettes with internal edges (darker means more similar). From left to right, joint angles (JA), external contour shape context (SC), internal contour pairwise edge (PE), and block sift features (fized sized silhouette bounding box) for different motions: *(a) Top row:* walking parallel to the image plane. Notice the periodicity as well as the higher frequencies in the (SC) matrix caused by half-cycle ambiguities for silhouettes; *(b) Middle row:* complex walk of a subject walking towards the camera and back; *(c) Bottom row:* conversations. The joint angle and image features correlate less intuitively.

3.1 High-dimensional Models

Comparisons: We compare our conditional Bayesian Mixture of Experts Models (BME) with competing methods: nearest neighbor (NN) or the relevance vector machine (RVM), a sparse Bayesian regressor [51]. We test several human activities obtained using motion-capture and artificially rendered. This provides ground truth and allows us to concentrate on the algorithms and factor out the variability given by the imperfections of our human model, or the noise in the silhouette extraction in real images. BME uses 5 modes, non-linear Gaussian kernel experts, with most probable mode selected. The results are shown in table 1. We run two comparisons, one by training separate models for each activity class and testing on it (top half of table 1), the other by training one global model on the entire database and using it to track all motion types (the bottom half of 1). Training and testing is run on motions from different subjects.

Testing separate activity models: We use several training sets: walking diagonal w.r.t. to the image plane (train 300, test 56), complex walk towards the camera and turning back (train 900, test 90), running parallel to the image plane (train 150, test 150), conversation involving some hand movement and turning (train 800, test 160), pantomime (1000 train, 100 test). During testing, we initialize from ground truth. This is necessary for single hypothesis methods (NN, RVM), which may immediately fail following and incorrect initialization, in the dynamic case $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$.

BME gives better average estimates and significantly lower maximum errors. The large maximum error for running is consistent across methods and corresponds to the right hand joint. For comparison we only consider the most probable BME prediction. While the correct solution is not always predicted as the most probable, it is often present among the top modes predicted, see fig. 13c. For probabilistic tracking, this ‘approximately correct’ behavior is desirable, because the correct solution is often propagated with significant probability.

Testing the global model: We have also built one global model using the entire 8262 motion database and tested on six motion types. We use 7238 samples to train the static state predictor and 7202 samples to train the dynamic predictor $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. Testing is based on 2-fold cross validation with test set sizes: normal walk - 55 Frames, complex walk - 100 frames, running - 150 frames, conversation – 100 frames, pantomime – 200 frames, dancing 270 frames. For these experiments *only* we use conditional models based on 10 linear (as opposed to Gaussian kernel) experts and a 200d shape context feature vector made of two 100d histograms computed separately for the contour and internal edge features (this improved performance over a global histogram computed on the entire silhouette). Results are shown in the bottom-half of table 1 and in fig. 9. As expected, all methods have larger errors compared to the easier case of separately trained and tested activity models. BME is robust and outperforms its competitors on this large and diverse database.

Sequence	$p(\mathbf{x}_t \vert \mathbf{r}_t)$			$p(\mathbf{x}_t \vert \mathbf{x}_{t-1}, \mathbf{r}_t)$		
	NN	RVM	BME	NN	RVM	BME
Normal walk	4 / 20	2.7 / 12	2 / 10	7 / 25	3.7 / 11.2	2.8 / 8.1
Complex walk	11.3 / 88	9.5 / 60	4.5 / 20	7.5 / 78	5.67 / 20	2.77 / 9
Running	7 / 91	6.5 / 86	5 / 94	5.5 / 91	5.1 / 108	4.5 / 76
Conversation	7.3 / 26	5.5 / 21	4.15 / 9.5	8.14 / 29	4.07 / 16	3 / 9
Pantomime	7 / 36	7.5 / 53	6.5 / 25	7.5 / 49	7.5 / 43	7 / 41
Normal walk	15.8 / 179.5	9.54 / 72.9	7.41 / 128.5	5.79 / 164.8	8.12 / 179.4	3.1 / 94.5
Complex walk	17.7 / 178.6	15 / 179.8	8.6 / 178.8	17.8 / 178.6	9.5 / 179.9	7.7 / 134.9
Running	20.1 / 178.2	10.6 / 76.8	5.9 / 177.4	9.3 / 64.9	8.64 / 76.8	3.3 / 59.5
Conversation	12.9 / 177.4	12.4 / 179.9	9.9 / 179.7	12.8 / 88.8	10.6 / 179.9	6.13 / 94.3
Pantomime	20.6 / 177.4	17.5 / 176.4	13.5 / 178.5	21.1 / 177.4	11.1 / 119.9	7.4 / 119.2
Dancing	18.4 / 179.9	20.3 / 179.9	14.3 / 179.9	25.6 / 179.9	14.9 / 149.8	6.26 / 124.6

Table 1: Comparative results showing RMS errors per joint angle (average error / maximum joint average error) in degrees for two conditional models, $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. We compare three different algorithms on motion-capture, synthetically generated test data (we select the best candidate for each test input, there is no probabilistic tracking, but $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ has memory). *The top table* shows result obtained by training separate activity models each sequence and testing on motions in their class (BME uses 5 Gaussian kernel experts). *Bottom table* **(motion types in bold)** shows results obtained by training one single global model on the entire 8262 sample database. BME models are based on 10 sparse linear experts, RVM uses one sparse linear expert. In all tests, accuracy is reported w.r.t. the most probable expert for BME, but see also fig. 9.

