Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 108 (2007) 243–260
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Combining object and feature dynamics in probabilistic tracking

Leonid Taycher *, John W. Fisher III, Trevor Darrell

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Received 7 May 2005; accepted 5 November 2006
Available online 9 May 2007
Communicated by James Maclean

Abstract

Objects can exhibit different dynamics at different spatio-temporal scales, a property that is often exploited by visual tracking algorithms. A local dynamic model is typically used to extract image features that are then used as inputs to a system for tracking the object using a global dynamic model. Approximate local dynamics may be brittle—point trackers drift due to image noise and adaptive background models adapt to foreground objects that become stationary—and constraints from the global model can make them more robust. We propose a probabilistic framework for incorporating knowledge about global dynamics into the local feature extraction processes. A global tracking algorithm can be formulated as a generative model and used to *predict* feature values thereby influencing the observation process of the feature extractor, which in turn produces feature values that are used in high-level inference. We combine such models utilizing a multichain graphical model framework. We show the utility of our framework for improving feature tracking as well as shape and motion estimates in a batch factorization algorithm. We also propose an approximate filtering algorithm appropriate for online applications and demonstrate its application to tasks in background subtraction, structure from motion and articulated body tracking.

Keywords: Probabilistic graphical models; Approximate models; Articulated body tracking; Background subtraction; Shape from motion

1. Introduction

Motion analysis algorithms are often structured in a multistage fashion, with each stage operating at a particular spatio-temporal scale and exploiting a different model of scene dynamics. Systems of this type are usually more computationally efficient than monolithic ones that jointly model local and global dynamics. They also have the advantage of modularity, as algorithms at each stage can be designed independently. Rather than using raw pixel data, high-level (large scale) stages treat the output of early, low-level ones as observations. For example, an algorithm may start by extracting local features (e.g., foreground/background labels or feature point tracks) from incoming frames, use these features to determine poses of the objects moving in the scene, and then analyze object interaction based on the individual objects' poses. High-level algorithms use models that are often too coarse (and/or approximate) for local motion estimation, but take into account global spatial relationships.

Low-level algorithms ignore global spatial relationships by modeling the evolution of each image patch (in feature extraction [22,25]) or object (in object tracking [17]) independently, and compensating for it with restrictive assumptions about the local behavior of the scene. Feature-point trackers usually assume that the image patch about the point of interest has a relatively stable appearance. Adaptive background subtraction modules typically assume that foreground objects do not remain stationary for extended periods of time. When these assumptions are violated, the resulting errors (e.g., so-called "sleeping man problem", Fig. 1), are propagated to higher-level modules, and these are not always able to correct them.

While algorithms operating at each stage are often formulated as inference in probabilistic generative models, most existing multi-stage systems are formed in an ad

* Corresponding author.
E-mail addresses: lodrion@csail.mit.edu (L. Taycher), fisher@csail.mit.edu (J.W. Fisher III), trevor@csail.mit.edu (T. Darrell).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.11.022

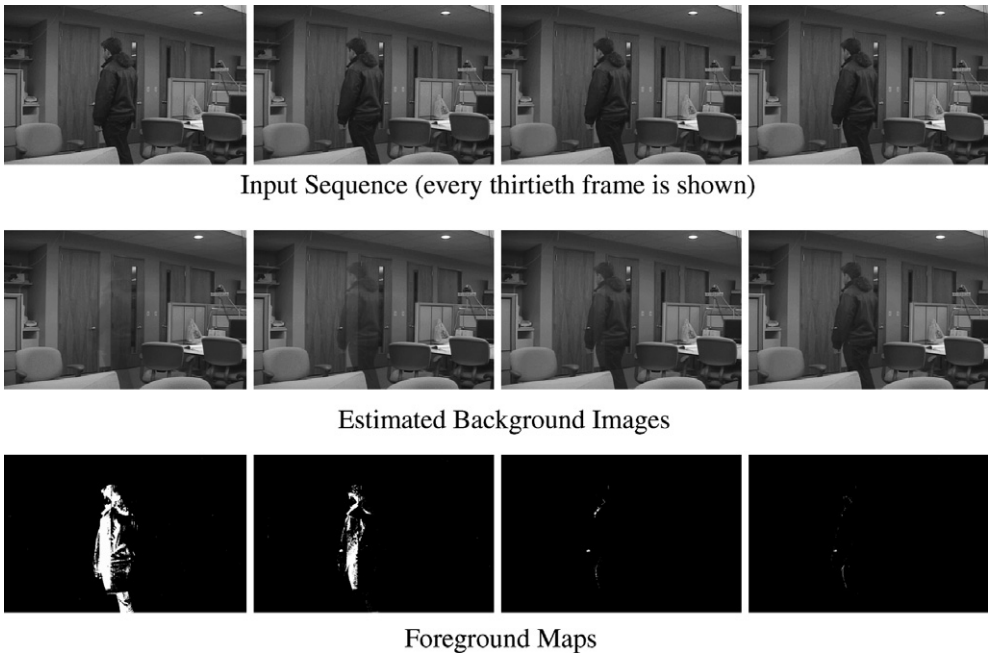

Fig. 1. An example of the "sleeping man" problem in adaptive background subtraction. Adaptive background maintenance systems make an implicit assumption that foreground objects do not remain stationary. When this is not the case (as in the sequence shown in the top row), the background model (middle row) adapts to motionless foreground objects which then "fade-away" (the computed foreground maps are in the bottom row).

hoc fashion and do not have a consistent probabilistic interpretation—e.g., the uncertainty information is propagated only in one direction, from low- to high-level models. The need to incorporate a feedback mechanism into multi-stage systems has long been recognized [20,1]. There are three desirable criteria for a viable feedback framework. First, it should preserve existing modularity (i.e., not be reduced to a monolithic model). Second, it is advantageous to be able to use existing algorithms with minimal modifications. Finally, it is critical to have consistent propagation of uncertainty from high- to low-level processing.

In this paper, we develop a framework that satisfies these requirements in the case when the constituent tracking algorithms can be interpreted as inference in dynamic generative models. Our motivation in building such a framework is based on the observation that when two modules interact, they share scene representation (which, with some abuse of terminology, we refer to as "features"). The features (e.g., foreground labels or individual object positions) are *latent* variables in the lower-level module, but are treated as *observed* at the higher-level. We can make these variables latent high-level generative models by explicitly modeling their dependency on the images. In a sense, each model can then be thought of as describing the evolution of the *features* across time with different approximations to the true dynamic. The models may then be combined by sharing these variables, in a manner similar to Product of Hidden Markov Models (PoHMMs) [3].

The resulting framework, which we call a Redundant-State Multi-Chain Model (RSMCM), may be thought of as performing probabilistic model-based regularization of low-level algorithms, similar to deterministic model-based regularization [1]. Since methods operating at different levels are coupled only through latent features, modularity is preserved with only minimal modification to the algorithms.

In the following discussion, we focus on systems that combine feature extraction and individual object tracking, but the conclusions may be extended to systems which model more than two levels. We demonstrate the advantages of our framework by applying it to such problems as structure from motion recovery, adaptive background subtraction and articulated body tracking.

2. Related work

Independence assumptions inherent in low-level tracking algorithms, combined with image noise, can lead to unreliable (or incorrect) results under unexpected noise conditions. Without relaxing the assumptions, the best that can be done is to propagate not only feature values but also an uncertainty about the measurements. For example, dissimilarity computations [21] and Kalman filtering [20] have been used to estimate uncertainty of feature-point tracking.

Tracking results may be improved by introducing dependency between features. This dependency can be represented both with and without using a higher-level motion model. Model-free methods such as multi-hypothesis tracking [6] and probabilistic data association filters [8] are computationally intensive and can model dependency only at the data association level (i.e., can be used to disambiguate feature tracks). These methods cannot correct feature drift, and have poor performance when dealing with long-duration occlusions.

Global dynamics models have been involved in feature extraction on multiple levels. On the lowest level, robust methods such as Least Median Squares have been used to reject feature locations that are deemed to be outliers [14]. Model parameters estimated at the previous frame were used to initialize current-frame feature tracking in [1]. The complete integration of feature extraction and object motion is achieved in monolithic systems [27,23], which jointly model foreground and background processes. Such systems are jointly rather than modularly designed, inference algorithms are tuned for particular models for reasons of efficiency, and replacing one of the system's components is usually complicated.

The framework proposed in this paper is most closely related to the intermediate integration approaches of [16] and [13]. These methods update both global and local models based on the feature match deterministically selected from among those predicted by the global and local motion models. If no matches were produced, the corresponding feature is dropped. In contrast to these methods, our approach allows feature extractors to use the global motion model to recover after multiple frames with no observations.

We adopt a paradigm of reconciling multiple generative models (each corresponding to a particular set of independence assumptions and dynamics representations) that describe the same set of observations. This is in contrast to sensor fusion techniques [26] that use a single dynamical model to interpret multiple streams of observations.

Representing complex distributions as products of simple ones has been proposed in Product of Experts (PoE) [11] and Product of HMMs (PoHMMs) [3] frameworks. The PoHMMs is based on co-training training multiple simple HMMs on the same set of training data, and assigning a novel sequence probability equal to the scaled product of probabilities assigned by each HMM. Our framework also uses renormalized products of tractable probability distributions to model data that satisfies constraints arising from different models.

There are two major differences between our redundant-state multi-chain model and PoHMMs. First, individual chains in our model share a *latent* rather than an observed variable, which enables a two-way flow of information between states of the individual modules (e.g., object tracker and background maintenance). In particular, global spatial relationships are introduced into low-level modules by using the feature predictions available from the high-level generative model. In our example, object positions predicted by the object interaction model influence individual object tracking; position and appearance predictions from independent object tracking model then modify the behavior of the adaptive background subtraction.

The second difference is that in order for the product approximation to be advantageous, the errors in the predictions by the individual chains have to be uncorrelated. This property is ensured during the training process in the PoHMMs model. We, on the other hand, assume that stochastic models combined in RSMCM are completely prespecified, and that the appearance feature hierarchy (if any) is known; we concern ourselves with inference on the combined model, rather than learning its structure. Under these assumptions, the decorrelation of errors (and thus the improvement in estimation) has to be demonstrated separately. This can be done analytically (as we do for a purely linear-Gaussian models), or empirically.

Our approach differs significantly from factorial models of [9,15,7]. These methods partition the state into independently evolving subsets that jointly generate the observation. Furthermore, Boyen and Koller [2] have shown the conditions under which the posterior distribution of the state can be viably approximated as a product of marginal distributions of subsets, which allows for more efficient inference. However the model is constrained to a single state (and evolution) model. It does not easily allow simultaneous use of multiple *alternative* ways to generate the same observations, which is the key property of our approach.

3. Developing a redundant state model

We pose probabilistic model-based regularization as a problem of reconciling two generative models describing evolution of the observations using the same latent variables. Feature extraction algorithms can often be seen as inference in a generative model with a structure similar to the one in Fig. 2(a). The feature set at time t, $F^t = \{F_k^t\}$, is generated based on the hidden low-level state R^t (e.g., a background model), and is in turn used to generate the observed image, I^t. Feature behavior is typically modeled as independent, with the state evolving according to local dynamics $p(R^{t+1}|R^t) = \prod_k p(R_k^{t+1}|R_k^t)$. The features are then generated according to $p(F^t|R^t) = \prod_k p(F_k^t|R_k^t)$. The objective of the algorithm is to infer F^ts that are then used as input for object-tracking algorithms. As a consequence of the independence assumption both the state prediction, $p(R^t|R^{t-1})$, and the prior over features, which is given by

$$p(F^t|I^{0\ldots t-1}) = \int p(F^t|R^t) \times \int p(R^t|R^{t-1}) p(R^{t-1}|I^{0\ldots t-1})\,\mathrm{d}R^{t-1}\,\mathrm{d}R^t$$

is overly broad making the system susceptible to unmodeled image variations (e.g., template warps).

Similarly, a probabilistic object tracking algorithm may be formulated as inference in the model shown in Fig. 2(b). The hidden high-level state, S^t, evolves according to global dynamics, $p(S^t|S^{t-1})$. The feature set, F^t, is generated at every frame based on the rendering model $p(F^t|S^t)$. This model treats features as observations, ignoring the fact that in reality they are obtained from images by a low-level feature-extraction process. Random variables and conditional distributions used in this discussion are summarized in Table 1.

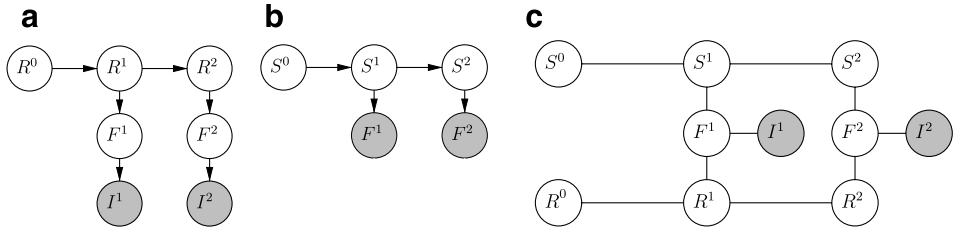

Fig. 2. Combining local and global dynamics for object tracking. (a) A generative model used in feature extraction algorithms. The low-level state, $R = \{R_k\}$, evolves according to the local dynamic model, $p(R^{t+1}|R^t) = \prod_k p(R_k^{t+1}|R_k^t)$. At time t, the observed image is drawn from $p(I^k|F^k)$, where the feature set, $F^t = \{F_k^t\}$, is generated from the state according to $p(F^t|R^t) = \prod_k p(F_k^t|R_k^t)$. (b) Generative model used for object tracking. The high-level state, S^t, contains pose and appearance information about moving object(s), and evolves according to global dynamic model, $p(S^{t+1}|S^t)$. The feature set, F^t, generated based on the appearance and pose is considered to be observed. (c) Combined model with potentials corresponding to the conditional probabilities in the individual models (e.g., $\phi(R^t, R^{t-1}) = p(R^t|R^{t-1})$, etc.).

Table 1
Summary of random variables and conditional distributions used in this paper

t	Time index
I^t	Image observed at time t
S^t	State of the high-level (tracking) generative model, e.g., 2D position and velocity of the object and its appearance
$p(S^t\|S^{t-1})$	High-level state evolution model
R^t	State of the low-level (feature extraction) generative model, e.g., per-pixel background models
$p(R^t\|R^{t-1})$	Low-level state evolution model
F^t	Latent instantaneous description of the world used by both models, e.g., pixels intensity values with corresponding foreground/background labels
$p(F^t\|S^t)$	The distribution used to generate latent features based on the high-level model state
$p(F^t\|R^t)$	The distribution used to generate latent features based on the low- level model state
$p(I^t\|F^t)$	Observation generation model

Both of the models described above are approximate. The question we address is how to combine them in a statistically consistent fashion so as to leverage complimentary properties of each. The local dynamic model ignores dependency between features, and the global dynamic model is usually too coarse to be of use for feature matching. By ignoring dependency between features, the feature extraction algorithm assumes that the joint distribution of the state and the appearance conditioned on all previous observations is

$$p(F^t, R^t|I^{0\ldots t-1}) = p(F^t|R^t) \int p(R^t|R^{t-1}) p(R^{t-1}|I^{0\ldots t-1})\, \mathrm{d}R^{t-1}, \qquad (1)$$

but the true distribution, which accounts for interfeature dependencies, is of the form

$$\begin{aligned} p(F^t, R^t|I^{0\ldots t-1}) &= q(F^t, R^t; I^{0\ldots t-1}) \\ &\quad \times \int p(R^t|R^{t-1}) p(R^{t-1}|I^{0\ldots t-1}) \mathrm{d}R^{t-1} \\ &\quad \times q(F^t, R^t; I^{0\ldots t-1}) \neq p(F^t|R^t). \end{aligned} \qquad (2)$$

That is, when the true dynamic model is used, F^t (and I^t) are independent from prior observations conditioned on R_i^t. However this is not the case when the approximate dynamic is used. Modeling the dependencies between F^t and prior observations that are unaccounted for by feature extraction model allows for better estimation of the state posterior. We chose the approximation to $q(F^t, R^t; I^{0\ldots t-1})$ that incorporates the information available to the object tracking model via a product

$$\hat{q}(F^t, R^t; I^{0\ldots t-1}) \propto p(F^t|R^t) \int p(F^t|S^t) p(S^t|I^{0\ldots t-1})\, \mathrm{d}S^t. \qquad (3)$$

This is equivalent to the *undirected* dual-chain model shown in Fig. 2(c), with potentials corresponding to conditional distributions from constituent models ($\phi(S^t, S^{t-1}) = p(S^t|S^{t-1})$, $\phi(F^t, S^t) = p(F^t|S^t)$, $\phi(I^t, F^t) = p(I^t|F^t)$, etc.). Sharing of the feature nodes between two individual models allows them to influence each other. For example, in the case of background subtraction, the background model would not be adapted to pixels that the tracking system predicts to be generated by the foreground objects; vice versa, pixels that are predicted to belong to the background would not be considered by the tracker. In the case of feature-point tracking, the prediction based on the global dynamic would serve as a data association filter, (e.g., it would mitigate individual point drift). The intuition behind this approximation from the modeling point of view is that while both models define broad priors over features, their product (similar to the fuzzy `and` operator) would be more narrow, making the overall system less sensitive to observation noise.

Although have so far we discussed the case when individual models use the same latent appearance features, it is possible to combine models with intersecting feature sets. In that case, the combined feature model would be the union of individual feature sets, and the likelihood poten-

tials are extended to produce uniform likelihoods for features that are not part of the original submodel. In general, when the feature sets are disjoint, the model would reduce to a PoHMMs model with non-interacting chains. Since we are interested in combining models that correspond to interacting stages of a feed-forward algorithm, we do not consider such cases.

3.1. Approximate filtering in the multi-chain model

Single-chain models are popular because there exist efficient algorithms for performing inference in them. While our proposed multi-chain model introduces loops (Fig. 3(a)), complicating inference in general, we take advantage of the fact that we are interested only in marginal distributions for the state nodes to propose an efficient algorithm for *filtering* in our multi-chain model.

Consider the model in Fig. 3(a). At time $t = 1$, we are concerned with nodes with superscripts (times) $t \leqslant 1$. If the initial states S^0 and R^0 are independent (as shown), then the resulting subgraph is a tree, and we can use the standard Belief Propagation [18] technique to compute exact marginal distributions at state nodes S^1 and R^1.

$$p(S^1|I^1) = \frac{1}{Z}\left[\int \phi(S^1,S^0)p(S^0)\,\mathrm{d}S^0\right]\left[\int \phi(F^1)\phi(F^1,S^1) \times \int \phi(F^1,R^1)\int \phi(R^1,R^0)p(R^0)\,\mathrm{d}R^0\,\mathrm{d}R^1\,\mathrm{d}F^1\right], \tag{4}$$

where $\phi(F^1) \equiv \phi(I^1,F^1)$. The expression for $p(R^1|I^1)$ can be similarly derived. Filtering at the next time step ($t = 2$) is more complex since the model now contains loops and the exact inference would require representing the joint $p(S^1,R^1|I^1)$:

$$p(S^2|I^1,I^2) = \frac{1}{Z}\int \phi(F^2)\phi(F^2,S^2)\int \phi(F^2,R^2) \times \int\int \phi(S^2,S^1)\phi(R^2,R^1)p(S^1,R^1|I^1) \times \mathrm{d}R^1\,\mathrm{d}S^1\,\mathrm{d}R^2\,\mathrm{d}F^2. \tag{5}$$

In order to simplify computations, we approximate the joint distribution, $p(S^1,R^1|I^1)$ with a product, $q(S^1)q(R^1)$. It can be easily shown that the best such approximation (in the KL-divergence sense) is the product of marginal distributions, $p(S^1|I^1)$ and $p(R^1|I^1)$. Substituting $p(S^1|I^1)p(R^1|I^1)$ for $p(S^1,R^1|I^1)$ in Eq. (5), we obtain an approximate inference equation:

$$p(S^2|I^2) = \frac{1}{Z}\int \phi(S^2,S^1)p(S^1)\,\mathrm{d}S^1\int \phi(F^2)\phi(F^2,S^2) \times \int \phi(F^2,R^2) \times \int \phi(R^2,R^1)p(R^1)\,\mathrm{d}R^1\,\mathrm{d}R^2\,\mathrm{d}F^2. \tag{6}$$

The similarity between Eqs. (4) and (6) suggests an approximate filtering algorithm that estimates marginal distributions of the state variables by recursively applying Belief Propagation to acyclic subgraphs of the form shown in Fig. 3(a), using the marginal state distribution obtained at time $t-1$ as priors at time t. It can be shown that this approximation preserves the main property of the exact model: the appearance features that are assigned zero probability under *any* of the constituent models are assigned zero probability in the computation of *all* of the marginal distributions. The messages exchanged between nodes during Belief Propagation are computed as described in Algorithm 1. Note that computations required for the prediction and update steps, as well as for part of the feature estimation step, are the same as those of individual object tracking and feature extraction algorithms.

If inference on constituent Markov chains were performed individually, it would still involve steps analogous to the prediction, update, and to part of the feature prediction steps of the approximate algorithm; consequently, combining models introduces very little additional complexity to the inference process.

3.2. Batch optimization in the multi-chain model

While filtering is appropriate for online tasks, some object-tracking problems are formulated as global optimizations in single-chain models such as the one in Fig. 2(b). For example, in structure-from-motion estimation we may be interested in computing the shape of the object based on *all* observed data, that is computing $\arg\max_{S^{0\ldots T}} p(F^{1\ldots T}|S^{0\ldots T})$. Once again, the algorithms devel-

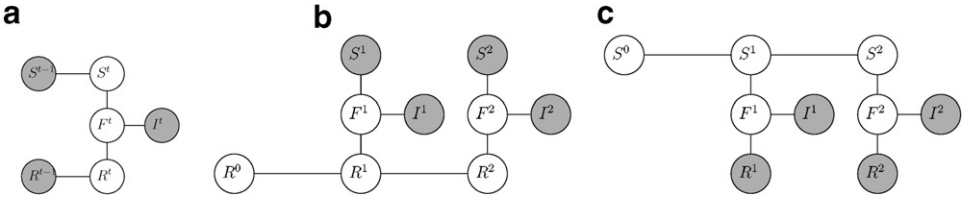

Fig. 3. Graph structures used in inference algorithms in the dual-chain model. (a) A tree-shaped subgraph on which a single step of approximate filtering is performed. The marginal distributions, $p(S^{t-1}|I^{0\ldots t-1})$ and $p(R^{t-1}|I^{0\ldots t-1})$, have been computed at the previous iteration, and are not modified; I^t is observed. (b and c) Subgraphs for coordinate ascent in the dual-chain model. By fixing values of states $S^{0\ldots T}$, the structure is reduced to the single-chain model shown in (b). Existing feature-extraction algorithms may be adapted to perform inference in this model with relatively little modifications. When $R^{0\ldots T}$ are fixed (c) an existing high-level optimization algorithm can be applied.

oped for single-chain models need to be modified to be of use in the dual-chain setting.