The dancing and pantomime sequences are the most difficult due to their inherently higher semantic variability (compared to walking say), and given our training and testing setting based on motions captured from different subjects. While BME's 'best expert' errors are sometimes large, these decrease substantially when measuring prediction error in any of the best-k most probable experts – qualitatively illustrated in fig. 9. The average error of the best (most probable) expert is $\approx 14.3^o$, but the error in the best 3 experts is under 10^o, and the error in the best 7 experts is under 5^o. This shows that a BME model can generalize well even for large motion variability at some decrease in the confidence of its most probable predictions. An illustration of BM^3E tracking (with 5-mode filtered posteriors), applied to a dancing sequence, is given in fig. 8. Notice the large joint angle trajectory separation in state space, and the different types of multimodality, including well separated paths (fig. 8a), bundles (b) or merge / splits (c).

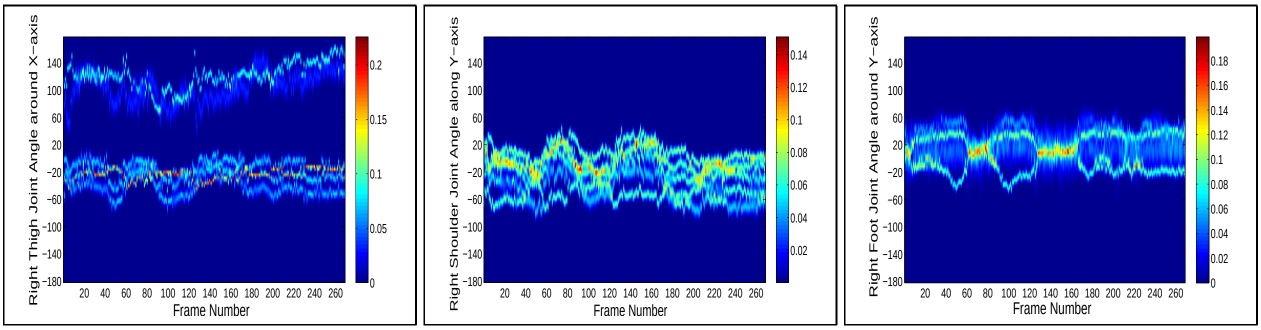

Figure 8: (Best viewed in color) Illustration of the BM^3E tracker in a dancing sequence, with 5 mode posterior, computed using (1). Time is unfolded on the horizontal axis, filtered density at timestep on the vertical (showing one selected variable), probability is color coded. Notice different types of multimodality, including well separated paths (*a*, *left*), bundles (*b*, *middle*) and merge / splits (*c*, *right*).

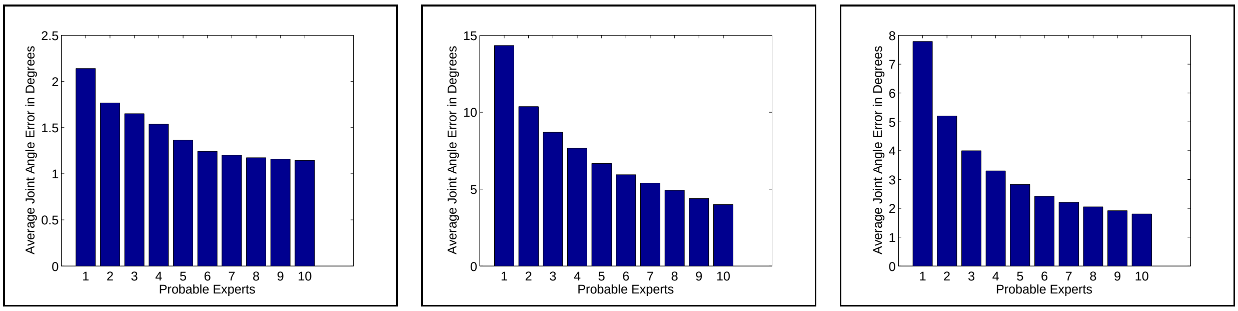

Figure 9: Reconstruction error in the ‘best k’ experts ($k = 1 \dots 10$) for a global model, trained on the 8262 sample database – see also the bottom half of table 1. Prediction accuracy is here *not* only computed only w.r.t. the most probable expert (1st bar on the left), but w.r.t. the best-k – we measure the expert prediction closest to the ground truth with cutoff at the k-th most probable (each error level is obtained for a different k). *(a) Left and (b) Middle:* train and test errors for dancing. *(c) Right:* test errors for a person walking towards the camera, turning 180^o and going back. In testing, the most probable expert may not always be reliable, but prediction from top ones is.

Real Image Sequences. Walking, Picking and Dancing: We track using BM^3E with 5 mode posteriors and local BME conditionals based on 5 experts, with RBF kernels and degree of sparsity varying between 5%-25%. Fig. 10 shows a succesful reconstruction of walking – the frames are from a 3s sequence, 60fps. Occasionally, there are leg assignment ambiguities that can confuse a single hypothesis tracker, as visible in fig. 5, or in the affinity matrix of the image descriptor (fig. 7). While the affinity matrices of 3d joint angles and image features for walking correlate well (fig. 7a) (far better that for other motions like conversations or complex walking), the higher frequency in the image affinity sub-diagonal bands illustrate the silhouette ambiguities at walking half-cycles.