Algorithm 1. Recursive Belief Propagation Algorithm for Filtering in a Dual-Chain Model

for all $t \geqslant 0$ **do**
PREDICT the current state of the object and states of individual features, compute messages:
$\mu_{S^{t-1}\to S^t} = \int \mathrm{d}S^{t-1}\phi(S^t, S^{t-1})p(S^{t-1}|I^{0\ldots t-1})$ and
$\mu_{R^{t-1}\to R^t} = \int \mathrm{d}R^{t-1}\phi(R^t, R^{t-1})p(R^{t-1}|I^{0\ldots t-1})$.
ESTIMATE feature distributions based on predicted states and current observations, compute messages:
$\mu_{S^t\to F^t} = \int \mathrm{d}S^t\phi(F^t, S^t)\mu_{S^{t-1}\to S^t}$,
$\mu_{R^t\to F^t} = \int \mathrm{d}R^t\phi(F^t, R^t)\mu_{R^{t-1}\to R^t}$,
$\mu_{F^t\to S^t} = \int \mathrm{d}F^t\mu_{R^t\to F^t}\phi(I^t, F^t)$, and
$\mu_{F^t\to R^t} = \int \mathrm{d}F^t\mu_{S^t\to F^t}\phi(I^t, F^t)$.
UPDATE object state using features predicted by feature extractor and state of the feature extractor using features predicted by object model:
$p(S^t|I^{0\ldots t}) \propto \mu_{S^{t-1}\to S^t}\mu_{F^t\to S^t}$ and
$p(R^t|I^{0\ldots t}) \propto \mu_{R^{t-1}\to R^t}\mu_{F^t\to R^t}$.
end for

Algorithm 2.Coordinate Ascent for Batch Optimization in a Dual-Chain Model

APPLY feature-extraction algorithm to all available observations.
while not converged **do**
APPLY the global optimization algorithm to object model
COMPUTE feature predictions from the object model for each time step.
APPLY feature-extraction algorithm to all available observations while incorporating predictions from the object model on the feature level.
end while

We base our optimization approach on a coordinate ascent algorithm that alternates between optimizing one set of states (either $R^{0\ldots T}$ or $S^{0\ldots T}$) while keeping the other one fixed. The dual-chain structure, with latent feature nodes separating states, naturally lends itself to this algorithm. Fixing one set of states reduces the problem to a single-chain optimization that can be performed with available algorithms, (cf., Figs. 3(b and c). The summary of our method is presented in Algorithm 2.

3.3. Analyzing approximation validity

The redundant-state model described above is quite general, in that it allows combining any two dynamics models sharing the same "feature" representation. It is clear that there are cases when one of the constituent models would produce better results than RSMCM. For example if the R and S are defined over the same state space, and shape the same dynamics, then the product model would amplify the errors rather than decrease them! Taking the product of the approximate temporal prior with itself results in a prior that is *more certain* (has smaller variance) about an incorrect estimate. While the Product of HMMs [3] model may suffer from the same drawback, it is specifically trained to reduce the correlation between individual models and reduce the probability of being overconfident. Individual models are predefined in our framework, we analytically define when it is appropriate to combine two single-chain models into a RSMCM.

We analyze a case where the underlying and both approximate models are linear-Gaussian in order to obtain a closed-form solution; this case is directly useful and provides intuition about more complicated cases.

We consider the system that is described by the following equations:

$$\begin{cases} F^t = g(F^{t-1}) + \omega_0^t, & \omega_0^t \sim N(\omega_0^t; 0, \Sigma_0) \\ I^t = F^t + v^t, & v^t \sim N(v^t; 0, \Sigma_v), \end{cases} \tag{7}$$

where $N(\cdot; \mu, \Sigma)$ is a multi-variate Gaussian distribution with mean μ and covariance Σ. The approximate models are described by

$$\begin{cases} S^t = \hat{g}_1(S^{t-1}) + \omega_1^t, & \omega_1^t \sim N(\omega_1^t; 0, \Sigma_1) \\ F^t = S^t \\ I^t = F^t + v^t, & v^t \sim N(v^t; 0, \Sigma_v), \end{cases} \tag{8}$$

and

$$\begin{cases} R^t = \hat{g}_2(R^{t-1}) + \omega_2^t, & \omega_2^t \sim N(\omega_2^t; 0, \Sigma_2) \\ F^t = R^t \\ I^t = F^t + v^t, & v^t \sim N(v^t; 0, \Sigma_v). \end{cases} \tag{9}$$

Both approximate models share the emission (image generation) equations with the true model, but incorporate approximate evolution functions $\hat{g}_1(\cdot)$ and $\hat{g}_2(\cdot)$ rather than the true function $g(\cdot)$. All functions are modeled as linear. We denote $\mu_1 = \hat{g}_1(F^{t-1}) - g(F^{t-1})$ and $\mu_2 = \hat{g}_2(F^{t-1}) - g(F^{t-1})$.

For ease of analysis we assume that both approximate estimators are unbiased, that is

$$E_{F^{t-1}}[\mu_1] = E_{F^{t-1}}[\mu_2] = 0 \tag{10}$$

and have the covariance structure

$$E_{F^{t-1}}\left[\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\quad(\mu_1 \quad \mu_2)\right] = \begin{pmatrix}P_1 & P_{12}\\ P_{12}^T & P_2\end{pmatrix} \tag{11}$$

with the expectation taken with respect to the marginal distribution $p(F^{t-1})$. Evolution equations of each model can be described via conditional distributions

$$p(F^t|F^{t-1}) = N(F^t; g(F^{t-1}), \Sigma_0) \tag{12}$$

$$q_1(F^t|F^{t-1}) = N(F^t; g(F^{t-1}), \Sigma_1) \tag{13}$$

$$q_2(F^t|F^{t-1}) = N(F^t; g(F^{t-1}), \Sigma_2) \tag{14}$$

by using the property $F^t = S^t$ and $F^t = R^t$ of approximate models. All models share the same emission model

$$p(I^t|F^t) = N(I^t; F^t, \Sigma_v) \tag{15}$$

Using these conditional distributions we can define posterior distributions

$$p(F^t|I^t,F^{t-1}) \propto p(I^t|F^t)p(F^t|F^{t-1}),$$
$$q_1(F^t|I^t,F^{t-1}) \propto p(I^t|F^t)q_1(F^t|F^{t-1}), \quad \text{and}$$
$$q_2(F^t|I^t,F^{t-1}) \propto p(I^t|F^t)q_2(F^t|F^{t-1}).$$

We define the cost of using an approximate evolution model as an expected value of KL-divergence between the optimal (i.e., using the correct model) and approximate posteriors.[1]

$$C(q) = E_{I^t,F^{t-1}}\left[D_{\mathrm{KL}}\left(p(F^t|I^t,F^{t-1})\|q_{F|I}(F^t|I^t,F^{t-1})\right)\right] \quad (16)$$

We assume that single-chain models combined into a RSMCM are optimal, in the sense that they use noise distributions that would, on average, result in the best posterior estimates. Lemma 1 describes the conditions under which $C(q_1)$ and $C(q_2)$ are optimal.

Lemma 1. *$C(q_1)$ and $C(q_2)$ are minimized by setting $\Sigma_1 = P_1 + \Sigma_0$ and $\Sigma_2 = P_2 + \Sigma_0$.*

Theorem 2 describes *sufficient* conditions under which the product approximation that uses the conditional distribution

$$q_*(F^t|F^{t-1}) \propto q_1(F^t|F^{t-1})q_2(F^t|F^{t-1})$$

has cost $C(q_*)$ that is less than the cost of each of the constituent models.

Theorem 2. *$C(q_*) < C(q_1)$ and $C(q_*) < C(q_2)$ if matrices*

$$\Gamma_1 = Q_{1\eta}P_1 - \left(Q_{1\eta}P_{12} + (Q_{1\eta}P_{12})^T + (Q_{1\eta}\Sigma_0)^T\right) \quad \text{and}$$
$$\Gamma_2 = Q_{2\eta}P_2 - \left(Q_{2\eta}P_{12}^T + (Q_{2\eta}P_{12}^T)^T + (Q_{2\eta}\Sigma_0)^T\right)$$

are both positive semidefinite when

$$Q_{1\eta} = (I + (\Sigma_0 + P_1)\Sigma_v^{-1})^{-1}$$
$$Q_{2\eta} = (I + (\Sigma_0 + P_2)\Sigma_v^{-1})^{-1}$$

The proofs of this theorem and Lemma 1 can be found in Appendix A.

Theorem 2 confirms our intuition that the models combined into RSMCM should be decorrelated. In the extreme case where the models are perfectly correlated, $P_1 = P_{12}$ and $\Gamma_1 = -(Q_{1\eta}^T(P_{12}^T + \Sigma_0^T))^T$ is not positive semidefinite.

While it is well understood that unbiased estimators, whose errors are uncorrelated, can be coherently combined to produce an improved estimate, the previous analysis is more specific. For the Gaussian case, Theorem 2 quanitfies the degree of correlation in the estimation errors which can be tolerated and still produce an improved using an RSMCM. It is instructive to consider a one-dimensional case when all constituent matrices become scalars. The sufficient conditions then reduce to

$$p_1 \geqslant 2p_{12} + \sigma_0^2 \quad \text{and}$$
$$p_2 \geqslant 2p_{12} + \sigma_0^2 \quad (17)$$

That is each of the diagonal terms on the covariance matrix of the estimators should be greater than the sum of the off-diagonal terms and the noise variance of the underlying model. The off-diagonal terms in this case are equal to $\sqrt{p_1p_2}\rho_{12}$ where ρ_{12} is the correlation coefficient. For the above conditions to be satisfied, it is necessary for the correlation coefficient to be less than 0.5.

4. Applications

We demonstrate the utility of our RSMCM framework in three different domains. We present a redundant articulated-body tracking approach combining rigid 2D head and hand motion model with articulated body dynamics. We also show how ubiquitous low-level methods such as adaptive background subtraction and feature-point tracking, used in many high-level motion-analysis algorithms (e.g, [28,22,4,12,1]) can benefit from spatial coherence information available to those high-level algorithms. In particular, we demonstrate that the results can be dramatically improved by using a RSMCM formulation to combine adaptive background subtraction and multi-object (blob) tracking. Finally, structure-from-motion estimates in a RSMCM framework that includes a feature-point tracker are shown to be superior to those in a feed-forward system.

4.1. Articulated body tracking

We have used the multi-chain framework for tracking human motion. We modeled the human upper body with an articulated tree with 13 degrees of freedom—2 in-plane translational dofs, 3 rotational dofs at the neck, 3 rotational dofs at each shoulder and 1 rotational dof at each elbow.

Since no good parametric form is known for body-pose distribution, we chose to use a sample-based density representation. Common sample-based particle-filtering approaches (e.g., CONDENSATION) compute a posterior state distribution at each time step by sampling from the distribution at the previous time step propagated by dynamics and reweighting samples by their likelihood. If the configuration space is complex, then this procedure results in many samples falling into areas of zero likelihood unless the dynamics are well known. This increases the number of samples that need to be drawn. An alternative is *likelihood sampling* [24], when pose samples are drawn from the pose likelihood function and are reweighted based on the temporal prior. Although this method results in greater per-sample complexity, it enables us to use fewer samples since they are placed more appropriately with respect to the posterior distribution.

To implement likelihood sampling we take advantage of the fact that we are able to not only evaluate, but also draw samples from observation likelihood definitions for the head and hand locations (in this case, mixtures of Gaussians corresponding to the face detector outputs and to

[1] KL-divergence is, for reasons detailed in [5], a natural way to measure differences between distributions.

detected flesh-colored blobs). We define observation likelihood using latent image observation likelihoods: face detector output for the head segment, flesh-color likelihoods for the hands, and occlusion edge map matching for the rest of the segments. Once the 2D face and hand position samples have been drawn, we use them together with inverse kinematics constraints to define a pose-proposal distribution. We then use this distribution in the importance sampling framework to obtain samples from the pose likelihood.

We define our proposal distribution as in [24]. In defining the proposal distribution, we take advantage of the fact that once head and hand positions and neck configuration are specified, then arm configurations (shoulder and elbow angles) are independent, and each arm has only two degrees of freedom. The complete description of likelihood pose-sampling may be found in [24].

While a tracker based on likelihood sampling can successfully operate with a small number of samples and is self recovering, it is extremely sensitive to feature detector failures (such as flesh-color misdetections). In this work, we combine a likelihood-sampling tracker with low-level flesh-blob tracking using robust Kalman filtering. These tracking systems share appearance features (flesh-colored blobs), enabling us to combine them in the RSMCM model.

We have applied our RSMCM tracker to three sample sequences, with results shown in Figs. 4 and 5. For each frame in the sequence, we have rendered 40 randomly drawn samples from the posterior state distribution (the frontal view overlaid on top of the input image is shown in the middle row, and side view is shown in the bottom row). The tracking results for the first sequence are also available in the submitted video file (rendered at one third of the framerate). In most frames, the tracker succeeded in estimating poses that contained significant out of plane components and self occlusions, and was able to recover from mistracks (e.g., around frame 61 in the third sequence).

In Fig. 6, we compare the performance of the enhanced RSMCM tracker using 1000 samples per frame (first column), likelihood-sampling tracker using 1000 samples (second column), CONDENSATION tracker with 5000 samples that runs as fast as the RSMCM tracker (third column), and finally CONDENSATION tracker with 15,000 samples (the smallest number of samples that enables CONDENSATION to perform with accuracy approaching RSMCM tracker performance). The results are presented using the same method as in Fig. 4, the frontal view is shown overlayed on top of the input image, with the side view to the right of it.

The RSMCM tracker was able to successfully track the body through the entire sequence. The likelihood-sampling tracker was generally able to correctly estimate the pose distribution, but failed on frames where image features were not correctly extracted (cf. frames 20, 60, etc.). The CONDENSATION variant with 5000 samples failed after 30 frames partly due to sample impoverishment (note that only a few distinct samples were drawn in frames 40 and later). Increasing the size of sample set to 15,000 (with similar increase in the running time) allowed CONDENSATION to successfully track through most of the sequence (see Fig. 6).

Our method improves upon likelihood-sampling, and compares favorably with the CONDENSATION algorithm in two ways. First, a monolithic approach using CONDENSATION requires a significantly greater number of samples in order to explore the configuration space suf-

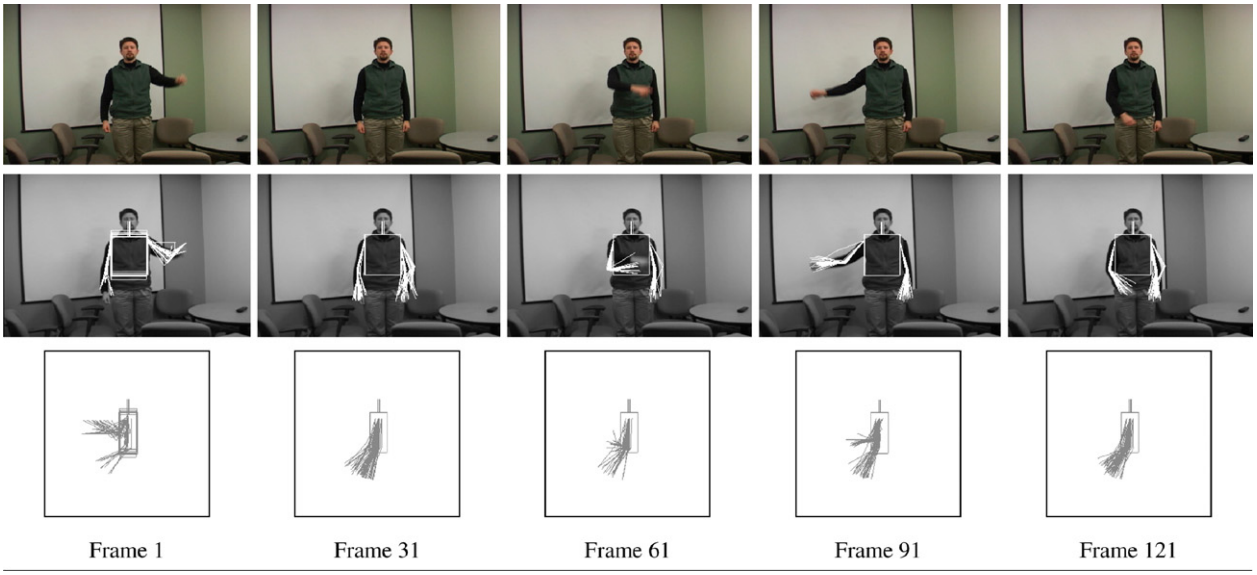

Fig. 4. Applying dual-chain tracking to sample sequence 1. The top row contains input frames. Forty random particles from the estimated posterior pose distributions are shown: in the middle row, the particles are rendered onto the input image (frontal view); in the bottom row they are rendered in the side view.

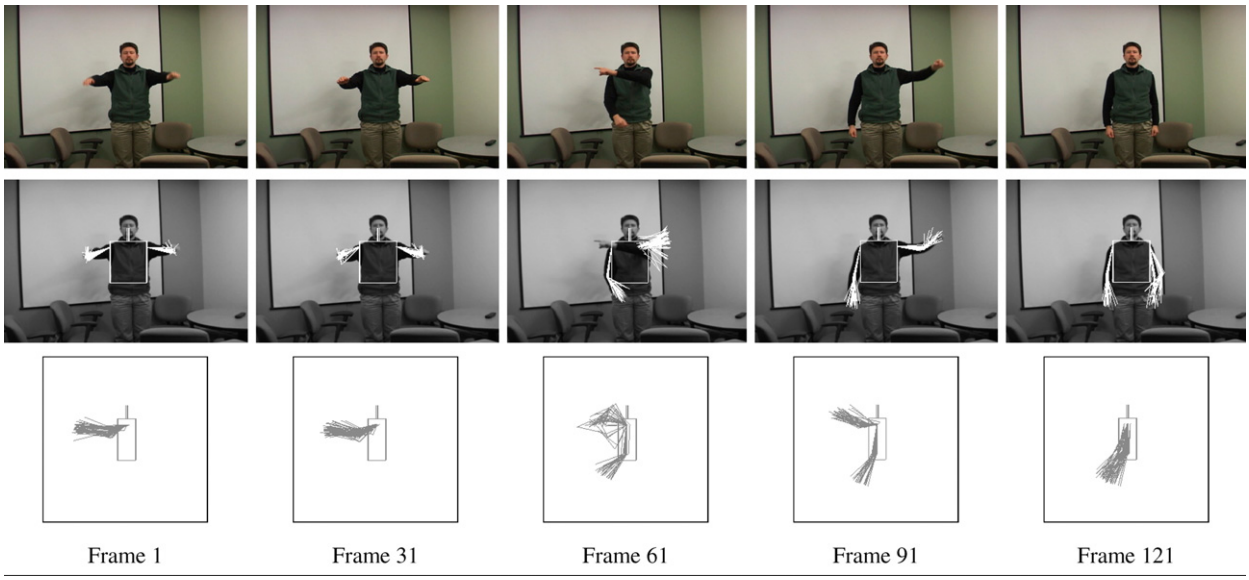

Fig. 5. Applying dual-chain tracking to sample sequence 3. The top row contains input frames. Forty random particles from the estimated posterior pose distributions are shown: in the middle row, the patricles are rendered onto the input image (frontal view); in the bottom row they are rendered in the side view. Note that while a mistrack has occurred on the third sequence near frame 61, the tracker was able to recover.

ficiently as compared to the RSMCM with likelihood sampling. Secondly, in the experiments presented, the estimate of the posterior state distribution more accurately represents the uncertainty of the upper-body pose than the alternative methods.

4.2. Improving adaptive background subtraction performance

Adaptive background models are popular since they are able to adjust to scene changes due to causes other than objects of interest (e.g., lighting variations). An important assumption made in all these models is that the background objects remain stationary for extended periods of time, while foreground objects tend to move frequently. So, when a foreground object stops for more than a few frames, the model adapts to it, causing it to "fade" into the background, and its location is no longer labeled as foreground (Fig. 1).

Common adaptive background algorithms similar to [22] can be represented as inference in a generative model that can then be incorporated into a RSMCM framework. This model maintains the background scene at time t as a set of independent per-pixel models $\{R_k^t\}$. A binary background label, B_k^t, is generated for every pixel according to the prior probability, $P(B_k^t)$. The latent pixel value, L_k^t, is generated according to the predicted model, R^t, if the pixel belongs to the background ($B_k^t = 1$) and by a uniform distribution otherwise. The value of L_j^t contaminated by observation noise is then observed as I_k^t. By denoting $F_k^t = (B_k^t, L_k^t)$, we obtain the form shown in Fig. 2(a).

The "fade-away" effect is caused, in part, by the use of constant $P(B_k^t)$, that governs the rate at which the background model is adapted to new observations. This problem may be alleviated by, modifying $P(B_k^t)$ based on feedback from an object (blob) tracking system. We achieve this by combining this background model with an object tracker (with the form shown in Fig. 2(b)) in the RSMCM framework.

We have used an object (blob) tracker with first-order linear dynamics similar to the one described in [22]. In this case, high-level state, S^t, contained appearances of the moving objects and their 2D positions and velocities. The background scene distribution was modeled with a single (per-pixel) Gaussian with fixed variance and variable mean. Model dynamics and observation noise were also represented with Gaussian distributions with fixed variances. Based on these modules, we implemented and compared the performance of the RSMCM algorithm and of the stand-alone background subtraction modules with different values of $P(B_k^t = 1)$. The resulting RSMCM implementation is able to solve the "sleeping man" problem described in Section 1. Compare the segmentation results from a stand-alone system in Fig. 1 and the redundant state system output in Fig. 7.

The systems were evaluated on datasets provided for the PETS 2001 workshop.[2] Algorithms were evaluated as follows: at every frame, we computed a raw foreground map by thresholding (at 0.5) the background probability value at every pixel and then extracted a set of connected components from this map.

We were interested in three common classes of errors: missing people, missing vehicles, and incorrectly detected "ghost" objects. We evaluated the following performance metrics: (1) less than 50% of a pedestrian covered by extracted components; (2) less than 50% of a vehicle cov-

[2] Available from ftp://pets.rdg.ac.uk/PETS2001/.

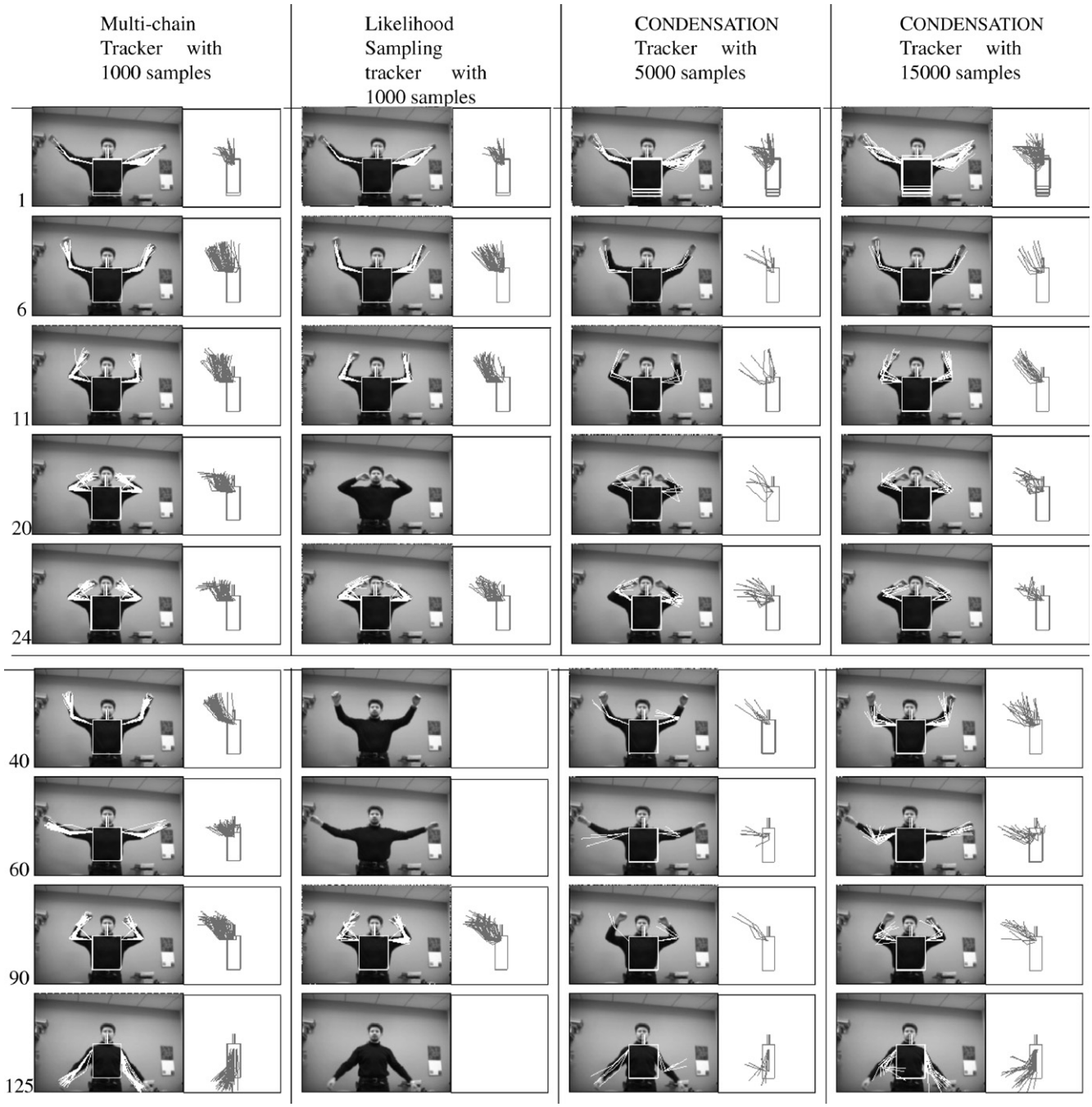

Fig. 6. Applying four tracking algorithms to a sample sequence. For each frame a set of 40 random pose samples were drawn from estimated posterior distribution and the corresponding skeletons were rendered (frontal view overlaid on the frame and side view below). Errors in feature detection caused likelihood-sampling tracker to fail on some of the frames (no samples were produced).

ered by extracted components; and (3) a foreground component detected in a location where no moving objects were present. Quantitative comparison results are summarized in Fig. 8. Sample frames from the first sequence with corresponding estimated background images and foreground components are shown in Fig. 11. The stand-alone background subtraction module suffers from the ''sleeping man'' problem and adapts to stationary vehicles (one in the middle of the screen and another in the bottom left corner). This may or may not be correct behavior for the car in the middle, since it does not move for the remainder of the sequence; it is clearly incorrect for the van in the bottom left, since it is lost by a tracker after the background model has adapted to it, and its subsequence motion results in mislabeled foreground regions. The RSMC model is not subject to these errors.