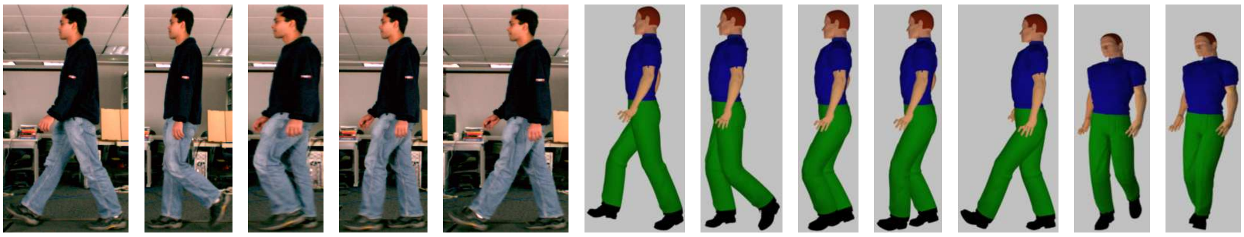

Figure 10: Reconstruction of walking. *(a) Left:* original images; *(b) Middle:* reconstruction rendered from the same viewpoint used in training. *(c) Right:* Images showing the reconstruction from a different viewpoint.

In fig. 12, we show reconstruction results from a 2s video filmed at 60 fps, where a subject mimicks the act of picking an object from the floor. We experiment with both Bayesian single hypothesis tracking (single expert local conditionals), propagated using (1), as well as BM^3E. The single hypothesis tracker follows the beginning of the sequence but fails shortly after, when its input kernels stop firing due to an out-of-range state input predicted from the previous timestep (fig. 11).[7]

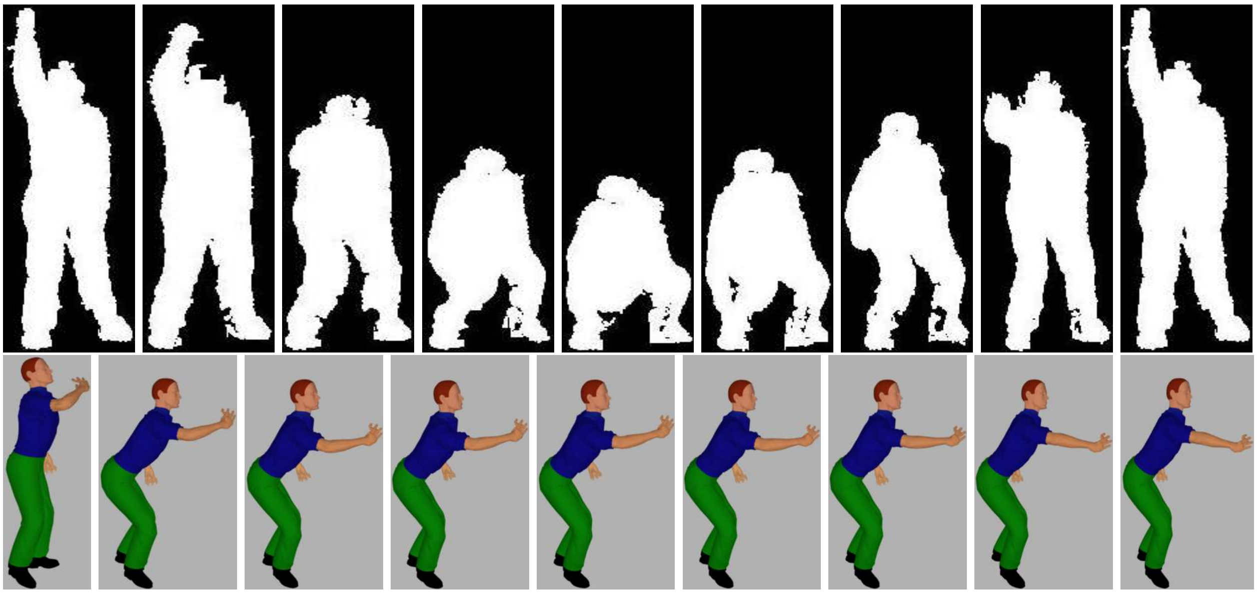

Figure 11: A single hypothesis Bayesian tracker based on (1) fails to reconstruct the sequence in fig. 12 *(bottom row)* even when presented with only the silhouettes *(top row)*. In the beginning, the tracker follows the motion, but fails shortly after, by generating a prediction out of its input kernel firing range. The track is lost with the expert locked to its bias joint angle values.

In fig. 12 we show results obtained with the 5 mode BM^3E tracker. This was also able to re-

[7] We initialize using a BME for $p(\mathbf{x}_t|\mathbf{r}_t)$. For single hypothesis tracking, we select the most probable component.

construct the pose based on silhouettes alone (shown in the top row of fig. 11), but here we show a more difficult sequence where the person has been placed on a dynamically changing background and the tracker had no silhouette information. In this case, we use block SIFT image features and 5 linear experts for local BME. The model is trained on 1500 samples from picking motions, syntehtically rendered on natural backgrounds, using our 3d human model. A rectangular bounding box containing both the person and the background is used as input and SIFT descriptors are extracted on a regular grid, typically over both foreground and background regions. Block features, linear experts and training with different image backgrounds are effective in order to build models that are resistent to clutter. Block descriptors are more appropriate than bag-of-feature, global histograms – during training the model learns to downgrade image blocks that contain only background. Regions with oscillatory foreground / background labels are assigned to different experts. The reconstruction is perceptually plausible, but there are imperfections, possibly reflecting the bias introduced by our training set – notice that the knee of the model is tilted outward whereas the knee of the human is tilted inward. We observe persistent multimodality for joints actively moving, *e.g.* the left and right femur and the right shoulder.

In fig. 14, we show reconstruction experiments for a real dancing sequence, with quantitative results given in fig. 13. We train on 300 synthetic motion samples and test on 100 images of a real video. Our test subject (an author of this paper) has watched the motion capture video and tried to imitate it. Given the complexity of the motion, the training and testing data is inherently different. Our tracker generalizes well and succeeds in capturing the real 3d motion in a perceptually plausible way. There are, however, noticeable imperfections in the reconstruction, *e.g.* in the estimates of the arms and legs.