Importantly, replacing the feed-forward tracking algorithm with a RSMCM framework did not result in a large performance penalty. In our experiments, the difference between running times of the RSMCM algorithm and the feed-forward system was less than 4%. Partially optimized code on a 2.8 GHz workstation was able to achieve 9.6 fps

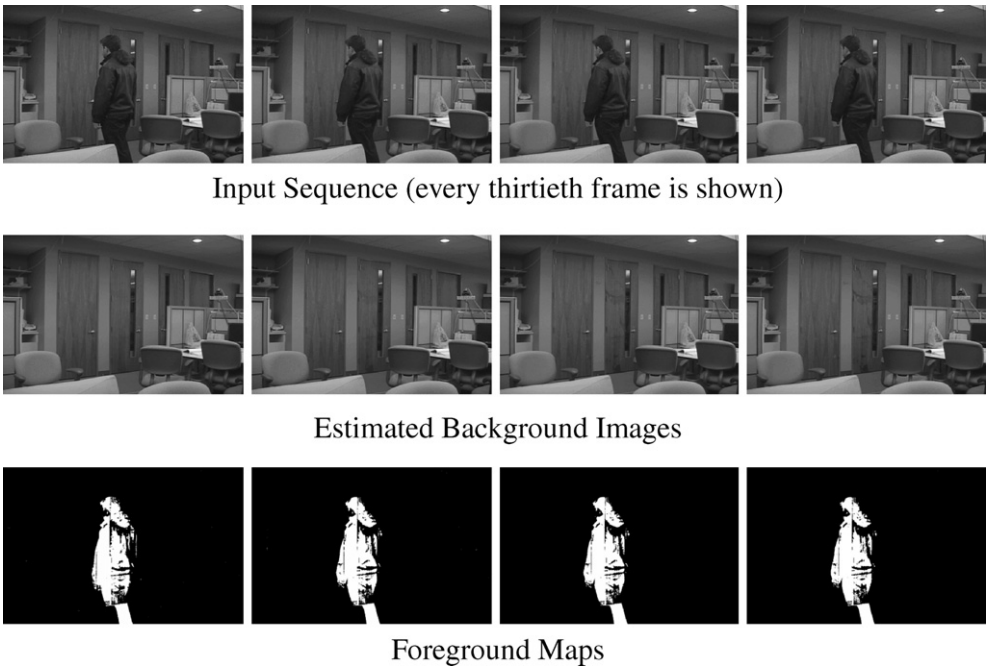

Fig. 7. Fixing "sleeping man" problem. Performance of the dual-chain system on the sequence shown in Fig. 1. Note that the correct background model and foreground maps are maintained while the person is stationary.

for sequential processing and 9.3 fps for RSMCM processing on 768×576 images (this time included reading images from the hard drive).

4.3. Structure from motion estimation

We have evaluated our batch optimization algorithm by applying it to the problem of extracting structure from motion sequences. Our algorithm combines a Kalman-filter based feature point tracker with structure-from-motion estimation [10].

Feature point tracking was implemented in a manner similar to that of [16]. The initial points of interest were located using Tomasi-Kanade feature point detector [25], and the 5×5 patches around the points were extracted. The points were then tracked using a first order Kalman filter, with the likelihood computed based on the normalized correlation scores around the location predicted by the filter. The concatenated states of individual point trackers were considered to be the state R of the feature-extraction chain, and the feature set F consisted of the 2d positions of the individual feature points and their appearances $F = \{(u_i, v_i, a_i)\}$.

Since the point tracking was part of a batch process, it was possible to further smooth point tracks using an RTS smoother [19].

A structure from motion estimation algorithm was implemented based on the variant of factor-analysis based method [10]. Denoting the 3D position of the ith point as (x_i, y_i, z_i), its projection at time t as (u_i^t, v_i^t), and first two rows of the homogeneous projection matrix at time t as $m^t = (m_1^t, \ldots, m_8^t)$, the noisy projection equations for P points in T frames are written by [10] as

$$\begin{pmatrix} u_1^1 & \ldots & u_P^1 & v_1^1 & \ldots & v_P^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_1^T & \ldots & u_P^T & v_1^T & \ldots & v_P^T \end{pmatrix} = \begin{pmatrix} m_1^1 & \ldots & m_8^1 \\ \vdots & \ddots & \vdots \\ m_1^P & \ldots & m_8^P \end{pmatrix} A + \eta_{[T \times 2P]}, \tag{18}$$

where

$$A = \begin{pmatrix} S & & \\ \mathbf{1}_{1\times P} & & \\ & S & \\ & \mathbf{1}_{1\times P} & \end{pmatrix}, \quad S = \begin{pmatrix} x_1 & \ldots & x_P \\ y_1 & \ldots & y_P \\ z_1 & \ldots & z_P \end{pmatrix}, \quad \eta_{ij} \sim N(0, \sigma_{ij}^2).$$

This equation is then solved using standard EM algorithm for factor analysis. The temporal coherence in pose estimates is enforced by adding second-order smoothness constraints over camera-motion parameters m^t:

$$\begin{aligned} m^t &= m^{t-1} + \dot{m}^{t-1} + \frac{1}{2}\ddot{m}^{t-1} + \epsilon_1, \\ \dot{m}^t &= \dot{m}^{t-1} + \ddot{m}^{t-1} + \epsilon_2, \\ \ddot{m}^t &= \ddot{m}^{t-1} + \epsilon_3. \end{aligned}$$

This algorithm may be converted to inference in the single-chain model in Fig. 2(b) by using $S^t = (\mathfrak{S}^t, m^t, \dot{m}^t, \ddot{m}^t)$, where $\mathfrak{S}^t = (x_1, y_1, z_1, \ldots, x_P, y_P, z_P)$ and $F^t = (u_1^t, \ldots, u_P^t, v_1^t, \ldots, v_P^t)$. The model dynamics are then

$$p(S^t|S^{t-1}) = \delta(\mathfrak{S}^t - \mathfrak{S}^{t-1}) N\left(\begin{pmatrix} m^t \\ \dot{m}^t \\ \ddot{m}^t \end{pmatrix}; \begin{pmatrix} \mathbf{1} & \mathbf{1} & \frac{\mathbf{1}}{\mathbf{2}} \\ 0 & \mathbf{1} & \mathbf{1} \\ & & \mathbf{1} \end{pmatrix} \begin{pmatrix} m^{t-1} \\ \dot{m}^{t-1} \\ \ddot{m}^{t-1} \end{pmatrix}, \Sigma_\epsilon\right), \tag{19}$$

where the first factor preserves the constancy of shape estimates across time and the second term describes the pose evolution. The feature generation model is

$$p(F^t|S^t) = N(F^t; m^t A, \Sigma_\eta), \tag{20}$$

with A defined in Eq. (18).

The feature tracking and structure estimation chains were combined as described in Algorithm 2. The feature tracking process was modified by replacing the Kalman prediction in the individual feature's prior by the product of the prediction available from the global model and the Kalman prediction. The effect of this combination was two-fold: it reduced the point drift and allowed for more robust handling of occlusions. If the feature point became occluded (i.e., the peak correlation value was below the threshold), the uncertainty in its position quickly became too large and it was dropped by the stand-alone tracker, and a new track was started when the point became visible again. In the RSMCM, the high-level prediction was, in effect, providing a virtual observation, which would preserve the track for longer periods of time. We have empirically verified that the errors in the low-level and high-level predictions have low correlation; the result of Section 3.3 is thus applicable.

We have experimented with RSMCM extensions of both the pure factor-analysis based algorithm and a variant that enforced pose coherence. In order to quantitatively compare the performance of these algorithms, we have created a synthetic dataset that emulates the behavior of common feature trackers on real data. Forty points randomly dis-

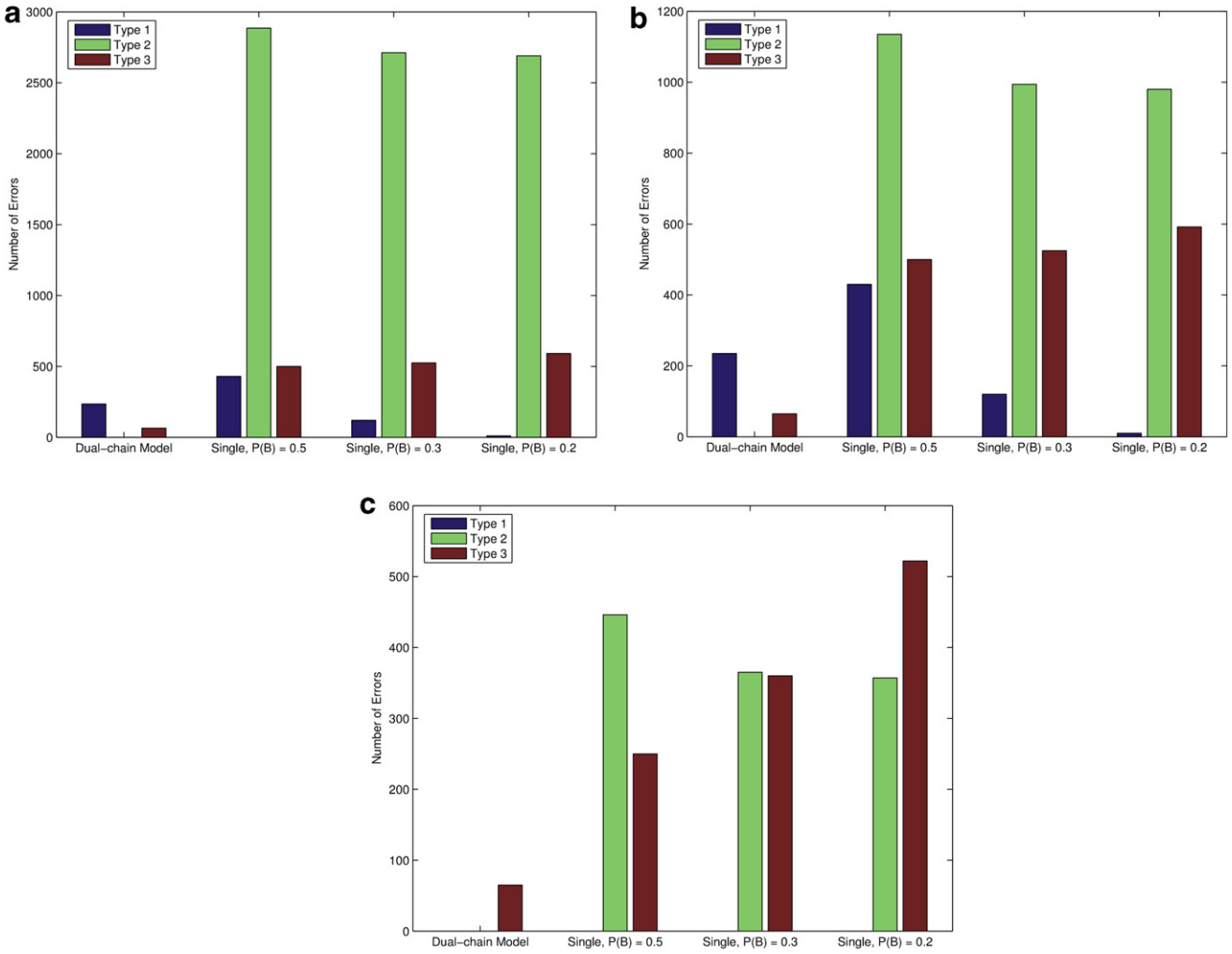

Fig. 8. Quantitative evaluation of background subtraction performance on PETS 2001 image sequences. Three error classes were differentiated. 1: no foreground components corresponding to a **pedestrian** have been detected. 2: no foreground components corresponding to a **vehicle** have been detected. 3: foreground component detected when no foreground object is present. Total number of errors in sequence 1 is presented in (a). Since one car in this sequence remains stationary after parking, its incorporation into the background model by single-chain trackers can be justified. The error chart in (b) shows results for sequence 1 ignoring type 2 errors corresponding to this car. Error statistics for sequence 2 are shown in (c). See the text for more details.

tributed on a unit cylinder were observed for 60 frames by a camera moving with constant angular velocity. To emulate occlusions and misdetections, every point changed state from visible to invisible in each frame with probability P(loose). To emulate template drift, consistent bias was introduced into each visible point for five frames with probability P(drift).