3.2 Low-dimensional Models

We learn kBME conditionals (§2.3) and reconstruct human pose in a low-dimensional kernel induced state space, using the kBM^3E tracker. Gaussian kernels are used for kernel PCA. We learn kBME with 6d kernel induced state spaces and 25d feature spaces. In fig. 15a), we evaluate the

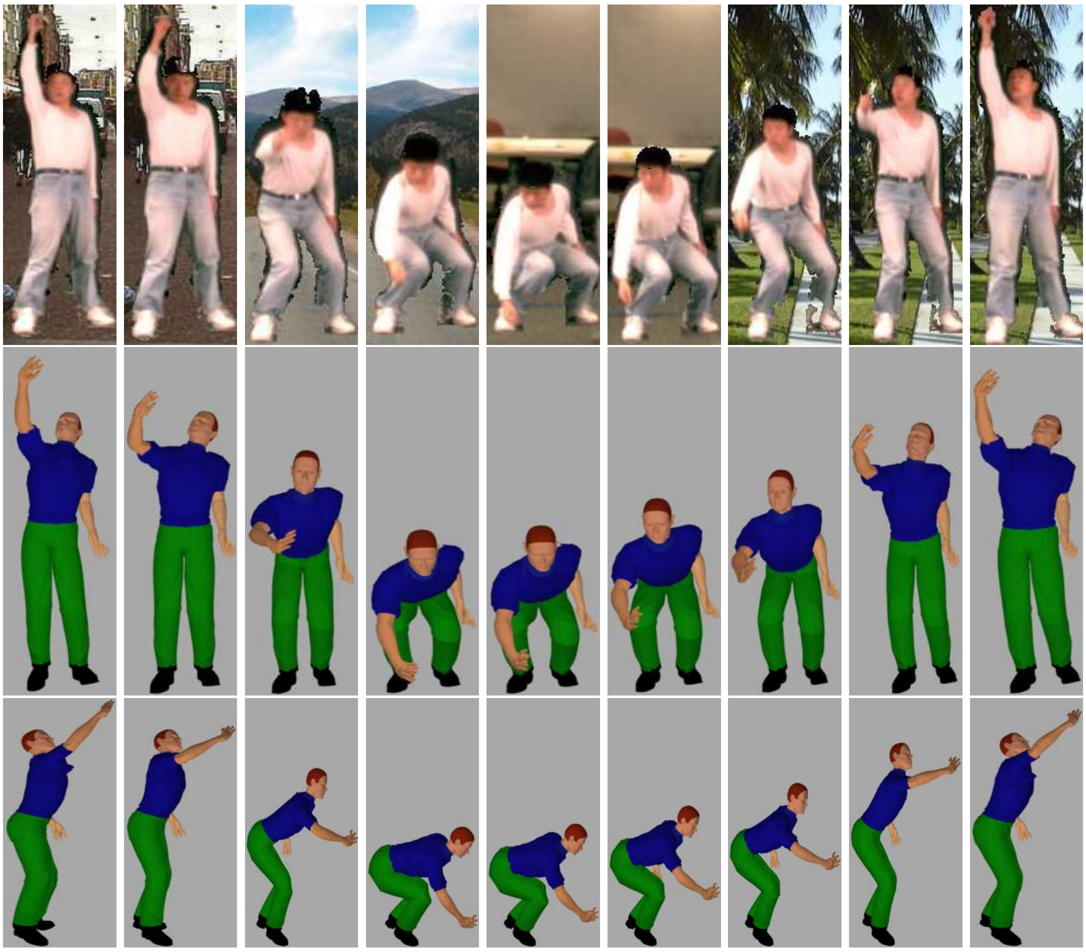

Figure 12: *(a) Top row:* Original image sequence showing the silhouette of the person in fig. 11, placed on different natural scene backgrounds. (The tracker is given a cluttered rectangular bounding box of the person, *not* its silhouette.) *(b) Middle row:* Reconstruction seen from the same viewpoint used for training, *(c) Bottom row:* Reconstruction seen from a synthetic viewpoint. Despite the variablly changing background, BM^3E can reconstruct the motion with reasonable perceptual accuracy. However, there are imperfections, *e.g.* the right knee of the subject is tilted inward, whereas the one of the model is tilted outward). A single hypothesis Bayesian tracker fails even when presented with only silhouettes, see fig. 11.

accuracy of kBME for different state dimensions in a dancing sequence (for this test only, we use a 50d observation descriptor). On dancing, which involves complex motions of the torso, arms and the legs, the non-linear model significantly outperforms alternative PCA methods and gives good predictions for compact, low-dimensional states. In table 2 and fig. 15, we show quantitative comparisons on artificially rendered silhouettes – 3d joint angle ground truth is available for systematic evaluation. The low-dimensional non-linear models kBME outperform PCA-based models,