Shapes recovered for $P(\text{loose}) = 0.1$, $P(\text{drift}) = 0.3$ are shown in Fig. 9. The shapes computed by the single-chain variants contain more points. This is due to the fact that each point on the cylinder has produced several partial tracks separated by occlusions. The inability of a feature tracker to recognize partial tracks as belonging to a single feature complicates shape recovery. Since RSMCM methods are able to use the global model for data association, their shape estimates are much more accurate.

A quantitative evaluation of this experiment is shown in Fig. 10. The errors in individual feature trackers' and structure-based predictions have been empirically verified to have low correlation, so, as we would expect from the analysis in Section 3.3, RSMCM estimates have significantly lower errors than those from a feed-forward sys-

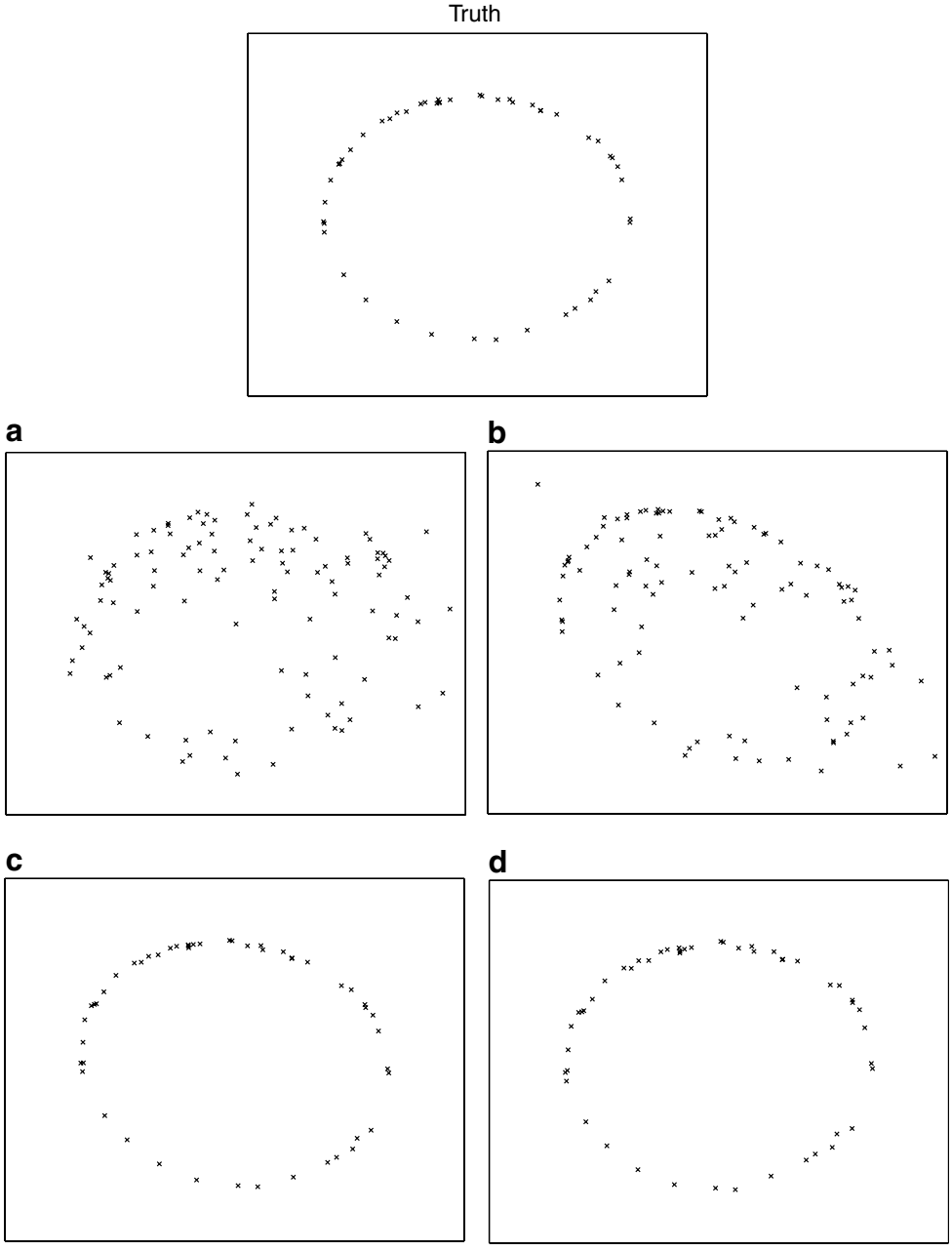

Fig. 9. Comparison of typical performance of factor analysis (a), factor analysis with temporal coherence (b), dual-chain factor analysis (c), and dual-chain factor analysis with temporal coherence (d) structure-from-motion algorithms on a synthetic sequence ($P(\text{loose}) = 0.1$, $P(\text{drift}) = 0.3$, see text for details). Single-chain methods produce much poorer results in the presence of occlusions due to their inability to establish correspondences between partial tracks of the same point.

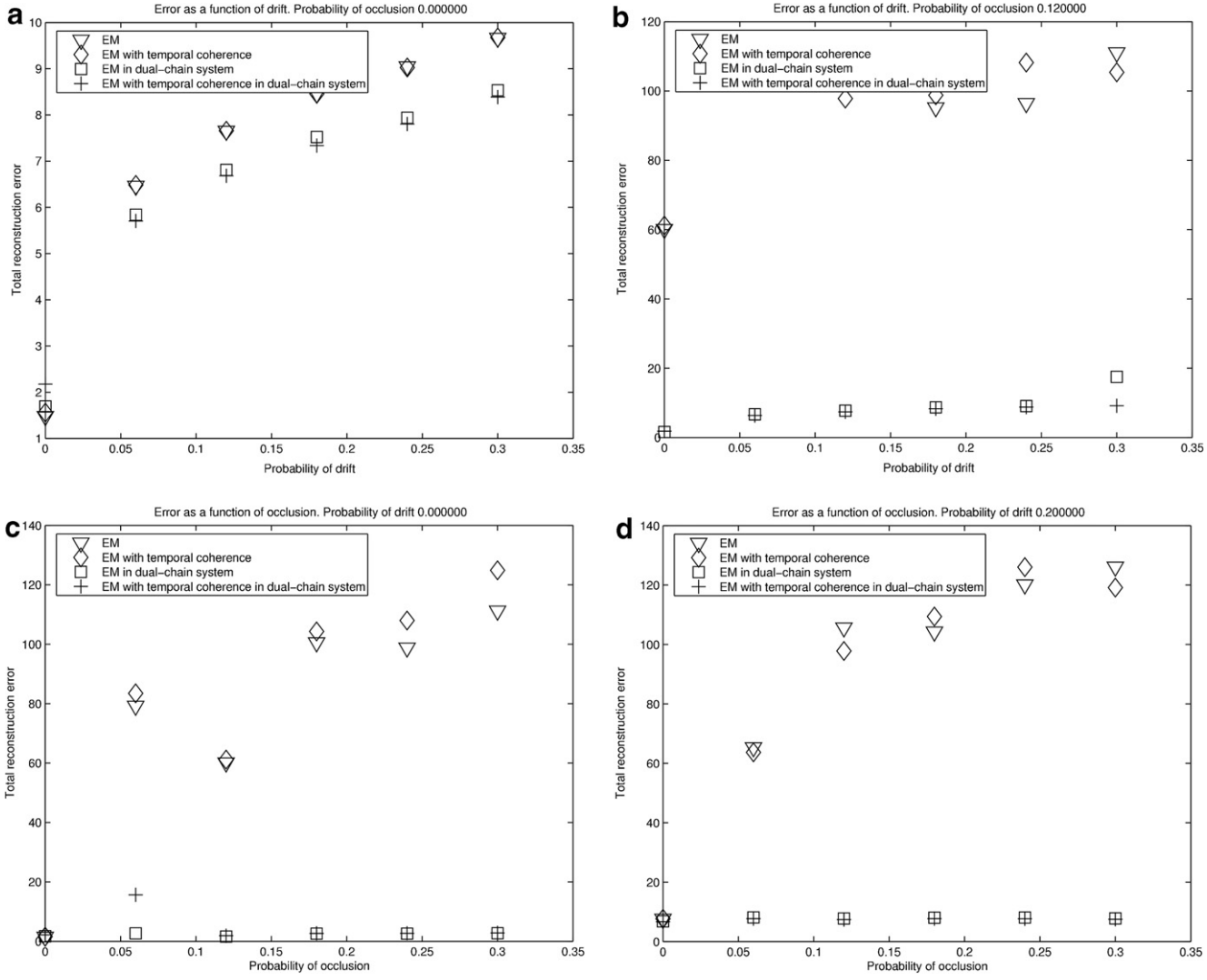

Fig. 10. Quantitative comparison of structure-from-motion recovery algorithms on the synthetic sequence with varying amounts of drift and occlusion. Top row—total reprojection error as a function of drift with no occlusion, i.e., $P(\text{loose}) = 0$ (left) and with 12% chance of occlusion, i.e., $P(\text{loose}) = 0.12$ (right). Bottom row—total reprojection error as a function of occlusion for $P(\text{drift}) = 0$ (left) and $P(\text{drift}) = 0.2$ (right). Dual-chain algorithms were able to approximately reconstruct shape in all cases. Single-chain methods failed for even small values of $P(\text{loose})$.

tem. Note that the number of occlusions (related to $P(\text{loose})$) had the greatest impact on the shape estimation. Neither of the single-chain approaches was able to deal with multiple partial tracks observed for one feature point. They failed to correctly recover the shape (signified by large reprojection errors), even for small values of $P(\text{loose})$.

The results of applying factor analysis with temporal coherence and its RSMCM variant to a fifty-frame video sequence[3] of a rotating box are shown in Fig. 12. The shape recovered by stand-alone factor analysis contains many spurious points, but the RSMCM framework succeeded in approximately estimating the correct shape.

5. Conclusions

We have proposed a method for combining probabilistic feature extraction and object tracking systems into a unified probabilistic model The approach was motivated by the observation that both of these models marginalize over an intermediate feature representation between state and observation. By making the feature representation explicit, we obtained a straightforward means of mediating between the constituent models. The resulting fused model has a clear probabilistic interpretation, reconciling multiple generative models that describe the same observations, each corresponding to a particular set of independence assumptions and dynamical model. In this paper we have concentrated on two-chain models with a single feature representation, although our framework is quite general and can incorporate multiple dynamic models and hierarchies of features.