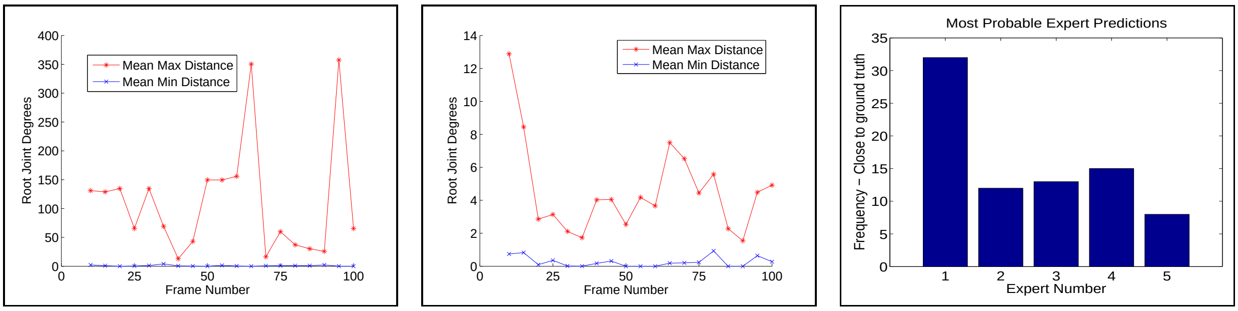

Figure 13: Quantitative 3d reconstruction results for a dancing sequence. *(a) Left:* shows the maximum and minimum distance between the modes of the root joint (vertical axis) rotation angle. The minimum distance is only informatively shown, it does not necessarily reflect modes that will survive the mixture simplification. Most likely, modes that cluster together collapse. *(b) Middle:* same as *(a)* for the left femur. *(c) Right:* shows the good accuracy of BME. Notice that occasionally, its most probable prediction is not the most accurate.

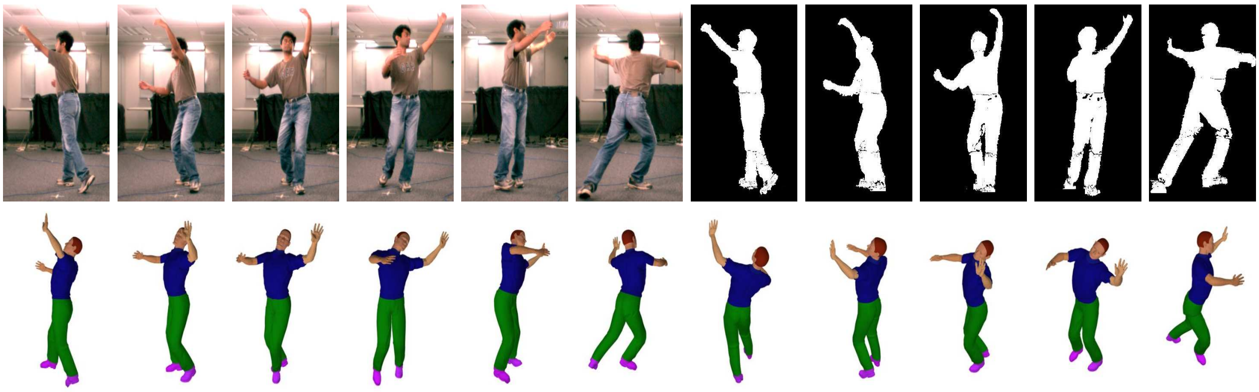

Figure 14: Tracking and 3d reconstruction of a dancing sequence. *(a) Top* row shows original images and silhouettes (the algorithms use both the silhouette contour and the internal image edges); *(b) Bottom* row shows reconstructions from training (left) and new synthetic viewpoint (right).

and give results competitive to high-dimensional BME predictors. But low-dimensionality makes training and tracking less expensive, *c.f.* (1). In fig. 16 and 17 we show human motion reconstructions based on two real image sequences. Fig. 16 shows a person performing an agile jump. Given the missing observations in a side view, the 3d reconstruction of occluded body parts would not be possible without prior knowledge. The sequence in fig. 17 shows simultaneous pose reconstruction for two people mimicking domestic activities – washing a window and picking an object. We track in a 12d state space, obtained by concatenating the 6d state of each person. We reconstruct successfully using only 5 hypotheses, although the results are not perfect – notice errors in the elbow

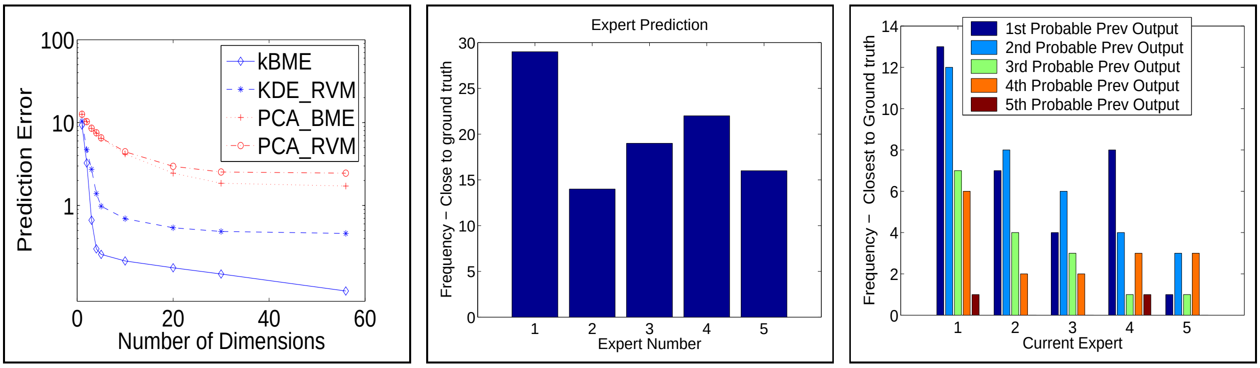

Figure 15: *(a) Left:* Evaluation of dimensionality reduction methods for an artificial dancing sequence (models trained on 300 samples). kBME is discussed in §2.3; KDE-RVM is a Kernel Dependency Estimator (KDE) with a Relevance Vector Machine (RVM) [51] feature space map; PCA-BME and PCA-RVM are models where mappings between feature spaces (obtained with PCA) are learned with BME and RVM. Due to non-linearity, kernel-based methods outperform PCA and give low prediction error for 5-6d models. *(b) Middle:* Histogram showing the accuracy of various expert kBME predictors – how many times the expert ranked as k-th most probable by the model (horizontal axis) is closest to the ground truth. The model is consistent (the most probable expert indeed is the most accurate most frequently), but occasionally less probable experts are better. *(c) Right:* Histograms show the dynamics of $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)$, *i.e.* how the probability mass is redistributed among experts between two successive time steps, in a conversation sequence.

and the bending of the knee of the subject at the left, or in the wrist orientation of the subject at the right.

	KDE-RR	RVM	KDE-RVM	BME	kBME
Walk and turn	10.46	4.95	7.57	4.27	4.69
Conversation	7.95	4.96	6.31	4.15	4.79
Run and turn left	5.22	5.02	6.25	5.01	4.92
Walk and turn back	7.59	6.9	7.15	3.6	3.72
Run and turn	17.7	16.8	16.08	8.2	8.01

Table 2: Comparison of average joint angle prediction error for different models. All kPCA models have 6 output dimensions. Testing is done on 100 video frames for each sequence, with artificially generated silhouette inputs, not in the training set. Existing 3d joint angle ground truth is used for evaluation. KDE-RR is a KDE model with a ridge regression (RR) feature space map, KDE-RVM uses an RVM. BME are the high and low-dimensional models discussed in §2.2 and §2.3. kernelPCA-based methods use kernel regressors for pre-images.

Running times for different models: On a Pentium 4 PC (3 GHz, 2 GB RAM), a full dimensional BME model with 5 experts takes 802s to train $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$, whereas a kBME (including the pre-

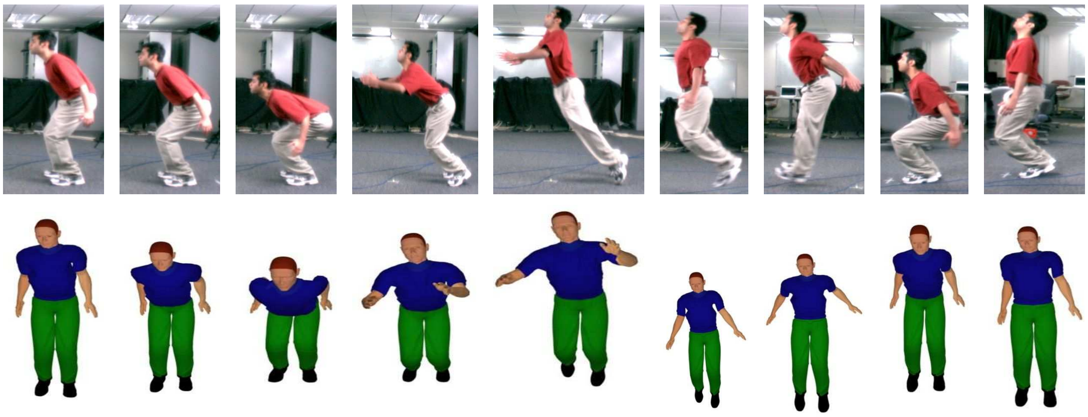

Figure 16: Reconstruction of a jump (scaled selected frames). *(a) Top row:* original image sequence. *(b) Bottom row:* 3D reconstruction seen from a synthetic viewpoint.

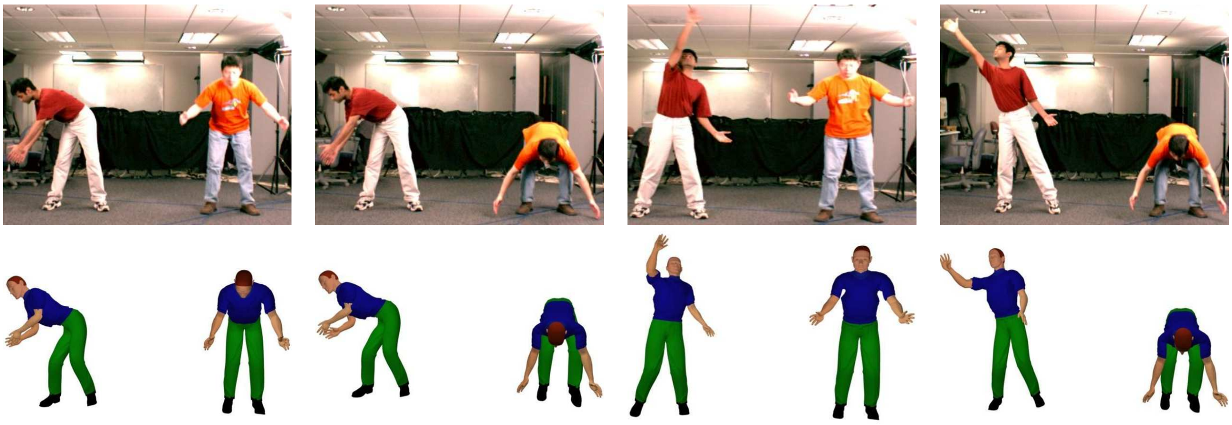

Figure 17: Reconstruction of domestic activities – 2 people operating in an 12d state space (each person has its own 6d state). *(a) Top row:* original image sequence. *(b) Bottom row:* 3d reconstruction seen from a synthetic viewpoint.

image) takes 95s to train $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)$. The prediction time is 13.7s for BME and 8.7s (including the pre-image cost 1.04s) for kBME. The integration in (1) takes 2.67s for BME and 0.31s for kBME. The speed-up of kBME is significant and likely to increase w.r.t. original models having higher dimensionality.