[3] We used part of an original sequence from http://www.cs.ucla.edu/

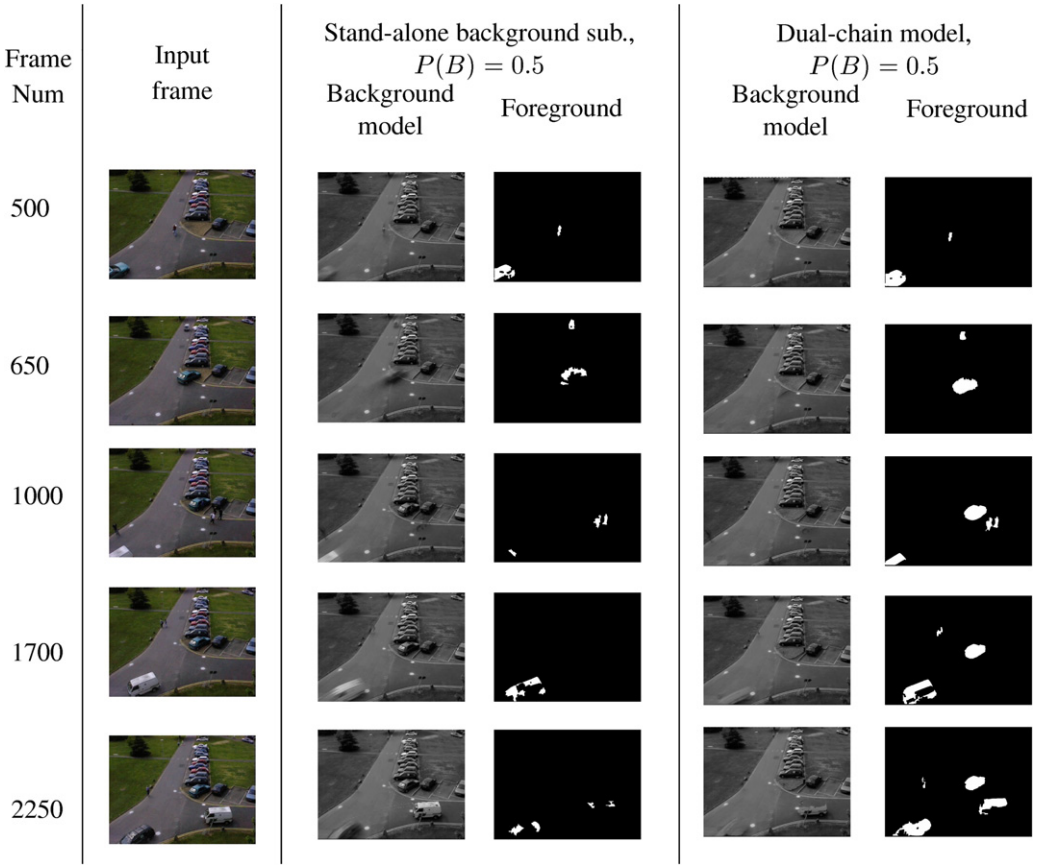

Fig. 11. Qualitative comparison of background subtraction performance on one PETS2001 image sequence. Second column holds input frames. Estimated background model and the computed foreground components are presented in the third and fourth columns for stand-alone background subtraction and in the fifth and sixth columns for dual-chain model. Note that while input images are in color, all computations were performed in grayscale. See text for more details.

Using the proposed framework requires some extra modeling in order to combine existing low- and high-level vision algorithms. An integrated model is enabled by the introduction of an explicit latent appearance model: this model is desirable for reasons of global consistency; however, exact inference on the resulting combined model is complicated by the introduction of loops. We have proposed two methods for adapting algorithms designed for constituent modules to operate in a combined system. An approximate inference method based on sequential inference on acyclic subgraphs provides a suitable alternative to exact filtering and was shown to perform well in online tracking applications. A coordinate-ascent based algorithm has been designed for the batch inference case and successfully applied to structure-from-motion estimation. Our method compared favorably to the pure feed-forward approaches in such diverse applications as articulated body tracking, background subtraction, and structure from motion estimation.

Appendix A. Proofs of analysis theorems

In order to prove Lemma 1 and Theorem 2 we first prove the following lemma

Lemma 3. *If*

$$p_y(y) = \int p_{y|x}(y|x)p_x(x)\,\mathrm{d}x,$$
$$p_{x|y}(x|y) = \frac{p_{y|x}(y|x)p_x(x)}{p_y(y)},$$
$$q_y(y) = \int p_{y|x}(y|x)q_x(x)\,\mathrm{d}x, \quad \textit{and}$$
$$q_{x|y}(x|y) = \frac{p_{y|x}(y|x)q_x(x)}{q_y(y)}$$

and all densities are absolutely continuous w.r.t. each other, then

$$E_{y\sim p_y(y)}[D_{\mathrm{KL}}(p_{x|y}(x|y)\|q_{x|y}(x|y))] = D_{\mathrm{KL}}(p_x(x)\|q_x(x)) - D_{\mathrm{KL}}(p_y(y)\|q_y(y))$$

Proof.

The lemma follows from [5] (p. 34, first equality). □

Using Lemma 3, we can re-express $C(q)$ as

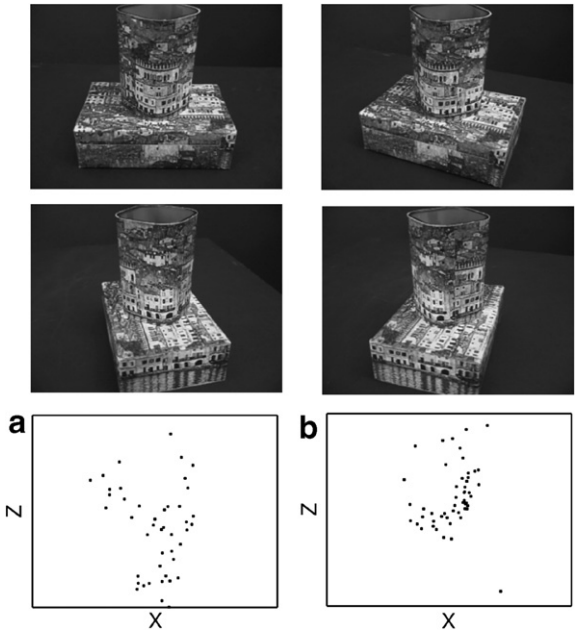

Fig. 12. Comparing shape points computed by the stand-alone factor-analysis with temporal coherence and its dual-chain variant. Top row: four frames of the input video sequence. Bottom row: (a) view from above onto the top part of the shape produced by factor-analysis and (b) view from above onto the top part of the shape produced by the dual-chain algorithm. Note that in the shape produced by factor analysis more than half of the points were spurious.

$$\begin{aligned} C(q) &= E_{I^t,F^{t-1}}[D_{\mathrm{KL}}(p(F^t|I^t,F^{t-1})\|q(F^t|I^t,F^{t-1}))] \\ &= E_{F^{t-1}}[E_{I^t}[D_{\mathrm{KL}}(p(F^t|I^t,F^{t-1})\|q(F^t|I^t,F^{t-1}))]] \\ &= E_{F^{t-1}}[D_{\mathrm{KL}}(p(F^t|F^{t-1})\|q(F^t|F^{t-1}))] \\ &\quad - E_{F^{t-1}}[D_{\mathrm{KL}}(p(I^t|F^{t-1})\|q(I^t|F^{t-1}))] \end{aligned} \tag{A.1}$$

Substituting expressions (12), (13) and (15) into Eq. (A.1), using a closed-form expression for KL divergence[4]

$$\begin{aligned} &D_{\mathrm{KL}}(N(x;m_1,S_1)\|N(x;m_2,S_2)) \\ &= \frac{1}{2}\left(\log\frac{|S_2|}{|S_1|} + Tr\left(S_1S_2^{-1} + S_2^{-1}(m_2-m_1)(m_2-m_1)^T\right) - d\right) \end{aligned}$$

and denoting $\mu_1 = \hat{g}_1(F^{t-1}) - g(F^{t-1})$, we obtain an expression for $C(q_1)$,

$$\begin{aligned} C(q_1) = &\frac{1}{2}E_{F^{t-1}}\left[\log\frac{|\Sigma_1|}{|\Sigma_0|} + \mathrm{Tr}\left(\Sigma_0\Sigma_1^{-1} + \Sigma_1^{-1}\mu_1\mu_1^T\right)\right] \\ &- \frac{1}{2}E_{F^{t-1}}\left[\log\frac{|\Sigma_1+\Sigma_v|}{|\Sigma_0+\Sigma_v|} + \mathrm{Tr}\Big((\Sigma_0+\Sigma_v)(\Sigma_1+\Sigma_v)^{-1}\right. \\ &\left.+ (\Sigma_1+\Sigma_v)^{-1}\mu_1\mu_1^T\Big)\right] \end{aligned} \tag{A.2}$$

$$\begin{aligned} &= \frac{1}{2}\left(\log\frac{|\Sigma_1|}{|\Sigma_0|} + \mathrm{Tr}(\Sigma_0\Sigma_1^{-1} + \Sigma_1^{-1}E_{F^{t-1}}[\mu_1\mu_1^T])\right. \\ &\quad - \log\frac{|\Sigma_1+\Sigma_v|}{|\Sigma_0+\Sigma_v|} - \mathrm{Tr}((\Sigma_0+\Sigma_v)(\Sigma_1+\Sigma_v)^{-1} \\ &\quad \left.+ (\Sigma_1+\Sigma_v)^{-1}E_{F^{t-1}}[\mu_1\mu_1^T])\right) \\ &= \frac{1}{2}\left(\log\frac{|\Sigma_1|}{|\Sigma_0|} + \mathrm{Tr}(\Sigma_1^{-1}(\Sigma_0+P_1)) - \log\frac{|\Sigma_1+\Sigma_v|}{|\Sigma_0+\Sigma_v|}\right. \\ &\quad \left.+\mathrm{Tr}((\Sigma_1+\Sigma_v)^{-1}(\Sigma_0+\Sigma_v+P_1))\right) \end{aligned}$$

The expression for $C(q_2)$ can be obtained in the similar manner.We can now prove Lemma 1.

Proof of Lemma 1. The derivative of $C(q_1)$ with respect to Σ_1 is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\Sigma_1}C(q_1) &= \frac{\mathrm{d}}{\mathrm{d}\Sigma_1}\frac{1}{2}\left(\log\frac{|\Sigma_1|}{|\Sigma_0|} + \mathrm{Tr}(\Sigma_1^{-1}(\Sigma_0+P_1))\right. \\ &\quad - \log\frac{|\Sigma_1+\Sigma_v|}{|\Sigma_0+\Sigma_v|} \\ &\quad \left.+\mathrm{Tr}((\Sigma_1+\Sigma_v)^{-1}(\Sigma_0+\Sigma_v+P_1))\right) \\ &= \frac{1}{2}\left(\frac{\mathrm{d}}{\mathrm{d}\Sigma_1}\log|\Sigma_1| + \frac{\mathrm{d}}{\mathrm{d}\Sigma_1}\mathrm{Tr}(\Sigma_1^{-1}(\Sigma_0+P_1))\right. \\ &\quad - \frac{\mathrm{d}}{\mathrm{d}\Sigma_1}\log|\Sigma_1+\Sigma_v| \\ &\quad \left.- \frac{\mathrm{d}}{\mathrm{d}\Sigma_1}\mathrm{Tr}((\Sigma_1+\Sigma_v)^{-1}(\Sigma_0+\Sigma_v+P_1))\right) \\ &= \frac{1}{2}\left(\Sigma_1^{-1} - \Sigma_1^{-1}(\Sigma_0+P_1)\Sigma_1^{-1} - (\Sigma_1+\Sigma_v)^{-1}.\right. \\ &\quad \left.+(\Sigma_1+\Sigma_v)^{-1}(\Sigma_0+\Sigma_v+P_1)(\Sigma_1+\Sigma_v)^{-1}\right) \\ &= \frac{1}{2}\left(\Sigma_1^{-1}(\Sigma_1-\Sigma_0-P_1)\Sigma_1^{-1}.\right. \\ &\quad \left.-(\Sigma_1+\Sigma_v)^{-1}(\Sigma_1-\Sigma_0-P_1)(\Sigma_1+\Sigma_v)^{-1}\right) \end{aligned} \tag{A.3}$$