4 Conclusions

We have introduced BM^3E, a framework for discriminative density propagation in continuous state spaces. We argued that existing discriminative methods do not offer a formal management of uncertainty, and explained why current representations cannot model multivalued mappings inherent in 3d perception. We contribute by establishing the density propagation rules in continuous conditional chain models, and by proposing models capable to represent feedforward, multivalued relations contextually. The combined system automatically self-initializes and recovers from failure – it can operate either stand-alone, or as a component to initialize generative inference algorithms. We show results on real and synthetically generated image sequences, and demonstrate significant performance gains with respect to nearest neighbor, regression, and structured prediction methods. Our study suggests that flexible conditional modeling and uncertainty propagation are both important in order to reconstruct complex 3d human motion in monocular video reliably. We hope that our research will provide a framework to analyze discriminative and generative tracking algorithms and stimulate a debate on their relative advantages within a common probabilistic setting. By virtue of its generality, we hope that the proposed methodology will be useful in other 3d visual inference and tracking problems.

Future Work: We plan to investigate alternative model state and observation descriptors that would make possible to reconstruct complex dynamic scenes with occlusion, partial body views, background clutter, and camera motion. We intend to study alternative learning and inference algorithms based on bound optimization. Combining the strengths of generative and discriminative methods remains a promising avenue for future research [46].

Appendix: Filtering and Joint Distribution for Conditional Chains

The filtering recursion (1). The following properties can be verified visually in fig. 1a, using a Bayes ball algorithm [23] (‘$\perp\!\!\!\perp$’ denotes independence, and ‘$|$’ conditioning on)[8]:

[8] The model is conditional, hence no attempt is made to model the observations, which can have arbitrary inner or temporal dependency structure. An arrow-reversed generative model as in fig. 1a, but without instantiated observations, will have a dependency structure with marginally independent temporal observations: $\mathbf{r}_t \perp\!\!\!\perp \mathbf{R}_{t-1}$. This has no effect in a conditional model, where observations are always instantiated. Contrast this with the conditional independence of temporal observations given the states, assumed by temporal generative models (fig. 1b).

$$\mathbf{x}_t \perp\!\!\!\perp \mathbf{X}_{t-2} | \mathbf{x}_{t-1} \tag{10}$$

$$\mathbf{x}_t \perp\!\!\!\perp \mathbf{R}_{t-1} | \mathbf{x}_{t-1}, \mathbf{r}_t \tag{11}$$

$$\mathbf{X}_{t-1} \perp\!\!\!\perp \mathbf{r}_t \tag{12}$$

$$\begin{align} p(\mathbf{x}_t|\mathbf{R}_t) &= \int p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t)\mathbf{dx}_{t-1} = \tag{13}\\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{R}_{t-1}, \mathbf{r}_t)p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t)\mathbf{dx}_{t-1} = \tag{14}\\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})\mathbf{dx}_{t-1} \tag{15} \end{align}$$

where in the last line we used:

$$\begin{align*} \text{(11)} \Rightarrow & p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{R}_{t-1}, \mathbf{r}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)\\ \text{(12)} \Rightarrow & p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) = p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) \end{align*}$$

Remark: It is also possible to use a generative model, but express the propagation rules in terms of discriminative-style conditionals in order to simplify inference [42]:

$$p(\mathbf{x}_t|\mathbf{R}_t) \propto \frac{p(\mathbf{x}_t|\mathbf{r}_t)}{p(\mathbf{x}_t)} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})\mathbf{dx}_{t-1} \tag{16}$$

where $p(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1})\mathbf{dx}_{t-1}$. Implementing (16) requires recursively propagating both $p(\mathbf{x}_t|\mathbf{R}_t)$ and $p(\mathbf{x}_t)$ (an equilibrium approximation could be precomputed[9]), two mixture sim-

[9] Alternatively, the ratio could be estimated.

plification levels, inside the integrand and outside it through the multiplication by $p(\mathbf{x}_t|\mathbf{r}_t)$ and a division by $p(\mathbf{x}_t)$ (see [42] for details).

The joint distribution (2)**.** Using basic conditioning:

$$p(\mathbf{X}_T, \mathbf{R}_T) = p(\mathbf{X}_{T-1}, \mathbf{R}_{T-1}) p(\mathbf{r}_T|\mathbf{X}_{T-1}, \mathbf{R}_{T-1}) p(\mathbf{x}_T|\mathbf{X}_{T-1}, \mathbf{R}_T) \tag{17}$$

The independence of (12) can be used to simplify (17):

$$p(\mathbf{r}_T|\mathbf{X}_{T-1}, \mathbf{R}_{T-1}) = p(\mathbf{r}_T) \tag{18}$$

$$p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{X}_{T-2}, \mathbf{R}_{T-1}, \mathbf{r}_T) = p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{r}_T) \tag{19}$$

Using (18) and (19) in (17), we obtain:

$$p(\mathbf{X}_T, \mathbf{R}_T) = p(\mathbf{X}_{T-1}, \mathbf{R}_{T-1}) p(\mathbf{r}_T) p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{r}_T) \tag{20}$$

and (2) is verified given:

$$p(\mathbf{X}_T|\mathbf{R}_T) = \frac{p(\mathbf{X}_T, \mathbf{R}_T)}{\prod_{t=1}^{T} \mathbf{r}_t} \tag{21}$$

and $p(\mathbf{x}_1|\mathbf{r}_1) = p(\mathbf{x}_1, \mathbf{r}_1)/p(\mathbf{r}_1)$.

Acknowledgments: This work has been supported in part by the National Science Foundation under award number IIS-0535140. CS thanks Allan Jepson for insightful discussions and feedback. The authors thank Zhiguo Li for useful comments, support with the human motion database creation, and for posing as a subject. We thank Ahmed Elgammal for providing statistical background subtraction code and the anonymous reviewers for valuable comments.

References

[1] CMU Human Motion Capture DataBase. Available online at http://mocap.cs.cmu.edu/search.html, 2003.

[2] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Workshop on Vision for Human Computer Interaction*, 2005.

[3] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[4] F. Aherne, N. Thacker, and P. Rocket. Optimal pairwise geometric histograms. In *British Machine Vision Conference*, 1997.

[5] G. Bakir, J. Weston, and B. Scholkopf. Learning to find pre-images. In *Advances in Neural Information Processing Systems*, 2004.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 2002.

[7] C. Bishop and M. Svensen. Bayesian mixtures of experts. In *Uncertainty in Artificial Intelligence*, 2003.

[8] M. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise Smooth Flow Fields. *Computer Vision and Image Understanding*, 6(1):57–92, 1996.

[9] M. Brand. Shadow Puppetry. In *IEEE International Conference on Computer Vision*, pages 1237–44, 1999.

[10] M. Bray, P. Kohli, and P. Torr. Posecut: Simoultaneous segmentation and and 3d pose estimation of humans using dynamic graph cuts. In *European Conference on Computer Vision*, 2006.

[11] K. Choo and D. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In *IEEE International Conference on Computer Vision*, 2001.

[12] O. Cula and K. Dana. 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60, 2004.

[13] W. DeSarbo and W. Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, (5):249–282, 1988.

[14] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000.

[15] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Foreground and background modeling using non-parametric kernel density estimation for visual surveillance. *Proc.IEEE*, 2002.

[16] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[17] N. Gordon, D. Salmond, and A. Smith. Novel Approach to Non-linear/Non-Gaussian State Estimation. *IEE Proc. F*, 1993.

[18] K. Grauman, G. Shakhnarovich, and T. Darell. Inferring 3D structure with a statistical image-based shape model. In *IEEE International Conference on Computer Vision*, 2003.

[19] N. Howe, M. Leventon, and W. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. *Advances in Neural Information Processing Systems*, 1999.

[20] M. Isard and A. Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 1998.

[21] T. Jaeggli, E. Koller-Meier, and L. Van Gool. Monocular tracking with a mixture of view-dependent learned models. In *IV Conference on Articulated Motion and Deformable Objects, AMDO*, pages 494–503, 2006.

[22] T. Jebara and A. Pentland. On reversing Jensen's inequality. In *Advances in Neural Information Processing Systems*, 2000.

[23] M. Jordan. *Learning in graphical models*. MIT Press, 1998.

[24] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, (6):181–214, 1994.

[25] I. Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion with Occlusion Prediction Based on Active Multi-Viewpoint Selection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 81–87, 1996.

[26] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems*, 2003.

[27] M. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[28] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[29] D. Mackay. Bayesian interpolation. *Neural Computation*, 4(5):720–736, 1992.

[30] D. Mackay. Comparison of Approximate Methods for Handling Hyperparameters. *Neural Computation*, 11(5), 1998.

[31] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *International Conference on Machine Learning*, 2000.

[32] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, 2002.

[33] R. Rosales and S. Sclaroff. Learning Body Pose Via Specialized Maps. In *Advances in Neural Information Processing Systems*, 2002.

[34] S. Roth, L. Sigal, and M. Black. Gibbs Likelihoods for Bayesian Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[35] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[36] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing. In *IEEE International Conference on Computer Vision*, 2003.

[37] H. Sidenbladh and M. Black. Learning Image Statistics for Bayesian Tracking. In *IEEE International Conference on Computer Vision*, 2001.

[38] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking Loose-limbed People. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[39] C. Sminchisescu and A. Jepson. Density propagation for continuous temporal chains. Generative and discriminative models. Technical Report CSRG-401, University of Toronto, October 2004.

[40] C. Sminchisescu and A. Jepson. Generative Modeling for Continuous Non-Linearly Embedded Visual Inference. In *International Conference on Machine Learning*, pages 759–766, Banff, 2004.

[41] C. Sminchisescu and A. Jepson. Variational Mixture Smoothing for Non-Linear Dynamical Systems. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 608–615, Washington D.C., 2004.

[42] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Learning to reconstruct 3D human motion from Bayesian mixtures of experts. A probabilistic discriminative approach. Technical Report CSRG-502, University of Toronto, October 2004.

[43] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *IEEE International Conference on Computer Vision*, volume 2, pages 1808–1815, 2005.

[44] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional Visual Tracking in Kernel Space. In *Advances in Neural Information Processing Systems*, 2005.

[45] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative Density Propagation for 3D Human Motion Estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 390–397, 2005.

[46] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Learning Joint Top-down and Bottom-up Processes for 3D Visual Inference. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.

[47] C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[48] C. Sminchisescu and B. Triggs. Kinematic Jump Processes for Monocular 3D Human Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 69–76, Madison, 2003.

[49] C. Sminchisescu and M. Welling. Generalized darting Monte Carlo. Technical Report CSRG-543, University of Toronto, October 2006.

[50] E. Sudderth, A. Ihler, W. Freeman, and A.Wilsky. Non-parametric belief propagation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[51] M. Tipping. Sparse Bayesian learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 2001.

[52] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In *IEEE International Conference on Computer Vision*, 2003.

[53] N. Ueda and Z. Ghahramani. Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15:1223–1241, 2002.

[54] R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking in small training sets. In *IEEE International Conference on Computer Vision*, 2005.

[55] S. Waterhouse, D.Mackay, and T.Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems*, 1996.

[56] J. Weston, O. Chapelle, A. Elisseeff, B. Scholkopf, and V. Vapnik. Kernel dependency estimation. In *Advances in Neural Information Processing Systems*, 2002.

[57] D. Wipf, J. Palmer, and B. Rao. Perspectives on Sparse Bayesian Learning. In *Advances in Neural Information Processing Systems*, 2003.