Setting the derivative to 0, we obtain the only minimum at $\Sigma_{1\mathrm{opt}} = \Sigma_0 + P_1$. Applying the similar analysis to the second approximation, we obtain $\Sigma_{2\mathrm{opt}} = \Sigma_0 + P_2$. □

The costs of the optimal approximations can be obtained by plugging in the optimal values for dynamic noise covariance into cost expressions:

$$\begin{aligned} C(q_{1\mathrm{opt}}) &= \log\frac{|\Sigma_{1\mathrm{opt}}|}{|\Sigma_0|} - \log\frac{|\Sigma_{1\mathrm{opt}}+\Sigma_v|}{|\Sigma_0+\Sigma_v|} \\ C(q_{2\mathrm{opt}}) &= \log\frac{|\Sigma_{2\mathrm{opt}}|}{|\Sigma_0|} - \log\frac{|\Sigma_{2\mathrm{opt}}+\Sigma_v|}{|\Sigma_0+\Sigma_v|} \end{aligned} \tag{A.4}$$

The product of optimal individual priors (for a particular F^{t-1}) is a normal distributions with mean μ_* and covariance Σ_*, where

[4] d is the dimensionality of the space.

$$\Sigma_* = \left(\Sigma_{1\text{opt}}^{-1} + \Sigma_{2\text{opt}}^{-1}\right)^{-1}$$
$$\mu_* = \mu_*(F^{t-1}) = \Sigma_*\Big(\Sigma_{1\text{opt}}^{-1}(\hat{g}_1(F^{-1}) - g(F^{-1})) + \Sigma_{2\text{opt}}^{-1}(\hat{g}_2(F^{-1}) - g(F^{-1}))\Big)$$

The cost of the approximation is

$$\begin{aligned}
C(q_*) &= E_{F^{t-1}}[D_{\mathrm{KL}}(p_x(x)||q_{*x}(x))] - E_{F^{t-1}}[D_{\mathrm{KL}}(p_y(y)||q_{*y}(y))] \\
&= \frac{1}{2}E_{F^{t-1}}\left[\log\frac{|\Sigma_*|}{|\Sigma_0|} - \mathrm{Tr}(\Sigma_*^{-1}\Sigma_0 + \Sigma_*^{-1}\mu_*\mu_*^T) - \log\frac{|\Sigma_* + \Sigma_v|}{|\Sigma_0 + \Sigma_v|}\right. \\
&\quad \left. + \mathrm{Tr}((\Sigma_* + \Sigma_v)^{-1}(\Sigma_0 + \Sigma_v) + (\Sigma_* + \Sigma_v)^{-1}\mu_*\mu_*^T)\right] \\
&= \frac{1}{2}E_{F^{t-1}}\left[\log\frac{|\Sigma_*|}{|\Sigma_0|} - \log\frac{|\Sigma_* + \Sigma_v|}{|\Sigma_0 + \Sigma_v|}\right. \\
&\quad + \mathrm{Tr}((\Sigma_* + \Sigma_v)^{-1}(\Sigma_0 + \Sigma_v) - \Sigma_*^{-1}\Sigma_0 \\
&\quad \left. + ((\Sigma_* + \Sigma_v)^{-1} - \Sigma_*^{-1})\mu_*\mu_*^T)\right] \\
&= \frac{1}{2}\left(\log\frac{|\Sigma_*|}{|\Sigma_0|} - \log\frac{|\Sigma_* + \Sigma_v|}{|\Sigma_0 + \Sigma_v|}\right. \\
&\quad + \mathrm{Tr}((\Sigma_* + \Sigma_v)^{-1}(\Sigma_0 + \Sigma_v) - \Sigma_*^{-1}\Sigma_0) \\
&\quad \left. + \mathrm{Tr}(((\Sigma_* + \Sigma_v)^{-1} - \Sigma_*^{-1})E_{F^{t-1}}[\mu_*\mu_*^T])\right)
\end{aligned} \tag{A.5}$$

Expressing

$$\begin{aligned}
E_{F^{t-1}}[\mu_*\mu_*^T] = \Sigma_*(&\Sigma_{1\text{opt}}^{-1}P_1\Sigma_{1\text{opt}}^{-1} + \Sigma_{1\text{opt}}^{-1}P_{12}\Sigma_{2\text{opt}}^{-1} + \Sigma_{2\text{opt}}^{-1}P_{12}^T\Sigma_{1\text{opt}}^{-1} \\
&+ \Sigma_{2\text{opt}}^{-1}P_2\Sigma_{2\text{opt}}^{-1})\Sigma_*,
\end{aligned}$$

the cost may be rewritten as

$$\begin{aligned}
C(q_*) &= \frac{1}{2}\left(\log\frac{|\Sigma_{1\text{opt}}|}{|\Sigma_0|} - \log\frac{|\Sigma_{1\text{opt}} + \Sigma_v|}{|\Sigma_0 + \Sigma_v|})\right) \\
&\quad - \frac{1}{2}\left(\log\frac{|\Delta_1|}{\Sigma_{2\text{opt}}} + \mathrm{Tr}(\Sigma_{2\text{opt}}\Delta_1^{-1} + \Delta_1^{-1}\Gamma_1) - d\right) \\
&= C(q_{1\text{opt}}) - \frac{1}{2}\left(\log\frac{|\Delta_1|}{\Sigma_{2\text{opt}}} + \mathrm{Tr}(\Sigma_{2\text{opt}}(\Delta_1)^{-1}\right. \\
&\quad \left. + \Delta_1^{-1}\Gamma_1) - d\right)
\end{aligned} \tag{A.6}$$

with

$$\begin{aligned}
\Delta_1 &= (\Sigma_{1\text{opt}}^{-1} + \Sigma_v^{-1})^{-1} + \Sigma_{2\text{opt}} \\
\Gamma_1 &= (\Sigma_{1\text{opt}}^{-1} + \Sigma_v^{-1})^{-1}\Sigma_{1\text{opt}}^{-1}(P_1 - P_{12}) \\
&\quad + (-\Sigma_0 - P_{12}^T)\Sigma_{1\text{opt}}^{-1}(\Sigma_{1\text{opt}}^{-1} + \Sigma_v^{-1})^{-1} \\
&= Q_{1\eta}P_1 - \left(Q_{1\eta}P_{12} + (Q_{1\eta}P_{12})^T + (Q_{1\eta}\Sigma_0)^T\right),
\end{aligned}$$

where $Q_{1\eta} = (I + (\Sigma_0 + P_1)\Sigma_v^{-1})^{-1}$

The proof of Theorem 2 follows from the observation that $C(q_{1\text{opt}}) > C(q_*)$ iff $\frac{1}{2}(\log\frac{|\Delta_1|}{\Sigma_{2\text{opt}}} + \mathrm{Tr}(\Sigma_{2\text{opt}}\Delta_1^{-1} + \Delta_1^{-1}\Gamma_1 - d) > 0$. If Γ_1 is positive semidefinite, then Γ_1 can be written as

$$\Gamma_1 = \sum_{i=1}^{d}(\sqrt{\lambda_i}e_i)(\sqrt{\lambda_i}e_i)^T$$

where (λ_i, e_i) are its eigenvalue/eigenvector pairs, and then

$$\begin{aligned}
C(q_{1\text{opt}}) - C(q_*) &= \frac{1}{2}\left(\log\frac{|\Delta_1|}{|\Sigma_{2\text{opt}}|} + \mathrm{Tr}(\Sigma_{2\text{opt}}\Delta_1^{-1} + \Delta_1^{-1}\Gamma_1) - d\right) \\
&= \frac{1}{d}\sum_{i=1}^{d}\frac{1}{2}\left(\log\frac{|\Delta_1|}{|\Sigma_{2\text{opt}}|} + \mathrm{Tr}(\Sigma_{2\text{opt}}\Delta_1^{-1}\right. \\
&\quad \left. + \Delta_1^{-1}(\sqrt{\mathrm{d}\lambda_i}e_i)(\sqrt{\mathrm{d}\lambda_i}e_i)^T) - d\right) \\
&= \frac{1}{d}\sum_{i=1}^{d}D_{\mathrm{KL}}\left(N(x;0,\Sigma_{2\text{opt}})\|N(x;\sqrt{\mathrm{d}\lambda_i}e_i,\Delta_1)\right) > 0
\end{aligned}$$

References

[1] S. Basu, I. Essa, A. Pentland, Motion regularization for model-based head tracking, in: Proc. ICPR, Vienna, Austria, 1996.

[2] Xavier Boyen, Daphne Koller, Tractable inference for complex stochastic processes, in: Proc. UAI, 1998, pp. 33–42.

[3] A. Brown, G.E. Hinton, Products of hidden markov models, in: Proc. Artificial Intelligence and Statistics, 2001, pp. 3–11.

[4] J.P. Costeira, T. Kanade, A multibody factorization method for independently moving objects, IJCV 29 (3) (1998) 159–179.

[5] T.M. Cover, J.A. Thomas, Elements of Information Theory, J. Wiley & Sons, Inc., New York, 1991.

[6] I.J. Cox, S.L. Hingorani, An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking, PAMI 18 (2) (1996) 138–150.

[7] A. Doucet, N. de Freitas, K. Murphy, S. Russell, Rao-Blackwellised particle filtering for dynamic Bayesian networks, in: Proc. UAI, 2000.

[8] T. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, IEEE Journal of Oceanic Engineering 8 (3) (1983) 173–183.

[9] Z. Ghahramani, M. Jordan, Factorial hidden markov models, Machine Learning 29 (1997) 245–273.

[10] A. Gruber, Y. Weiss, Factorization with uncertainty and missing data: exploiting temporal coherence, in: Proc. NIPS, 2003.

[11] G.E. Hinton, Products of experts, in: Ninth International Conference on Artificial Neural Networks, 1999, pp. 1–6.

[12] D.W. Jacobs, Linear fitting with missing data for structure-from-motion, CVIU 82 (1) (2001) 57–81.

[13] T. Jebara, A. Pentland, Parametrized structure from motion for 3d adaptive feedback tracking of faces, Technical Report, MIT Media Lab, 1997.

[14] T. Kurata, J. Fujiki, M. Kourogi, K. Sakaue, A fast and robust approach to recovering structure and motion from live video frames, in: Proc. CVPR, 2000, pp. 528–535.

[15] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: Proc. ECCV (2), 2000, pp. 3–19.

[16] P.F. McLauchlan, I.D. Reid, D.W. Murray, Recursive affine structure and motion from image sequences, in: Proc. ECCV (1), 1994, pp. 217–224.

[17] N.M. Oliver, B. Rosario, A. Pentland, A bayesian computer vision system for modeling human interactions, PAMI 22 (8) (2000) 831–843.

[18] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufman, 1997.

[19] H. Rauch, F. Tung, C. Striebel, Maximum likelihood estimates of linear dynamic systems, American Institute of Aeronautics and Astronautics Journal 3 (8) (1965) 1445–1450.

[20] I. Reid, D. Murray, Active tracking of foveated feature clusters using affine structure, IJCV 18 (1) (1996) 41–60.

[21] J. Shi, C. Tomasi, Good features to track, in: Proc. CVPR, Seattle, June 1994.

[22] C. Stauffer, W.E.L Grimson, Adaptive background mixture models for real-time tracking, in: Proc. CVPR, 1999.

[23] Hai Tao, Harpreet S. Sawhney, Rakesh Kumar, Object tracking with bayesian estimation of dynamic layer representations, PAMI 24 (1) (2002) 75–89.
[24] L. Taycher, T. Darrell, Bayesian articulated tracking using single frame pose sampling, in: Proc. SCTV, October 2003.
[25] C. Tomasi, T. Kanade, Detection and tracking of point features, Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.
[26] K. Toyama, E. Horvitz, Bayesian modality fusion: probabilistic integration of multiple vision algorithms for head tracking, in: Proc. ACCV'00, 2000.
[27] C.R. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, PAMI 19 (7) (1997) 780–785.
[28] Q. Zhou, J. Aggarwal, Tracking and classifying moving objects from videos, in: Proc. PETS, 2001.